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	February 07, 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While I have made diligent efforts to ensure the accuracy of the content, there is a potential for errors, outdated information, or inaccuracies, especially in sections sourced from ChatGPT.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

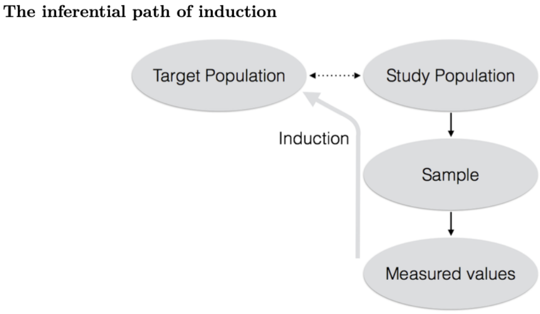


Figure 1: The inferential path of induction

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:

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

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:

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

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

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

Example 2.1

The population average is location equivariant:

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

But is the population variance location equivariant? No!

Definition 2.6

Scale Invariance and Equivariance

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

- scale invariant if

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

- scale equivariant if

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

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

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

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

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

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 shuold 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.

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

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

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

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

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 $\mathcal{P} = \{y_1, y_2, \dots, y_N\}$.

- Partition the range of the population into k non-overlapping intervals, called bins, $I_j = [a_{j-1}, a_j)$, for $j = 1, 2, \dots, k$ and then calculate the number (frequency) or proportion (relative frequency) of observations in the jth bin for $j = 1, \dots, k$.
- 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 popula- tion.

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:

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

Histograms using different rules for bin size selection:

- the first row is `Number of farms` and
- the second row is `log(Number of farms+1)`.

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")
```

The above content is for Lecture 3 on Jan 16, 2024

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.).

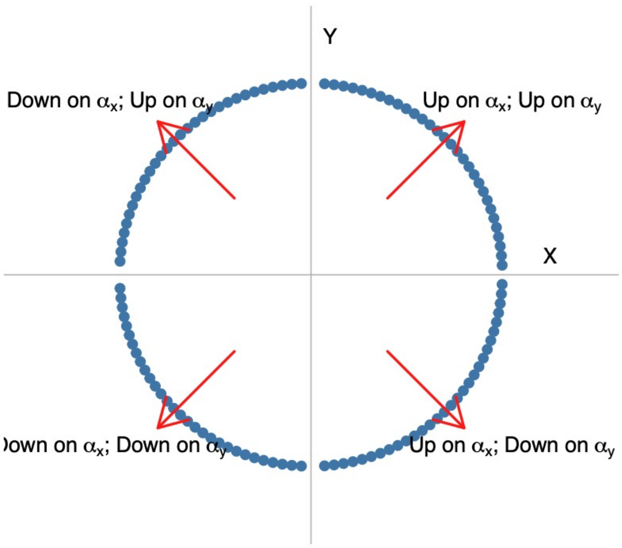
alpha	**ladder**
⋮	up
2	———
1	original values
$\frac{1}{2}$	———
$\frac{1}{3}$	———
0	———
$-\frac{1}{3}$	———
$-\frac{1}{2}$	———
-1	———
-2	———
⋮	down

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:

Each quadrant shows a monotonic curved relation


Y
X
Down on α_x; Up on α_y
Up on α_x; Up on α_y
Down on α_x; Down on α_y
Up on α_x; Down on α_y

Direction of the bump suggests ladder moves

Figure 2: Direction of the bump suggests ladder moves

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:

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

  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:

$$r_1, r_2, \dots, r_N$$

which are the ranks of the variate values $y_1, y_2, \dots, y_N$ 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
  - $p \in \left\{\frac{1}{N}, \frac{2}{N}, \dots, 1\right\}$ 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.

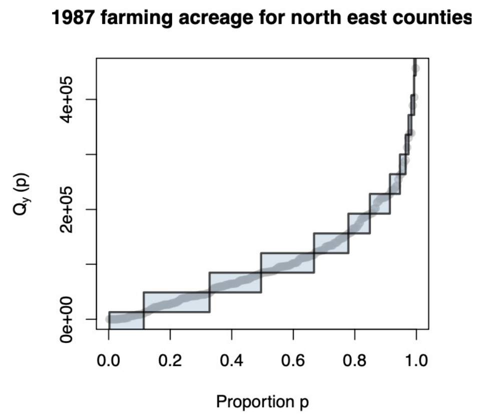
Quantiles that measure spread

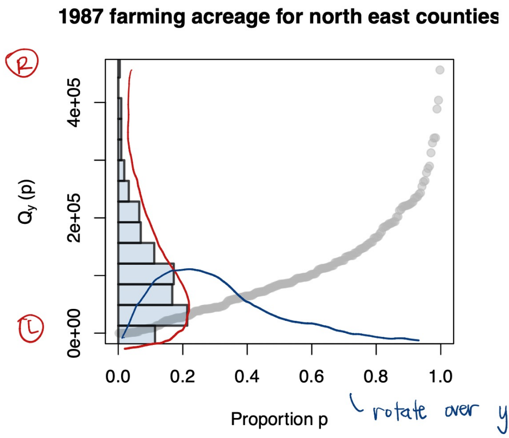
- The quantile function can also be used to provide some natural measures of spread for the variate y:
  - the range: $Q_y(1) - Q_y\left(\frac{1}{N}\right)$
  - the inter-quartile range: $IQR_y = Q_y\left(\frac{3}{4}\right) - Q_y\left(\frac{1}{4}\right)$
  - the central $100 \times p\%$ range
- Alternatively, 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!





rotate over $y=x$

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:

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

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

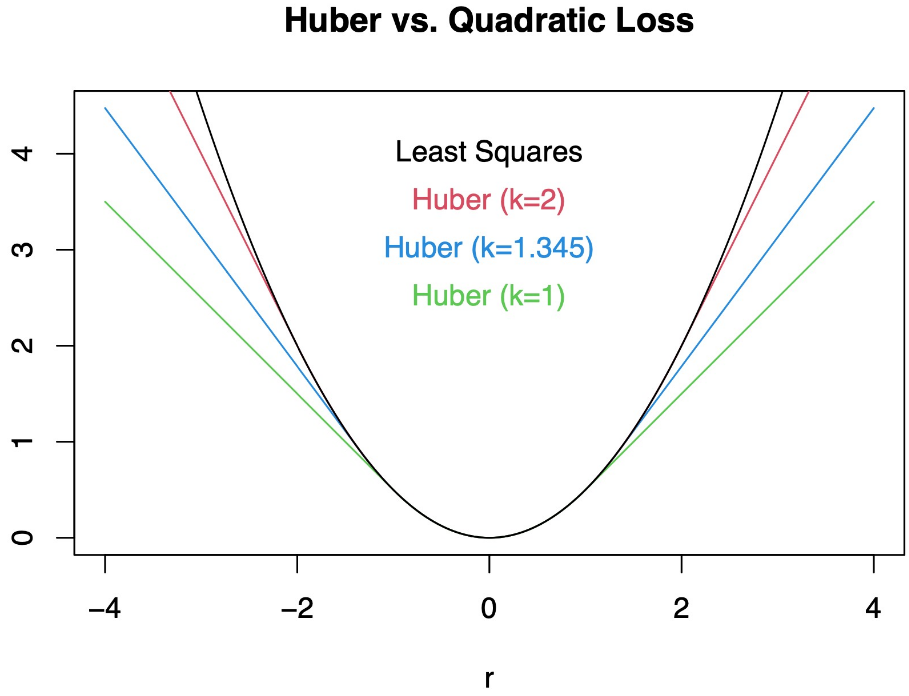
$$(\hat{\alpha}, \hat{\beta}) = \arg \min_{(\alpha,\beta)\in\mathbb{R}^2} \sum_{u\in P} \rho(y_u - \alpha - \beta(x_u - c))$$

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 $\rho(\hat{\theta}_{i+1}; P)$.

The Gradient Descent Algorithm

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

```
1. Initialize $i; \hat{\theta}_0$;
2. LOOP:
   (a) Gradient:
         $g_i = \nabla \rho(\theta; P)\Big|_{\theta=\hat{\theta}_i}$
   (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:

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

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

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

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

- 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 P into H non-overlapping groups (batches)

  $$P = B_1 \cup B_2 \cup \ldots \cup B_H$$

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

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

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 $\mathcal{P} = \{B_1, \ldots, B_H\}$
- 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:
    $$\hat{g}_i = \nabla \rho(\theta; B_{i \mod H})|_{\theta=\hat{\theta}_i}$$
    (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 $i = H + 1, thebatchisactuallyB_1$.

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.

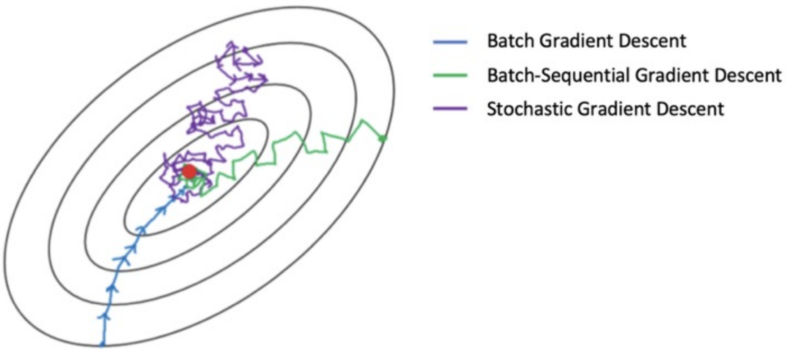
- Like batch-sequential gradient descent, the estimate $\hat{\theta}$ is updated after each batch (sample). Denoting the sample size by M, note that setting $M = 1$ gives rise to what is often simply referred to as stochastic gradient descent.

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


Batch Gradient Descent
Batch-Sequential Gradient Descent
Stochastic Gradient Descent

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).

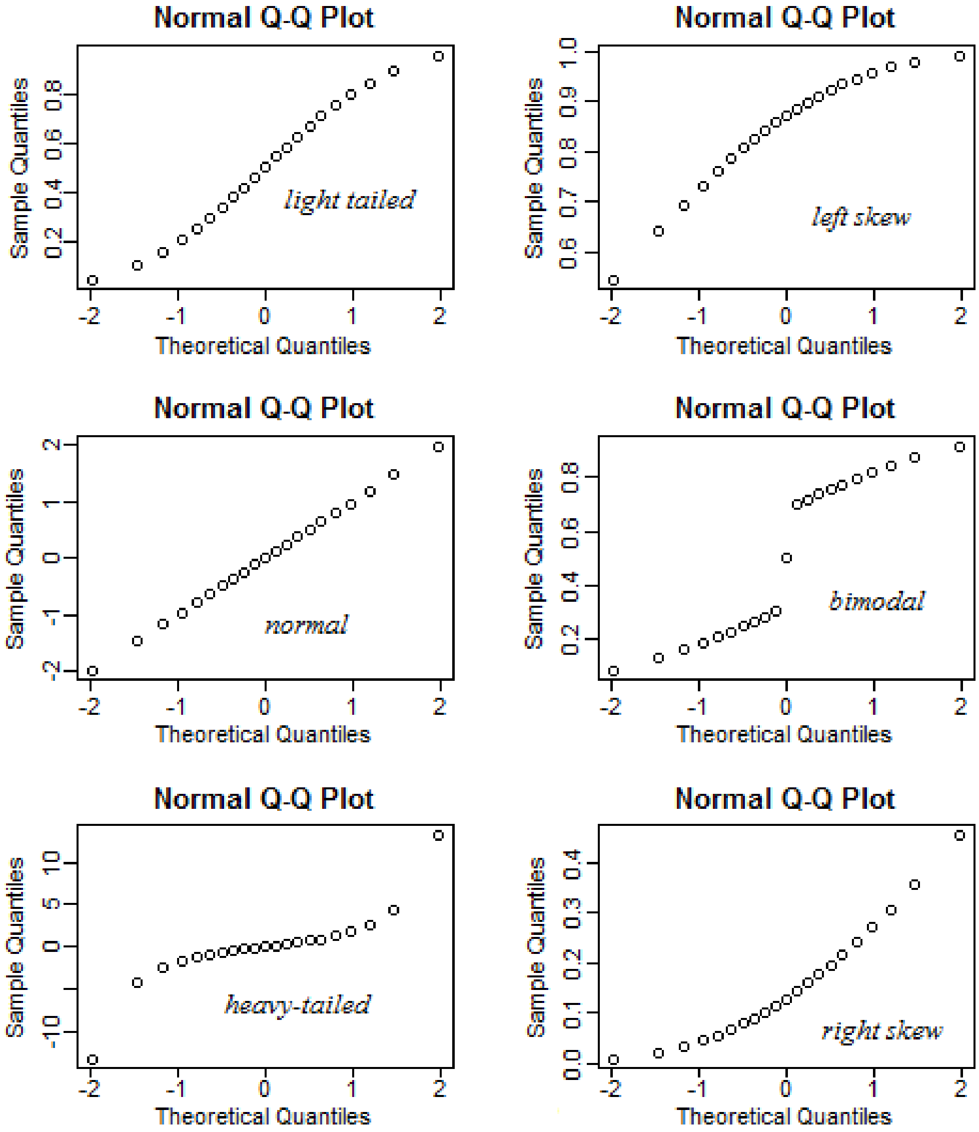


Figure 3: Q-Q plot