University of Waterloo

STAT 341 - Computational Statistics and Data Analysis

Winter 2024

Personal Course Notes

Brandon Zhou

BASIC INFO

Author	Brandon Zhou
Course Code	STAT 341
Course Name	Computational Statistics and Data Analysis
Days & Times	TTh 2:30PM - 3:50PM
Section	001
Date Created	January 05, 2024
Last Modified	April 08, 2024
Final Exam Date	TBA

DISCLAIMER

These course notes are intended to supplement primary instructional materials and facilitate learning. It's worth mentioning that some sections of these notes might have been influenced by ChatGPT, an OpenAI product. Segments sourced or influenced by ChatGPT, where present, will be clearly indicated for reference.

While I have made diligent efforts to ensure the accuracy of the content, there is a potential for errors, outdated information, or inaccuracies, especially in sections sourced from ChatGPT. I make no warranties regarding the completeness, reliability or accuracy of the notes contained in this notebook. It's crucial to view these notes as a supplementary reference and not a primary source.

Should any uncertainties or ambiguities arise from the material, I strongly advise consulting with your course instructors or the relevant course staff for comprehensive understanding. I apologize for any potential discrepancies or oversights.

Any alterations or modifications made to this notebook after its initial creation are neither endorsed nor recognized by me. For any doubts, always cross-reference with trusted academic resources.

TABLE OF CONTENTS

CHAPTER 1: Introduction

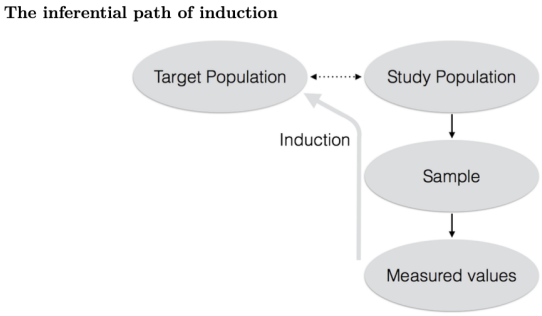


Figure 1: The inferential path of induction

The above content is for Lecture 1 on Jan 9, 2024

CHAPTER 2: Populations

2.1 Populations

Definition 2.1

Here we aim to describe a population using attributes.

- A population is a finite (though possibly huge) set $\mathcal{P}$ of elements.
  - Elements of a population are called units $u \in \mathcal{P}$
  - Variates are functions $x(u), y(u)$, etc. on the individual units $u \in \mathcal{P}$. For simplicity we will more often use the notation x_u, y_u, etc. when referring to the realized values of these variates for the unit $u = 1, \ldots, N$.
- We will define and explore interesting population attributes, denoted generally as $a(\mathcal{P})$.

2.2 Explicitly Defined Population Attributes

2.2.1 Population Attributes

Definition 2.2

Some definitions we need to know:

- The population is typically a set or collection of units, each with one or more variates that we can measure.
- Variates are characteristics of each unit in the population, and they can take on numerical or categorical values.
  - The values of variates typically differ from unit to unit.
  - If we are only interested in the variate y's we might write the population as

$$\mathcal{P} = \{y_1, y_2, \ldots, y_N\}$$

- Population attributes are summaries describing characteristics of the population.
  - Formally, an attribute is a function applied to the entire population and determined by the variate values observed for each of the population's units.

$$a(\mathcal{P}) = f(y_1, y_2, \ldots, y_N)$$

- Some examples of attributes are
  - the population total:

$$a(\mathcal{P}) = \sum_{u \in \mathcal{P}} y_u$$

– or various counts over the population

$$a(\mathcal{P}) = \sum_{u \in \mathcal{P}} I_A(y_u)$$

where $I_A(y)$ is the indicator function

$$I_A(y) = \begin{cases} 1 & \text{if } y \in A \\ 0 & \text{if } y \notin A \end{cases}$$

In general, attributes can be numerical or graphical – as long as they summarize the whole population.

Definition 2.3

Location Attributes measure or describe the centre of the distribution of variate values in a dataset.

- the population average:
$$a(\mathcal{P}) = \bar{y} = \frac{1}{N} \sum_{u \in \mathcal{P}} y_u$$
- the population proportion:
$$a(\mathcal{P}) = \frac{1}{N} \sum_{u \in \mathcal{P}} I_A(y_u)$$
- Other examples include the mode, the median, etc.

Spread Attributes measure variability or spread of the variate values in a data set. Some are

- the population variance:
$$a(\mathcal{P}) = \frac{1}{N} \sum_{u \in \mathcal{P}} (y_u - \bar{y})^2$$
- the population standard deviation:
$$a(\mathcal{P}) = SD_{\mathcal{P}}(y) = \sqrt{\frac{1}{N} \sum_{u \in \mathcal{P}} (y_u - \bar{y})^2}$$
- coefficient of variation:
$$a(\mathcal{P}) = \frac{SD_{\mathcal{P}}(y)}{\bar{y}}$$
- *Note*: the population variance or standard deviation could also be defined using $N - 1$ in the denominator.
- Other examples include the range, the inter-quartile range, etc.

Order Statistics

- Population attributes can also be based on an indexed collection of values,

$$y_{(1)} \leq y_{(2)} \leq \cdots \leq y_{(N)}$$

which are the variate values $y_u \in \mathcal{P}$ ordered from smallest to largest (including ties).

Location Attributes based on Order Statistics

These attributes measure or describe the centre of the distribution of variate values in a data set.

- the population minimum:

$$a(\mathcal{P}) = \min_{u \in \mathcal{P}} y_u = y_{(1)}$$

- the population maximum:

$$a(\mathcal{P}) = \max_{u \in \mathcal{P}} y_u = y_{(N)}$$

- the population mid-range:

$$a(\mathcal{P}) = \frac{1}{2}\left[\min_{u \in \mathcal{P}} y_u + \max_{u \in \mathcal{P}} y_u\right] = \frac{y_{(1)} + y_{(N)}}{2}$$

- the population median:

$$a(\mathcal{P}) = \text{median}_{u \in \mathcal{P}} y_u = \begin{cases} y_{\left(\frac{N+1}{2}\right)}, & \text{if } N \text{ is odd} \\ \frac{y_{\left(\frac{N}{2}\right)} + y_{\left(\frac{N}{2}+1\right)}}{2}, & \text{if } N \text{ is even} \end{cases}$$

- the population quartiles:
  - Q_1 is 25^{th} percentile, or the first quartile,
  - Q_2 is 50^{th} percentile, or the median, and
  - Q_3 is 75^{th} percentile, or the third quartile.

Variability Attributes based on Order Statistics

- The population range:

$$a(\mathcal{P}) = \max_{u \in \mathcal{P}} y_u - \min_{u \in \mathcal{P}} y_u = y_{(N)} - y_{(1)}$$

- The population inter-quartile range IQR:

$$a(\mathcal{P}) = Q_3 - Q_1$$

where Q_1 and Q_3 are 25^{th} and 75^{th} percentiles or the first and third quartiles, as above. Notice these are functions of entire population.

- The Median Absolute Deviation (MAD) is the median of the absolute differences between each

y_u and the median:

$$a(\mathcal{P}) = \text{median}_{u\in\mathcal{P}} |y_u - \text{median}_{u\in\mathcal{P}} y_u|$$

Skewness Attributes

These are measures of asymmetry in a population. A symmetric distribution of population values should result in a skewness attribute of zero.

- Pearson's moment coefficient of Skewness:

$$a(\mathcal{P}) = \frac{\frac{1}{N}\sum_{u\in\mathcal{P}}(y_u - \bar{y})^3}{[SD_{\mathcal{P}}(y)]^3}$$

- Pearson's second skewness coefficient (median skewness) given by:

$$a(\mathcal{P}) = \frac{3 \times (\bar{y} - \text{median}_{u\in\mathcal{P}} y_u)}{SD_{\mathcal{P}}(y)}$$

- Bowley's measure of skewness based on the quartiles:

$$a(\mathcal{P}) = \frac{(Q_3 + Q_1)/2 - Q_2}{(Q_3 - Q_1)/2}$$

NAs in R: Note that many programs in R accommodate missing data (represenated as `NA`s) and do something appropriate (typically they omit them).

- For your own code and analyses, you either need to decide what to do with `NA`s or ensure that the data do not have any `NA`s.
- If you choose to simply omit `NA`s, for example, the function `na.omit(...)` may be helpful (it will remove rows which contain an `NA` from a data set). For other possibilities see `help("na.omit")` in R.

2.2.2 Attribute Properties

Definition 2.4

A population attribute is a function of measured variates y_u:

$$a(\mathcal{P}) = f(y_1, y_2, \ldots, y_N)$$

and the variates y_u are typically associated with some measurement units.

Definition 2.5

Location Invariance and Equivariance

For an attribute $a(\mathcal{P}) = a(y_1, \ldots, y_N)$ we say that for any $m > 0$ and $b \in \mathbb{R}$, that the attribute is

- location invariant if

$$a(y_1 + b, \ldots, y_N + b) = a(y_1, \ldots, y_N)$$

- location equivariant if

$$a(y_1 + b, \ldots, y_N + b) = a(y_1, \ldots, y_N) + b$$

Example 2.1

The population average is location equivariant:

$$\begin{aligned} a(\mathcal{P}) = a(y_1, y_2, \ldots, y_N) &= \frac{1}{N}\sum_{i=1}^{N} y_i \\ a(y_1 + b, y_2 + b, \ldots, y_N + b) &= \frac{1}{N}\sum_{i=1}^{N}(y_i + b) \\ &= \frac{1}{N}\sum_{i=1}^{N} y_i + \frac{Nb}{N} = a(\mathcal{P}) + b \end{aligned}$$

But is the population variance location equivariant? No!

Definition 2.6

Scale Invariance and Equivariance

For an attribute $a(\mathcal{P}) = a(y_1, \ldots, y_N)$ we say that for any $m > 0$ and $b \in \mathbb{R}$, that the attribute is

- scale invariant if

$$a(m \times y_1, \ldots, m \times y_N) = a(y_1, \ldots, y_N)$$

- scale equivariant if

$$a(m \times y_1, \ldots, m \times y_N) = m \times a(y_1, \ldots, y_N)$$

- location-scale invariant if it is both location invariant and scale invariant, i.e.

$$a(m \times y_1 + b, \ldots, m \times y_N + b) = a(y_1, \ldots, y_N)$$

- location-scale equivariant if it is both location equivariant and scale equivariant, i.e.

$$a(m \times y_1 + b, \ldots, m \times y_N + b) = m \times a(y_1, \ldots, y_N) + b$$

Example 2.2

The population average is location-scale equivariant

$$\begin{aligned} a(my_1 + b, my_2 + b, \ldots, my_N + b) &= \frac{1}{N}\sum_{i=1}^{N}(my_i + b) \\ &= \frac{m}{N}\sum_{i=1}^{N} y_i + \frac{Nb}{N} \\ &= ma(\mathcal{P}) + b \end{aligned}$$

Definition 2.7
Replication

Another invariance/equivariance property of interest for population attributes is replication invariance and replication equivariance.

If a population $\mathcal{P}$ is duplicated $k-1$ times (so that there are k copies of it), how does the attribute change on this new population denoted by $\mathcal{P}^k$?

$$\mathcal{P}^k = \{y_1, y_2, \ldots, y_N, y_1, y_2, \ldots, y_N, \ldots, y_1, y_2, \ldots, y_N\} = \underbrace{\{x_1, x_2, \ldots, x_{kN}\}}_{kN \text{ elements}}$$

The attribute $a(\mathcal{P})$ is

- replication invariant whenever $a(\mathcal{P}^k) = a(\mathcal{P})$
- replication equivariant whenever $a(\mathcal{P}^k) = k \times a(\mathcal{P})$

Attribute properties

- A location attribute should be location equivariant, because it should reflect the change in the centre of the distribution of data. A location invariant location attribute cannot reflect the change in the centre, which would render it useless.
- A variance attribute should be location invariant, because the spread of the data's distribution does not depend on its centre. For example, in $N(\mu, 1)$ distribution, the standard deviation is 1 regardless of the value of μ, because the mean (location parameter) does not influence the spread of the distribution.
- A variance attribute should be scale equivariant, because scaling influences the spread of the data's distribution. As a variance attribute measures the spread of the distribution, it should reflect the influence of scaling accordingly. For example, consider the five points $\{1, 2, 3, 4, 5\}$. The sample standard deviation is approximately 1.58. However, once each point is multiplied by 2, the sample standard deviation doubles to approximately 3.16.
- A skewness attribute should be location invariant, because the asymmetry of distribution is independent of location, which affects every point equally. The influence of location is different from that of scaling, which affects points of larger magnitude more.
- A skewness attribute should be scale invariant, because scaling affects points of different magnitudes

differently, as mentioned above.

Example 2.3
The population average is replication invariant.

$$a(\mathcal{P}^k) = \frac{1}{kN}\sum_{j=1}^{kN} y_j = \frac{1}{kN}\sum_{i=1}^{N} k y_i = \frac{1}{N}\sum_{i=1}^{N} y_i = a(\mathcal{P})$$

2.2.3 Influence, Sensitivity Curves, and Breakdown Points

Definition 2.8
Influence(outlier detection)

- If we remove variate y_u (i.e. remove unit u) then the influence of that variate on the population attribute is quantified by

$$\Delta(a,u) = a(\underbrace{y_1,\ldots,y_{u-1},y_u,y_{u+1},\ldots,y_N}_{\textit{population with the unit } u}) - a(\underbrace{y_1,\ldots,y_{u-1},y_{u+1},\ldots,y_N}_{\textit{population without the unit } u})$$

- Ideally, no single unit's value should have greater influence than any other.
- If a unit has larger influence than the rest;
  1. it would require further investigation as it might be in error, or
  2. it might be the most interesting unit in the population.

The population average, $a(y_1, y_2, \ldots, y_n) = \bar{y}$ and the average without unit u can be written as

$$a(y_1,\ldots,y_{u-1},y_{u+1},\ldots,y_N) = \frac{1}{N-1}\sum_{\substack{k\in\mathcal{P},\\ k\neq u}} y_k = \frac{\sum_{k\in\mathcal{P}} y_k - y_u}{N-1} = \frac{N\bar{y}-y_u}{N-1}$$

and $\Delta(a,u)$, the influence for a given u, is:

$$\Delta(a,u) = \bar{y} - \frac{N\bar{y}-y_u}{N-1} = \frac{(N-1)\bar{y}-(N\bar{y}-y_u)}{N-1} = \frac{y_u-\bar{y}}{N-1}$$

The above content is for Lecture 2 on Jan 11, 2024

Definition 2.9
Sensitivity Curve

- We can also examine the effect on an attribute when we add a variate. To examine this effect,
  - suppose we have a population of size $N-1$ and
  - add a variate with the value y.

  - Then our new population with N elements is $\{y_1, \ldots, y_{N-1}, y\}$.

- We define the *sensitivity curve* of an attribute as

$$\begin{aligned} SC(y; a(\mathcal{P})) &= \frac{a(y_1, \ldots, y_{N-1}, y) - a(y_1, \ldots, y_{N-1})}{\frac{1}{N}} \\ &= N\left[a(y_1, \ldots, y_{N-1}, y) - a(y_1, \ldots, y_{N-1})\right] \end{aligned}$$

- We can then plot the *sensitivity curve* as a function of the new variate value y.
  - the sensitivity curve gives a scaled measure of the effect that a single variate value y has on the value of a population attribute $a(\mathcal{P})$.
- We can explore the sensitivity curve for any attribute. These can be determined *mathematically* in general, but can also be determined *computationally* for any particular population and any particular attribute.

The following is a general-purpose sensitivity curve function in R which accommodates any population and any attribute:

```
sc = function(y.pop, y, attr, ...) {
    N = length(y.pop) + 1
    sapply(y, function(y.new) { N * (attr(c(y.new, y.pop), ...) - attr(y.pop,
    ...)) })
}
# ... means "carry through any additional arguments".
```

Example 2.4

Derive the sensitivity curve for Arithmetic Mean

$$a(y_1, \ldots, y_N) = \frac{1}{N}\sum_{i=1}^{N} y_i = \bar{y}$$

$$
\begin{aligned}
P &= \{y_1, \ldots, y_{N-1}\} \\
P^* &= \{y_1, \ldots, y_{N-1}, y\} \\
a(P) &= \frac{1}{N-1} \sum_{i=1}^{N-1} y_i = \overline{y}_{N-1} \\
a(P^*) &= \frac{1}{N} \left[\sum_{i=1}^{N-1} y_i + y \right] \\
&= \frac{(N-1)\overline{y}_{N-1} + y}{N} \\
\therefore \mathrm{SC}(y, a) &= N\left[a(P^*) - a(P)\right] \\
&= N\left[\frac{(N-1)\overline{y}_{N-1} + y}{N} - \overline{y}_{N-1} \right] \\
&= (N-1)\overline{y}_{N-1} + y - N\overline{y}_{N-1} \\
&= y - \overline{y}_{N-1}
\end{aligned}
$$

Notes:

- A single observation can change the average by a huge (even infinite) amount.
- Averages may not be the best choice for a population attribute representing the location of a population – particularly if extreme values exist in the population.

Example 2.5

Derive the sensitivity curve for maximum

$$a(y_1, \ldots, y_N) = \max\{y_1, \ldots, y_N\} = y_{(N)}$$

$$
\begin{aligned}
P &= \{y_1, \ldots, y_{N-1}\} \\
P^* &= \{y_1, \ldots, y_{N-1}, y\} \\
a(P) &= y_{(N-1)} \\
a(P^*) &= \begin{cases} y_{(N-1)} & \text{if } y \le y_{(N-1)} \\ y & \text{if } y > y_{(N-1)} \end{cases} \\
\therefore \mathrm{SC}(y, \alpha) &= N\left[a(P^*) - a(P)\right] \\
&= \begin{cases} 0 & \text{if } y \le y_{(N-1)} \\ N[y - y_{(N-1)}] & \text{if } y > y_{(N-1)} \end{cases}
\end{aligned}
$$

If we draw the sensitivity curve for the maximum, we would find out it is unbounded for large y, the maximum is very sensitive to large outliers.

Example 2.6
Derive the sensitivity curve for 2^{nd} Order Statistic

$$a(y_1, \ldots, y_N) = y(2)$$

$$P = \{y_1, \ldots, y_{N-1}\}$$
$$a(P) = y(2)$$
$$P^* = \{y_1, \ldots, y_{N-1}, y\}$$
$$a(P^*) = \begin{cases} y_{(1)} & \text{if } y < y_{(1)} \\ y & \text{if } y_{(1)} \leq y < y_{(2)} \\ y_{(2)} & \text{if } y \geq y_{(2)} \end{cases}$$
$$\therefore \text{SC}(y, a) = N\left[a(P^*) - a(P)\right]$$
$$= \begin{cases} N(y_{(1)} - y_{(2)}) & \text{if } y < y_{(1)} \\ N(y - y_{(2)}) & \text{if } y_{(1)} \leq y < y_{(2)} \\ 0 & \text{if } y \geq y_{(2)} \end{cases}$$

Definition 2.10
Breakdown Points

Another measure of robustness that exists is called the breakdown point.

- It gives an assessment of just how large a proportion of the data must be contaminated before the statistic breaks down (and becomes useless).
- The breakdown point of a statistic is the smallest possible fraction of the observations that can be changed to something very extreme (i.e., plus or minus infinity) to make the error large (infinite)
- e.g. the break-point for
  - the average is $1/N$ (or asymptotically zero), and
  - the median is $1/2$ (i.e., that is half of the data has to go to infinity before the median breaks down).
- Attributes with high breakdown points are called resistant or robust.

2.2.4 Graphical Attributes

Population attributes can also be entirely graphical as in

- histograms of y_u values (univariate graphical summaries)
- bar plots of y_u values (univariate graphical summaries)
- box plots of y_u values (univariate graphical summaries)

- scatter-plots of pairs (x_u, y_u) (bivariate graphical summaries)
- scatter-plots of quantiles and ranks of y_u (bivariate graphical summaries)

Each of these plots summarizes the entire population, and so they're all attributes.

Histograms

Consider the population $\mathcal{P} = \{y_1, y_2, \ldots, y_N\}$.

- Partition the range of the population into k non-overlapping intervals, called bins, $I_j = [a_{j-1}, a_j)$, for $j = 1, 2, \ldots, k$ and then calculate the number (frequency) or proportion (relative frequency) of observations in the jth bin for $j = 1, \ldots, k$.
- Histograms help determine how the values are concentrated.

We can define bins two ways:

- bins of equal size, or (most common)
- bins with equal number of elements but varying size. ("equal area" histogram)
- Below are some examples of histograms with equal-sized bins (top row) and bins of varying sizes (bottom row)

```
x = agpop$farms87
par(mfrow=c(2,3), mar=2.5*c(1,1,1,0.1))
rx = range(x)
hist(x, breaks=seq(rx[1], rx[2], length.out=4), prob=TRUE, main="3 Bins", col
    = "grey")
hist(x, breaks=seq(rx[1], rx[2], length.out=5), prob=TRUE, main="4 Bins", col
    = "grey")
hist(x, breaks=seq(rx[1], rx[2], length.out=16), prob=TRUE, main="15 Bins",
    col = "grey")

# For the histograms in the bottom row, the areas of all rectangles in each
    panel are the same.
hist(x, breaks=quantile(x, p=seq(0, 1, length.out=4)), prob=TRUE, main="3 Bins
    ", col = "grey")
hist(x, breaks=quantile(x, p=seq(0, 1, length.out=5)), prob=TRUE, main="4 Bins
    " , col = "grey")
hist(x, breaks=quantile(x, p=seq(0, 1, length.out=16)), prob=TRUE, main="15
    Bins", col = "grey")
```

The bins with equal numbers of elements but varying size can help identify asymmetry in the population.

Rules for the Number of Bins

- Sturges rule:

$$\text{the number of bins should be } = \lceil \log_2(N) + 1 \rceil$$

- Freedman–Diaconis rule:

$$\text{Bin size } = 2\frac{\text{IQR}(x)}{N^{1/3}}$$

- Scott's rule:

$$\text{Bin size } = 3.5\frac{\sigma}{N^{1/3}}$$

Question: Which scale would you prefer to work with? The original scale or the transformed scale?

Answer: Advantages

- Raw data: data values are easily interpretable
- Transformed data: symmetric data are often easier to work with, statistically speaking

Scatter-plots

- A scatter-plot is a plot of the points (x_u, y_u) for all units in the population.
  - It is used to see whether two variates x and y are related in some way.
- A scatter-plot of the number of farms and total acreage of farming in 1987 by US county is below.

```
par(mfrow=c(1,2))
plot(agpop$farms87, agpop$acres87, pch = 19, cex=0.5, col=adjustcolor("black",
     alpha = 0.3), xlab = "Number of farms", ylab = "Total acreage of farming"
    , main = "US counties 1987")

plot(agpop$acres87, agpop$farms87, pch = 19, cex=0.5, col=adjustcolor("black",
     alpha = 0.3), ylab = "Number of farms", xlab = "Total acreage of farming"
    , main = "US counties 1987")
```

- Sometimes, the scatter-plot of a transformed version of the data provides more insight.

```
par(mfrow=c(1,2))
plot(log(agpop$farms87+1), log(agpop$acres87+1), pch = 19, cex=0.5, col=
    adjustcolor("black", alpha = 0.3), xlab = "log(Number of farms + 1)", ylab
     = "log(Total acreage of farming + 1)", main = "US counties 1987")
plot(log(agpop$acres87+1),log(agpop$farms87+1), pch = 19, cex=0.5, col=
    adjustcolor("black", alpha = 0.3), ylab = "log(Number of farms + 1)", xlab
     = "log(Total acreage of farming + 1)", main = "US counties 1987")
```

Sometimes the variates are integer-valued making duplicate values difficult to identify in Scatter-plots, there are two solutions:

- Try changing the shading and varying the size of bullets.
- Another way to deal with discreteness (multiple points at the same coordinate) is to add jitter to the population values. Jitter seperates duplicate points slightly (provided it makes sense to do so).

$$y_u^* = y_u + noise$$

```
y.example <- rep(1,100)
par(mfrow=c(1,3)) #dividing the panel into 1 row and 3 columns for 3 plots
title.seq = c('Raw data (all values are 1)', 'Raw Data + Jitter', 'Raw Data + More
    Jitter')
fact.seq = c(0, 1, 2)
for (i in 1:3) {
  plot(jitter(y.example, factor= fact.seq[i]),
    main=title.seq[i],
    ylim=c(0.9,1.1), ylab='measured values', cex.lab=1.5, cex.axis=1.5, pch=19)
}
```

2.2.5 Power Transformations

- For any variate y, it is sometimes helpful to re-express the values in a non-linear way via a transformation $T(y)$ so that on the transformed scale location/scale attributes are easier to define, to understand, or simply to determine.
- A commonly used transformation when $y > 0$ is the family of **power transformations** which is indexed by a power α. The general form is

$$T_\alpha(y) = \begin{cases} y^\alpha & \alpha > 0 \\ \log(y) & \alpha = 0 \end{cases}$$

- These transformations are monotonic, in the sense that

$$y_u < y_v \iff T_\alpha(y_u) < T_\alpha(y_v)$$

  That is, they preserve the order of the variate values associated with the units u and v.

  – What does change, often dramatically, is the relative positions of the variate values.

- What is the effect of varying the power transformation?
  1. Different values of α change the "spacing" between observations.
  2. Changing the spacing impacts how symmetric the histogram is
- Note: the most common purpose of a transformation is to change the shape of the histogram so that it is more symmetric.
  – We mentioned that if $y > 0$, the family of power transformations indexed by a power α is defined as

$$T_\alpha(y) = \begin{cases} y^\alpha & \text{if } \alpha > 0 \\ \log(y) & \text{if } \alpha = 0 \end{cases}$$

- An alternative mathematical form is

$$T_\alpha(y) = \frac{y^\alpha - 1}{\alpha} \quad \forall \alpha$$

  Note that the following limit gives rise to the $\alpha = 0$ case above:

$$\lim_{\alpha \to 0} T_\alpha(y) = \log(y)$$

- Yet another power transformation specification (with minimal potential for calculation errors) is the following:

$$T_\alpha(y) = \begin{cases} y^\alpha & \text{if } \alpha > 0 \\ \log(y) & \text{if } \alpha = 0 \\ -(y^\alpha) & \text{if } \alpha < 0 \end{cases}$$

- The effect of α changes on histogram
  - Decrease α: bump on histogram moves to the right
  - Increase α: bump on the histogram moves left

How to pick α ?

Two different, but related, effects of transformation are often of interest:

- First, producing a more symmetric looking histogram
- Second, producing roughly linear scatter-plots
  - Imagine (for all $u \in P$) a scatter-plot of all pairs (x_u, y_u).
  - Can we change the powers α_x and α_y for each such that the scatter-plot of the re-expressed pairs $(T_{\alpha_x}(x), T_{\alpha_y}(y))$ linearly on a straight line?
- Fortunately, for each of these effects there is a corresponding "bump rule" that indicates the direction (up or down) to move on Tukey's ladder to achieve it.

Bump Rule 1: Making histograms more symmetric

- The rule is that the location of the "bump" in the histogram (where the points are concentrated) tells you which way to "move" on the ladder.
  - If the bump is on "lower" values, then move the power "lower" on the ladder;
  - If it is on the "higher" values, then move the power "higher" on the ladder (John Tukey suggested (Tukey 1977) imagining that the set of powers were arranged in a "ladder" with the smallest powers on the bottom and the largest on the top.).

alpha	ladder
⋮	up
2	———
1	original values
$\frac{1}{2}$	———
$\frac{1}{3}$	———
0	———
$-\frac{1}{3}$	———
$-\frac{1}{2}$	———
-1	———
-2	———
⋮	down

Bump Rule 2: Straightening Scatter-plots

A scatter-plot of (x_u, y_u) for $u \in P$ may be "straightened" by applying (possibly) different power transformations to each coordinate to give a new (hopefully straighter looking) scatter-plot of the re-expressed data $(T_{\alpha_x}(x_u), T_{\alpha_y}(y_u))$.

- Because each of the coordinates has its own power transformation, there will be two different ladders of transformation
  - the x ladder and
  - the y ladder.
- As with histograms, there is a "bump rule" to tell you how to move on the ladder.
  - In the case of scatter-plots, the "bump" corresponds to the curvature appearing in the scatter-plot.
  - This is only approximate in practice, but reduces to one of four different possibilities:

Also, at least one of α_x, α_y needs to change to the expected direction and none of them should go to the opposite direction when straightening the line.

2.2.6 Order, Rank, and Quantiles

Definition 2.11

Population attributes can also be an indexed collection of values. For example, consider the following different attributes

- Recall the order statistics:

$$y_{(1)} \leq y_{(2)} \leq \cdots \leq y_{(N)}$$

  which are the ordered values (including ties) of the variate values $y_u \in \mathcal{P}$. $y_{(k)} = k^{th}$ smallest value of y.

- The rank statistics:

$$r_1, r_2, \ldots, r_N$$

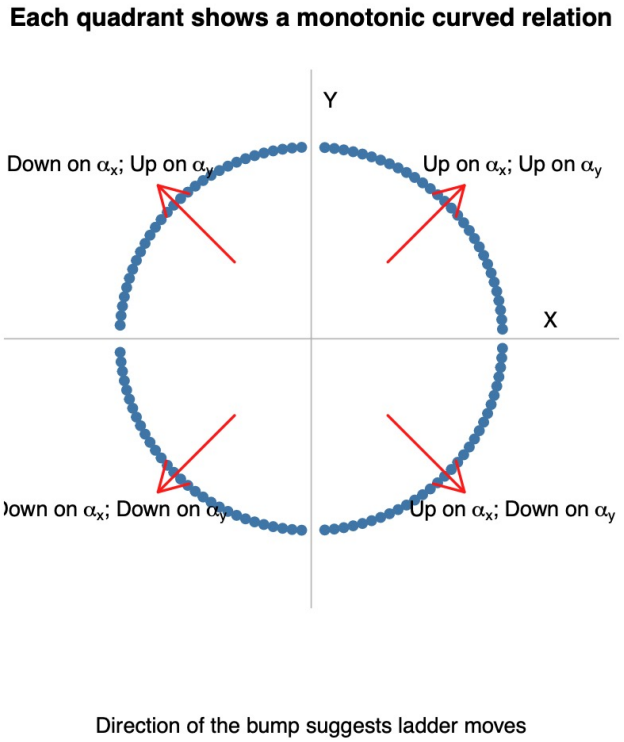


Figure 2: Direction of the bump suggests ladder moves

which are the ranks of the variate values $y_1, y_2, \ldots, y_N$ from the $y_u \in \mathcal{P}$. r_i = rank of unit i.

- For example, if $y_i = y_{(k)}$ then y_i is the k^{th} smallest value and so y_i has rank $r_i = k$. This means that

$$y_{(r_u)} = y_u \quad \forall u \in \mathcal{P}$$

Definition 2.12
Quantiles

- Rather than using ranks, it can be more convenient to use the proportion of units in the population having a value less-than-or-equal-to y.
  - So instead of plotting the pairs (r_u, y_u), we could equivalently plot the pairs (p_u, y_u) where

$$p_u = \frac{r_u}{N}$$

    is the proportion of the units $i \in \mathcal{P}$ whose value $y_i \leq y_u$.

- Notes
  - The middle value or proportion equal to $\frac{1}{2}$ corresponds to the median.

  - The values on the y-axis are the quantiles.
- Strictly speaking, the plotted points are $(p, Q_y(p))$ where
  - $p \in \left\{\frac{1}{N}, \frac{2}{N}, \ldots, 1\right\}$ and
  - $Q_y(p)$ is the p^{th} quantile of y

    $$Q_y(p) = y_{(N \times p)}$$

    and is sometimes called the quantile function of y for all $p \in \left[\frac{1}{N}, 1\right]$.
- The quantile function is a population attribute which can be used to generate a number of other interesting population attributes:
  - the quantile $Q_y(p)$ for any p locates the variate values in the population, and is thus a measure of location.
  - most (but not all) location measures try to capture central tendency.

Quantiles that measure center

- the median: $Q_y(1/2)$
- the mid-hinge (average of the first and third quartiles):

  $$\frac{Q_y(1/4) + Q_y(3/4)}{2}$$

- the mid-range (average of the minimum and maximum):

  $$\frac{Q_y(1/N) + Q_y(1)}{2}$$

- the trimean:

  $$\frac{Q_y(1/4) + 2 \times Q_y(1/2) + Q_y(3/4)}{4}$$

These can be readily obtained from the quantile plot.

- Reading off the vertical location of $Q_y(p)$ for any pre-determined p provides some measure of location.

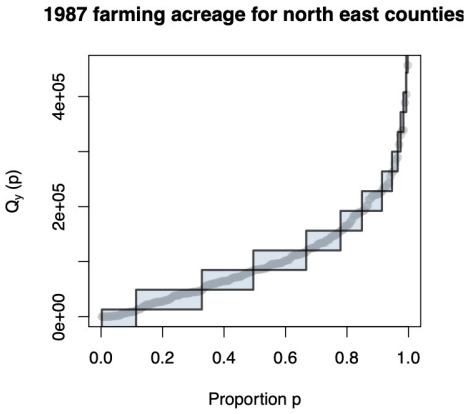
Quantiles that measure spread

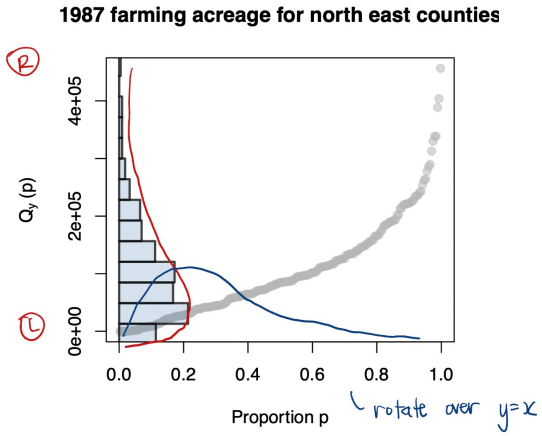
- The quantile function can also be used to provide some natural measures of spread for the variate y:
  - the range: $Q_y(1) - Q_y\left(\frac{1}{N}\right)$
  - the inter-quartile range: $IQR_y = Q_y\left(\frac{3}{4}\right) - Q_y\left(\frac{1}{4}\right)$
  - the central $100 \times p\%$ range
- Alternatively, the difference between any two quantiles might be divided by the difference in the corresponding p values.
  - That is, the slope of the line segment joining any two points $(p_1, Q_y(p_1))$ and $(p_2, Q_y(p_2))$ for $p_1 < p_2$ provides a measure of spread.

Concentration in Quantile Plots

Flatter regions in a quantile plot indicate areas where the variate values appear to be concentrated.

- To quantify this we could draw a box with fixed height and see how many elements are within the box.
- The width of the box is proportional to the number of elements it contains.
  - The greater the width, the greater the concentration.
- We can produce all such boxes, with fixed height, to see how the concentration changes with p.
- So how do we interpret these boxes on the histogram? What happens if we move them all to the left edge of the plot? A rotated histogram!
- A histogram of the acreage (or any y variate) is formed from the boxes that identify concentrations on the quantile plot!





2.3 Implicity Defined Attributes

2.3.1 The Minimum of a Function

In most practical situations we are interested in a (possibly vector-valued) attribute $\boldsymbol{\theta}$ which minimizes some function $\rho(\theta; \mathcal{P})$ of the variates in the population.

- That is, we want the value $\hat{\theta}$ which satisfies

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmin}}\ \rho(\theta; \mathcal{P})$$

  where the possible values of θ may be constrained to be in some set Θ.

- Note that maximizing a function is the same as minimizing its negation:

$$-\max_{\theta \in \Theta} \rho(\theta; \mathcal{P}) = \min_{\theta \in \Theta} -\rho(\theta; \mathcal{P})$$

  and so

$$\underset{\theta \in \Theta}{\operatorname{argmax}}\ \rho(\theta; \mathcal{P}) = \underset{\theta \in \Theta}{\operatorname{argmin}}\ -\rho(\theta; \mathcal{P})$$

Therefore, we only need to consider minimization here.

The most common form for $\rho(\theta, \mathcal{P})$ is a sum of functions $\rho(\theta, u)$ evaluated at each unit $u \in \mathcal{P}$:

$$\rho(\theta, \mathcal{P}) = \sum_{u \in \mathcal{P}} \rho(\theta, u)$$

Example 2.7
Scalar valued attributes

Some familiar examples for a scalar valued attribute $\theta \in \mathbb{R}$ and $u \in \mathcal{P}$ include:

- **Least-squares:** If $\rho(\theta; u) = (y_u - \theta)^2$ then

$$\hat{\theta} = \underset{\theta \in \mathbb{R}}{\operatorname{argmin}}\ \rho(\theta, \mathcal{P}) = \underset{\theta \in \mathbb{R}}{\operatorname{argmin}} \sum_{u \in \mathcal{P}} \rho(\theta, u) = \underset{\theta \in \mathbb{R}}{\operatorname{argmin}} \sum_{u \in \mathcal{P}} (y_u - \theta)^2 = \bar{y}$$

- **Weighted least-squares:** If $\rho(\theta; u) = w_u(y_u - \theta)^2$ then

$$\hat{\theta} = \underset{\theta \in \mathbb{R}}{\operatorname{argmin}}\ \rho(\theta, \mathcal{P}) = \underset{\theta \in \mathbb{R}}{\operatorname{argmin}} \sum_{u \in \mathcal{P}} \rho(\theta, u) = \underset{\theta \in \mathbb{R}}{\operatorname{argmin}} \sum_{u \in \mathcal{P}} w_u(y_u - \theta)^2 = \frac{\sum_{u \in \mathcal{P}} w_u y_u}{\sum_{u \in \mathcal{P}} w_u}$$

- **Least absolute deviations:** If $\rho(\theta; u) = |y_u - \theta|$ then

$$\hat{\theta} = \underset{\theta \in \mathbb{R}}{\operatorname{argmin}}\ \rho(\theta, \mathcal{P}) = \underset{\theta \in \mathbb{R}}{\operatorname{argmin}} \sum_{u \in \mathcal{P}} \rho(\theta, u) = \underset{\theta \in \mathbb{R}}{\operatorname{argmin}} \sum_{u \in \mathcal{P}} |y_u - \theta| = Q_y(1/2)$$

- **Least generalized-absolute deviations:** If for some $q \in (0, 1)$ we define the vee function

$$\rho_q(\theta; u) = \begin{cases} q(y_u - \theta) & \text{if } y_u \geq \theta \\ (q-1)(y_u - \theta) & \text{if } y_u < \theta \end{cases}$$

then

$$\hat{\theta} = \underset{\theta \in \mathbb{R}}{\operatorname{argmin}}\ \rho(\theta, \mathcal{P}) = \underset{\theta \in \mathbb{R}}{\operatorname{argmin}} \sum_{u \in \mathcal{P}} \rho(\theta, u) = \underset{\theta \in \mathbb{R}}{\operatorname{argmin}} \sum_{u \in \mathcal{P}} \rho_q(\theta; u) = Q_y(q)$$

Example 2.8
(Vector valued attributes): Simple Linear Regression

A familiar **vector valued attribute** is the vector of coefficients associated with the following simple linear regression:

$$y_u = \alpha + \beta(x_u - c) + r_u \quad \forall u \in \mathcal{P}$$

The attribute of interest is $\theta = (\alpha, \beta)$.

Note that a re-centering of the x_u values in a linear regression is not uncommon. Typically c is chosen to

be a meaningful value in the data set such as the average x_u value (i.e., $c = \bar{x}$), for example. Different choices of c give rise to different interpretations for α. Not all such interpretations have practical relevance.

- These coefficients are determined implicitly by

$$\hat{\theta} = (\hat{\alpha}, \hat{\beta}) = \underset{(\alpha,\beta)\in\mathbb{R}^2}{\operatorname{argmin}} \sum_{u\in\mathcal{P}} (y_u - \alpha - \beta(x_u - c))^2$$

- It can be shown that

$$\hat{\alpha} = \bar{y} - \hat{\beta}(\bar{x} - c) \quad \text{and} \quad \hat{\beta} = \frac{\sum_{u\in\mathcal{P}}(x_u - \bar{x})(y_u - \bar{y})}{\sum_{u\in\mathcal{P}}(x_u - \bar{x})^2}$$

- The resulting estimates determine the **least-squares fitted line**:

$$y = \hat{\alpha} + \hat{\beta}(x - c)$$

- The equation of the fitted values, defined for all $u \in \mathcal{P}$, is:

$$\hat{y}_u = \hat{\alpha} + \hat{\beta}(x_u - c)$$

- The residuals are

$$\hat{r}_u = y_u - \hat{\alpha} - \hat{\beta}(x_u - c)$$

Each residual is the signed vertical distance between the point (x_u, y_u) and the point $(x_u, \hat{y}_u) = (x_u, \hat{\alpha} + \hat{\beta}(x_u - c))$. The latter point is the value of the fitted line, defined by $\hat{\theta} = (\hat{\alpha}, \hat{\beta})$, calculated at $x = x_u$.

2.3.2 Dealing with Influential Units in Linear Regression

When there are some units that are quite different than others, we can either

- do nothing (not good)
- remove the units (not good)
- assign weights to the observations according to their variation
- use a method to find the regression line which is *robust* to potential outliers.

Rather than removing the problematic units, we can consider giving these units less weight.

Definition 2.13
Weighted Least Squares

In Weighted Least Squares (WLS), the fitted line minimizes the following objective function

$$(\hat{\alpha}, \hat{\beta}) = \arg \min_{(\alpha,\beta)\in\mathbb{R}^2} \sum_{u\in P} w_u[y_u - \alpha - \beta(x_u - c)]^2$$

It is assumed the weights w_u are known but, as we will see, the residuals from an ordinary LS regression model can help us determine sensible values.

As in ordinary LS regression a choice for c needs to be made. In this setting it is common to either set $c = 0$ or define c to be the weighted average of the x_u values:

$$c = \bar{x}_w = \frac{\sum_{u\in P} w_u x_u}{\sum_{u\in P} w_u}$$

Given the values of the w_u's and c, we determine $\hat{\Theta} = (\hat{\alpha}, \hat{\beta})$ by taking derivatives of $\rho(\Theta; P)$ with respect to each parameter and then setting the resulting gradient equal to zero and solving the system of equations.

$$\sum_{u\in P} w_u \begin{bmatrix} 1 \\ x_u - c \end{bmatrix} [y_u - \alpha - \beta(x_u - c)] = 0$$

Doing so yields the following estimates (show this):

$$\hat{\alpha} = \bar{y}_w - \hat{\beta}(\bar{x}_w - c)$$

and

$$\hat{\beta} = \frac{\sum_{u\in P} w_u(x_u - \bar{x}_w)(y_u - \bar{y}_w)}{\sum_{u\in P} w_u(x_u - \bar{x}_w)^2}$$

where $\bar{y}_w = \frac{\sum_{u\in P} w_u y_u}{\sum_{u\in P} w_u}$ and $\bar{x}_w$ are respectively the weighted averages of the y and x values.

Both of the procedures for dealing with outliers discussed so far (deletion and re-weighting) have been very manual. It would be nice to have a more automatic procedure to do this.

Definition 2.14
Robust Regression

Robust regression has the same goal. That is, points that are far from the linear line should have less weight in the objective function. We can modify the least square objective function to be

$$(\hat{\alpha}, \hat{\beta}) = \arg \min_{(\alpha,\beta)\in\mathbb{R}^2} \sum_{u\in P} \rho(y_u - \alpha - \beta(x_u - c))$$

where different forms of the function $\rho(\cdot)$ give rise to different fitted lines:

- Least Squares Regression: equal weight on every single unit

$$\rho(y_u - \alpha - \beta(x_u - c)) = [y_u - \alpha - \beta(x_u - c)]^2$$

- Weighted Least Squares Line: give less weight to units with LS residuals that are large (in magnitude)

$$\rho(y_u - \alpha - \beta(x_u - c)) = w_u[y_u - \alpha - \beta(x_u - c)]^2$$

In robust regression we modify the loss function $\rho(y_u - \alpha - \beta(x_u - c))$ so that

- it gives lower weight than least squares to units with large residuals (i.e., u such that $|r_u| \gg 0$),
- and that it is quadratic near 0 and hence behaves similarly to LS for units with small residuals (i.e., u such that $|r_u| \approx 0$).

The Huber Loss Function achieves these goals by combining the quadratic and absolute value functions:

$$\rho_k(r) = \begin{cases} \frac{1}{2}r^2 & \text{for } |r| \le k \\ k\left(|r| - \frac{1}{2}k\right) & \text{for } |r| > k \end{cases}$$

Note: r or r_u here means $y_u - \alpha - \beta(x_u - c)$

An attribute (e.g., regression coefficients) based on this function will be affected by the scale of r, and so...

- we might let $k = cS$ where S is a (possibly robust) measure of scale.

In practice it is common

- to satisfy a theoretical balance between efficiency and resistance to outliers, we set $k \approx 1.345S$.
- Other common choices include $k = 1.5$ or 2.

Note: as k increases, the robust regression with Huber function imposes a larger penalty on larger residuals, hence approaches the least squares fit ($k = \infty$).

Another form of robust regression involves defining the loss function in terms of **least absolute deviations (LAD)**:

$$(\hat{\alpha}, \hat{\beta}) = \arg \min_{(\alpha,\beta)\in\mathbb{R}^2} \sum_{u\in P} |y_u - \alpha - \beta(x_u - c)|$$

where $r_u = y_u - \alpha - \beta(x_u - c)$

However, in both Huber and LAD-based regression the attribute $(\hat{\alpha}, \hat{\beta})$ cannot be solved for in closed form, which means that there is no explicit algebraic expression or formula for directly calculating the optimal values of the parameters $(\hat{\alpha}, \hat{\beta})$.

When the attribute of interest is

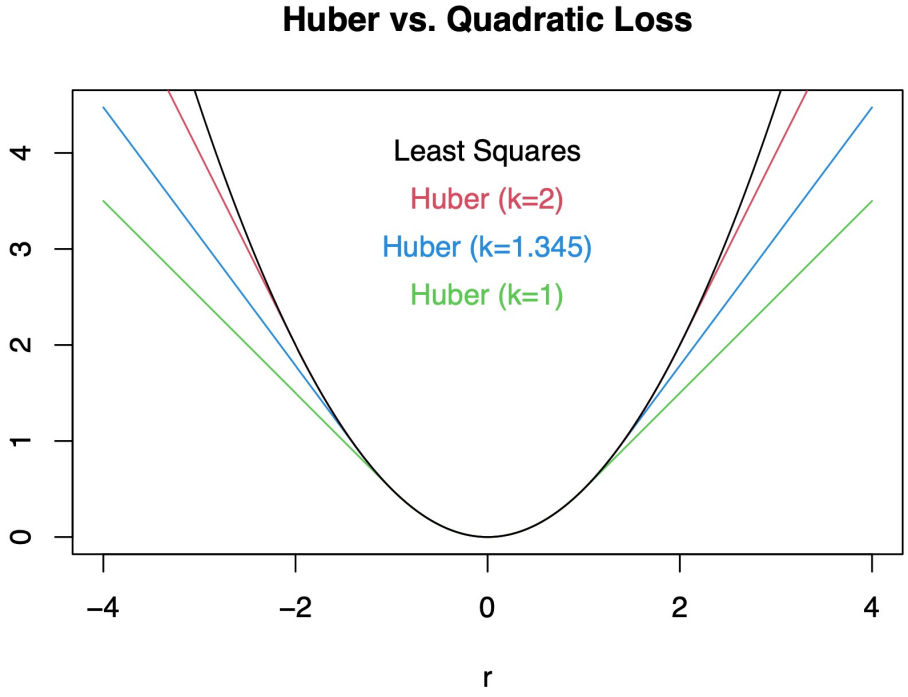
$$(\hat{\alpha}, \hat{\beta}) = \arg \min_{(\alpha,\beta)\in\mathbb{R}^2} \sum_{u\in P} \rho(y_u - \alpha - \beta(x_u - c))$$

but the definition of $\rho(\cdot)$ precludes straightforward calculation, we consider the following optimization methods:

- Gradient descent
- Newton-Raphson
- Iteratively reweighted least-squares

The algorithms above are employed very generally to handle attributes defined implicitly as

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}\in\Theta} \rho(\boldsymbol{\theta}; \mathcal{P})$$



2.3.3 Gradient Descent

Direction and Step Size

If $\rho(\theta; \mathcal{P})$ is a differentiable function of $\theta = (\theta_1, \theta_2, \ldots, \theta_k) \in \mathbb{R}^k$ then we can calculate the gradient for any value of θ:

$$g = g(\theta) = \nabla\rho(\theta; \mathcal{P}) = \begin{bmatrix} \frac{\partial\rho(\theta;\mathcal{P})}{\partial\theta_1} \\ \frac{\partial\rho(\theta;\mathcal{P})}{\partial\theta_2} \\ \vdots \\ \frac{\partial\rho(\theta;\mathcal{P})}{\partial\theta_k} \end{bmatrix}$$

- Note that we will typically distinguish among the gradient calculations at each iteration.

- At iteration i, when $\hat{\theta}_i$ is our best guess at the solution, we denote the gradient by $g_i = g(\hat{\theta}_i)$.

By definition, the normalized gradient

$$d_i = \frac{g_i}{\|g_i\|}$$

provides the direction in which $\rho(\theta; \mathcal{P})$ increases or decreases fastest. (Recall that $\|x\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_k^2}$ for $x \in \mathbb{R}^k$). In particular:

- d_i indicates the direction of steepest ascent, and
- $-d_i$ indicates the direction of steepest descent.

We iterate and obtain a new estimate of θ by

- moving in the direction of $-d_i$ and
- taking a step of size $\lambda_i > 0$

$$\hat{\theta}_{i+1} = \hat{\theta}_i - \lambda_i d_i.$$

Note that the step size λ at each iteration can be chosen in a variety of ways:

1. We could choose a fixed value for all i such as $\lambda_i = 0.1$
2. We could define a fixed sequence such as $\lambda_i = 0.1 + \frac{1}{i}$
3. We could perform a line search and algorithmically choose the value of λ_i that minimizes

$$\rho(\hat{\theta}_i - \lambda_i d_i)$$

In other words, which step size in the direction $-d_i$, away from $\hat{\theta}_i$, minimizes $\rho(\hat{\theta}_{i+1}; \mathcal{P})$.

The Gradient Descent Algorithm

Given some initial value $\hat{\theta}_0$

1. Initialize $i; \hat{\theta}_0$;
2. LOOP:
   (a) Gradient:
   $$g_i = \nabla \rho(\theta; \mathcal{P})\Big|_{\theta=\hat{\theta}_i}$$
   (b) Gradient direction:
   $$d_i \leftarrow \frac{g_i}{\|g_i\|}$$
   (c) Line search: Find the step size $\hat{\lambda}_i$
   $$\hat{\lambda}_i = \arg\min_{\lambda>0} \rho(\hat{\theta}_i - \lambda d_i)$$

(d) Update the iterate:

$$\hat{\theta}_{i+1} \leftarrow \hat{\theta}_i - \hat{\lambda}_i d_i$$

(e) Converged?

- if the iterates are not changing, then Return
- else $i \leftarrow i + 1$ and repeat LOOP.

3. Return: $\hat{\theta} = \hat{\theta}_i$;

We stop when two iterates are sufficiently close to one another, where "sufficiently" depends on a tolerance ε. That is,

$$\|\hat{\theta}_{i+1} - \hat{\theta}_i\|_1 < \varepsilon$$

where $\|\cdot\|_1$ is the L_1 norm defined by

$$\|z\|_1 = \sum_{j=1}^{k} |z_j| \text{ where } z \text{ is a vector with dimension } k.$$

We could also measure this on a relative scale:

$$\frac{\|\hat{\theta}_{i+1} - \hat{\theta}_i\|_1}{\|\hat{\theta}_i\|_1} < \varepsilon$$

Remark: We could change the L_1 norm to any other distance metric (such as L_2 norm).

Functions containing their own data environment are called closures.

- Every function has a local environment where variables may be defined; this is the closure of the function.
- Functions also have access to the environment in which they were created (that's why functions can access values in the global environment).

Encapsulation of data within a function is an important and powerful construct. Yes, like the concept of encapsulation in OOP!

Factory Functions: Functions that make Functions

Here is a simple example of a function that defines and returns a quadratic function.

```
createQuadratic <- function(a, b, c) {
  ## Return this function
  function(x) {
    fx = a * x^2 + b * x + c
    return(fx)
  }
}

## our function-creating-function in action
```

```
x = seq(-3, 3, length.out = 100)
f1 = createQuadratic(a = 1, b = 1, c = 1)
f2 = createQuadratic(a = -2, b = 1, c = 5) f3 = createQuadratic(a = -2, b = -2, c
    = 10)
plot(x, f1(x), type = "l", ylab = "f(x)") lines(x, f2(x), col = 2)
lines(x, f3(x), col = 3)
legend("topleft", lty = 1, col = 1:3, legend = c("f1", "f2", "f3"), bty = "n")
```

2.3.4 Gradient Descent in Batches

Batch Gradient Descent

In practice, many of the objective functions minimized during statistical analyses have the following form:

$$\rho(\theta, \mathcal{P}) = \sum_{u \in \mathcal{P}} \rho(\theta; u)$$

in which case the gradient g can simply be written as the sum of the unit-specific contributions to the objective function:

$$g = g(\theta) = \nabla\rho(\theta; \mathcal{P}) = \sum_{u \in \mathcal{P}} \nabla\rho(\theta; u) = \sum_{u \in \mathcal{P}} g(\theta; u)$$

Thus when $\rho(\cdot)$ is a sum over $u \in \mathcal{P}$:

- the gradient g is composed of N 'smaller' independent gradient calculations
- these individual gradient calculations $g(\theta; u)$ can be done in any order
  - this can be very handy when N is large and the individual gradients are expensive to calculate
  - we may wish to perform the gradient computations in several batches which are distributed across different machines
  - this is important in many "Big Data" applications

In this situation the terms *batch gradient descent* and *gradient descent* are used interchangeably.

- The appropriateness of the term *batch* becomes clear when we explicitly partition the population $\mathcal{P}$ into H non-overlapping groups (batches)

  $$\mathcal{P} = \mathcal{B}_1 \cup \mathcal{B}_2 \cup \ldots \cup \mathcal{B}_H$$

  each containing M_k units ($k = 1, 2, \ldots, H$):

  $$g = \sum_{u \in \mathcal{P}} g(\theta; u) = \sum_{k=1}^{H} \sum_{u \in \mathcal{B}_k} g(\theta; u)$$

Batch gradient descent lends itself well to parallel computation. But what if the gradient calculations are sufficiently complex and even parallelization isn't fast enough?

Gradient Descent Using Subsets of the Population

When computing the gradient g is computationally expensive we could consider using only a subset of the available data – as opposed to using all of it.

- If the run time for batch gradient descent based on all N units is too long, consider *estimating* the gradient using just $M < N$ units.
- In such situations we typically do not optimize for the step size λ and instead use a fixed step size λ^* (Why?)
  - The gradients are just an approximation, so we don't optimize for step size in a potentially wrong direction
  - λ^* is often referred to as the learning rate.
  - Consequently, we will always use `relative = TRUE` in our test of convergence.
- Two common approaches to do this are batch-sequential and batch-stochastic gradient descent
  - *these approaches differ only in the manner in which the subsets of size M are chosen.*

Batch-Sequential Gradient Descent

- Suppose we can divide the population of size N to H batches of size M, i.e., $N = H \times M$ and $\mathcal{P} = \{\mathcal{B}_1, \ldots, \mathcal{B}_H\}$
- In this approach we sequentially move through the H batches and update our estimate $\hat{\theta}$ after each batch.
  - Note that this is different from ordinary batch gradient descent; in that case the gradients are still calculated in batches, but the $\hat{\theta}$ is only updated after observing all batches.
- If convergence takes more than H iterations then the batches are iteratively sequenced through until convergence.

The batch-sequential gradient descent algorithm is the following:

Given some initial values $\hat{\theta}_0$ and a fix step size λ^*

1. Initialize; $i \leftarrow 0$;
2. LOOP:
   (a) Gradient:
   $$g_i = \nabla\rho(\theta; \mathcal{B}_{i \mod H})|_{\theta=\hat{\theta}_i}$$
   (b) Gradient direction:
   $$d_i \leftarrow \frac{g_i}{\|g_i\|}$$
   (c) Update the iterate:
   $$\hat{\theta}_{i+1} \leftarrow \hat{\theta}_i - \lambda^* d_i$$
   (d) Converged?

- if the iterates are not changing, then Return
- else $i \leftarrow i + 1$ and repeat LOOP.
- (using relative = TRUE for test of convergence)

3. Return: $\hat{\theta} = \hat{\theta}_i$;

For example, when $i = H + 1$, the batch is actually $\mathcal{B}_1$, and $\mathcal{B}_0 \equiv \mathcal{B}_H$.

Batch-Stochastic Gradient Descent

In this approach, each iteration of the gradient is calculated from a sample (batch) $\mathcal{S}$ selected randomly from the population P.

- Like batch-sequential gradient descent, the estimate $\hat{\theta}$ is updated after each batch (sample). Denoting the sample size by M, note that setting $M = 1$ gives rise to what is often simply referred to as stochastic gradient descent.

The batch-stochastic gradient descent algorithm is the following:

Given some initial values $\hat{\theta}_0$ and a fix step size λ^*

1. Initialize; $i \leftarrow 0$;
2. LOOP:
   (a) Gradient: Given a new random sample $\mathcal{S} \in P$

$$g_i = \nabla \rho(\theta; \mathcal{S})|_{\theta=\hat{\theta}_i}$$

   (b) Gradient direction:

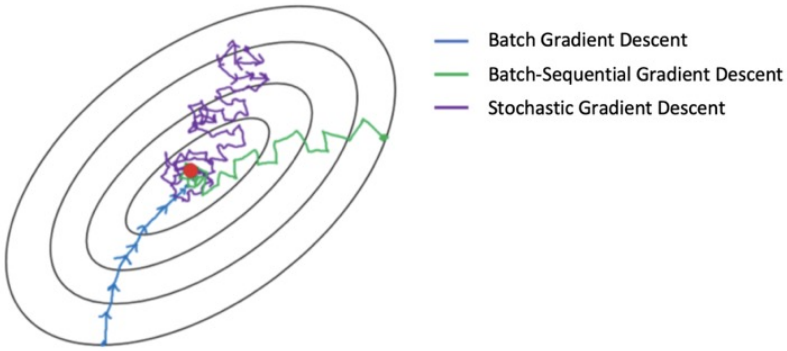
$$d_i \leftarrow \frac{g_i}{\|g_i\|}$$

   (c) Update the iterate:

$$\hat{\theta}_{i+1} \leftarrow \hat{\theta}_i - \lambda^* d_i$$

   (d) Converged?
   - if the iterates are not changing, then Return
   - else $i \leftarrow i + 1$ and repeat LOOP.
   - (using relative)
3. Return: $\hat{\theta} = \hat{\theta}_i$;

Comparing the Algorithms



Here, Batch Gradient Descent also refers to the Ordinary Gradient Descent, and the Stochastic Gradient Descent is when $M = 1$. This plot communicates efficiency in number of steps, NOT in terms of time. Batch-Sequential and Batch-Stochastic GD may require more steps but ultimately be faster because each step takes less time.

In conclusion, in batch-sequential gradient descent, we divide in the population into H batches. Then in each iteration perform one step of gradient descent using one of the H batches. Each iteration uses a different batch and we move through the population, batch-by-batch sequentially. In batch-stochastic gradient descent, at each iteration we randomly sample a batch and then perform one step of gradient descent using that batch (sample).

Remark 2.1
If you are confused among Gradient Descent, Batch Gradient Descent, Batch-Sequence Gradient Descent, Batch Stochastic Gradient Descent and Stochastic Gradient Descent,

- As long as the objective function ρ can be written in the form of summation over units, then essentially the Batch Gradient Desent is the Gradient Descent. The reason for singling it out is that, since we compute the gradient between each unit without affecting each other, we can do parallel computation.
- Unlike the normal Batch Gradient Descent, in the Batch-Sequential Gradient Descent approach we update $\hat{\theta}$ once after each batch.
- We fix step size λ^* in Batch-Sequence Gradient Descent and Batch Stochastic Gradient Descent instead of line search (Why? Simplifies the computation, and because not all units are considered at the same time, the results obtained by line search can be imprecise).
- If each unit is a separate batch (batch size=1), then each iteration is equivalent to randomly picking a point from the population, and this particular algorithm can be called Stochastic Gradient Descent.

2.3.5 Systems of Equations

We have defined an implicit attribute θ as the solution to an optimization problem. Until now we have cast such an optimization problem as the minimization of some appropriately defined objective function $\rho(\theta; P)$:

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \rho(\theta; P)$$

In all situations we saw that such a solution was obtained by solving (either algorithmically or in closed form) the equation

$$\nabla \rho(\theta; P) = 0$$

When the dimension of the vector valued attribute θ is k, such an equation is actually a system of k independent equations with k unknowns. Thus an implicit attribute $\theta \in \Theta$ can be defined slightly more generally as the solution to a system of equations

$$\psi(\theta; P) = 0$$

Thus we work directly with $\psi(\theta; P)$ which, in most cases, equals $\nabla \rho(\theta; P)$.

Unsurprisingly, $\psi(\theta; P)$ is often defined as a sum over $u \in P$:

$$\psi(\theta; P) = \sum_{u \in P} \psi(\theta; u)$$

in which case the attribute of interest $\hat{\theta}$ is the value $\theta \in \Theta$ that solves

$$\sum_{u \in P} \psi(\theta; u) = 0$$

Example 2.9
Scalar valued attributes

The average is the value of θ which solves

$$\psi(\theta; \mathcal{P}) = \sum_{u \in \mathcal{P}} \psi(\theta; u) = \sum_{u \in \mathcal{P}} y_u - n\theta = 0$$

The weighted average is the value of θ which solves

$$\psi(\theta; \mathcal{P}) = \sum_{u \in \mathcal{P}} \psi(\theta; u) = \sum_{u \in \mathcal{P}} w_u(y_u - \theta) = 0$$

The q^{th} quantile is the smallest value of θ which solves

$$\psi(\theta; \mathcal{P}) = \sum_{u \in \mathcal{P}} \frac{1}{N} I(y_u \leq \theta) - q = 0$$

Vector valued attributes

Least squares

$$\psi(\theta; \mathcal{P}) = \sum_{u \in \mathcal{P}} \psi(\theta; u) = \sum_{u \in \mathcal{P}} (y_u - \alpha - \beta(x_u - c)) \left(\frac{1}{x_u - c} \right) = 0$$

Weighted least squares

$$\psi(\theta; \mathcal{P}) = \sum_{u \in \mathcal{P}} \psi(\theta; u) = \sum_{u \in \mathcal{P}} w_u (y_u - \alpha - \beta(x_u - c)) \left(\frac{1}{x_u - c} \right) = 0$$

Robust regression

$$\psi(\theta; \mathcal{P}) = \sum_{u \in \mathcal{P}} \psi(\theta; u) = \sum_{u \in \mathcal{P}} \rho'_k (y_u - \alpha - \beta(x_u - c)) \left(\frac{1}{x_u - c} \right) = 0$$

The class of methods used to find such solutions to systems of equations are generally referred to as **root finding** methods.

Definition 2.15
Newton's Method

Suppose we have a *differentiable function* $f(x)$ and we wish to find $x = x^*$ which solves

$$f(x) = 0$$

Given an initial value x_0

- we can use a linear function to approximate $f(x)$ in the vicinity of x_0

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0)$$

- then we find the root of the linear approximation to iterate to the next value of x:

$$0 = f(x_0) + f'(x_0)(x - x_0) \Rightarrow x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

- We continue in this way until x_{i+1} does not differ much from x_i

For our implicitly defined scalar attribute θ, we want to find $\theta \in \Theta$ such that

$$\psi(\theta; \mathcal{P}) = 0$$

Given the current guess $\hat{\theta} = \hat{\theta}_i$, a first order approximation is

$$\psi(\theta; \mathcal{P}) \approx \psi(\hat{\theta}_i; \mathcal{P}) + \psi'(\hat{\theta}_i; \mathcal{P}) \times (\theta - \hat{\theta}_i).$$

Then the update is the root of the linear approximation and is given by

$$\hat{\theta}_{i+1} = \hat{\theta}_i - \frac{\psi(\hat{\theta}_i; \mathcal{P})}{\psi'(\hat{\theta}_i; \mathcal{P})}$$

The Newton Algorithm

Given an initial value $\hat{\theta}_0$

1. Initialize: $i \leftarrow 0$;
2. **LOOP**:
   (a) Update the iterate:
   $$\hat{\theta}_{i+1} \leftarrow \hat{\theta}_i - \frac{\psi(\hat{\theta}_i; \mathcal{P})}{\psi'(\hat{\theta}_i; \mathcal{P})}$$
   (b) Converged?
   if the iterates are not changing, then return
   else $i \leftarrow i + 1$ and repeat **LOOP**.
3. Return: $\hat{\theta} = \hat{\theta}_i$;

R Code for Newton's Method

```
Newton <- function(theta = 0,
                   psiFn, psiPrimeFn,
                   testConvergenceFn = testConvergence,
                   maxIterations = 100,    # maximum number of iterations
                   tolerance = 1E-6,       # parameters for the test
                   relative = FALSE        # for convergence function
) {
  ## Initialize
  converged <- FALSE
  i <- 0
  ## LOOP
  while (!converged & i <= maxIterations) {
    ## Update theta
    thetaNew <- theta - psiFn(theta)/psiPrimeFn(theta)
    ##
```

```
    ## Check convergence
    converged <- testConvergenceFn(thetaNew, theta,
                                   tolerance = tolerance,
                                   relative = relative)
    ## Update iteration
    theta <- thetaNew
    i <- i + 1
  }
  ## Return last value and whether converged or not
  list(theta = theta,
       converged = converged,
       iteration = i,
       fnValue = psiFn(theta)
       )
}

testConvergence <- function(thetaNew, thetaOld, tolerance = 1e-10, relative =
    FALSE) {
    sum(abs(thetaNew - thetaOld)) < if (relative)
        tolerance * sum(abs(thetaOld)) else tolerance
}
```

2.3.6 The Newton-Raphson Method

Notation Note: In mathematical notation, especially within vectors and matrices, the prime symbol $'$ denotes the transpose of a vector or a matrix.

Definition 2.16

When $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_k)'$ is a vector-valued attribute, the multivariate analog to Newton's Method is referred to as the Newton-Raphson Method. In this case we want to determine the vector which solves

$$\psi(\boldsymbol{\theta}; \mathcal{P}) = \mathbf{0}$$

Since $\psi(\boldsymbol{\theta}; \mathcal{P})$ is a differentiable $k \times 1$ vector $(\psi_1, \psi_2, \ldots, \psi_k)'$ we let

$$\psi'(\boldsymbol{\theta}; \mathcal{P}) = \frac{\partial \psi(\boldsymbol{\theta}; \mathcal{P})}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial \psi_1}{\partial \theta_1} & \frac{\partial \psi_1}{\partial \theta_2} & \cdots & \frac{\partial \psi_1}{\partial \theta_k} \\ \frac{\partial \psi_2}{\partial \theta_1} & \frac{\partial \psi_2}{\partial \theta_2} & \cdots & \frac{\partial \psi_2}{\partial \theta_k} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \psi_k}{\partial \theta_1} & \frac{\partial \psi_k}{\partial \theta_2} & \cdots & \frac{\partial \psi_k}{\partial \theta_k} \end{bmatrix}$$

be the $k \times k$ matrix of partial derivatives. This matrix is called the *Jacobian matrix* of ψ.

Given a current guess $\hat{\boldsymbol{\theta}}$, we can use a linear function to approximate the function $\psi(\boldsymbol{\theta}; \mathcal{P})$

- A first-order approximation of $\psi(\boldsymbol{\theta}; \mathcal{P})$ in the vicinity of $\hat{\boldsymbol{\theta}}$ can be written as

$$\psi(\boldsymbol{\theta}; \mathcal{P}) \approx \psi(\hat{\boldsymbol{\theta}}; \mathcal{P}) + \psi'(\hat{\boldsymbol{\theta}}; \mathcal{P}) \times (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})$$

- Then the vector at which the linear approximation is equal to zero is

$$\boldsymbol{\theta} \approx \hat{\boldsymbol{\theta}} - [\psi'(\hat{\boldsymbol{\theta}}; \mathcal{P})]^{-1} \psi(\hat{\boldsymbol{\theta}}; \mathcal{P})$$

which suggests the iterative Newton-Raphson algorithm for root finding.

Notice the differences!

- Newton's method is for single-variable functions, while the Newton-Raphson method can be extended to multivariate functions.
- Newton's method uses the derivative of the function, while the multivariate Newton-Raphson method uses the Jacobian matrix of partial derivatives.
- The update step in the multivariate case involves matrix inversion and multiplication, which is more complex than the single-variable derivative division.

The Newton-Raphson Algorithm

Given an initial value $\hat{\boldsymbol{\theta}}_{\mathbf{0}}$

1. Initialize: $i \leftarrow 0$;
2. **LOOP**:
   (a) Update the iterate:
   $$\hat{\boldsymbol{\theta}}_{\boldsymbol{i+1}} \leftarrow \hat{\boldsymbol{\theta}}_{\boldsymbol{i}} - [\boldsymbol{\psi'}(\hat{\boldsymbol{\theta}}_{\boldsymbol{i}}; \boldsymbol{\mathcal{P}})]^{-1} \boldsymbol{\psi}(\hat{\boldsymbol{\theta}}_{\boldsymbol{i}}; \boldsymbol{\mathcal{P}})$$
   (b) Converged?
   if the iterates are not changing, then return
   else $i \leftarrow i + 1$ and repeat **LOOP**.
3. Return: $\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}_{\boldsymbol{i}}$;

R Code for the Newton-Raphson Algorithm

```
NewtonRaphson <- function(theta, psiFn, psiPrimeFn, dim, testConvergenceFn =
    testConvergence, maxIterations = 100, tolerance = 1e-06, relative = FALSE) {
    if (missing(theta)) {
        ## need to figure out the dimensionality
        if (missing(dim)) {
            dim <- length(psiFn())
        }
        theta <- rep(0, dim)
    }
```

```
    converged <- FALSE
    i <- 0
    while (!converged & i <= maxIterations) {
        thetaNew <- theta - solve(psiPrimeFn(theta), psiFn(theta))
        converged <- testConvergenceFn(thetaNew, theta, tolerance = tolerance,
            relative = relative)

        theta <- thetaNew
        i <- i + 1
    }
    ## Return last value and whether converged or not
    list(theta = theta, converged = converged, iteration = i, fnValue = psiFn(
    theta))
}
```

Remark 2.2
Connection between Newton-Raphson and Gradient Descent

- Notice that when the objective function of interest is $\rho(\boldsymbol{\theta}; \mathcal{P})$ then $\psi(\boldsymbol{\theta}; \mathcal{P}) = \nabla\rho(\boldsymbol{\theta}; \mathcal{P}) = \mathbf{g}$ and the system of equations

$$\boldsymbol{\psi}(\boldsymbol{\theta}; \mathcal{P}) = 0$$

is equivalent to

$$\nabla\rho(\boldsymbol{\theta}; \mathcal{P}) = 0$$

- Also notice that the updating equation associated with Newton-Raphson is given by

$$\hat{\boldsymbol{\theta}}_{\boldsymbol{i+1}} = \hat{\boldsymbol{\theta}}_{\boldsymbol{i}} - [\boldsymbol{\psi}'(\hat{\boldsymbol{\theta}}_{\boldsymbol{i}}; \boldsymbol{\mathcal{P}})]^{-1}\boldsymbol{\psi}(\hat{\boldsymbol{\theta}}_{\boldsymbol{i}}; \boldsymbol{\mathcal{P}})$$

which, in light of the previous point, can be rewritten as

$$\hat{\boldsymbol{\theta}}_{\boldsymbol{i+1}} = \hat{\boldsymbol{\theta}}_{\boldsymbol{i}} - \mathbf{H}_i^{-1}\mathbf{g}_i$$

where

$$\mathbf{H}_i = \begin{bmatrix} \frac{\partial^2\rho}{\partial\theta_1^2} & \frac{\partial^2\rho}{\partial\theta_1\partial\theta_2} & \cdots & \frac{\partial^2\rho}{\partial\theta_1\partial\theta_k} \\ \frac{\partial^2\rho}{\partial\theta_2\partial\theta_1} & \frac{\partial^2\rho}{\partial\theta_2^2} & \cdots & \frac{\partial^2\rho}{\partial\theta_2\partial\theta_k} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2\rho}{\partial\theta_k\partial\theta_1} & \frac{\partial^2\rho}{\partial\theta_k\partial\theta_2} & \cdots & \frac{\partial^2\rho}{\partial\theta_k^2} \end{bmatrix}\Bigg|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{\boldsymbol{i}}}$$

is the *Hessian* of ρ (i.e., the matrix of second partial derivatives of ρ with respect to $\theta_1, \theta_2, \ldots, \theta_k$)

- Recall that the updating equation associated with gradient descent is given by

$$\hat{\boldsymbol{\theta}}_{\boldsymbol{i+1}} = \hat{\boldsymbol{\theta}}_{\boldsymbol{i}} - \lambda_i\mathbf{d}_i$$

$$= \hat{\boldsymbol{\theta}}_{\boldsymbol{i}} - \frac{\lambda_i}{\|\mathbf{g}_i\|}\mathbf{g}_i$$

Thus Newton-Raphson applied to the gradient of an objective function is essentially equivalent to gradient descent applied directly to that objective function but with step sizes modulated by the Hessian of the objective function.

2.3.7 Iteratively Reweighted Least Squares

Definition 2.17

When $\boldsymbol{\theta} = (\alpha, \beta)'$ is a vector-valued attribute associated with a linear regression, *iteratively reweighted least squares (IRLS)* provides a convenient iterative method for finding $\hat{\boldsymbol{\theta}} = (\hat{\alpha}, \hat{\beta})'$, the solution to

$$\arg\min_{\boldsymbol{\theta}\in\Theta} \sum_{u\in\mathcal{P}} \rho(y_u - \alpha - \beta(x_u - c))$$

Differentiating this with respect to α and β and setting the result equal to zero yields

$$\sum_{u\in\mathcal{P}} \rho'(y_u - \alpha - \beta(x_u - c)) \begin{pmatrix} 1 \\ (x_u - c) \end{pmatrix} = \mathbf{0}$$

and $\hat{\boldsymbol{\theta}} = (\hat{\alpha}, \hat{\beta})'$ is found by solving this system of equations.

If we let $r_u = y_u - \alpha - \beta(x_u - c)$ and $\boldsymbol{z}_u = (1, x_u - c)'$, then the system above becomes

$$\sum_{u\in\mathcal{P}} \rho'(r_u)\boldsymbol{z}_u = \mathbf{0}$$

Note that in WLS $\rho(r_u) = w_u r_u^2$ and $\rho'(r_u) = 2w_u r_u$ and so the system reduces to

$$\sum_{u\in\mathcal{P}} w_u r_u \boldsymbol{z}_u = \mathbf{0}$$

With the IRLS algorithm, we will exploit the fact that the general objective function minimization problem can be recast as a weighted least squares problem.

The general equation can be made to look like the WLS equation as follows:

$$\begin{aligned} \mathbf{0} &= \sum_{u\in\mathcal{P}} \rho'(r_u)\boldsymbol{z}_u \\ &= \sum_{u\in\mathcal{P}} \frac{\rho'(r_u)}{r_u} r_u \boldsymbol{z}_u \\ &= \sum_{u\in\mathcal{P}} w_u r_u \boldsymbol{z}_u \end{aligned}$$

where $w_u = \frac{\rho'(r_u)}{r_u}$ is the weight for unit u provided $r_u \neq 0$.

Why did we do this? Because WLS has a closed-form solution.

If we had some initial value for the residuals r_u (and hence the weights w_u) we could solve this equation in closed form yielding a value for $\boldsymbol{\theta}$, call it $\hat{\boldsymbol{\theta}}_1 = (\hat{\alpha}_1, \hat{\beta}_1)'$

- This requires initial values for α and β
- Since $\boldsymbol{\theta}_0 = (\alpha_0, \beta_0)$ is likely far from $\hat{\boldsymbol{\theta}} = (\hat{\alpha}, \hat{\beta})'$ we should iterate and update our values of the residuals (and hence weights) with $\boldsymbol{\theta}_1 = (\hat{\alpha}_1, \hat{\beta}_1)'$
- This process can be repeated until the estimates of α and β converge
- This process is called *iteratively reweighted least squares*

We can generalize this algorithm by expressing the problem in terms of the vector-valued attribute $\boldsymbol{\theta} = (\alpha, \beta)'$: $r_u = y_u - \boldsymbol{z}_u'\boldsymbol{\theta}$.

The Algorithm

Given an initial value $\boldsymbol{\theta}_0$

1. Initialize: $i \leftarrow 0$;
2. **LOOP**:
   (a) Construct residuals and weights for all $u \in \mathcal{P}$

$$r_u = y_u - \mathbf{z}_u'\boldsymbol{\theta}_i \quad \text{and} \quad w_u = \frac{\rho'(r_u)}{r_u}$$

   (b) Solve the weighted least squares problem, i.e., find $\hat{\boldsymbol{\theta}}$, the value of θ such that:

$$\sum_{u \in \mathcal{P}} w_u (y_u - \mathbf{z}_u'\boldsymbol{\theta})\mathbf{z}_u = \mathbf{0}$$

   (c) Update the parameter:

$$\hat{\boldsymbol{\theta}}_{i+1} \leftarrow \hat{\boldsymbol{\theta}}$$

   (d) Converged?

   if the iterates are not changing, then return

   else $i \leftarrow i + 1$ and repeat **LOOP**.
3. Return: $\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}_i$;

The algorithm above works for any vector-valued attribute $\boldsymbol{\theta}$ that arises in the context of a linear response model:

$$y_u = \mathbf{z}_u'\boldsymbol{\theta} + r_u$$

```
irls <- function(y, x, theta, rhoPrimeFn,
                 dim = 2, delta = 1E-10,
                 testConvergenceFn = testConvergence,
```

```
                  maxIterations = 100,    # maximum number of iterations
                  tolerance = 1E-6,       # parameters for the test
                  relative = FALSE        # for convergence function
){if (missing(theta)) {theta <- rep(0, dim)}
  ## Initialize
  converged <- FALSE
  i <- 0
  N <- length(y)
  wt <- rep(1,N)
  ## LOOP
  while (!converged & i <= maxIterations) {
    ## get residuals
    resids <- getResids(y, x, wt, theta)
    ## update weights  (should check for zero resids)
    wt <- getWeights(resids, rhoPrimeFn, delta)
    ## solve the least squares problem
    thetaNew <- getTheta(y, x, wt)
    ##
    ## Check convergence
    converged <- testConvergenceFn(thetaNew, theta,
                                   tolerance = tolerance,
                                   relative = relative)
    ## Update iteration
    theta <- thetaNew
    i <- i + 1
  }
  ## Return last value and whether converged or not
  list(theta = theta, converged = converged, iteration = i)
}
```

CHAPTER 3: Samples

3.1 Samples

If we have a *sample* or a subset $\mathcal{S}$ of $n \ll N$ units,

- Then the attribute $a(\mathcal{S})$ calculated based on this sample is an *estimate* of its population counterpart $a(\mathcal{P})$.

$$a(S) = \widehat{a(\mathcal{P})} = a(\hat{\mathcal{P}})$$

- The second equality emphasizes that $\mathcal{S}$ is an estimate of $\mathcal{P}$.

When using a sample instead of the entire population, we might consider

- sample error, and
- Fisher consistency.

Sample error

- Any difference between the actual values of the estimate $a(\mathcal{S})$ and the quantity being estimated (the estimand) $a(\mathcal{P})$ is an error.

$$\text{sample error} = a(\mathcal{S}) - a(\mathcal{P})$$

That is, the difference between the estimated and true values.

- The nature of this error will depend on the sample and the attribute.
- Quantifying error;
  - for numerical attributes, this is determined mathematically;
  - for graphical attributes, it is not precise though still conceptually applicable.

For obvious reasons, an attribute with lower sampling error is preferable.

Fisher Consistency

- If the sample $\mathcal{S}$ is equal to the population $\mathcal{P}$ then the sample error should be zero (or non-existent), i.e. $a(\mathcal{P}) = a(\mathcal{S})$.

  As $n \to N, a(\mathcal{S}) \to a(\mathcal{P})$, i.e., sample error $\to 0$
- This would mean that the estimation is in some sense consistent.
  - This type of consistency is sometimes called *Fisher consistency* in the statistical literature,
  - Named after the statistical scientist Ronald A. Fisher who in 1922 identified this consistency as an important criterion for estimation.

3.2 All Possible Samples

3.2.1 All Possible Samples

Suppose the population $\mathcal{P}$ was of size N and that the sample $\mathcal{S}$ was of size n.

- Then there are $\binom{N}{n}$ different possible samples $\mathcal{S}$ of size n.

The average sample error overall all possible samples of size n is

$$\text{Average sample error} = \left(\frac{1}{M}\sum_{i=1}^{M} a(\mathcal{S}_i)\right) - a(\mathcal{P}) = \frac{1}{M}\sum_{i=1}^{M}\left[a(\mathcal{S}_i) - a(\mathcal{P})\right]$$

3.2.2 Consistency and the Effect of Sample Size

The nature of sample error depends largely on the sample size.

- As the sample size increases, the sample approaches the population
- Attribute values will concentrate even more around the population value
  - Regardless of n, $a(\mathcal{S})$ values concentrate around $a(\mathcal{P})$. But as n increases, dispersion decreases.
- To quantify this concentration we could look at

$$\left\|a(\mathcal{S}) - a(\mathcal{P})\right\|_1 = \left\|\frac{1}{n}\sum_{u\in\mathcal{S}} y_u - \frac{1}{N}\sum_{u\in\mathcal{P}} y_u\right\|_1 < c$$

  for some $c > 0$
  - Then we could calculate the proportion of samples that satisfy this.
- Consider a population $\mathcal{P}$ of size $N < \infty$.
  - For each n, we can construct the set of all possible samples.

$$\mathcal{P}_{\mathcal{S}}(n) = \{\mathcal{S} : \mathcal{S} \subset \mathcal{P} \text{ and } |\mathcal{S}| = n\}$$

- For any $c > 0$,

$$\mathcal{P}_a(c, n) = \{\mathcal{S} : \mathcal{S} \subset \mathcal{P}_{\mathcal{S}}(n) \text{ and } \left\|a(\mathcal{S}) - a(\mathcal{P})\right\|_1 < c\}$$

  that is, all samples for which absolute sample error is less than c, and define the proportion

$$p_a(c, n) = \frac{|\mathcal{P}_a(c, n)|}{|\mathcal{P}_{\mathcal{S}}(n)|}$$

  for all $c > 0$, and $n \leq N$.

$p_a(c, n)$ increases with n for a fixed c. Note this notion of consistency is different and separate from Fisher consistency.

As c increases, the proportion $p_a(c, n) \to 1$ for all n. But it does so much more quickly for large n. This

signifies the sample attributes cluster more tightly around the true population attributes for large samples.

3.2.3 Comparisons across attributes

The concentration of the sample attribute values around the true value represents another kind of consistency. We numerically measure this form of consistency using the proportion of attribute values some distance from the true value.

Previously we defined consistency in terms of absolute sample error. This allowed us to evaluate the impact of sample size on concentration.

However, if we want to compare *different* attributes, we use the relative absolute sample error. For any $c > 0$, let

$$\mathcal{P}_a^*(c,n) = \left\{ \mathcal{S} : \mathcal{S} \subset \mathcal{P}_{\mathcal{S}}(n) \text{ and } \frac{\|a(\mathcal{S}) - a(\mathcal{P})\|_1}{\|a(\mathcal{P})\|_1} < c \right\}$$

which is scale-free, and define the corresponding proportion, for all $c > 0$, and $n \leq N$

$$p_a^*(c,n) = \frac{|\mathcal{P}_a^*(c,n)|}{|\mathcal{P}_{\mathcal{S}}(n)|}$$

$p_a^*(c,n)$ measures the consistency of the sample attribute with respect to the *same* population attribute.

When making comparisons between attributes, we are evaluating each attribute on how well its sample values track its population value on the *same scale*.

3.3 Selecting Samples

The sampling distribution of the attribute $a(\mathcal{S})$ gives insight into understanding the magnitude of error that can be expected. Properties of this distribution can be determined

- exactly, when all possible samples are available
- approximately, when a subset of all possible samples is considered
- in expectation, when a probabilistic sampling mechanism is used to draw a single sample

3.3.1 Randomly Selecting m Samples

Consider drawing samples of size n from the population $\mathcal{P}$. The population $\mathcal{P}_{\mathcal{S}}$ of all such samples has size $M = \binom{N}{n}$ and is denoted

$$\mathcal{P}_{\mathcal{S}} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_M\}$$

- Any attribute $a(\mathcal{S}_i)$ is now just a variate on that unit!
  - We then have a population of attributes

$$\mathcal{P}_{a(\mathcal{S})} = \{a(\mathcal{S}_1), a(\mathcal{S}_2), \ldots, a(\mathcal{S}_M)\}$$

    which is sampling distribution

Interesting Aside: The Distribution of a Histogram

- Suppose the histogram based on all possible samples has K bins

$$B_1 = (b_0, b_1], B_2 = (b_1, b_2], \ldots, B_K = (b_{K-1}, b_K]$$

  where the k^{th} bin B_k contains $M_k \geq 0$ of the attribute values $a(S_i)$ for $i = 1, \ldots, M$

  - The bins contain the attribute values of all of the $S_i \in \mathcal{P}_S$ so that $\sum_{k=1}^{K} M_k = M$.

- Now suppose that we select m samples at random from $\mathcal{P}_S$ such that the probability that any given sample is selected is

$$p(S) = \frac{1}{M}$$

  - Let m_k be the number of the m selected samples whose attribute value falls in B_k, with $m = \sum_{k=1}^{K} m_k$.

With this notation,

- the histogram based on all possible samples has heights $M_1, \ldots, M_K$ and
- the histogram based on a sample of m of the possible samples has heights $m_1, \ldots, m_K$.

The probability of any particular histogram arising from a random selection of m samples is therefore a multivariate hypergeometric probability

$$P = \frac{\binom{M_1}{m_1}\binom{M_2}{m_2}\cdots\binom{M_K}{m_K}}{\binom{M}{m}}$$

which, when $m \ll M$ can be approximated by the multinomial probability

$$\left(\frac{m}{m_1 m_2 \ldots m_K}\right) p_1^{m_1} p_2^{m_2} \ldots p_K^{m_K}$$

with probabilities $p_k = \frac{M_k}{M}$ for $k = 1, \ldots, K$.

- From the multinomial, the expected value of the number of attribute values in each bin B_k is proportional to M_k. i.e., $mp_k = \frac{m}{M} M_k = E(m_k)$, where $E(m_k) \propto M_k$.
  - The *frequency* histogram based on m samples is (in expectation) a scaled version of that of all possible samples.
  - The *density* histogram based on m samples is (in expectation) identical to that of all possible samples.

3.3.2 Quantifying Sample Error

In principle we select a sample S from the population $\mathcal{P}_\mathcal{S}$ containing all possible samples.

- We do so with some probability $p(\mathcal{S}) \geq 0$ of being selected. We require of course that

$$\sum_{\mathcal{S} \in \mathcal{P}_\mathcal{S}} p(\mathcal{S}) = 1.$$

- For any sample $\mathcal{S} \in \mathcal{P}_\mathcal{S}$, we have its sample error

$$\text{Sample Error} = a(\mathcal{S}) - a(\mathcal{P}).$$

  - Recall, for any collection of samples (or population of samples) $\mathcal{P}_\mathcal{S}$ containing M samples, we can calculate the average sample error

$$\text{Average Sample Error} = \frac{1}{M} \sum_{\mathcal{S} \in \mathcal{P}_\mathcal{S}} [a(\mathcal{S}) - a(\mathcal{P})].$$

We can quantify the concentration of sample errors in expectation using quantities such as sampling bias, sampling variance, and sampling mean squared error.

- By sampling $\mathcal{S}$ randomly from $\mathcal{P}_\mathcal{S}$ with probability $p(\mathcal{S})$ we define the sampling bias as

$$\begin{aligned}
\text{Sampling Bias} &= E[a(\mathcal{S}) - a(\mathcal{P})] \quad \text{where } a(\mathcal{S}) \text{ is a random variable} \\
&= E[a(\mathcal{S})] - a(\mathcal{P}) \\
&= \sum_{\mathcal{S} \in \mathcal{P}_\mathcal{S}} a(\mathcal{S}) p(\mathcal{S}) - a(\mathcal{P}) \quad \text{recall } E[x] = \sum_{x \in A} x Pr(X = x) \\
&= \sum_{\mathcal{S} \in \mathcal{P}_\mathcal{S}} a(\mathcal{S}) p(\mathcal{S}) - a(\mathcal{P})(1) \\
&= \sum_{\mathcal{S} \in \mathcal{P}_\mathcal{S}} a(\mathcal{S}) p(\mathcal{S}) - a(\mathcal{P}) \sum_{\mathcal{S} \in \mathcal{P}_\mathcal{S}} p(\mathcal{S}) \\
&= \sum_{\mathcal{S} \in \mathcal{P}_\mathcal{S}} a(\mathcal{S}) p(\mathcal{S}) - \sum_{\mathcal{S} \in \mathcal{P}_\mathcal{S}} a(\mathcal{P}) p(\mathcal{S}) \\
&= \sum_{\mathcal{S} \in \mathcal{P}_\mathcal{S}} [a(\mathcal{S}) - a(\mathcal{P})] p(\mathcal{S})
\end{aligned}$$

  - Sampling bias is just the expected sample error induced by the repeated random sampling of $\mathcal{S}$ from $\mathcal{P}_\mathcal{S}$. If $p(\mathcal{S}) = \frac{1}{M}$, the sampling bias is identical to the average sample error of $a(\mathcal{P})$.
  - The sampling bias depends on the attribute $a(\cdot)$, the set of possible samples $\mathcal{P}_\mathcal{S}$, and the sample probabilities $p(\mathcal{S})$.
  - Note: If sampling bias is zero, then $a(\mathcal{S})$ is called an unbiased estimator of $a(\mathcal{P})$.

- The sampling variance is defined as

$$Var[a(\mathcal{S})] = E\left[(a(\mathcal{S}) - E[a(\mathcal{S})])^2\right] = \sum_{\mathcal{S} \in \mathcal{P}_\mathcal{S}} \left[(a(\mathcal{S}) - E[a(\mathcal{S})])^2\right] p(\mathcal{S})$$

  Recall $Var[x] = E[(x - E(x))^2]$

  - This quantifies dispersion in the sample errors.
  - We may also use the sampling standard deviation defined as the square root of the variance.

- Given a sample $\mathcal{S}$, we would like $a(\mathcal{S})$ and $a(\mathcal{P})$ to be as close as possible. We can use the mean squared error to quantify the expected squared distance between these two quantities

$$MSE[a(\mathcal{S})] = E\left[(a(\mathcal{S}) - a(\mathcal{P}))^2\right] = Var[a(\mathcal{S})] + [\text{Sampling Bias}]^2$$

  - Ideally, we would like to choose $p(\mathcal{S})$ and/or $\mathcal{P}_{\mathcal{S}}$, so that both the square of sampling bias and the sampling variance are as small as possible.

Note that all expectations are taken with respect to the probabilities $p(\mathcal{S})$ of choosing a sample $\mathcal{S}$ from $\mathcal{P}_{\mathcal{S}}$.

If the MSE associated with a sampling design d_2 is smaller, it is to be preferred as that tells us that the sample attribute values are expected to be closer to the true population value than with another sampling design, let's say d_1.

Attribute as an Estimator

- Thinking of the sampling distribution of an attribute $a(\mathcal{S})$ gives rise to the notion of an attribute as an estimator (i.e., as a random variable).
- We can introduce a random variable, say A, that takes values a from the distinct values of $a(\mathcal{S})$ for all $\mathcal{S} \in \mathcal{P}_{\mathcal{S}}$. The induced probability distribution is

$$Pr(A = a) = \sum_{\mathcal{S} \in \mathcal{P}_{\mathcal{S}}} p(\mathcal{S}) \times I(a(\mathcal{S}) = a)$$

  - $I(a(\mathcal{S}) = a)$ is an indicator function = 1 if $a(\mathcal{S}) = a$
  - It is weighted sum of all possible samples whose attribute value is a.
  - It follows that A is a discrete random variable.
  - Probability statements about its values can be made using its distribution, including its expectation, variance, etc.
  - Each of the definitions above (sampling bias, sampling variance, sampling MSE) can be defined in terms of this random variable and the corresponding probability distribution.

3.3.3 Sampling Mechanisms

Rather than select samples at random from all possible samples, the same outcome is effected by sampling the units that will appear in any particular sample.

In other words: rather than select $\mathcal{S}$ with probability $p(\mathcal{S})$ from $\mathcal{P}_{\mathcal{S}} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_M\}$ we form $\mathcal{S}$ by selecting n units $u_{i1}, u_{i2}, \ldots, u_{in}$, directly from the population of units $\mathcal{P} = \{u_1, u_2, \ldots, u_N\}$.

- Each unit u in a sample $\mathcal{S}$ is selected one at a time from the population $\mathcal{P}$.

- A sequence of the first k units u_i selected from $\mathcal{P}$ is

$$s_k = (u_{i1}, u_{i2}, \ldots, u_{ik})$$

- A sampling mechanism is defined by the probabilities $\Pr(u)$ and $\Pr(u|k, s_{k-1})$ (probability of selecting unit u on the k^{th} drawn given we've observed sequence s_{k-1}, i.e., the probability unit u is the k^{th} one drawn into sample) where
  - the first unit is selected with probability $\Pr(u)$, and
  - the probability of the sequence of the first k units selected is

$$\Pr(s_k) = \Pr(u_{i1}) \times \Pr(u_{i2}|2, s_1) \times \Pr(u_{i3}|3, s_2) \times \ldots \times \Pr(u_{ik}|k, s_{k-1}).$$

which is equivalent to $\Pr(u_{i1}$ and u_{i2} and u_{i3} and $\ldots$ $u_{ik})$

To determine $p(\mathcal{S})$ from a sampling mechanism:

- Recognize that the order in which the units appear does not matter, i.e., any of the $n!$ permutations of the elements of s_n counts as $\mathcal{S}$
- $p(\mathcal{S})$ is thus the sum of $\Pr(s_n)$ over all permutations s_n.

Simple Random Sampling Without Replacement (SRSWOR)

- The sampling mechanism is

$$\Pr(u) = \frac{1}{N} \text{ and } \Pr(u \mid k, s_{k-1}) = \frac{1}{N-k+1}$$

- The probability of the sequence s_n is

$$\Pr(s_n) = \frac{1}{N} \times \frac{1}{N-1} \times \frac{1}{N-2} \times \ldots \times \frac{1}{N-n+1}$$

  which is the same for all $n!$ permutations, so

$$p(\mathcal{S}) = \frac{n!}{N(N-1)(N-2)\ldots(N-n+1)} = \frac{1}{\binom{N}{n}} = \frac{1}{M} = n! \times \Pr(s_n)$$

  This probability is the same as we had before for selecting n distinct units from a population of N distinct units.

However, we now have a mechanism that allows us to select a sample without first enumerating all $M = \binom{N}{n}$ possible samples in $\mathcal{P}_{\mathcal{S}}$.

Simple Random Sampling With Replacement (SRSWR)

- The sampling mechanism is

$$\Pr(u) = \frac{1}{N} = \Pr(u|k, s_{k-1})$$

  and thus a sample $\mathcal{S}$ can contain one or more replicated units

- The probability of the sequence s_n is
$$\Pr(s_n) = \left(\frac{1}{N}\right)^n$$
Note that unlike in the case of SRSWOR, we typically treat each s_n as an ordered sample and so
$$p(\mathcal{S}) = \Pr(s_n) = \frac{1}{N^n}$$
The population of all samples $\mathcal{P}_S$ in this case contains $M = N^n$ ordered samples.
- If we treat each s_n as an unordered sample similar to SRSWOR we obtain
$$p(\mathcal{S}) = \frac{n!}{n_1!n_2!\ldots n_N!} \times \frac{1}{N^n} = \Pr(s_n) \times \frac{n!}{n_1!n_2!\ldots n_N!}$$
where n_u is the number of duplicates of unit u in the sample. e.g., if $n_4 = 2$ then unit 4 was in the sequence (sample) twice.

A Weird Hybrid Sampling Mechanism (Let's call it SRSWH)

- The following mechanism was first explored by Basu (1958).
- Suppose we perform simple random sampling with replacement except that we remove any duplicate units.
  - The samples produced will have sizes anywhere from 1 to n according to how many distinct units were selected in a sample (sampling with replacement).

Note that since the number of duplicates is a random variable, the actual sample size (n minus the number of duplicates) is also a random variable here!

Comparing Sampling Mechanisms – Australian Shark Encounters

For a population of size N:

- there exist $\binom{N}{n}$ samples without replacement
- there exist $\binom{N+n-1}{n}$ samples with replacement (order does not matter)
- there exist somewhere between $\binom{N}{n}$ and $\binom{N+n-1}{n}$ samples with replacement but no duplicates

Through an example, we may have the following conclusion: SRSWOR is superior in terms of MSE; SRSWR is worst in terms of MSE; SRSWH produces samples on higher quality than SRSWR since they do not have redundant (i.e., duplicate) information. But these samples are smaller than SRSWOR ones, so they are of less quality comparatively.

Note that different sample sizes can result for SRSWH since it will remove duplicates even if we say we want a sample size of n.

3.3.4 Unit Inclusion Probabilities

The inclusion probability for unit u is the probability of unit u being included in the sample.

$$\pi_u = \Pr(u \in \mathcal{S}) = \sum_{\mathcal{S} \in \mathcal{P}_\mathcal{S}} p(\mathcal{S}) \times I(u \in \mathcal{S})$$

Now, consider the following random variable

$$D_u = \begin{cases} 1 & \text{if } u \in \mathcal{S} \\ 0 & \text{otherwise} \end{cases}$$

D_u takes the value 1 with probability $\Pr(u \in \mathcal{S})$, and it takes on the value 0 with probability $\Pr(u \notin \mathcal{S})$. The expectation of this random variable is

$$E[D_u] = 1 \times \Pr(D_u = 1) + 0 \times \Pr(D_u = 0) = \Pr(u \in \mathcal{S}) = \pi_u$$

The variance of this random variable is

$$Var[D_u] = E[D_u^2] - (E[D_u])^2 = 1^2 \times \Pr(D_u = 1) + 0^2 \times \Pr(D_u = 0) - \pi_u^2 = \pi_u - \pi_u^2 = \pi_u(1 - \pi_u)$$

The probability that u and v are in the sample $\mathcal{S}$ is called the joint inclusion probability and is given by

$$\pi_{uv} = \Pr(u \in \mathcal{S} \text{ and } v \in \mathcal{S}) = \sum_{\mathcal{S} \in \mathcal{P}_\mathcal{S}} p(\mathcal{S}) I[u \in \mathcal{S}, v \in \mathcal{S}]$$

And we have the following relationship

$$E[D_u \times D_v] = 1 \times \Pr(D_u = 1, D_v = 1) = \Pr(u \in \mathcal{S}, v \in \mathcal{S}) = \pi_{uv}$$

where $D_u \times D_v = 0$ if either of them is 0.

$$Cov(D_u, D_v) = E[D_u \times D_v] - E[D_u] \times E[D_v] = E[D_u \times D_v] - \pi_u \times \pi_v = \pi_{uv} - \pi_u \times \pi_v$$

Recall $Cov[X, Y] = E(XY) - E[X]E[Y]$.

Simple Random Sampling Without Replacement (SRSWOR)

- The inclusion probability is

$$\pi_u = \Pr(u \in \mathcal{S}) = \frac{1 \times \binom{N-1}{n-1}}{\binom{N}{n}} = \frac{n}{N}$$

- The joint inclusion probability is

$$\pi_{uw} = \Pr(u \in \mathcal{S} \cap v \in \mathcal{S}) = \frac{1 \times 1 \times \binom{N-2}{n-2}}{\binom{N}{n}} = \frac{n(n-1)}{N(N-1)}$$

Simple Random Sampling With Replacement (SRSWR)

- The inclusion probability is

$$\pi_u = \Pr(u \in \mathcal{S}) = 1 - \Pr(u \notin \mathcal{S}) = 1 - \left(1 - \frac{1}{N}\right)^n = 1 - \left(\frac{N-1}{N}\right)^n$$

  So here by specifying $\Pr(u \notin \mathcal{S})$, we only consider $n_u = 0$.

- The joint inclusion probability is

$$\begin{aligned}\pi_{uw} &= 1 - 2\left(\frac{N-1}{N}\right)^n + \left(\frac{N-2}{N}\right)^n \\ &= \Pr(u \in \mathcal{S} \cap v \in \mathcal{S}) \\ &= 1 - \Pr(u \notin \mathcal{S} \cup v \notin \mathcal{S}) \\ &= 1 - [\Pr(u \notin \mathcal{S}) + \Pr(v \notin \mathcal{S}) - \Pr(u \notin \mathcal{S}, v \notin \mathcal{S})]\end{aligned}$$

The Weird Hybrid Sampling Mechanism (SRSWH)

The inclusion probabilities for sampling with replacement but using only the unique units selected (i.e., the "weird hybrid" mechanism discussed earlier due to Basu) are *identical* to simple random sampling with replacement.

3.3.5 Estimating Totals

Many attributes are either a total

$$a(\mathcal{P}) = \sum_{u \in \mathcal{P}} y_u$$

of some variate y_u observed on every unit $u \in \mathcal{P}$, or a function of such a total

$$a(\mathcal{P}) = f\left(\sum_{u \in \mathcal{P}} y_u\right)$$

(Recall that a variate y is any function that when applied to any unit $u \in \mathcal{P}$ returns a value $y(u) = y_u$)

The Horvitz-Thompson Estimate

- A natural (and very commonly used) estimate of a population total

$$a(\mathcal{P}) = \sum_{u \in \mathcal{P}} y_u$$

  is the Horvitz-Thompson estimate (due to Daniel G. Horvitz and Donovan J. Thompson, 1952) defined

as

$$\widehat{a(\mathcal{P})} = a_{HT}(\mathcal{S}) = \sum_{u \in \mathcal{S}} \frac{y_u}{\pi_u}$$

where the contribution for each unit in the sample is weighted inversely by π_u, its probability of inclusion in $\mathcal{S}$.

- if the probability of inclusion is small, then the weight will be high
- if the probability of inclusion is large, then the weight will be low

- Note that we use the subscript HT to distinguish the Horvitz-Thompson estimate of $a(\mathcal{P})$ from other estimates based on the sample $\mathcal{S}$.

The Horvitz-Thompson Estimator

Here we consider properties of the Horvitz-Thompson estimator (i.e., the random variable), $\tilde{a}_{HT}(\mathcal{S})$. Such properties inform what can be expected under repeated sampling.

Bias, Variance, and Mean Squared Error

In what follows it will be convenient to work with the random variable

$$D_u = \begin{cases} 1 & \text{if } u \in \mathcal{S} \\ 0 & \text{otherwise.} \end{cases}$$

Note that with this defined the Horvitz-Thompson estimator can be written as

$$\tilde{a}_{HT}(\mathcal{S}) = \sum_{u \in \mathcal{S}} \frac{y_u}{\pi_u} = \sum_{u \in \mathcal{P}} D_u \times \frac{y_u}{\pi_u}$$

The following properties of D_u that we derived earlier will also be useful:

$$E[D_u] = \pi_u$$

$$Var[D_u] = \pi_u(1 - \pi_u)$$

$$\text{Cov}[D_u, D_v] = \pi_{uv} - \pi_u \pi_v = \Delta_{uv}$$

Let us derive the bias of the HT estimator:

$$\begin{aligned}
E[\tilde{a}_{HT}(\mathcal{S}) - a(\mathcal{P})] &= E[\tilde{a}_{HT}(\mathcal{S})] - a(\mathcal{P}) \\
&= E\left[\sum_{u\in\mathcal{P}} D_u \times \frac{y_u}{\pi_u}\right] - a(\mathcal{P}) \\
&= \sum_{u\in\mathcal{P}} \frac{y_u}{\pi_u} E[D_u] - a(\mathcal{P}) \\
&= \sum_{u\in\mathcal{P}} \frac{y_u}{\pi_u} \times \pi_u - a(\mathcal{P}) \\
&= \sum_{u\in\mathcal{P}} y_u - a(\mathcal{P}) \\
&= 0
\end{aligned}$$

The variance of the Horvitz-Thompson estimator is given by:

$$Var[\tilde{a}_{HT}(\mathcal{S})] = \sum_{u\in\mathcal{P}}\sum_{v\in\mathcal{P}} (\pi_{uv} - \pi_u\pi_v) \frac{y_u y_v}{\pi_u\pi_v} = \sum_{u\in\mathcal{P}}\sum_{v\in\mathcal{P}} \Delta_{uv} \frac{y_u y_v}{\pi_u\pi_v}$$

Recall: $Var\left[\sum a_i X_i\right] = \sum\sum Cov[a_i X_i, a_j X_j] = \sum\sum a_i a_j Cov[X_i, X_j]$

So the above variance is derived from

$$Var[\tilde{a}_{HT}(\mathcal{S})] = Var\left[\sum_{u\in\mathcal{P}} \frac{y_u}{\pi_u} \times D_u\right] = \sum_{u\in\mathcal{P}}\sum_{v\in\mathcal{P}} \left(\frac{y_u}{\pi_u}\right)\left(\frac{y_v}{\pi_v}\right) Cov[D_u, D_v]$$

This variance can be equivalently written in the Yates-Grundy or the Sen-Yates-Grundy formulation:

$$Var[\tilde{a}_{HT}(\mathcal{S})] = -\frac{1}{2}\sum_{u\in\mathcal{P}}\sum_{v\in\mathcal{P}} \Delta_{uv}\left(\frac{y_u}{\pi_u} - \frac{y_v}{\pi_v}\right)^2$$

Note that because the HT estimator is unbiased the mean squared error is simply equal to the variance.

The HT Estimate of the Variance of the HT Estimator

- For the HT estimate we will need the following quantity

  $$q_{u,v} = \Delta_{u,v} \frac{y_u y_v}{\pi_u\pi_v}$$

  defined over $\mathcal{P}_{uv}$ which is the population of all pairs (u, v) where $u, v \in \mathcal{P}$.

- The variance of the HT estimator can be written as

  $$Var[\tilde{a}_{HT}(\mathcal{S})] = \sum_{u\in\mathcal{P}}\sum_{v\in\mathcal{P}} \Delta_{u,v} \frac{y_u y_v}{\pi_u\pi_v} = \sum_{(u,v)\in\mathcal{P}_{uv}} q_{u,v}$$

- The HT estimate of the variance of the HT estimator is:

$$\widehat{Var}[\tilde{a}_{HT}(\mathcal{S})] = \sum_{(u,v)\in\mathcal{S}_{uv}} \frac{q_{u,v}}{\pi_{uv}} = \sum_{u\in\mathcal{S}}\sum_{v\in\mathcal{S}} \frac{\Delta_{u,v}}{\pi_{uv}} \frac{y_u y_v}{\pi_u \pi_v} = \sum_{u\in\mathcal{S}}\sum_{v\in\mathcal{S}} \left(\frac{\pi_{uv} - \pi_u\pi_v}{\pi_{uv}}\right) \frac{y_u y_v}{\pi_u \pi_v}$$

  - Note that the sample $\mathcal{S}_{uv}$ is obtained by sampling from the population $\mathcal{P}_{uv}$ of all possible pairs (u, v). The probability that any particular pair (u, v) is included in the sample is $\pi_{uv} > 0$
  - Note also that the square root of this variance estimate or HT estimate of the standard deviation is commonly referred to the standard error of the estimate

$$SE(\tilde{a}_{HT}(\mathcal{S})) = \sqrt{\widehat{Var}[\tilde{a}_{HT}(\mathcal{S})]} = \widehat{SD}[\tilde{a}_{HT}(\mathcal{S})]$$

- Thus, using Horvitz-Thompson estimation we are able to construct
  - an estimate of the population total *and*
  - an estimate of the variance of this estimator *and*
  - both estimators are unbiased.

We obtain the following interval which (under an assumption of normality) approximately contains 95% of the HT estimates

$$a(\mathcal{P}) \pm 2\sqrt{Var[\tilde{a}_{HT}(\mathcal{S})]}$$

A sample estimate of this interval is

$$a_{HT}(\mathcal{S}) \pm 2\sqrt{\widehat{Var}[\tilde{a}_{HT}(\mathcal{S})]} = a_{HT}(\mathcal{S}) \pm 2\widehat{SD}[\tilde{a}_{HT}(\mathcal{S})] = a_{HT}(\mathcal{S}) \pm 2SE[\tilde{a}_{HT}(\mathcal{S})]$$

3.4 Sampling Designs

The pair $(\mathcal{P}_{\mathcal{S}}, p(\mathcal{S}))$ is called a sampling design

- Together they determine which samples are possible and with what probability they are selected
- The SRSWOR, SRSWR and SRSWH frameworks provide examples of different sampling designs
- The sampling design is ours to choose
  - We may choose $\mathcal{P}_{\mathcal{S}}$ so that the values $a(\mathcal{S})$ for $\mathcal{S} \in \mathcal{P}_S$ are constrained to be near $a(\mathcal{P})$
  - We may choose $p(\mathcal{S})$ so that samples $\mathcal{S} \in \mathcal{P}_S$ that have $a(\mathcal{S})$ close to $a(\mathcal{P})$ have higher probability $p(\mathcal{S})$ of being selected
  - Within the Horvitz-Thompson framework we know that

$$MSE[\tilde{a}_{HT}(\mathcal{S})] = Var[\tilde{a}_{HT}(\mathcal{S})] = -\frac{1}{2}\sum_{u\in\mathcal{P}}\sum_{v\in\mathcal{P}} \Delta_{uv}\left(\frac{y_u}{\pi_u} - \frac{y_v}{\pi_v}\right)^2$$

    which provides insight into how we might best choose a sampling design.

* For example, if we could choose $\pi_u \propto y_u$ then the variance (and MSE) will be zero!
* Perhaps there is another variate x_u that is highly positively correlated with y_u for all $u \in \mathcal{P}$. Then choosing $\pi_u \propto x_u$ could reduce MSE.
* If we knew when $y_u \approx y_v$ we could choose $\pi_u \approx \pi_v$ and this might reduce MSE (e.g., stratified sampling tries to do this).

CHAPTER 4: Inference

4.1 Inductive Inference

Any type of error can comprise the quality of our inference.

4.1.1 Target and Study populations

The target population is the population about which we would like to draw an inference, but the study population is the population from which samples are taken.

The difference between the attribute evaluated on the two populations is the study error:

$$\text{Study Error} = a(\mathcal{P}_{\text{study}}) - a(\mathcal{P}_{\text{target}}).$$

Then if we obtain a sample $\mathcal{S}$ to draw inferences about the target population $\mathcal{P}_{\text{target}}$, the error for a given attribute $a(\cdot)$ is

$$a(\mathcal{S}) - a(\mathcal{P}_{\text{target}}) = [a(\mathcal{S}) - a(\mathcal{P}_{\text{study}})] + [a(\mathcal{P}_{\text{study}}) - a(\mathcal{P}_{\text{target}})] = (\text{Sample error}) + (\text{Study error}).$$

We use probabilistic sampling to control the sample error but not the study error.

4.1.2 Measurement Error

The inductive path of inference includes the set of measured values. Errors made in measurement can also affect conclusions drawn about attributes.

Measurement systems

- Every measuring system has at least three sources of potential error:
  1. the measuring device (sometimes called the gauge)
  2. the person reading or recording the measurement (sometimes called the operator)
  3. the method followed to take the measurement

4.2 Comparing Sub-Populations

Suppose we have the population $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2\}$ and interest lies in comparing some attribute across the two sub-populations:

- We could compare the two sub-population attributes by way of a difference:

$$a(\mathcal{P}_1) \text{ vs. } a(\mathcal{P}_2)$$

- We could compare the two sub-population attributes by way of a difference:

$$a(\mathcal{P}_1) - a(\mathcal{P}_2)$$

- We could compare the two sub-population attributes by way of a ratio:

$$\frac{a(\mathcal{P}_1)}{a(\mathcal{P}_2)}$$

- If the attribute is graphical (a histogram or quantile plot, for example) we could compare the two sub-populations by displaying the figures beside one another or overlaying them on top of one another.

Note: that when the attribute is a measure of location, sub-population comparisons are typically based on differences and when the attribute is a measure of spread, comparisons are typically based on ratios.

Randomly Mixing Sub-Populations

- If the two sub-populations are *essentially the same*
  - then the sub-populations observed should not look too different if we were to mix them up with one another
  - in other words, swapping units would not dramatically change the features of the resulting sub-populations.
- On the other hand, if the two sub-populations were *very different*
  - then shuffling the units could dramatically change the features of the resulting sub-populations.
- Here we combine the two sub-populations together into one $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2\}$ (size $N = N_1 + N_2$) and then
  - randomly draw two new sub-populations $\mathcal{P}_1^*$ and $\mathcal{P}_2^*$
  - ensuring that the sub-population sizes are kept the same
- We then compare the attributes of $\{\mathcal{P}_1, \mathcal{P}_2\}$ with $\{\mathcal{P}_1^*, \mathcal{P}_2^*\}$:
  - e.g., $a(\mathcal{P}_1)$ to $a(\mathcal{P}_1^*)$, $a(\mathcal{P}_2)$ to $a(\mathcal{P}_2^*)$, or
  - e.g., $a(\mathcal{P}_1) - a(\mathcal{P}_2)$ to $a(\mathcal{P}_1^*) - a(\mathcal{P}_2^*)$, or
  - e.g., $\frac{a(\mathcal{P}_1)}{a(\mathcal{P}_2)}$ to $\frac{a(\mathcal{P}_1^*)}{a(\mathcal{P}_2^*)}$, or
  - some other measure of difference among the sub-populations.
- If the sub-populations were similar to begin with, there shouldn't be a very large difference between attributes calculated on $\{\mathcal{P}_1, \mathcal{P}_2\}$ versus those calculated on $\{\mathcal{P}_1^*, \mathcal{P}_2^*\}$.

4.2.1 Anatomy of a Significance Test

We would like to quantify, numerically, how unusual the difference between $a(\mathcal{P}_1)$ and $a(\mathcal{P}_2)$ is relative to randomly mixed sub-populations.

- If the two sub-populations are actually similar, we want to provide numerical evidence in favour of the notion that the two sub-populations are similar to randomly mixed sub-populations.
- If the two sub-populations are actually different, we want to provide numerical evidence against the notion that the two sub-populations are similar to randomly mixed sub-populations.

The following steps are used to gather such evidence:

1. We suppose the sub-populations were randomly drawn from the same population. This is known as the null hypothesis.
2. We construct a discrepancy measure (AKA test statistics) that quantifies how inconsistent our data is with the null hypothesis
   - where large values indicate evidence against the null hypothesis
3. We obtain the observed discrepancy by calculating
   - the discrepancy measure on the two observed (i.e., unshuffled) sub-populations
4. To evaluate the extremity of the observed discrepancy, we compare its value to an approximation of the sampling distribution. We obtain this approximation through repeated random shuffling. This comparison is formalized with a p-value.
5. Finally, we obtain the observed p-value by calculating
   - the probability that a randomly shuffled sub-population has a discrepancy measure at least as large as the observed discrepancy
   - where small values indicate evidence against the null hypothesis.

The Null Hypothesis

- Each of the following (equivalent) statements constitutes the null hypothesis we are testing.
  - H_0: The sub-populations $\mathcal{P}_1$ and $\mathcal{P}_2$ were randomly drawn from the same population.
  - H_0: $\mathcal{P}_1$ and $\mathcal{P}_2$ were created by randomly assigning units in the same population to one of the two sub-populations.
  - H_0: $\mathcal{P}_1$ and $\mathcal{P}_2$ were generated by random mixing.
  - H_0: $\mathcal{P}_1$ and $\mathcal{P}_2$ are indistinguishable
- Regardless of how the null hypothesis is stated, the alternative hypothesis H_A is the complement of H_0.
- Note that we do not state the null hypothesis in terms of the equivalence of attribute values, i.e., $a(\mathcal{P}_1) = a(\mathcal{P}_2)$.
  - Although such a statement is true if H_0 holds, it is weaker and so we avoid using it.

The Discrepancy Measure

- A discrepancy measure $D(\mathcal{P}_1, \mathcal{P}_2)$ quantifies how inconsistent our data is with the null hypothesis, and is defined so that large values indicate evidence against the null hypothesis.

  - As a point of interest, the discrepancy measure is technically an attribute for the population $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2\}$ and so we could consider properties such as equivariance and invariance.
  - In other statistical texts the discrepancy measure is often referred to as a test statistic.
- The form of $D(\mathcal{P}_1, \mathcal{P}_2)$ depends on *how* we want to compare $\mathcal{P}_1$ and $\mathcal{P}_2$
  - If we want to compare measures of location the discrepancy measure is typically based on *differences*:
$$a(\mathcal{P}_1) - a(\mathcal{P}_2)$$
  - If we want to compare measures of spread the discrepancy measure is typically based on *ratios*:
$$\frac{a(\mathcal{P}_1)}{a(\mathcal{P}_2)}$$

The Observed Discrepancy

- The observed discrepancy, d_{obs}, is the value of discrepancy measure D calculated on the two observed (i.e., unshuffled) sub-populations:
$$d_{\text{obs}} = D(\mathcal{P}_1, \mathcal{P}_2)$$
- It's important to recognize that the discrepancy measure quantifies only one type of discrepancy between the populations
  - e.g., discrepancy in averages
  - e.g., discrepancy in standard deviations
- All other differences are completely ignored. For instance, $\mathcal{P}_1$ and $\mathcal{P}_2$ may be very similar with respect to their means, but very different with respect to their variation.

The Observed p-value

The probability of observing a discrepancy at least as extreme as the one we observed if H_0 was true.

- The observed p-value is the probability that a randomly shuffled sub-population has a discrepancy measure at least as large as the observed discrepancy
$$\text{p-value} = \Pr(D \geq d_{\text{obs}} | H_0 \text{ is true})$$
where sampling distribution of D values approximated via random shuffling (i.e., the null distribution).
- If the p-value is very small then either
  - the null hypothesis is true and we have observed a very unusual value of d_{obs}
  - OR the null hypothesis is false.
- The smaller the p-value, the greater the evidence against the null hypothesis.
  - p-value < 0.001 means that there is very strong evidence against H_0

  - $0.001 \leq$ p-value < 0.01 means that there is strong evidence against H_0
  - $0.01 \leq$ p-value < 0.05 means that there is evidence against H_0
  - $0.05 \leq$ p-value < 0.1 means that there is weak evidence against H_0
  - p-value ≥ 0.1 means that there is no evidence against H_0
  - In the extreme case where p-value $= 0$, then we have observed something impossible and the hypothesis must therefore be false – this would be a proof by contradiction.
- In order to calculate the p-value *exactly* one must consider all $\left(\binom{N}{N_1} = \binom{N}{N_2}\right)$ possible permutations of the observed data (permutation test)
  - The exact p-value is the fraction of $D(\mathcal{P}_1^*, \mathcal{P}_2^*)$ values greater than or equal to d_{obs}
- Because $\binom{N}{N_1} = \binom{N}{N_2}$ is in practice too many permutations to consider, we typically just use M (a large number) of them
  - In particular we generate M shuffled pairs:

$$(\mathcal{P}_{1,1}^*, \mathcal{P}_{2,1}^*), (\mathcal{P}_{1,2}^*, \mathcal{P}_{2,2}^*), \ldots, (\mathcal{P}_{1,M}^*, \mathcal{P}_{2,M}^*)$$

  - The p-value is then *approximated* as

$$\frac{1}{M} \sum_{i=1}^{M} I(D(\mathcal{P}_{1,i}^*, \mathcal{P}_{2,i}^*) \geq d_{\text{obs}}) \quad \text{where} \quad d_{\text{obs}} = D(\mathcal{P}_1, \mathcal{P}_2)$$

  - If $M = \binom{N}{N_1} = \binom{N}{N_2}$ and we've considered all possible shuffles the equation above would yield the *exact* p-value.

Putting it All Together

Test of Significance Algorithm

1. State the null hypothesis: $H_0 : \mathcal{P}_1$ and $\mathcal{P}_2$ are drawn from the same population.
2. Construct a measure of discrepancy $D = D(\mathcal{P}_1, \mathcal{P}_2)$ where large values indicate evidence against the null hypothesis.
3. Calculate the observed discrepancy $d_{\text{obs}} = D(\mathcal{P}_1, \mathcal{P}_2)$.
4. Shuffle the sub-populations M times and calculate the observed p-value:

$$p\text{-value} = \Pr(D \geq d_{\text{obs}} | H_0 \text{ is true}) \approx \frac{1}{M} \sum_{i=1}^{M} I(D(\mathcal{P}_{1,i}^*, \mathcal{P}_{2,i}^*) \geq d_{\text{obs}})$$

Significance Testing Errors

Courtroom Analogy

Decision	defendant is innocent	the defendant is guilty
Convicted	Error (Type I Error)	Correct
Acquitted	Correct	Error (Type II Error)

In Hypothesis Testing

Decision	H_0 is true	H_0 is false
Reject H_0	Error (Type I Error)	Correct
Do Not Reject H_0	Correct	Error (Type II Error)

Important Remarks

- The observed p-value provides a common (probabilistic) scale on which to measure the evidence against the null hypothesis.
- The observed p-value does not measure evidence in favour of the null hypothesis.
  - in science, we try to falsify hypotheses and entertain only those which remain standing;
  - absence of evidence $\neq$ evidence of absence
- A test of significance therefore neither accepts nor rejects a null hypothesis; it simply provides a measure of the evidence against it.
  - the decision taken in light of this evidence is the choice of the researcher.
- There is no magic level for a p-value such as 0.05 or 0.01,
  - there is no practical or scientific difference between p-value = 0.048 and p-value = 0.051, for example.
- The fact that the evidence against the null hypothesis is statistically significant based on some discrepancy measure does not imply that the discrepancy is practically significant.
  - the p-value measures how unusual a discrepancy of that size might be when the null hypothesis holds,
  - it says nothing about whether a discrepancy of that size matters for any practical or scientific purpose.
  - e.g., for the shark lengths data, the average shark length in fatal vs. non-fatal encounters differed by 3.25 feet. Is that difference of practical importance?
- Every test of significance is based on some measure of discrepancy and different discrepancy measures can detect different departures from the null hypothesis, so one needs to understand the nature of the departure from the hypothesis that the discrepancy is trying to measure.

4.2.2 A t-like Discrepancy Measure

Introduction

When comparing two sub-populations on the basis of a measure of location, one particularly useful discrep-

ancy measure is

$$D(\mathcal{P}_1, \mathcal{P}_2) = \frac{a(\mathcal{P}_1) - a(\mathcal{P}_2)}{SD[a(\mathcal{P}_1) - a(\mathcal{P}_2)]}$$

This discrepancy measure is "physically dimensionless"

- Whatever scale the numerator is measured in, the scale of the denominator will match, leaving the ratio free of any measurement scale.
- This naturally makes this discrepancy measure scale-invariant.

The challenge is determining the denominator of the discrepancy measure.

- In rare cases, the denominator might be known and then this discrepancy measure is a rescaling of $a(\mathcal{P}_1) - a(\mathcal{P}_2)$ and would not yield different results.
- However, more commonly, we will estimate the denominator using information from $\mathcal{P}_1$ and $\mathcal{P}_2$.

Suppose that the populations $\mathcal{P}_1$ and $\mathcal{P}_2$ are random and independently drawn from the same larger population. Then the denominator should be instead

$$SD[\tilde{a}(\mathcal{P}_1) - \tilde{a}(\mathcal{P}_2)] = \sqrt{Var[\tilde{a}(\mathcal{P}_1)] + Var[\tilde{a}(\mathcal{P}_2)]}$$

There would need to be a covariance term here if $\mathcal{P}_1$ and $\mathcal{P}_2$ are not independently drawn.

But determining the form of $SD[\tilde{a}(\mathcal{P}_1) - \tilde{a}(\mathcal{P}_2)]$ can also be difficult

- Except in the common special case when $a(\mathcal{P})$ is an average.

Differences in Averages

Suppose we were interested in differences in averages:

- In this case $a(\mathcal{P}_i) = \bar{Y}_i$ and $\mathcal{P}_i$ has size N_i, $i = 1, 2$
- And the discrepancy measure becomes:

$$D(\mathcal{P}_1, \mathcal{P}_2) = \frac{\bar{Y}_1 - \bar{Y}_2}{\sqrt{\frac{\tilde{\sigma}^2}{N_1} + \frac{\tilde{\sigma}^2}{N_2}}}$$

  where $\tilde{\sigma}$ (combined estimator) is an estimator of the standard deviation of the Y values in the population $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2\}$.

- If $\tilde{\sigma}_1$ and $\tilde{\sigma}_2$ denote the estimators of the standard deviations from each of $\mathcal{P}_1$ and $\mathcal{P}_2$ respectively, then the pooled estimator of σ would be

$$\tilde{\sigma} = \sqrt{\frac{(N_1 - 1)\tilde{\sigma}_1^2 + (N_2 - 1)\tilde{\sigma}_2^2}{(N_1 - 1) + (N_2 - 1)}}$$

  Student's t-test!

- Note: if it were inappropriate to assume the variability in the two sub-populations was equivalent we

could instead use the denominator

$$\sqrt{\frac{\tilde{\sigma}_1^2}{N_1} + \frac{\tilde{\sigma}_2^2}{N_2}}$$

Welch's t-test!

This just looks like the t-test

- This is the "two-sample" Student t statistic used to test the equality of the means of two normal distributions with common (but unknown) standard deviation σ.
- If the Y values were in fact normally distributed, the discrepancy measure would follow a Student t distribution with $N_1 + N_2 - 2$ degrees of freedom under the null hypothesis that the means were identical.
- Note, however, in our procedure of randomly mixing the populations we make no such normality assumption.
  - We simply proceed with this discrepancy measure just as we did with the earlier measures. The only difference is that now we need to first calculate the denominator (the standard error).
- Below is a factory function that will return a function that calculates this discrepancy measure for any two sub-populations for a given variate var (assuming the variance of the sub-populations are equal)

```
### The t statistic

getDiscrepancyFn <- function(var) {
function(pop) {
    ## First sub-population
    pop1 <- pop$pop1
    n1 <- nrow(pop1)
    m1 <- mean(pop1[, var])
    v1 <- var(pop1[, var])

    ## Second sub-population
    pop2 <- pop$pop2
    n2 <- nrow(pop2)
    m2 <- mean(pop2[, var])
    v2 <- var(pop2[, var])

    ## Pool the variances
    v <- ((n1 - 1) * v1 + (n2 - 1) * v2)/(n1 + n2 - 2)

    ## Determine the t-statistic
    t <- (m1 - m2)/sqrt(v * ((1/n1) + (1/n2)))

    ## Return the t-value
    t
}
```

```
}
```

In many instances, even when no normality distribution is assumed, the Student t distribution will roughly approximate the histogram that arises from randomly mixing the sub-populations.

This in fact was one of the early justifications (by R.A. Fisher) for using the t distribution broadly in application; namely that it approximated the random mixing procedure.

4.2.3 Multiple Testing

- In general, to compare sub-populations we could use any number of discrepancy measures $D_1, D_2, \ldots, D_K$
  - Each with an associated p-value: $p_1, p_2, \ldots, p_K$.
- However, when a *family* of statistical inferences is considered simultaneously one encounters the *multiple testing problem* (also known as the *multiple comparison problem*).
  - The more inferences made, the more likely an error is going to occur.
  - e.g., Even if H_0 is true, the more discrepancy measures (and hence tests) we consider, the more likely it becomes that one of them will erroneously suggest that null hypothesis should be rejected. Recall: Type I error: reject a true H_0.
- If a single test has probability α of yielding a Type I Error, then this probability becomes inflated when considering K simultaneous tests.
- Such an inflation is commonly quantified by two metrics:
  - The *family-wise error rate* (FWER) is the probability of making a Type I Error on *any* of the K tests.
  - The *false discovery rate* (FDR) is the expected number of Type I Errors in K tests.
- When multiple testing cannot be avoided, many statistical methods have been developed to control FWER and FDR at acceptable values:
  - Bonferroni Correction
  - sidak Correction
  - Holm-Bonferroni Method
  - Benjamini-Hochberg Procedure
  - ...
- However, we have been considering a special case in which we are testing the same hypothesis using different discrepancy values.
  - And so we can employ a more tailored solution which simply *combines* the information gained by each of the K discrepancy measures rather than considering them in isolation.

Combining Information Across Tests

- To consider the p-values collectively we might consider the smallest of them as measuring the combined evidence against the null hypothesis, i.e.

$$p\text{-value}_{\min} = \min_{k=1,\ldots,K} p_k$$

  - Note that this is appropriate only because the p-values are on a common (interpretable) scale, i.e., they're probabilities
  - We could not, for instance, combine discrepancy measures in the same way
- The smaller the value of $p\text{-value}_{\min}$, the greater is the evidence against the null hypothesis.
- Note: $p\text{-value}_{\min}$ is not a p-value in the traditional sense
  - but it is a measure of the evidence against the hypothesis.
- Thus we can construct a discrepancy measure out of it:

$$D^* = 1 - p\text{-value}_{\min}$$

  - D^* is defined so that large values, again, indicate evidence against the null hypothesis (unlike the p-value)
  - Therefore, D^* is a discrepancy measure.
- If the observed value of D^* is d^*_{obs}, then the p-value that describes this combined evidence is denoted by

$$p\text{-value}^* = \Pr(D^* \geq d^*_{\text{obs}} \mid H_0 \text{ is true})$$

- $p\text{-value}^*$ will necessarily be larger than $p\text{-value}_{\min}$ because
  - $p\text{-value}_{\min}$ is the smallest p-value among $p_1, p_2, \ldots, p_K$ and so
  - $p\text{-value}_{\min}$ exaggerates the evidence against the hypothesis and is misleading as a p-value.
- Given the data, all probabilities are proportions, hence D^* and $p\text{-value}^*$ can be calculated.

Estimating d^*_{obs}

- Suppose we have K discrepancy measures $D_1, D_2, \ldots, D_K$.
  - The combined discrepancy measure is $D^* = 1 - p\text{-value}_{\min}$.
- For $i = 1, \ldots, M_{\text{inner}}$ and each discrepancy $k = 1, \ldots, K$
  - randomly mix the two sub-populations $\mathcal{P}_1$ and $\mathcal{P}_2$ yielding $\mathcal{P}^*_{1,i}$ and $\mathcal{P}^*_{2,i}$
  - calculate $d_{k,i} = D_k(\mathcal{P}^*_{1,i}, \mathcal{P}^*_{2,i})$
  - then we estimate each p-value labelled as p_k with

$$\hat{p}_k = \frac{1}{M_{\text{inner}}} \sum_{i=1}^{M_{\text{inner}}} I(d_{k,i} \geq d_{\text{obs},k})$$

- Finally, we estimate p-value$_{\min}$ and hence d^*_{obs}:

$$\hat{d}^*_{\text{obs}} = 1 - \min_{k=1,\ldots,K} \hat{p}_k$$

Estimating p-value*

- In order to estimate p-value* we need an idea of how extreme $\hat{d}^*_{\text{obs}}$ is if H_0 were true.
- Thus we need to generate a distribution for D^* so that p-value* can be estimated.
  - This is achieved by repeating the steps above over and over again.
  - Conceptually this requires nested looping; an *inner* loop to calculate $\hat{d}^*_{\text{obs}}$ and an *outer* loop to generate a distribution of $\hat{d}^*_{\text{obs}}$ values.
- In particular, we repeat the following steps M_{outer} times:
  - randomly construct two sub-populations and
  - estimate $\hat{d}^*_j$ by the same procedure used to calculate d^*_{obs} (see above).
- then we estimate the p-value* with

$$p\text{-value}^* = \frac{1}{M_{\text{outer}}} \sum_{j=1}^{M_{\text{outer}}} I(d^*_j \geq \hat{d}^*_{\text{obs}})$$

4.3 Interval Estimation

4.3.1 Revisiting Sampling Distributions

Recall that when we look at $a(\mathcal{S})$ for all possible samples $\mathcal{S}$ of some size n from a population $\mathcal{P}$ and that the values of $a(\mathcal{S})$ have a distribution.

- We called this the sampling distribution of $\tilde{a}(\mathcal{S})$.

The normal distribution that best approximates the sampling distribution is the one with mean and standard deviation equal to the mean and standard deviation from all possible $\mathcal{P}(\mathcal{S})$ values.

The normal approximation provides a model for the sampling distribution and can be used as a basis to construct confidence intervals for population averages.

But, many attributes will have sampling distributions that are not approximately normal, so we are going to need another method too.

4.3.2 Random vs. Observed Intervals

- Suppose the attribute of interest is the population average $a(\mathcal{P}) = \mu = \frac{1}{N}\sum_{u\in\mathcal{P}} y_u = \bar{y}$
- Recall (from STAT 231) that the estimator $\tilde{a}(\mathcal{S}) = \tilde{\mu} = \bar{Y}$ (a random variable) has the following properties:

$$E[\bar{Y}] = \mu \quad \text{and} \quad \text{Var}[\bar{Y}] = \frac{\sigma^2}{n}$$

where

$$\sigma^2 = \frac{1}{N}\sum_{u\in\mathcal{P}}(y_u - \mu)^2$$

is the population variance, n is the sample size.

- If the normality assumption holds (this may be appropriate due to the CLT) then the estimator $\tilde{a}(\mathcal{P}) = \bar{Y}$ has the following distribution:

$$\bar{Y} \sim N\left(\mu, \frac{\sigma^2}{n}\right)$$

- Standardizing this random variable yields

$$Z = \frac{\bar{Y} - \mu}{\sigma/\sqrt{n}} \sim N(0,1)$$

- Note: Strictly speaking we are dealing with a finite population and so our variance term should include the finite population correction:

$$\text{Var}[\bar{Y}] = \frac{\sigma^2}{n}\left(\frac{N-n}{N-1}\right)$$

  - However, when $N \gg n$ then $\frac{N-n}{N-1} \to 1$ in which case this term can be ignored
  - This is why most texts omit the correction factor, giving rise to the more familiar formulation

$$\text{Var}[\bar{Y}] = \frac{\sigma^2}{n}$$

  - For completeness we will henceforth include the finite population correction, in which case

$$\bar{Y} \sim N\left(\mu, \frac{\sigma^2}{n}\left(\frac{N-n}{N-1}\right)\right)$$

and

$$Z = \frac{\bar{Y} - \mu}{\sigma/\sqrt{n}\sqrt{\frac{N-n}{N-1}}} \sim N(0,1)$$

Random Intervals

- Using the standardized random variable and specified $p \in (0,1)$ we can find a constant $c > 0$ such that

$$\begin{aligned}
1 - p &= \Pr(-c \le Z \le c) \\
1 - p &= \Pr\left(-c \le \frac{\bar{Y} - \mu}{\sigma/\sqrt{n}\sqrt{\frac{N-n}{N-1}}} \le c\right) \\
&= \Pr\left(\bar{Y} - c \times \frac{\sigma}{\sqrt{n}}\sqrt{\frac{N-n}{N-1}} \le \mu \le \bar{Y} + c \times \frac{\sigma}{\sqrt{n}}\sqrt{\frac{N-n}{N-1}}\right).
\end{aligned}$$

- Rearranging this statement yields a random interval which contains μ with probability $1-p$:

$$\left[\bar{Y} - c \times \frac{\sigma}{\sqrt{n}}\sqrt{\frac{N-n}{N-1}}, \bar{Y} + c \times \frac{\sigma}{\sqrt{n}}\sqrt{\frac{N-n}{N-1}}\right]$$

- This interval is considered random because it's never actually observed - it is defined in terms of random variables.
- However, observed intervals calculated by substituting $\bar{Y}$ with $\bar{y}$ are guaranteed to contain μ, $100(1-p)\%$ of the time.
  - $1-p$ is therefore called the coverage probability.
  - μ is contained in (or covered by) such an interval $100(1-p)\%$ of the time.
  - All these intervals have the same width, they just have different (random) centres.

A Note on determining c:

- The normal distribution is symmetric about its mean μ, so p and c are related through

$$1 - p = \Pr(-c \leq Z \leq c)$$

  or, equivalently,

$$1 - p/2 = \Pr(Z \leq c)$$

  where $Z \sim N(0,1)$ is a standard normal random variable.

- Therefore, given any $p \in (0,1)$ the value of c can be determined from the quantile function of a standard normal random variable:

$$c = Q_Z\left(1 - \frac{p}{2}\right)$$

  which in R is calculated as `qnorm(1 - p/2)`.
  - e.g., $c \approx 1.96$ when $1-p = 0.95$ for a standard normal random variable.

Observed Intervals

- In practice, we will have only one sample (this is the one you can actually calculate given a sample of data)
  - And thus a single numerical average $\bar{y}$
  - And thus a single instance of these randomly generated intervals:

$$\left[\bar{y} - c \times \frac{\sigma}{\sqrt{n}}\sqrt{\frac{N-n}{N-1}}, \bar{y} + c \times \frac{\sigma}{\sqrt{n}}\sqrt{\frac{N-n}{N-1}}\right]$$

- Such observed intervals are referred to as confidence intervals and you must take care to distinguish them conceptually from a random interval.
  - In the context of random intervals, probabilistic statements are sensible.

  - However, an observed confidence interval is not random – it either contains μ or it doesn't.
  - The probability that your interval contains μ is 0 or 1!

- Probability statements are in reference to the *method used to generate the intervals*, NOT to the particular interval we have observed.
- If the normality assumption holds up, $100(1-p)\%$ of such intervals will contain μ,
  - We thus have some *confidence* that our particular observed interval will contain μ as well, but unfortunately we'll never know if it does.
- The larger $1-p$, the *more confident* we are that the interval will contain μ.

4.3.3 Student t Based Intervals

Standard Error vs. Standard Deviation

- In the previous section the confidence intervals we calculated assumed $SD[\bar{Y}]$ was known
  - This is an unrealistic assumption.
  - Only very rarely would we have this value.
- However, for many sample attributes $a(\mathcal{S})$ (e.g., Horvitz-Thompson estimators), we can estimate the standard deviation $SD[\tilde{a}(\mathcal{S})]$ of the sampling distribution of $a(\mathcal{S})$.
  - The standard error is an estimate of the standard deviation of the corresponding estimator:

$$SE[\tilde{a}(\mathcal{S})] = \widehat{SD}[\tilde{a}(\mathcal{S})]$$

  - We could use

$$\frac{a(\mathcal{S}) - a(\mathcal{P})}{SE[\tilde{a}(\mathcal{S})]}$$

    instead of

$$\frac{a(\mathcal{S}) - a(\mathcal{P})}{SD[\tilde{a}(\mathcal{S})]}$$

    as pivotal quantities.
    - Note that using the estimate SE in place of SD will increase the variability of the random intervals.
    - The corresponding estimator has much more variability - since we have to estimate SD now as well.

If the data are normally distributed we have the following distributional result

$$\frac{\bar{Y} - \mu}{\frac{\tilde{\sigma}}{\sqrt{n}}\sqrt{\frac{N-n}{N-1}}} \sim t_{n-1}$$

This statistic is called a pivotal quantity because

- it is a function of the sample data $(Y_u, u \in \mathcal{S})$ and unknown parameter (μ) and its sampling distribution is completely known.

Now if we suppose that

$$\frac{\tilde{a}(\mathcal{S}) - a(\mathcal{P})}{\widetilde{SD}[\tilde{a}(\mathcal{S})]} \sim t_{n-1}$$

Then we choose a $p \in (0, 1)$ and a corresponding $c > 0$ with

$$1 - p = \Pr\left(-c \leq \frac{\tilde{a}(\mathcal{S}) - a(\mathcal{P})}{\widetilde{SD}[\tilde{a}(\mathcal{S})]} \leq c\right) = \Pr\left(\tilde{a}(\mathcal{S}) - c \times \widetilde{SD}[\tilde{a}(\mathcal{S})] \leq \mu \leq \tilde{a}(\mathcal{S}) + c \times \widetilde{SD}[\tilde{a}(\mathcal{S})]\right)$$

This random interval has both a random center and a random length.

The value of c is determined using the t distribution with $n - 1$ degrees of freedom.

- In R use `qt (1-p/2, df = n-1)` to get the value of c
- You can check that $c \approx 2.78$ when $1 - p = 0.95$ for a t_4 random variable.
- $c = Q_t(1 - p/2)$

In the special case that $a(\mathcal{P}) = \bar{Y}$ is the population average then the standard deviation is

$$SD[\tilde{a}(\mathcal{S})] = \frac{\sigma}{\sqrt{n}}\sqrt{\frac{N-n}{N-1}}$$

which is a true unknown value, and the standard deviation estimator is

$$\widetilde{SD}[\tilde{a}(\mathcal{S})] = \frac{\tilde{\sigma}}{\sqrt{n}}\sqrt{\frac{N-n}{N-1}}$$

which is a random variable, and the standard deviation estimate (the standard error) is

$$SE[\tilde{a}(\mathcal{S})] = \widehat{SD}[\tilde{a}(\mathcal{S})] = \frac{\hat{\sigma}}{\sqrt{n}}\sqrt{\frac{N-n}{N-1}}$$

which is an observed estimated value.

Thus the random interval in this case is

$$\left[\bar{Y} - c \times \frac{\tilde{\sigma}}{\sqrt{n}}\sqrt{\frac{N-n}{N-1}}, \bar{Y} + c \times \frac{\tilde{\sigma}}{\sqrt{n}}\sqrt{\frac{N-n}{N-1}}\right]$$

And the corresponding observed confidence interval is

$$\left[\bar{y} - c \times \frac{\hat{\sigma}}{\sqrt{n}}\sqrt{\frac{N-n}{N-1}}, \bar{y} + c \times \frac{\hat{\sigma}}{\sqrt{n}}\sqrt{\frac{N-n}{N-1}}\right]$$

Note: In the formulae above σ denotes the population standard deviation, which may be estimated in a

sample by

$$\hat{\sigma} = \sqrt{\frac{\sum_{u \in \mathcal{S}} (y_u - \bar{y})^2}{n}}$$

A Note on Pivotal Quantities:

- Pivotal quantities are the basis for constructing random intervals.
- Many random intervals are constructed via pivotal quantities such as

$$\frac{\tilde{a}(\mathcal{S}) - a(\mathcal{P})}{\widetilde{SD}[\tilde{a}(\mathcal{S})]}$$

  - The term *pivot* comes from the fact that with this quantity (which is a function of both $\mathcal{S}$ and $\mathcal{P}$) we are able to pivot and isolate for $a(\mathcal{P})$
  - This is the general prescription for constructing random intervals.
- This particular pivotal is a t-like pivotal, but it's not the *only* form of pivotal quantity.
  - One common pivotal quantity used for scale attributes $s(\cdot)$, are of the form

$$\frac{\tilde{s}(\mathcal{S})}{s(\mathcal{P})}$$

  - e.g., this gives rise to χ^2-based intervals for population variances.

Note:

- Random intervals for population attributes can be constructed by picking an appropriate pivotal quantity and pivoting.
- The corresponding CI is obtained by substituting sample estimates for estimators.
- However, all of this assumes we know the distribution of the pivotal quantity.

4.4 Resampling

Resampling methods aim to *mimic* this process by repeatedly sampling $\mathcal{S}$ as if it were $\mathcal{P}$

- In particular, we draw B samples $\mathcal{S}_1^*, \ldots, \mathcal{S}_B^*$ of size n independently from a population $\mathcal{P}^*$.
- Ideally, $\mathcal{P}^*$ would be the study population $\mathcal{P}$, but as already mentioned this would require repeated sampling from the population
- Instead we use a sample $\mathcal{S}$ as an estimate of the population $\mathcal{P}$, i.e., $\hat{\mathcal{P}} = \mathcal{S}$, or in usual bootstrap notation $\mathcal{P}^* = \mathcal{S}$

The sample population has only n units, so the without-replacement sampling mechanism will immediately exhaust the population. Therefore, we sample with replacement.

Thus an approximate sampling distribution is obtained by drawing B samples $\mathcal{S}_1^*, \ldots, \mathcal{S}_B^*$ of size n from $\mathcal{P}^*$ with replacement and on each bootstrap sample calculate the attribute value: $a(\mathcal{S}_1^*), \ldots, a(\mathcal{S}_B^*)$

4.4.1 The Bootstrap Method

The distribution of any attribute over the bootstrap samples $\mathcal{S}_i^*$ ($i = 1, 2, \ldots, B$) from $\mathcal{P}^*$ is a bootstrap estimate of the distribution of the same attribute over all possible samples $\mathcal{S}_i$ from $\mathcal{P}$.

This approach to mimicking the sampling distribution was named the bootstrap method when it was first proposed in 1979 by Bradley Efron.

The word "bootstrap" conveys the notion of starting something up from nothing as in "pulling oneself over a fence by one's bootstraps".

- It suggests something for nothing, or something impossible to achieve.

Bootstrap Standard Deviation

In general, for any attribute $a(\mathcal{P})$, the standard deviation of the corresponding sample attribute's estimator can be estimated from the bootstrap distribution with

$$\widehat{SD}_*[a(\mathcal{S}^*)] = \sqrt{\frac{\sum_{b=1}^{B}(a(\mathcal{S}_b^*) - \hat{a}^*)^2}{B-1}}$$

where $\hat{a}^* = \frac{1}{B}\sum_{b=1}^{B} a(\mathcal{S}_b^*)$ is the average of the attribute over the bootstrap samples $\mathcal{S}_1^*, \ldots, \mathcal{S}_B^*$.

Since B is usually large, it does not make any practical difference whether we use B or $B-1$ in the denominator of the standard deviation.

This is an estimate of the standard deviation of the sampling distribution for the attribute $a(\mathcal{S})$, which we called the standard error.

Since B is usually large, it does not make any practical difference whether we use B or $B-1$ in the denominator of the standard deviation.

This is an estimate of the standard deviation of the sampling distribution for the attribute $a(\mathcal{S})$, which we called the standard error.

Special Case: Inference for a Population Average

In the special case of the arithmetic average $a(\mathcal{S}) = \frac{1}{n}\sum_{u\in\mathcal{S}} y_u$ the bootstrap estimate of its standard deviation is

$$\widehat{SD}_*[\bar{Y}] = \sqrt{\frac{\sum_{b=1}^{B}(\bar{y}_b^* - \bar{y}^*)^2}{B-1}}$$

where $\bar{y}^* = \frac{1}{B}\sum_{b=1}^{B} \bar{y}_b^*$.

But we also know that the standard deviation can be estimated with

$$\widehat{SD}[\bar{Y}] = \frac{\hat{\sigma}}{\sqrt{n}}\sqrt{\frac{N-n}{N-1}}$$

where $\hat{\sigma} = \sqrt{\frac{\sum_{u\in\mathcal{S}}(y_u - \bar{y})^2}{n}}$.

Note: if the sample is not a good representation of the population, these two numbers might be quite

different.

A Comment on n Versus $n-1$

In the calculations above the function sdn(...) was introduced and used which is an implementation of

$$\widehat{\sigma} = \sqrt{\frac{\sum_{u\in\mathcal{S}}(y_u - \bar{y})^2}{n}}$$

which has n as a divisor. The built-in function sd(...) in R is an implementation of

$$\widehat{\sigma} = \sqrt{\frac{\sum_{u\in\mathcal{S}}(y_u - \bar{y})^2}{n-1}}$$

For bootstrap confidence interval calculations, a divisor of n is preferred (since we are treating the sample as a population)

- This version has the advantage of being replication invariant.
- Replication invariant estimates are preferred and are often called plug in estimates in the bootstrap literature e.g., see Efron and Tibshirani (1994).

However, when n is reasonably large, there will be little practical difference between the two.

4.4.2 Bootstrap Confidence Intervals

Naive Normal-Theory Intervals

Recall that confidence intervals for sample averages tend to have the following structure:

$$[\bar{Y} - c\widehat{SD}(\bar{Y}), \bar{Y} + c\widehat{SD}(\bar{Y})]$$

Under an assumption of normality we might pick c such that $P(Z \leq c) = 1 - p/2$ generates a $100(1-p)\%$ confidence interval.

If the bootstrap distribution is approximately normal, rather than estimating $\widehat{SD}(\bar{Y})$ by $\frac{\widehat{\sigma}}{\sqrt{n}}\sqrt{\frac{N-n}{N-1}}$, we might estimate $\widehat{SD}(\bar{Y})$ using the standard deviation of the bootstrap distribution of Y. The attraction of this approach, if it works, is that the same approach could be used for any attribute $a(\mathcal{S})$.

A 95% naive normal theory bootstrap interval for a population attribute $a(\mathcal{P})$ is

$$a(\mathcal{S}) \pm 1.96\widehat{SD}_*[a(\mathcal{S})]$$

where $\widehat{SD}_*$ is the bootstrap estimate of the standard deviation.

4.4.3 Bootstrap-t Confidence Intervals

Introduction

- When $a(\mathcal{S}) = \bar{y}$, we have seen that the quantity

$$Z = \frac{\widetilde{a}(\mathcal{S}) - a(\mathcal{P})}{\widehat{SD}[\widetilde{a}(\mathcal{S})]}$$

  - is approximately pivotal and
  - its sampling distribution (over all possible samples) is well approximated by a t-density.

- We can confirm this approximation, using simulation by
  - generating $\mathcal{S}_1, \ldots, \mathcal{S}_M$
  - then for each sample calculate

$$Z_i = \frac{a(\mathcal{S}_i) - a(\mathcal{P})}{\widehat{SE}[a(\mathcal{S}_i)]} = \frac{a(\mathcal{S}_i) - a(\mathcal{P})}{\widehat{SD}[\widetilde{a}(\mathcal{S}_i)]}$$

- Then an estimate of t_{n-1} can be constructed with $\{Z_1, \ldots Z_M\}$.
- This means we require an estimate of the standard deviation of the estimator, i.e., a standard error.
- e.g., when $a(\mathcal{S})$ is the average, one estimate is

$$SE[\widetilde{a}(\mathcal{S}_i)] = \widehat{SD}[\widetilde{a}(\mathcal{S}_i)] = \frac{\hat{\sigma}}{\sqrt{n}} \times \sqrt{\frac{N-n}{N-1}}$$

The General Approach

For a given sample $\mathcal{S}$, attribute $a(\mathcal{S})$, and standard error $\widehat{SD}[\widetilde{a}(\mathcal{S})]$:

- Calculate $a(\mathcal{S})$ and $\widehat{SD}[\widetilde{a}(\mathcal{S})]$
- Generate B bootstrap samples $\mathcal{S}_1^*, \ldots, \mathcal{S}_B^*$ from $\mathcal{S}$

For each of the B bootstrap samples from above:

- Calculate $a(\mathcal{S}_b^*)$ and $\widehat{SD}[\widetilde{a}(\mathcal{S}_b^*)]$ and then

$$z_b^* = \frac{a(\mathcal{S}_b^*) - a(\mathcal{S})}{\widehat{SD}[\widetilde{a}(\mathcal{S}_b^*)]}$$

From the values $z_1^*, \ldots, z_B^*$, find:

- $c_{\text{lower}} = Q_z(p/2)$
- $c_{\text{upper}} = Q_z(1 - p/2)$

Then a $100(1-p)\%$ bootstrap-t confidence interval is:

$$\Big(a(\mathcal{S}) - c_{\text{upper}} \times \widehat{SD}[\widetilde{a}(\mathcal{S})],\ a(\mathcal{S}) - c_{\text{lower}} \times \widehat{SD}[\widetilde{a}(\mathcal{S})]\Big)$$

A Note on standard errors:

- This method (so far) requires an analytic form to calculate $\widehat{SD}[\widetilde{a}(\mathcal{S})]$ (i.e., the standard deviation of the estimator given a single sample)
- Another interval can be constructed using the bootstrap estimate of the standard error $\widehat{SD}_*[\widetilde{a}(\mathcal{S})]$
- When the bootstrap is used to calculate $\widehat{SD}_*[\widetilde{a}(\mathcal{S}_b^*)]$, this is called the double bootstrap which we will look at in next section.

4.4.4 The Double Bootstrap

When $a(\mathcal{S}) = \bar{y}$ we have an analytic form for its standard deviation:

$$SD[\bar{y}] = \frac{\sigma}{\sqrt{n}} \times \sqrt{\frac{N-n}{N-1}}$$

where replacing σ by $\widehat{\sigma}$ gives an estimate $\widehat{SD}[\bar{y}]$ based on the sample values y_u for $u \in \mathcal{S}$.

More generally, when $a(\mathcal{S})$ is a Horvitz-Thompson estimate we also have an analytic form for $\widehat{SD}[\widetilde{a}(\mathcal{S})]$.

However, very often an analytic solution is not available for $\widehat{SD}[\widetilde{a}(\mathcal{S})]$:

- In which case an estimate can be obtained by using the bootstrap by generating $\mathcal{S}_1^*, \ldots, \mathcal{S}_B^*$ and calculating

$$\widehat{SD}_*[\widetilde{a}(\mathcal{S})] = \sqrt{\frac{\sum_{b=1}^{B}(a(\mathcal{S}_b^*) - \bar{a}^*)^2}{B-1}}$$

  where $\bar{a}^* = \frac{1}{B}\sum_{b=1}^{B} a(\mathcal{S}_b^*)$

But in the bootstrap-t, we need an estimate of $SD[\widetilde{a}(\mathcal{S}_b^*)]$ for each bootstrap sample $\mathcal{S}_b^*$!

In order to determine $\widehat{SD}[\widetilde{a}(\mathcal{S}_b^*)]$ for each bootstrap sample $\mathcal{S}_b^*$ we use the double bootstrap:

- We apply the bootstrap method to each bootstrap sample $\mathcal{S}_b^*$.
- To apply a bootstrap within a bootstrap for each bootstrap sample $\mathcal{S}_b^*$:
  - We generate D bootstrap samples, $\mathcal{S}_1^{**}, \ldots, \mathcal{S}_D^{**}$, each with replacement from a population now defined as $P^{**} = \mathcal{S}_b^*$.
  - The standard deviation of the corresponding values $a(\mathcal{S}_1^{**}), \ldots, a(\mathcal{S}_D^{**})$ will provide the estimate $\widehat{SD}_*[a(\mathcal{S}_b^*)]$.
- This estimate $\widehat{SD}_*[\widetilde{a}(\mathcal{S}_b^*)]$ is then substituted into The General Approach for bootstrap-t confidence intervals.

4.4.5 The Percentile Method

The sampling distribution of $\widetilde{a}(\mathcal{S})$ and can be estimated using a sample $\mathcal{S}$ and the bootstrap.

- So why not simply use quantiles from the bootstrap distribution to directly construct a confidence interval?

The **percentile method** for bootstrap confidence intervals is the following:

- For a given sample $\mathcal{S}$ generate B bootstrap samples $\mathcal{S}_1^*, \ldots, \mathcal{S}_B^*$ by sampling with replacement from the sample $\mathcal{S}$.
- For the b^{th} bootstrap sample (b = 1, ..., B), calculate $a_b = a(\mathcal{S}_b^*)$.
- From the values $a_1, \ldots, a_B$, find $a_{\text{lower}} = Q_a(p/2)$ and $a_{\text{upper}} = Q_a(1 - p/2)$.
- Then the $100(1-p)\%$ confidence interval is $[a_{\text{lower}}, a_{\text{upper}}]$.

This approach is **equivariant** to any 1:1 transformation of the attribute, say $T(a(\mathcal{P}))$:

- For an increasing function $T(\cdot)$: the corresponding interval for $T(a(\mathcal{P}))$ is simply $[T(a_{\text{lower}}), T(a_{\text{upper}})]$.
- For a decreasing function $T(\cdot)$: the corresponding interval for $T(a(\mathcal{P}))$ is simply $[T(a_{\text{upper}}), T(a_{\text{lower}})]$.
- So, we only need to determine the values a_{lower} and a_{upper} once for $a(\mathcal{P})$ and we have them for any $T(a(\mathcal{P}))$.

Comments:

- Simplicity is the attraction of this method, and explains its continued popularity.
- This method is transformation equivariant.
- The coverage probability is often incorrect if the distribution of the estimator is not nearly symmetric.
  - Coverage may be improved by considering "bias-corrected" versions of the percentile method. This, however, is outside the scope of this course. For more information on these alternatives see Section 11.3 in the book *Computer Age Statistical Inference* by Efron and Hastie.

CHAPTER 5: Prediction

5.1 Accuracy of Prediction

One measure of inaccuracy over the population $\mathcal{P}$ (of size N) is the average prediction squared error (APSE)

$$\frac{1}{N}\sum_{u\in\mathcal{P}}(y_u - \hat{\mu}(x_u))^2$$

Note that the quantity above is proportional to the more familiar *residual sum of squares*

$$\sum_{i=1}^{N} r_i^2 = \sum_{i=1}^{N}(y_i - \hat{\mu}(x_i))^2$$

which sometimes also referred to as the estimated residual mean squared error.

5.1.1 Example: Loblolly Pine Trees

An example of analysis on a dataset can be found from the course notes.

5.1.2 Example: Global Temperature Data

An example of analysis on a dataset can be found from the course notes.

5.1.3 Measuring Inaccuracy (Fairly)

So far, we have been fitting models of various complexities to data, and comparing them on the basis of their average prediction squared error (APSE):

$$\frac{1}{N}\sum_{u\in\mathcal{P}}(y_u - \hat{\mu}(x_u))^2$$

HOWEVER: In this approach we estimate the predictor function and measure its accuracy using the exact same set of observations.

Our current inaccuracy measure can thus be written as

$$APSE(\mathcal{P}, \hat{\mu}_{\mathcal{P}}) = \frac{1}{N}\sum_{u\in\mathcal{P}}(y_u - \hat{\mu}_{\mathcal{P}}(x_u))^2 = \frac{1}{N}\sum_{u\in\mathcal{P}}(y_u - \hat{\mu}_{\mathcal{P}}(x_u))^2$$

where the notation $\hat{\mu}_{\mathcal{P}}(x_u)$ emphasizes the fact that the predictor function was determined from the entire population.

This isn't the most honest way to estimate a predictor function's accuracy.

- This method will underestimate the APSE for predictions at values of x not existing in the data (i.e., new/different values).

- The dataset used to fit (train) the model is the same dataset we use to evaluate (test) the model.
- This (bad) approach leads to a problem known as overfitting which will be discussed in more detail below.

Ideally (to provide a fair evaluation of prediction accuracy) we would use different data to train vs. test the model and our measure of inaccuracy would reflect this.

- We should estimate the predictor function using a sample $\mathcal{S}$ (sometimes called the training set)
- AND measure the inaccuracy over the population $\mathcal{P}$, or over the units in the population not included in the sample: $\mathcal{T} = \mathcal{P} \setminus \mathcal{S}$ (sometimes called the test set)

In this case we could write the APSE as

$$APSE(\mathcal{P}, \hat{\mu}_{\mathcal{S}}) = \frac{1}{N} \sum_{u \in \mathcal{P}} (y_u - \hat{\mu}_{\mathcal{S}}(x_u))^2$$

This notation emphasizes that the estimate of the predictor function $\hat{\mu}$ is based on a sample $\mathcal{S}$

Since $\mathcal{P} = \mathcal{S} \cup \mathcal{T}$ and $\mathcal{S} \cap \mathcal{T} = \varnothing$ (i.e., T is the complement set of $\mathcal{S}$ in $\mathcal{P}$) the APSE as defined in this way can be decomposed into a sum of two pieces:

$$\begin{aligned} APSE(\mathcal{P}, \hat{\mu}_{\mathcal{S}}) &= \frac{1}{N} \sum_{u \in \mathcal{P}} (y_u - \hat{\mu}_{\mathcal{S}}(x_u))^2 \\ &= \left(\frac{n}{N}\right) \frac{1}{n} \sum_{u \in \mathcal{S}} (y_u - \hat{\mu}_{\mathcal{S}}(x_u))^2 + \left(\frac{N-n}{N}\right) \frac{1}{N-n} \sum_{u \in \mathcal{T}} (y_u - \hat{\mu}_{\mathcal{S}}(x_u))^2 \\ &= \left(\frac{n}{N}\right) APSE(\mathcal{S}, \hat{\mu}_{\mathcal{S}}) + \left(\frac{N-n}{N}\right) APSE(\mathcal{T}, \hat{\mu}_{\mathcal{S}}) \end{aligned}$$

Given that interest often lies in the quality of the predictions outside of the sample

- we might exclusively calculate average prediction squared error over T

$$APSE(\mathcal{T}, \hat{\mu}_{\mathcal{S}}) = \frac{1}{N-n} \sum_{u \in \mathcal{T}} (y_u - \hat{\mu}_{\mathcal{S}}(x_u))^2.$$

- But clearly if $n \ll N$, the value $APSE(\mathcal{T}, \hat{\mu}_{\mathcal{S}})$ will not be that different from $APSE(\mathcal{P}, \hat{\mu}_{\mathcal{S}})$.

Note that it is not expected that the APSE over $\mathcal{S}$ is higher than the APSE over $\mathcal{T}$

We find that $\text{APSE}(\mathcal{S}, \hat{\mu}_{\mathcal{S}}) > \text{APSE}(\mathcal{T}, \hat{\mu}_{\mathcal{S}})$, but this is unusual. Typically, we'd expect this to be the other way around. This will be more apparent with more complicated models.

Generally speaking, as model complexity increases, $\text{APSE}(\mathcal{S}, \hat{\mu}_{\mathcal{S}})$ keeps decreasing while $\text{APSE}(\mathcal{T}, \hat{\mu}_{\mathcal{S}})$ decreases to some point and then increases.

At around degree equal to x (the elbow) when fitting a large set of polynomials model to the data, we may find that increasing complexity does not improve prediction accuracy.

- Increasing complexity will continue to improve predictions on the sample but not on the rest of the

population.

- This effect is what we previously referred to as overfitting:
  - the predictor function has been too closely tailored to the peculiarities of the sample. (training set)
  - the complexity of the model has been increased so far that it has compromised the *out-of-sample* prediction performance. (poor performance on unobserved data)

5.1.4 The Importance of a Good Sample

Since $\hat{\mu}_{\mathcal{S}}(x)$ is based on a single sample $\mathcal{S}$ the quality of the predictor function depends crucially on the quality of the sample

- if the sample is not a good/fair representation of the population, then any predictor function is bound to perform poorly

It is important, then, to recognize that the performance (i.e., APSE) associated with $\hat{\mu}_{\mathcal{S}}(x)$ could vary quite a lot from one sample to another.

In practice we tend to assume our sample is a good representation of the population

- But in case that's not true it's important to choose a predictor function that performs well no matter which sample was used to estimate it.
- *Simpler is often better.*

5.2 Prediction Over Multiple Samples

5.2.1 Calculating APSE Over Many Samples

The inaccuracy of a predictor function $\hat{\mu}(x)$ is measured by its average prediction squared error:

$$APSE(\mathcal{P}, \hat{\mu}_{\mathcal{S}}) = \frac{1}{N} \sum_{u \in \mathcal{P}} (y_u - \hat{\mu}_{\mathcal{S}}(x_u))^2$$

- where $\mathcal{S}$ is the sample we used to fit the function, and
- $\mathcal{T} = \mathcal{P} \setminus \mathcal{S}$ is the complement set of $\mathcal{S}$ such that the population $\mathcal{P} = \mathcal{S} \cup \mathcal{T}$ and $\mathcal{S} \cap \mathcal{T} = \varnothing$

The function $\hat{\mu}_{\mathcal{S}}(x)$ is an estimate of $\mu(x)$ based on the single sample $\mathcal{S}$ and

- its performance depends highly on the particular choice of sample
- its performance could vary quite a lot from one sample to another

It is important to choose a predictor function that performs well no matter which sample was used to estimate it.

Suppose that we have many (perhaps all possible) samples $\mathcal{S}_j$ for $j = 1, \ldots, M$.

- For each sample, we can calculate $\hat{\mu}_{\mathcal{S}_j}(x)$ and hence

$$APSE(\mathcal{P}, \hat{\mu}_{\mathcal{S}_j})$$

- We will have a sampling distribution for APSE based on M samples

The average $APSE$ over all M samples should be a better measure of the quality of a predictor function.

$$APSE(\mathcal{P}, \hat{\mu}) = \frac{1}{M}\sum_{j=1}^{M} APSE(\mathcal{P}, \hat{\mu}_{\mathcal{S}_j}) = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{N}\sum_{u\in\mathcal{P}}(y_u - \hat{\mu}_{\mathcal{S}_j}(x_u))^2$$

Note that in $APSE(\mathcal{P}, \hat{\mu})$, the estimator notation $\hat{\mu}$ is used to emphasize that the function is looking at the values of μ over many (perhaps all possible) samples $\mathcal{S}_j$.

5.2.2 Decomposing $APSE(\mathcal{P}, \tilde{\mu})$

Preamble

- Until now we have thought of prediction in the context of the response model

$$y = \mu(x) + \text{error}$$

  where y and $x = (x_1, \ldots, x_p)$ respectively represent our *observed* response and explanatory variate values

- The (user-specified) predictor function $\mu(x)$ is meant to approximate the *true underlying relationship* between y and x:

$$y = \tau(x)$$

  - the error term in the response model accounts for *model misspecification*, i.e., the difference between $\mu(x)$ and $\tau(x)$

- Note that when we observe the whole population $\mathcal{P}$, there is no uncertainty in the relationship between y and x
  - That is, we observe $\tau(x)$
- However, there may still be some uncertainty if, for example, the population contains duplicate x values that produce different y values
- For this reason, we define $\tau(x)$ to be the conditional average of y given x
  - Suppose that there are K different values of x in the population P: $x_1, \ldots, x_K$
  - Thus the population P can be partitioned according to the different values of x as

$$P = \bigcup_{k=1}^{K} \mathcal{A}_k$$

where

$$A_k = \{u : u \in P, x_u = x_k\}$$

is the collection of units who all have $x = x_k$, for the $k = 1, \ldots, K$

– The conditional average $\tau(x)$ can thus be expressed for each distinct x_k as

$$\tau(x_k) = \frac{1}{n_k} \sum_{u \in \mathcal{A}_k} y_u$$

where n_k is the number of units in $\mathcal{A}_k$.

5.3 Back to Reality: Predictions With a Single Sample

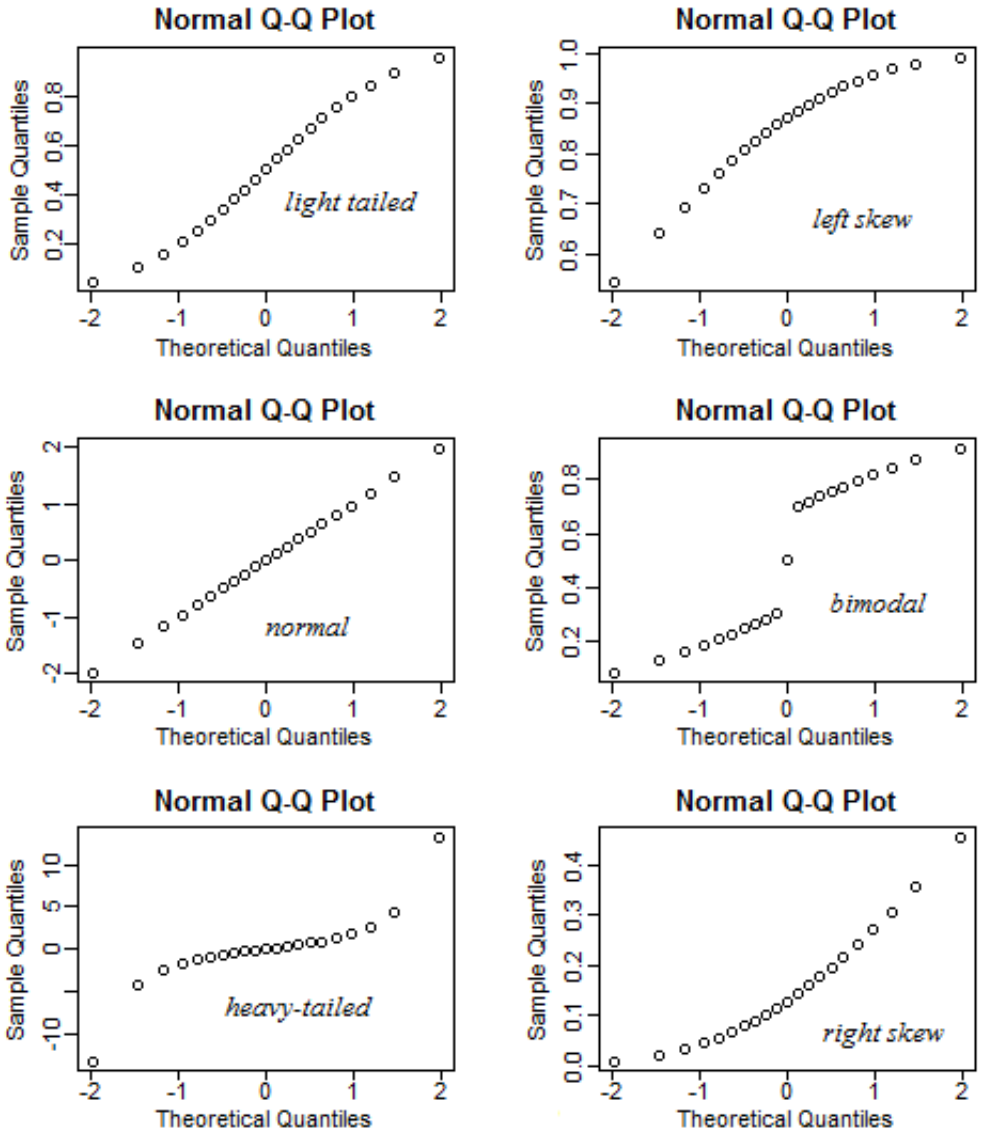


Figure 3: Q-Q plot