A Guide on Data Analysis

Mike Nguyen

2021-05-10

Contents

III RAMIFICATIONS 345

IV EXPERIMENTAL DESIGN 461

Chapter 1

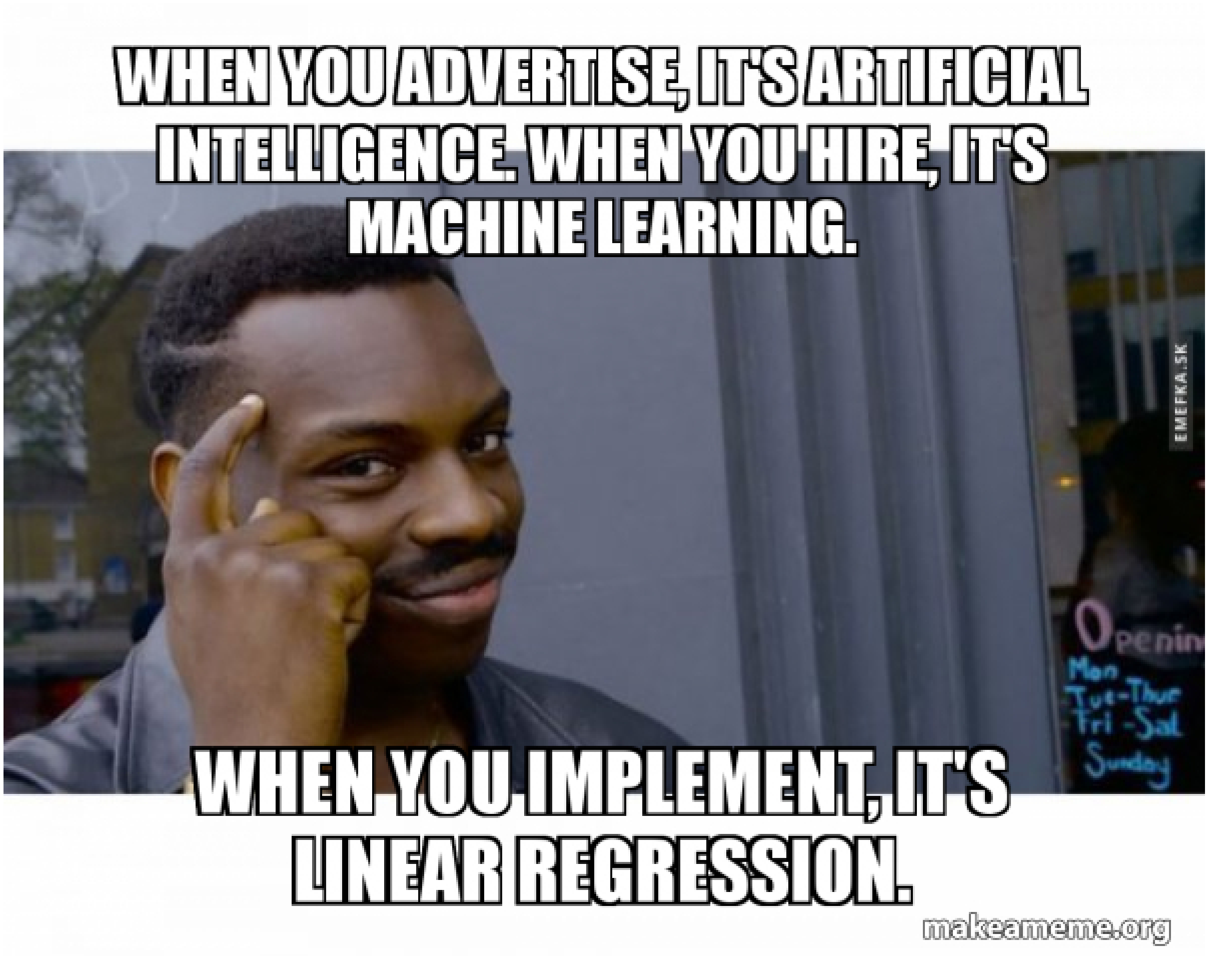
Introduction

This guide is an attempt to streamline and demystify the data analysis process.

By no mean this is an ultimate guide, or I am a great source of knowledge, or I claim myself to be a statistician/ econometrician, but I am a strong proponent of learning by teaching, and doing. Hence, this is more like a learning experience for both you and me.

Since the beginning of the century, we have been bombarded with amazing advancements and inventions, especially in the field of statistics, information technology, and computer science. However, I believe the downside of this introduction is that we use **big** and **trendy** words too often (i.e., big data, machine learning, deep learning).

It's all fun and exciting when I learned these new tools. But I have to admit that I hardly retain any of these new inventions.However, writing down from the beginning till the end of a data analysis process is the solution that I came up with. Accordingly, let's dive right in.


WHEN YOU ADVERTISE, IT'S ARTIFICIAL INTELLIGENCE. WHEN YOU HIRE, IT'S MACHINE LEARNING.
WHEN YOU IMPLEMENT, IT'S LINEAR REGRESSION.
EMEFKA.SK
Openin
Mon
Tue-Thur
Fri -Sat
Sunday
makeameme.org

Some general recommendation:

- The more you practice/habituate/condition, more line of codes that you write, more function that you memorize, I think the more you will like this journey.
- Readers can follow this book several ways:
  - If you are interested in particular methods/tools, you can jump to that section by clicking the section name.
  - If you want to follow a traditional path of data analysis, read the Linear Regression section.
  - If you want to create your experiment and test your hypothesis, read the Analysis of Variance (ANOVA) section.
- Alternatively, if you rather see the application of models, and disregard any theory or underlying mechanisms, you can skip to summary and ap-

plication portion of each section.

- If you don't understand a part, search the title of that part of that part on Google, and read more into that subject. This is just a general guide.
- If you want to customize your code beyond the ones provided in this book, run in the console `help(code)` or `?code`. For example, I want more information on `hist` function, I'll type in the console `?hist` or `help(hist)`.

- Another way is that you can search on Google. Different people will use different packages to achieve the same result in R. Accordingly, if you want to create a histogram, search on Google `histogram in R`, then you should be able to find multiple ways to create histogram in R.

Information in this book are from various sources, but most of the content is based on several courses that I have taken formally. I'd like to give professors credit accordingly.

Course	Professor
Data Analysis I	Erin M. Schliep
Data Analysis II	Christopher Wikle
Applied Econometric	Alyssa Carlson

Tools of statistics

- Probability Theory
- Mathematical Analysis
- Computer Science
- Numerical Analysis
- Database Management

Setup Working Environment

```r
if (!require("pacman"))
    install.packages("pacman")
if (!require("devtools"))
    install.packages("devtools")
library("pacman")
library("devtools")
```

Chapter 2

Prerequisites

This chapter is just a quick review of Matrix Theory and Probability Theory

If you feel you do not need to brush up on these theories, you can jump right into Descriptive Statistics

2.1 Matrix Theory

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \tag{2.1}$$

$$A' = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix} \tag{2.2}$$

$$(\mathbf{ABC})' = \mathbf{C}'\mathbf{B}'\mathbf{A}'\mathbf{A}(\mathbf{B}+\mathbf{C}) = \mathbf{AB}+\mathbf{AC}\mathbf{AB} \neq \mathbf{BA}(\mathbf{A}')' = \mathbf{A}(\mathbf{A}+\mathbf{B})' = \mathbf{A}'+\mathbf{B}'(\mathbf{AB})' = \mathbf{B}'\mathbf{A}'(\mathbf{AB})^{-\mathbf{1}} = \mathbf{B}^{-\mathbf{1}}$$

If A has an inverse, it is called **invertible.** If A is not invertible it is called **singular.**

$$\begin{aligned} \mathbf{A} &= \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} \\ &= \begin{pmatrix} a_{11}b_{11}+a_{12}b_{21}+a_{13}b_{31} & \sum_{i=1}^{3} a_{1i}b_{i2} & \sum_{i=1}^{3} a_{1i}b_{i3} \\ \sum_{i=1}^{3} a_{2i}b_{i1} & \sum_{i=1}^{3} a_{2i}b_{i2} & \sum_{i=1}^{3} a_{2i}b_{i3} \end{pmatrix} \end{aligned} \tag{2.3}$$

Let **a** be a 3 x 1 vector, then the quadratic form is

$$\mathbf{a}'\mathbf{B}\mathbf{a} = \sum_{i=1}^{3}\sum_{i=1}^{3} a_i b_{ij} a_j$$

Length of a vector
Let **a** be a vector, $||\mathbf{a}||$ (the 2-norm of the vector) is the length of vector **a**, is the square root of the inner product of the vector with itself:

$$||\mathbf{a}|| = \sqrt{\mathbf{a}'\mathbf{a}}$$

2.1.1 Rank

- Dimension of space spanned by its columns (or its rows).

- Number of linearly independent columns/rows

For a n x k matrix **A** and k x k matrix **B**

- $rank(A) \leq min(n, k)$
- $rank(A) = rank(A') = rank(A'A) = rank(AA')$
- $rank(AB) = min(rank(A), rank(B))$
- **B** is invertible if and only if rank(B) = k (non-singular)

2.1.2 Inverse

In scalar, a = 0 then 1/a does not exist. In matrix, a matrix is invertible when it's a non-zero matrix.

A non-singular square matrix A is invertible if there exists a non-singular square matrix B such that,

$$AB = I$$

Then $A^{-1} = B$. For a 2x2 matrix,

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

$$A^{-1} = \frac{1}{ad - bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

For the partition matrix,

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{(A - BD^{-1}C)^{-1}} & \mathbf{-(A - BD^{-1}C)^{-1}BD^{-}1} \\ \mathbf{-DC(A - BD^{-1}C)^{-1}} & \mathbf{D^{-1} + D^{-1}C(A - BD^{-1}C)^{-1}BD^{-1}} \end{bmatrix} \tag{2.4}$$

Properties for a non-singular square matrix

- $\mathbf{A^{-1}} = A$
- for a non-zero scalar b, $(\mathbf{bA})^{\mathbf{-1}} = \mathbf{b^{-1}A^{-1}}$
- for a matrix B, $(BA)^{-1} = B^{-1}A^{-1}$ only if B is non-singular
- $(\mathbf{A^{-1}})' = (\mathbf{A}')^{\mathbf{-1}}$
- Never notate $\mathbf{1/A}$

2.1.3 Definiteness

A symmetric square k x k matrix, **A**, is Positive Semi-Definite if for any non-zero k x 1 vector **x**,

$$\mathbf{x'Ax \geq 0}$$

A symmetric square k x k matrix, **A**, is Negative Semi-Definite if for any non-zero k x 1 vector **x**

$$\mathbf{x'Ax \leq 0}$$

A is indefinite if it is neither positive semi-definite or negative semi-definite.

The identity matrix is positive definite

Example Let $\mathbf{x} = (x_1 x_2)'$, then for a 2 x 2 identity matrix,

$$\begin{aligned} \mathbf{x'Ix} &= (x_1 x_2) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ &= (x_1 x_2) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ &= x_1^2 + x_2^2 > 0 \end{aligned} \tag{2.5}$$

Definiteness gives us the ability to compare matrices $\mathbf{A} - \mathbf{B}$ is PSD This property also helps us show efficiency (which variance covariance matrix of one estimator is smaller than another)

Properties

- any variance matrix is PSD
- a matrix **A** is PSD if and only if there exists a matrix **B** such that $\mathbf{A} = \mathbf{B'B}$
- if **A** is PSD, then $\mathbf{B'AB}$ is PSD
- if A and C are non-singular, then A-C is PSD if and only if $\mathbf{C^{-1}} - \mathbf{A^{-1}}$
- if A is PD (ND) then A^{-1} is PD (ND)

Note

- Indefinite A is neither PSD nor NSD. There is no comparable concept in scalar.
- If a square matrix is PSD and invertible then it is PD

Example:

1. Invertible / Indefinite

$$\begin{bmatrix} -1 & 0 \\ 0 & 10 \end{bmatrix}$$

2. Non-invertible/ Indefinite

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

3. Invertible / PSD

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

4. Non-Invertible / PSD

$$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

2.1.4 Matrix Calculus

$y = f(x_1, x_2, ..., x_k) = f(x)$ where x is a 1 x k row vector. The Gradient (first order derivative with respect to a vector) is,

$$\frac{\partial f(x)}{\partial x} = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ ... \\ \frac{\partial f(x)}{\partial x_k} \end{pmatrix}$$

The **Hessian** (second order derivative with respect to a vector) is,

$$\frac{\partial^2 f(x)}{\partial x \partial x'} = \begin{pmatrix} \frac{\partial^2 f(x)}{\partial x_1 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & ... & \frac{\partial^2 f(x)}{\partial x_1 \partial x_k} \\ \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \frac{\partial^2 f(x)}{\partial x_2 \partial x_2} & ... & \frac{\partial^2 f(x)}{\partial x_2 \partial x_k} \\ ... & ... & & ... \\ \frac{\partial^2 f(x)}{\partial x_k \partial x_1} & \frac{\partial^2 f(x)}{\partial x_k \partial x_2} & ... & \frac{\partial^2 f(x)}{\partial x_k \partial x_k} \end{pmatrix}$$

Define the derivative of $f(\mathbf{X})$ with respect to $\mathbf{X}_{(n \times p)}$ as the matrix

$$\frac{\partial f(\mathbf{X})}{\partial \mathbf{X}} = (\frac{\partial f(\mathbf{X})}{\partial x_{ij}})$$

Define **a** to be a vector and **A** to be a matrix which does not depend upon **y**. Then

$$\frac{\partial \mathbf{a}'\mathbf{y}}{\partial \mathbf{y}} = \mathbf{a}$$

$$\frac{\partial \mathbf{y}'\mathbf{y}}{\partial \mathbf{y}} = 2\mathbf{y}$$

$$\frac{\partial \mathbf{y}'\mathbf{A}\mathbf{y}}{\partial \mathbf{y}} = (\mathbf{A} + \mathbf{A}')\mathbf{y}$$

If **X** is a symmetric matrix then

$$\frac{\partial |\mathbf{X}|}{\partial x_{ij}} = \begin{cases} X_{ii}, i = j \\ X_{i}j, i \neq j \end{cases}$$

where X_{ij} is the (i,j)th cofactor of **X**

If **X** is symmetric and **A** is a matrix which does not depend upon **X** then

$$\frac{\partial tr\mathbf{X}\mathbf{A}}{\partial \mathbf{X}} = \mathbf{A} + \mathbf{A}' - diag(\mathbf{A})$$

If **X** is symmetric and we let $\mathbf{J}_{ij}$ be a matrix which has a 1 in the (i,j)th position and 0s elsewhere, then

$$\frac{\partial \mathbf{X}6{-}1}{\partial x_{ij}} = \begin{cases} -\mathbf{X}^{-1}\mathbf{J}_{ii}\mathbf{X}^{-1}, i = j \\ -\mathbf{X}^{-1}(\mathbf{J}_{ij} + \mathbf{J}_{ji})\mathbf{X}^{-1}, i \neq j \end{cases}$$

2.1.5 Optimization

	Scalar Optimization	**Vector Optimization**
First Order Condition	$\frac{\partial f(x_0)}{\partial x} = 0$	$\frac{\partial f(x_0)}{\partial x} = \begin{pmatrix} 0 \\ . \\ . \\ . \\ 0 \end{pmatrix}$

	Scalar Optimization	**Vector Optimization**
Second Order Condition **Convex → Min**	$\frac{\partial^2 f(x_0)}{\partial x^2} > 0$	$\frac{\partial^2 f(x_0)}{\partial x x'} > 0$
Concave → Max	$\frac{\partial^2 f(x_0)}{\partial x^2} < 0$	$\frac{\partial^2 f(x_0)}{\partial x x'} < 0$

2.2 Probability Theory

2.2.1 Axiom and Theorems of Probability

1. Let S denote a sample space of an experiment P[S]=1
2. $P[A] \geq 0$ for every event A
3. Let $A_1, A_2, A_3, ...$ be a finite or an infinite collection of mutually exclusive events. Then $P[A_1 \cup A_2 \cup A_3 ...] = P[A_1] + P[A_2] + P[A_3] + ...$
4. $P[\emptyset] = 0$
5. $P[A'] = 1 - P[A]$
6. $P[A_1 \cup A_2] = P[A_1] + P[A_2] - P[A_1 \cap A_2]$

Conditional Probability

$$P[A|B] = \frac{A \cap B}{P[B]}$$

Independent Events Two events A and B are independent if and only if:

1. $P[A \cap B] = P[A]P[B]$
2. $P[A|B] = P[A]$
3. $P[B|A] = P[B]$

A finite collection of events $A_1, A_2, ..., A_n$ is independent if and only if any subcollection is independent.

Multiplication Rule $P[A \cap B] = P[A|B]P[B] = P[B|A]P[A]$

Bayes' Theorem Let $A_1, A_2, ..., A_n$ be a collection of mutually exclusive events whose union is S.
Let b be an event such that $P[B] \neq 0$
Then for any of the events A_j, j = 1,2,...,n

$$P[A_|B] = \frac{P[B|A_j]P[A_j]}{\sum_{i=1}^{n} P[B|A_j]P[A_i]}$$

Jensen's Inequality

- If g(x) is convex $E(g(X)) \geq g(E(X))$
- If g(x) is concave $E(g(X)) \leq g(E(X))$

2.2.1.1 Law of Iterated Expectations

$E(Y) = E(E(Y|X))$

2.2.1.2 Correlation and Independence

Independence

- $f(x,y) = f_X(x)f_Y(y)$
- $f_{Y|X}(y|x) = f_Y(y)$ and $f_{X|Y}(x|y) = f_X(x)$
- $E(g_1(X)g_2(Y)) = E(g_1(X))E(g_2(Y))$

Mean Independence (implied by independence)

- Y is mean independent of X if and only if $E(Y|X) = E(Y)$
- $E(Xg(Y)) = E(X)E(g(Y))$

Uncorrelated (implied by independence and mean independence)

- $Cov(X,Y) = 0$
- $Var(X+Y) = Var(X) + Var(Y)$
- $E(XY) = E(X)E(Y)$

$$Strongest \downarrow Independence \downarrow MeanIndependence \downarrow Uncorrelated \downarrow Weakest$$

2.2.2 Central Limit Theorem

Let $X_1, X_2, ..., X_n$ be a random sample of size n from a distribution (not necessarily normal) X with mean μ and variance σ^2. then for large n ($n \geq 25$),

1. $\bar{X}$ is approximately normal with with mean $\mu_{\bar{X}} = \mu$ and variance $\sigma^2_{\bar{X}} = Var(\bar{X}) = \frac{\sigma^2}{n}$

2. $\hat{p}$is approximately normal with $\mu_{\hat{p}} = p, \sigma^2_{\hat{p}} = \frac{p(1-p)}{n}$

3. $\hat{p_1} - \hat{p_2}$ is approximately normal with $\mu_{\hat{p_1}-\hat{p_2}} = p_1 - p_2, \sigma^2_{\hat{p_1}-\hat{p_2}} = \frac{p_1(1-p)}{n_1} + \frac{p_2(1-p)}{n_2}$

4. $\bar{X_1} - \bar{X_2}$ is approximately normal with $\mu_{\bar{X_1}-\bar{X_2}} = \mu_1 - \mu_2, \sigma^2_{\bar{X_1}-\bar{X_2}} = \frac{\sigma^2_1}{n_1} + \frac{\sigma^2_2}{n_2}$

5. The following random variables are approximately standard normal:

   - $\frac{\bar{X}-\mu}{\sigma/\sqrt{n}}$
   - $\frac{\hat{p}-p}{\sqrt{\frac{p(1-p)}{n}}}$
   - $\frac{(\hat{p_1}-\hat{p_2})-(p_1-p_2)}{\sqrt{\frac{p_1(1-p_1)}{n_1}-\frac{p_2(1-p_2)}{n_2}}}$
   - $\frac{(\bar{X_1}-\bar{X_2})-(\mu_1-\mu_2)}{\sqrt{\frac{\sigma_1^2}{n_1}-\frac{\sigma_2^2}{n_2}}}$

If $\{x_i\}_{i=1}^n$ is an iid random sample from a probability distribution with finite mean μ and finite variance σ^2 then the sample mean $\bar{x} = n^{-1}\sum_{i=1}^n x_i$ scaled by $\sqrt{n}$ has the following limiting distribution

$$\sqrt{n}(\bar{x}-\mu) \to^d N(0,\sigma^2)$$

or if we were to standardize the sample mean,

$$\frac{\sqrt{n}(\bar{x}-\mu)}{\sigma} \to^d N(0,1)$$

- holds for most random sample from any distribution (continuous, discrete, unknown).

- extends to multivariate case: random sample of a random vector converges to a multivariate normal.

- Variance from the limiting distribution is the asymptotic variance (Avar)

$$Avar(\sqrt{n}(\bar{x}-\mu)) = \sigma^2 \lim_{n\to\infty} Var(\sqrt{n}(\bar{x}-\mu)) = \sigma^2 Avar(.) \neq lim_{n\to\infty}Var(.)$$

2.2.3 Random variable

	Discrete Variable	Continuous Variable
Definition	A random variable is discrete if it can assume at most a finite or countably infinite number of possible values	A random variable is continuous if it can assume any value in some interval or intervals of real numbers and the probability that it assumes any specific value is 0

	Discrete Variable	Continuous Variable
Density Function	A function f is called a density for X if: (1) $f(x) \geq 0$ (2) $\sum_{all\ x} f(x) = 1$ (3) $f(x) = P(X = x)$ for x real	A function f is called a density for X if: (1) $f(x) \geq 0$ for x real (2) $\int_{-\infty}^{\infty} f(x)\,dx = 1$ (3) $P[a \leq X \leq] = \int_a^b f(x)\,dx$ for a and b real
Cumulative Distribution Function for x real	$F(x) = P[X \leq x]$	$F(x) = P[X \leq x] = \int_{-\infty}^{\infty} f(t)dt$
$E[H(X)]$	$\sum_{all\ x} H(x)f(x)$	$\int_{-\infty}^{\infty} H(x)f(x)$
$\mu = E[X]$	$\sum_{all\ x} xf(x)$	$\int_{-\infty}^{\infty} xf(x)$
Ordinary Moments the kth ordinary moment for variable X is defined as: $E[X^k]$	$\sum_{all\ x \in X}(x^k f(x))$	$\int_{-\infty}^{\infty}(x^k f(x))$
Moment generating function (mgf) $m_X(t) = E[e^{tX}]$	$\sum_{all\ x \in X}(e^{tx} f(x))$	$\int_{-\infty}^{\infty}(e^{tx} f(x)dx)$

Expected Value Properties:

- E[c] = c for any constant c
- E[cX] = cE[X] for any constant c
- E[X+Y] = E[X] = E[Y]
- E[XY] = E[X].E[Y] (if X and Y are independent)

Expected Variance Properties:

- $Var(c) = 0$ for any constant c
- $Var(cX) = c^2 Var(X)$ for any constant c
- $Var(X) \geq 0$
- $Var(X) = E(X^2) - (E(X))^2$
- $Var(X + c) = Var(X)$
- $Var(X + Y) = Var(X) + Var(Y)$ (if X and Y are independent)

Standard deviation $\sigma = \sqrt{(}\sigma^2) = \sqrt{(}VarX)$

Suppose $y_1, ..., y_p$ are possibly correlated random variables with means $\mu_1, ..., \mu_p$. then

$$\mathbf{y} = (y_1, ..., y_p)' E(\mathbf{y}) = (\mu_1, ..., \mu_p)' =$$

Let $\sigma_{ij} = cov(y_i, y_j)$ for $i, j = 1, .., p$.

Define

$$= (\sigma_{ij}) = \begin{pmatrix} \sigma_{11} & \sigma_{12} & ... & \sigma_{1p} \\ \sigma_{21} & \sigma_{22} & ... & \sigma_{2p} \\ . & . & . & . \\ \sigma_{p1} & \sigma_{p2} & ... & \sigma_{pp} \end{pmatrix}$$

Hence, is the variance-covariance or dispersion matrix. And is symmetric with $(p+1)p/2$ unique parameters.

Alternatively, let $u_{p\times 1}$ and $v_{v\times 1}$ be random vectors with means ${}_{\mathbf{u}}$ and ${}_{\mathbf{v}}$. then

$${}_{\mathbf{uv}} = cov(\mathbf{u}, \mathbf{v}) = E[(\mathbf{u} - {}_{\mathbf{u}})(\mathbf{v} - {}_{\mathbf{v}})']$$

$\Sigma_{uv} \neq \Sigma_{vu}$ (but $\Sigma_{uv} = \Sigma'_{vu}$)

Properties of Covariance Matrices

1. Symmetric: $' =$
2. Eigendecomposition (spectral decomposition,symmetric decomposition): $=$, where is a matrix of eigenvectors such that $' = \mathbf{I}$ (orthonormal), and is a diagonal matrix with eigenvalues $(\lambda_1, ..., \lambda_p)$ on the diagonal.
3. Non-negative definite, $\mathbf{a}\,\mathbf{a} \geq 0$ for any $\mathbf{a} \in R^p$. Equivalently, the eigenvalues of , $\lambda_1 \geq ... \geq \lambda_p \geq 0$
4. $|\ | = \lambda_1, ...\lambda_p \geq 0$ (generalized variance)
5. $trace(\ ) = tr(\ ) = \lambda_1 + ... + \lambda_p = \sigma_{11} + ... + \sigma_{pp}$= sum of variances (total variance)

Note: is usually required to be positive definite. This implies that all eigenvalues are positive, and has an inverse $^{-1}$, such that $^{-1} \ = \mathbf{I}_{p\times p} = \ ^{-1}$

Correlation Matrices

Define the correlation ρ_{ij} and the correlation matrix by

$$\rho_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}}$$

$$\mathbf{R} = \begin{pmatrix} \rho_{11} & \rho_{12} & \cdots & \rho_{1p} \\ \rho_{21} & \rho_{22} & \cdots & \rho_{2p} \\ \cdot & \cdot & \cdot & \cdot \\ \rho_{p1} & \rho_{p2} & \cdots & \rho_{pp} \end{pmatrix}$$

where $\rho_{ii} = 1$ for all i.

Let x and y be random vectors with means μ_x and μ_y and variance-covariance matrices Σ_x and Σ_y. Let $\mathbf{A}$ and $\mathbf{B}$ be matrices of constants and c and d be vectors of constants. Then,

- $E(\mathbf{Ay} + \mathbf{c}) = \mathbf{A}\ _{\mathbf{y}} + c$
- $var(\mathbf{Ay} + \mathbf{c}) = \mathbf{A}var(\mathbf{y})\mathbf{A}' = \mathbf{A}\ _{\mathbf{y}}\mathbf{A}'$
- $cov(\mathbf{Ay} + \mathbf{c}, \mathbf{By} + \mathbf{d} = \mathbf{A}\ _{\mathbf{y}}\mathbf{B}'$

2.2.4 Moment generating function

Moment generating function properties:

(a) $\frac{d^k(m_X(t))}{dt^k}|_{t=0} = E[X^k]$

(b) $\mu = E[X] = m'_X(0)$

(c) $E[X^2] = m''_X(0)$

mgf Theorems

Let $X_1, X_2, ...X_n, Y$ be random variables with moment-generating functions $m_{X_1}(t), m_{X_2}(t), ..., m_{X_n}(t), m_Y(t)$

1. If $m_{X_1}(t) = m_{X_2}(t)$ for all t in some open interval about 0, then X_1 and X_2 have the same distribution
2. If $Y = \alpha + \beta X_1$, then $m_Y(t) = e^{\alpha t} m_{X_1}(\beta t)$
3. If $X_1, X_2, ...X_n$ are independent and $Y = \alpha_0 + \alpha_1 X_1 + \alpha_2 X_2 + ... + \alpha_n X_n$ (where $\alpha_0, ..., \alpha_n$ are real numbers), then $m_Y(t) = e^{\alpha_0 t} m_{X_1}(\alpha_1 t) m_{X_2}(\alpha_2 t) ... m_{X_n}(\alpha_n t)$

4. Suppose $X_1, X_2, ...X_n$ are independent normal random variables with means $\mu_1, \mu_2, ...\mu_n$ and variances $\sigma_1^2, \sigma_2^2, ..., \sigma_n^2$. If $Y = \alpha_0 + \alpha_1 X_1 + \alpha_2 X_2 + ... + \alpha_n X_n$ (where $\alpha_0, ..., \alpha_n$ are real numbers), then Y is normally distributed with mean $\mu_Y = \alpha_0 + \alpha_1\mu_1 + \alpha_2\mu_2 + ... + \alpha_n\mu_n$ and variance $\sigma_Y^2 = \alpha_1^2\sigma_1^2 + \alpha_2^2\sigma_2^2 + ... + \alpha_n^2\sigma_n^2$

2.2.5 Moment

Moment	Uncentered	Centered
1st	$E(X) = \mu = Mean(X)$	

Moment	Uncentered	Centered
2nd	$E(X^2)$	$E((X-\mu)^2) = Var(X) = \sigma^2$
3rd	$E(X^3)$	$E((X-\mu)^3)$
4th	$E(X^4)$	$E((X-\mu)^4)$

Skewness(X) $= E((X-\mu)^3)/\sigma^3$

Kurtosis(X) $= E((X-\mu)^4)/\sigma^4$

Conditional Moments

$$E(Y|X=x) = \begin{cases} \sum_y y f_Y(y|x) & \text{for discrete RV} \\ \int_y y f_Y(y|x) dy & \text{for continous RV} \end{cases}$$

$$Var(Y|X=x) = \begin{cases} \sum_y (y - E(Y|x))^2 f_Y(y|x) & \text{for discrete RV} \\ \int_y (y - E(Y|x))^2 f_Y(y|x) dy & \text{for continous RV} \end{cases}$$

2.2.5.1 Multivariate Moments

$$E = \begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} E(X) \\ E(Y) \end{pmatrix} = \begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix} \tag{2.6}$$

$$\begin{aligned} Var \begin{pmatrix} X \\ Y \end{pmatrix} &= \begin{pmatrix} Var(X) & Cov(X,Y) \\ Cov(X,Y) & Var(Y) \end{pmatrix} \\ &= \begin{pmatrix} E((X-\mu_X)^2) & E((X-\mu_X)(Y-\mu_Y)) \\ E((X-\mu_X)(Y-\mu_Y)) & E((Y-\mu_Y)^2) \end{pmatrix} \end{aligned} \tag{2.7}$$

Properties

- $E(aX + bY + c) = aE(X) + bE(Y) + c$
- $Var(aX + bY + c) = a^2 Var(X) + b^2 Var(Y) + 2ab Cov(X,Y)$
- $Cov(aX + bY, cX + bY) = ac Var(X) + bd Var(Y) + (ad + bc) Cov(X,Y)$
- Correlation: $\rho_{XY} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y}$

2.2.6 Distributions

Conditional Distributions

$$f_{X|Y}(X|Y=y) = \frac{f(X,Y)}{f_Y(y)}$$

$f_{X|Y}(X|Y=y) = f_X(X)$ if X and Y are independent

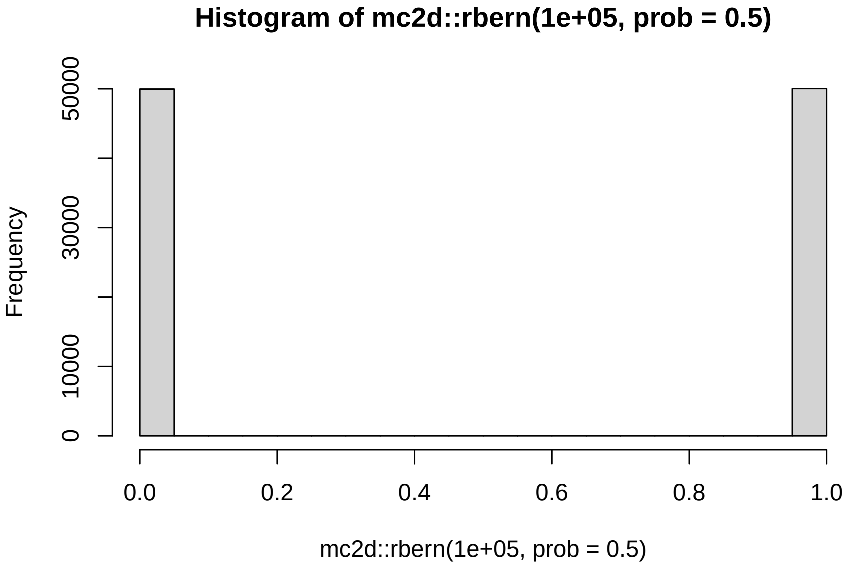
2.2.6.1 Discrete

CDF: Cumulative Density Function
MGF: Moment Generating Function

2.2.6.1.1 Bernoulli *Bernoulli*(p)

PDF

```
hist(mc2d::rbern(100000, prob=.5))
```



2.2.6.1.2 Binomial $B(n, p)$

- the experiment consists of a fixed number (n) of Bernoulli trials, each of which results in a success (s) or failure (f)
- The trials are identical and independent, and probability of success (p) and probability of failure (q = 1- p) remains the same for all trials.
- The random variable X denotes the number of successes obtained in the n trials.

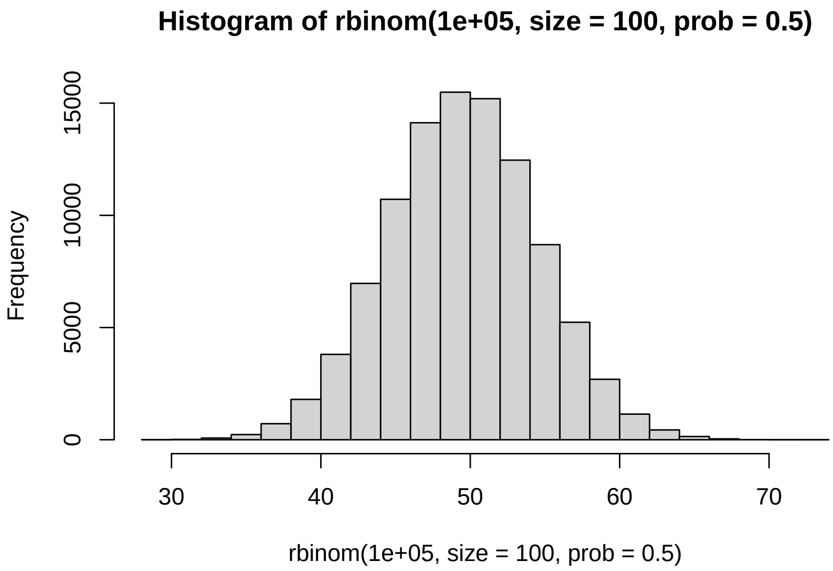
Density

$$f(x) = \binom{n}{x} p^x q^{n-x}$$

CDF
You have to use table

PDF

```r
# Histogram of 100000 random values from a sample of 100 with probability of 0.5
hist(rbinom(100000, size = 100, prob = 0.5))
```



MGF

$$m_X(t) = (q + pe^t)^n$$

Mean

$$\mu = E(x) = np$$

Variance

$$\sigma^2 = Var(X) = npq$$

2.2.6.1.3 Poisson $Pois(\lambda)$

- Arises with Poisson process, which involves observing discrete events in a continuous “interval” of time, length, or space.
- The random variable X is the number of occurrences of the event within an interval of s units

- The parameter λ is the average number of occurrences of the event in question per measurement unit. For the distribution, we use the parameter $k = \lambda s$

Density

$$f(x) = \frac{e^{-k}k^x}{x!}$$

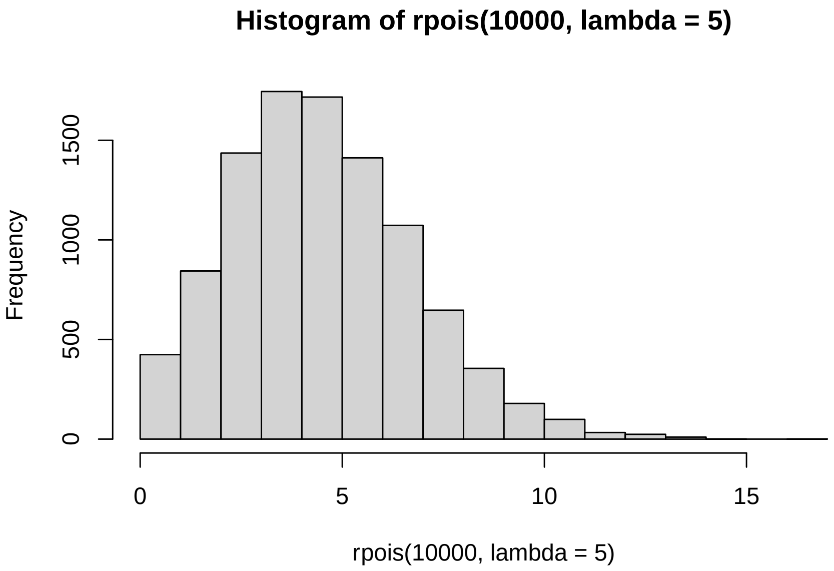
,k> 0, x =0,1,...

CDF

Use table

PDF

```r
# Poisson dist with mean of 5 or Poisson(5)
hist(rpois(10000, lambda = 5))
```



MGF

$$m_X(t) = e^{k(e^t-1)}$$

Mean

$$\mu = E(X) = k$$

Variance

$$\sigma^2 = Var(X) = k$$

2.2.6.1.4 Geometric

- The experiment consists of a series of trails. The outcome of each trial can be classed as being either a "success" (s) or "failure" (f). (This is called a Bernoulli trial).
- The trials are identical and independent in the sense that the outcome of one trial has no effect on the outcome of any other. The probability of success (p) and probability of failure (q=1-p) remains the same from trial to trial.
- lack of memory
- X: the number of trials needed to obtain the first success.

Density

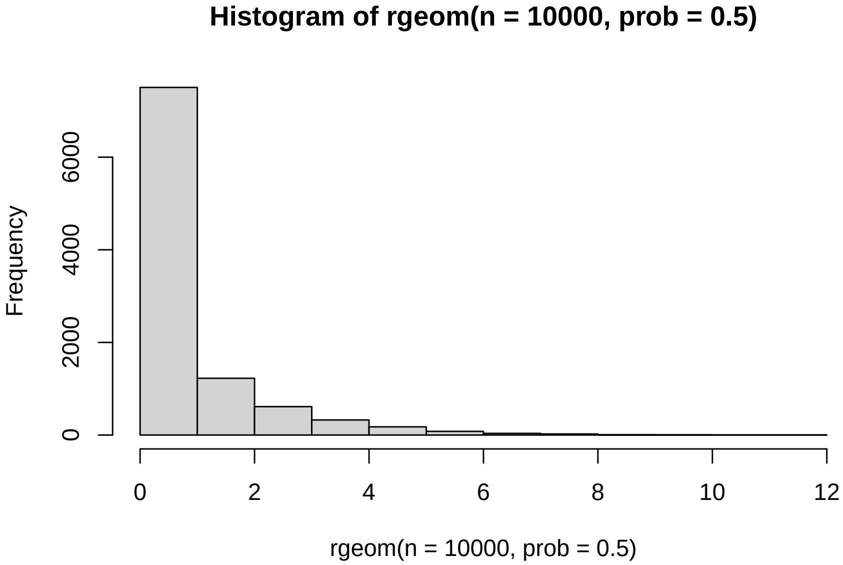
$$f(x) = pq^{x-1}$$

CDF

$$F(x) = 1 - q^x$$

PDF

```r
# hist of Geometric distribution with probability of success = 0.5
hist(rgeom(n = 10000, prob = 0.5))
```



MGF

$$m_X(t) = \frac{pe^t}{1 - qe^t}$$

for $t < -ln(q)$

Mean

$$\mu = \frac{1}{p}$$

Variance

$$\sigma^2 = Var(X) = \frac{q}{p^2}$$

2.2.6.1.5 Hypergeometric

- The experiment consists of drawing a random sample of size n without replacement and without regard to order from a collection of N objects.
- Of the N objects, r have a trait of interest; N-r do not have the trait
- X is the number of objects in the sample with the trait.

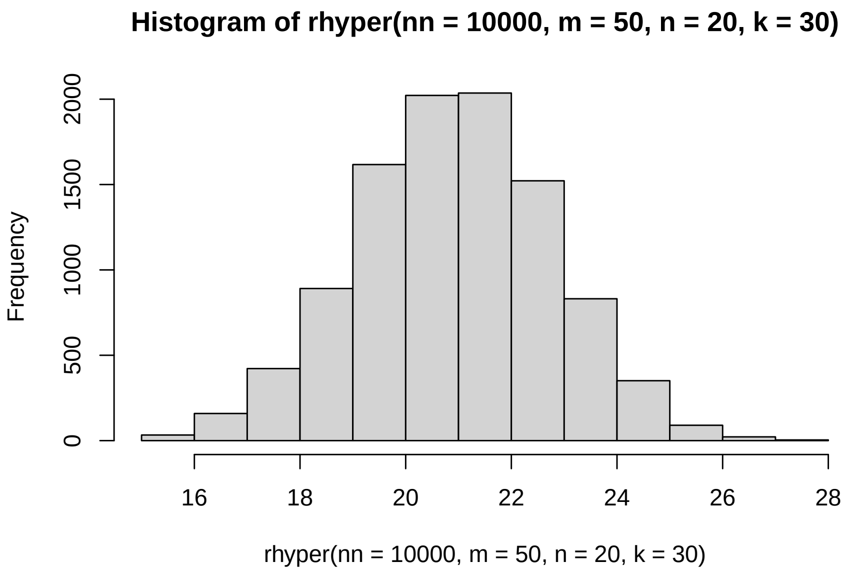
Density

$$f(x) = \frac{\binom{r}{x}\binom{N-r}{n-x}}{\binom{N}{n}}$$

where $max[0, n-(N-r)] \leq x \leq min(n, r)$

PDF

```
# hist of hypergeometric distribution with the number of white balls = 50, and the num
hist(rhyper(nn = 10000 , m=50, n=20, k=30))
```



Mean

$$\mu = E(x) = \frac{nr}{N}$$

Variance

$$\sigma^2 = var(X) = n(\frac{r}{N})(\frac{N-r}{N})(\frac{N-n}{N-1})$$

Note For large N (if $\frac{n}{N} \leq 0.05$), this distribution can be approximated using a Binomial distribution with $p = \frac{r}{N}$

2.2.6.1.6

2.2.6.2 Continuous

2.2.6.2.1 Uniform

- Defined over an interval (a,b) in which the probabilities are "equally likely" for subintervals of equal length.

Density

$$f(x) = \frac{1}{b-a}$$

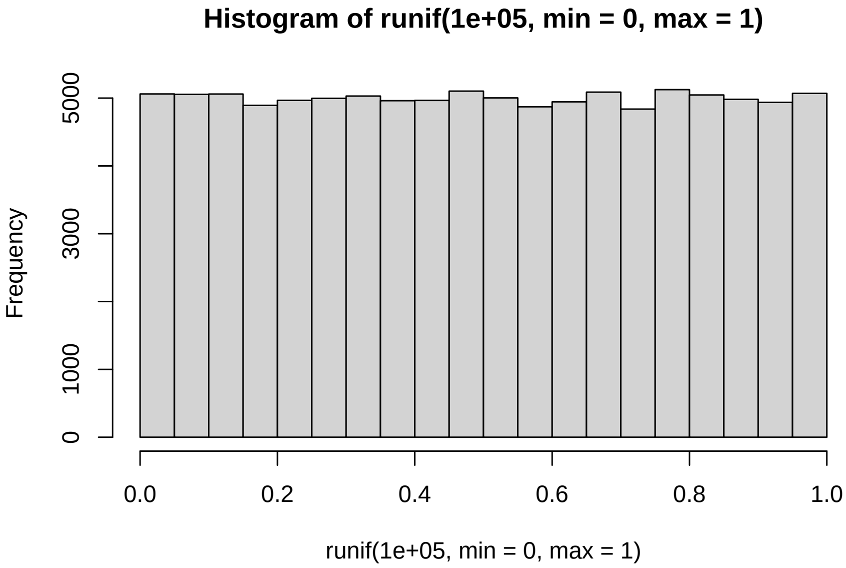
for a < x < b

CDF

$$\begin{cases} 0 & \text{if x <a} \\ \frac{x-a}{b-a} & \text{if } a \leq x \leq b \\ 1 & \text{if x >b} \end{cases}$$

PDF

```r
hist(runif(100000, min = 0, max = 1))
```



MGF

$$\begin{cases} \frac{e^{tb}-e^{ta}}{t(b-a)} & \text{if } t \neq 0 \\ 1 & \text{if } t \neq 0 \end{cases}$$

Mean

$$\mu = E(X) = \frac{a+b}{2}$$

Variance

$$\sigma^2 = Var(X) = \frac{(b-a)^2}{12}$$

2.2.6.2.2 Gamma

- is used to define the exponential and chi-squared distributions
- The gamma function is defined as:

$$\Gamma(\alpha) = \int_0^\infty z^{\alpha-1}e^{-z}dz$$

where $\alpha > 0$

- Properties of The Gamma function:

  - $\Gamma(1) = 1$ + For $\alpha > 1$, $\Gamma(\alpha) = (\alpha-1)\Gamma(\alpha-1)$ + If n is an integer and $n > 1$, then $\Gamma(n) = (n-1)!$

Density

$$f(x) = \frac{1}{\Gamma(\alpha)\beta^\alpha}x^{\alpha-1}e^{-x/\beta}$$

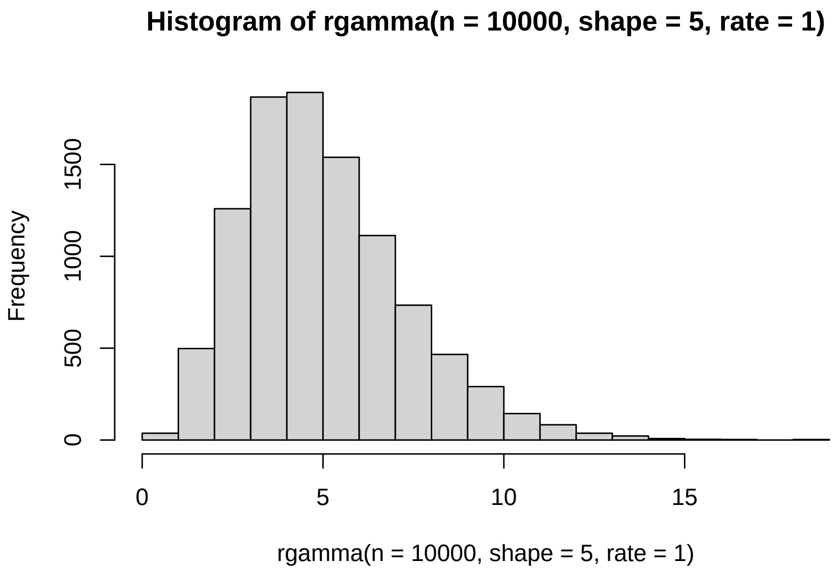
CDF

$$F(x,n,\beta) = 1 - \sum_{k=0}^{n-1}\frac{(\frac{x}{\beta})^k e^{-x/\beta}}{k!}$$

for x>0, and $\alpha = n$ (a positive integer)

PDF

```
hist(rgamma(n = 10000, shape = 5, rate = 1))
```



MGF

$$m_X(t) = (1 - \beta t)^{-\alpha}$$

where $t < \frac{1}{\beta}$

Mean

$$\mu = E(X) = \alpha\beta$$

Variance

$$\sigma^2 = Var(X) = \alpha\beta^2$$

2.2.6.2.3 Normal $N(\mu, \sigma^2)$

- is symmetric, bell-shaped curve with parameters μ and σ^2
- also known as Gaussian.

Density

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$$

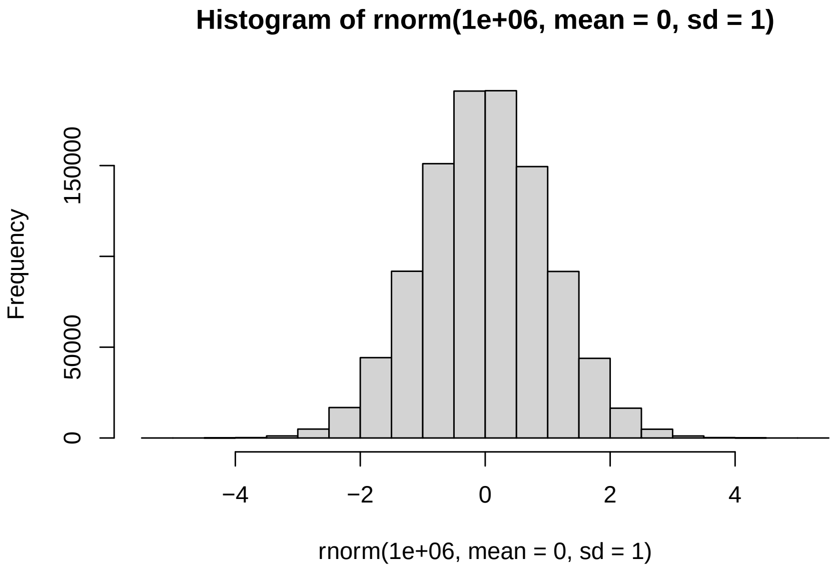
for $-\infty < x, \mu < \infty, \sigma > 0$

CDF

Use table

PDF

```
hist(rnorm(1000000, mean = 0, sd = 1))
```



MGF

$$m_X(t) = e^{\mu t + \frac{\sigma^2 t^2}{2}}$$

Mean

$$\mu = E(X)$$

Variance

$$\sigma^2 = Var(X)$$

Standard Normal Random Variable

- The normal random variable Z with mean $\mu = 0$ and standard deviation $\sigma = 1$ is called standard normal
- Any normal random variable X with mean μ and standard deviation σ can be converted to the standard normal random variable $Z = \frac{X-\mu}{\sigma}$

Normal Approximation to the Binomial Distribution

Let X be binomial with parameters n and p. For large n (so that $(A)p \leq .5$ and $np > 5$ or (B) $p > .5$ and $nq > 5$), X is approximately normally distributed with mean $\mu = np$ and standard deviation $\sigma = \sqrt{npq}$

When using the normal approximation, add or subtract 0.5 as needed for the continuity correction

Discrete	Approximate Normal (corrected)
P(X = c)	P(c -0.5 < Y < c + 0.5)
P(X < c)	P(Y < c - 0.5)
P(X c)	P(Y < c + 0.5)
P(X > c)	P(Y > c + 0.5)
P(X c)	P(Y > c - 0.5)

Normal Probability Rule

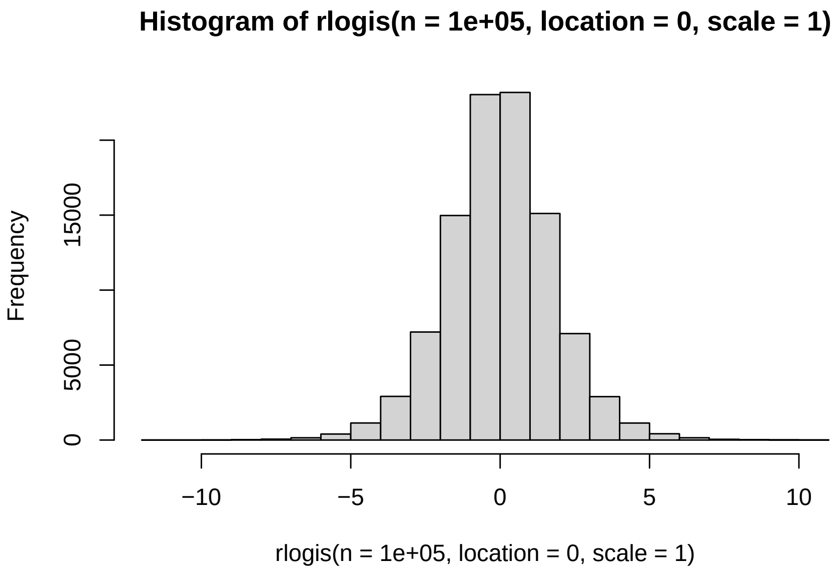
If X is normally distributed with parameters μ and σ, then

- $P(-\sigma < X - \mu < \sigma) \approx .68 * P(-2\sigma < X - \mu < 2\sigma) \approx .95 * P(-3\sigma < X - \mu < 3\sigma) \approx .997$

2.2.6.2.4 Logistic $Logistic(\mu, s)$

PDF

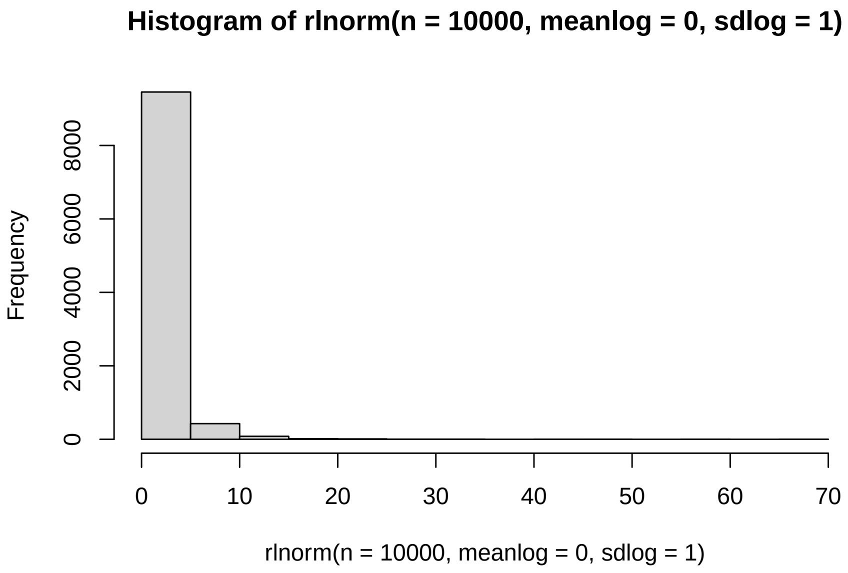
```
hist(rlogis(n = 100000, location = 0, scale = 1))
```



2.2.6.2.5 Lognomral $lognormal(\mu, \sigma^2)$

PDF

```
hist(rlnorm(n = 10000, meanlog = 0, sdlog = 1))
```



2.2.6.2.6 Exponential $Exp(\lambda)$

- A special case of the gamma distribution with $\alpha = 1$
- Lack of memory
- λ = rate Within a Poisson process with parameter λ, if W is the waiting tine until the occurrence of the first event, then W has an exponential distribution with $\beta = 1/\alpha$

Density

$$f(x) = \frac{1}{\beta} e^{-x/\beta}$$

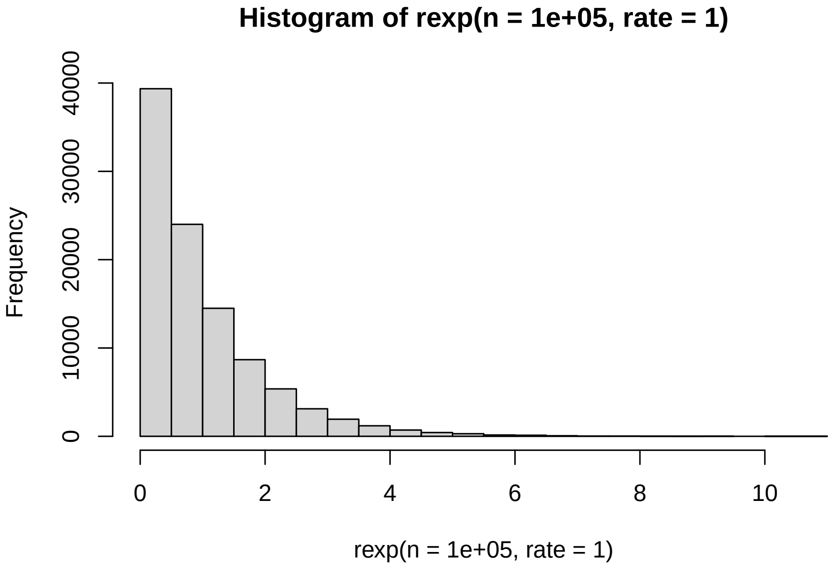
for $x, \beta > 0$

CDF

$$F(x) = \begin{cases} 0 & \text{if } x \le 0 \\ 1 - e^{-x/\beta} & \text{if } x > 0 \end{cases} \tag{2.8}$$

PDF

```
hist(rexp(n = 100000, rate = 1))
```



MGF

$$m_X(t) = (1 - \beta t)^{-1}$$

for $t < 1/\beta$

Mean

$$\mu = E(X) = \beta$$

Variance

$$\sigma^2 = Var(X) = \beta^2$$

2.2.6.2.7 Chi-squared $\chi^2 = \chi^2(k)$

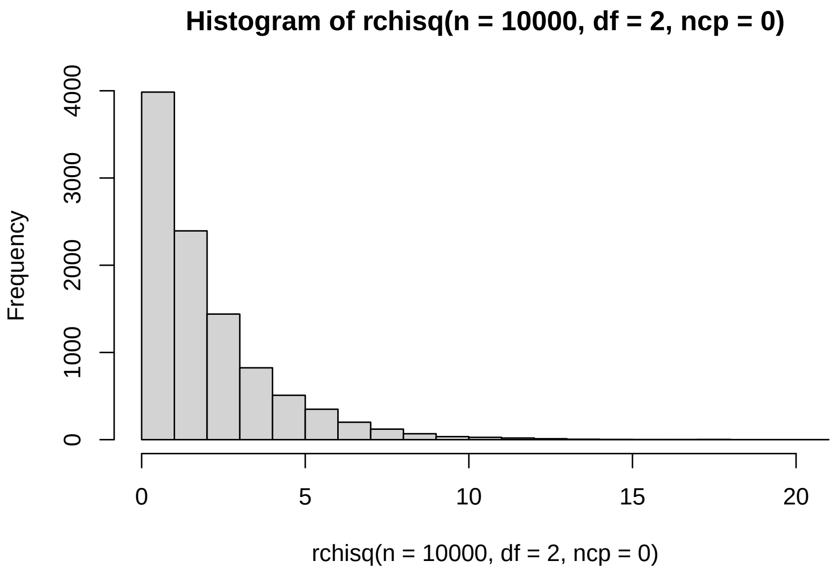
- A special case of the gamma distribution with $\beta = 2$, and $\alpha = \gamma/2$ for a positive integer γ
- The random variable X is denoted χ^2_γ and is said to have a chi-squared distribution with γ degrees of freedom.

Density Use density for Gamma Distribution with $\beta = 2$ and $\alpha = \gamma/2$

CDF Use table

PDF

```
hist(rchisq(n = 10000, df=2, ncp = 0))
```



MGF

$$m_X(t) = (1 - 2t)^{-\gamma/2}$$

Mean

$$\mu = E(X) = \gamma$$

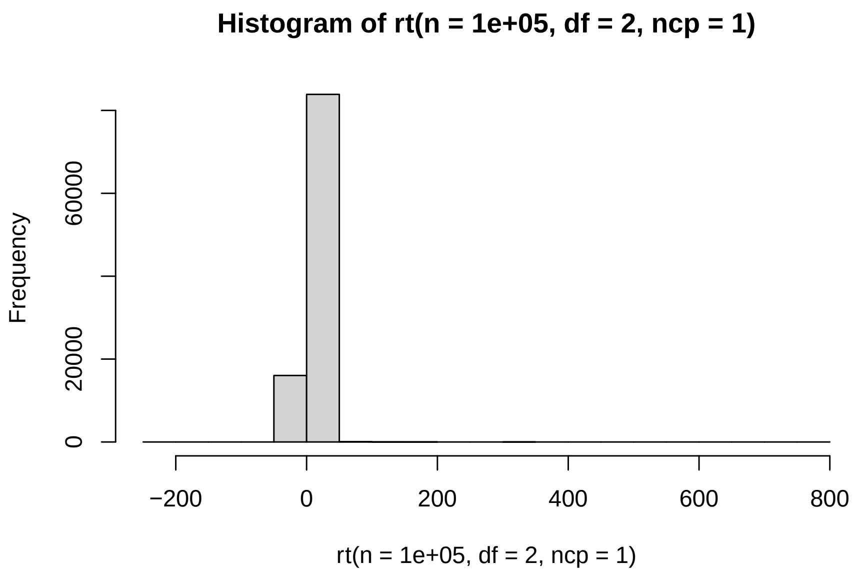
Variance

$$\sigma^2 = Var(X) = 2\gamma$$

2.2.6.2.8 Student T $T(v)$

- $T = \frac{Z}{\sqrt{\chi^2_\gamma/\gamma}}$, where Z is standard normal follows a student-t distribution with γ dof
- The distribution is symmetric, bell-shaped , with a mean of $\mu = 0$

```
hist(rt(n = 100000, df=2, ncp =1))
```



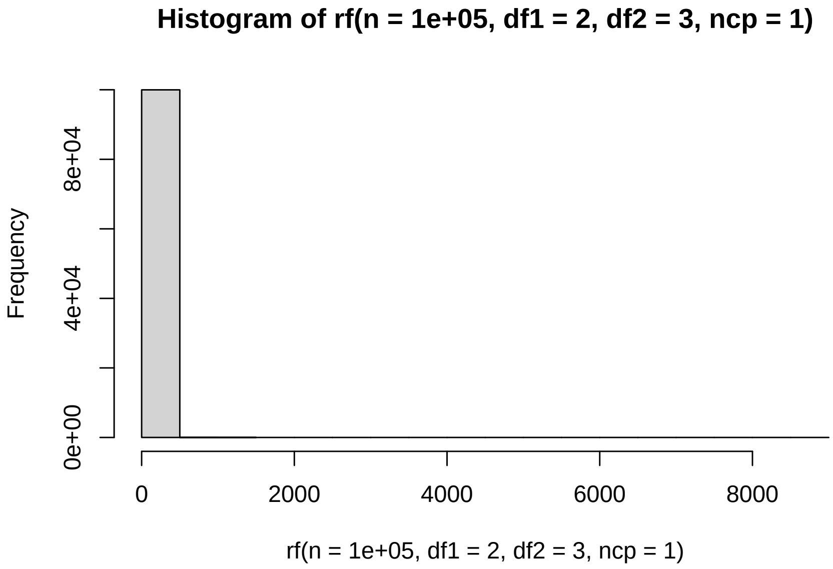
2.2.6.2.9 F-Distribution $F(d_1, d_2)$

- F distribution is strictly positive

- $F = \frac{\chi^2_{\gamma_1}/\gamma_1}{\chi_{\gamma_2^2}/\gamma_2}$ follows an F distribution with dof γ_1 and γ_2, where $\chi^2_{\gamma_1}$ and $\chi^2_{\gamma_2}$ are independent chi-squared random variables.
- The distribution is asymmetric and never negative.

PDF

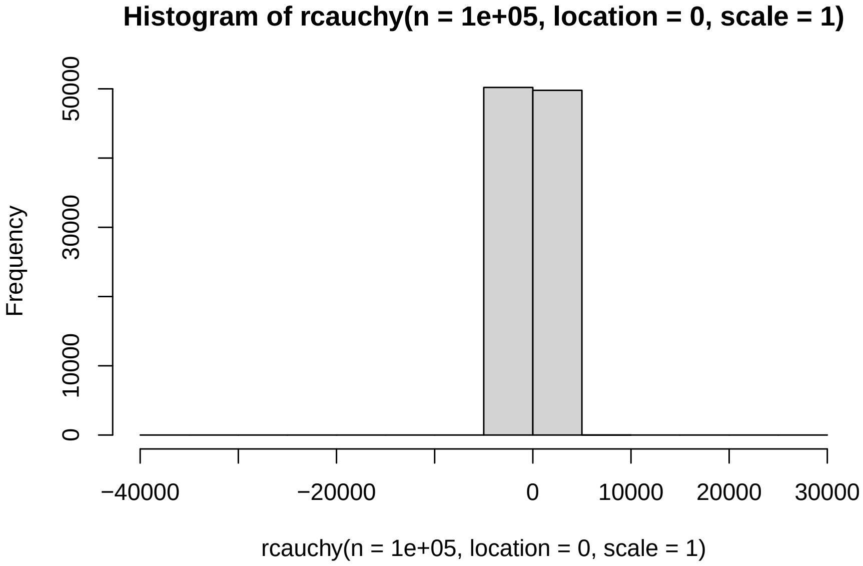
```
hist(rf(n = 100000, df1=2, df2=3, ncp=1))
```



2.2.6.2.10 Cauchy Central Limit Theorem and Weak Law do not apply to Cauchy because it does not have finite mean and finite variance

PDF

```
hist(rcauchy(n = 100000, location = 0, scale = 1))
```



2.2.6.2.11 Multivariate Normal Distribution Let y be a p-dimensional multivariate normal (MVN) rv with mean μ and variance Σ. Then, the density of y is

$$f(\mathbf{y}) = \frac{1}{(2\pi)^{p/2}|\quad|^{1/2}} exp(-\frac{1}{2}(\mathbf{y}-\quad)'\ ^{-\mathbf{1}}(\mathbf{y}-\quad))$$

We have $\mathbf{y} \sim N_p(\ ,\ )$

Properties:

- Let $\mathbf{A}_{r\times p}$ be a fixed matrix. then $\mathbf{Ay} \sim N_r(\mathbf{A}\ ,\mathbf{A}\ \mathbf{A}')$. Note that $r \leq p$ and all rows of A must be linearly independent to guarantee that $\mathbf{A}\ \mathbf{A}'$ is non-singular.

- Let $\mathbf{G}$ be a matrix such that $\ ^{-\mathbf{1}} = \mathbf{GG}'$. then, $\mathbf{G}'\mathbf{y} \sim N_p(\mathbf{G}'\ ,\mathbf{I})$ and $\mathbf{G}'(\mathbf{y}-\ ) \sim N_p(\mathbf{0},\mathbf{I})$.

- Any fixed linear combination of $y_1,...,y_p$ say $\mathbf{c}'\mathbf{y}$, follows $\mathbf{c}'\mathbf{y} \sim N_1(\mathbf{c}'\ ,\mathbf{c}'\ \mathbf{c})$

Large Sample Properties

Suppose that $y_1,...,y_n$ are a random sample from some population with mean μ and variance-variance matrix Σ

$$\mathbf{Y} \sim MVN(\ , \ )$$

Then

- $\bar{\mathbf{y}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{y}_i$ is a consistent estimator for
- $\mathbf{S} = \frac{1}{n-1}\sum_{i=1}^{n} (\mathbf{y_i} - \bar{\mathbf{y}})(\mathbf{y_i} - \bar{\mathbf{y}})'$ is a consistent estimator for
- Multivariate Central Limit Theorem: Similar to the univariate case, $\sqrt{n}(\bar{\mathbf{y}} - \ ) \sim N_p(\mathbf{0}, \ )$ when n is large relative to p (e.g., $n \geq 25p$), which is equivalent to $\bar{y} \sim N_p(\ , \ /\mathbf{n})$
- Wald's Theorem: $n(\bar{\mathbf{y}} - \ )'\mathbf{S^{-1}}(\bar{\mathbf{y}} - \ ) \sim \chi^2_{(p)}$ when n is large relative to p.

2.2.6.2.12

2.3 General Math

Chebyshev's Inequality Let X be a random variable with mean μ and standard deviation σ. Then for any positive number k:

$$P(|X - \mu| < k\sigma) \geq 1 - \frac{1}{k^2}$$

Chebyshev's Inequality does not require that X be normally distributed

Maclaurin series expansion for

$$e^z = 1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} + ...$$

Geometric series:

$$s_n = \sum_{k=1}^{n} ar^{n-1} = \frac{a(1-r^n)}{1-r}$$

if |r| < 1

$$s = \sum_{k=1}^{\infty} ar^{n-1} = \frac{a}{1-r}$$

2.3.1 Law of large numbers

Let $X_1, X_2, ...$ be an infinite sequence of independent and identically distributed (i.i.d) Then, the sample average is

$$\bar{X}_n = \frac{1}{n}(X_1 + ... + X_n)$$

converges to the expected value ($\bar{X}_n \to \mu$) as $n \to \infty$

$$Var(X_i) = Var(\frac{1}{n}(X_1 + ... + X_n)) = \frac{1}{n^2} Var(X_1 + ... + X_n) = \frac{n\sigma^2}{n^2} = \frac{\sigma^2}{n}$$

The difference between Weak Law and Strong Law regards the mode of convergence

2.3.1.1 Weak Law

The sample average converges in probability towards the expected value

$$\bar{X}_n \to^p \mu$$

when $n \to \infty$

$$\lim_{n \to \infty} P(|\bar{X}_n - \mu| > \epsilon) = 0$$

The sample mean from a iid random sample ($\{x_i\}_{i=1}^n$) from any population with a finite mean and finite variance σ^2 is ca consistent estimation for the population mean μ

$$plim(\bar{x}) = plim(n^{-1} \sum_{i=1}^{n} x_i) = \mu$$

2.3.1.2 Strong Law

The sample average converges almost surely to the expected value

$$\bar{X}_n \to^{a.s} \mu$$

when $n \to \infty$

Equivalently,

$$P(\lim_{n \to \infty} \bar{X}_n = \mu) = 1$$

2.3.2 Law of Iterated Expectation

Let X, Y be random variables. Then,

$$E(X) = E(E(X|Y))$$

means that the expected value of X can be calculated from the probability distribution of X|Y and Y

2.3.3 Convergence

2.3.3.1 Convergence in Probability

- $n \to \infty$, an estimator (random variable) that is close to the true value.
- The random variable θ_n converges in probability to a constant c if

$$\lim_{n \to \infty} P(|\theta_n - c| \geq \epsilon) = 0$$

for any positive ϵ

Notation

$$plim(\theta_n) = c$$

Equivalently,

$$\theta_n \to^p c$$

Properties of Convergence in Probability

- Slutsky's Theorem: for a continuous function g(.), if $plim(\theta_n) = \theta$ then $plim(g(\theta_n)) = g(\theta)$
- if $\gamma_n \to^p \gamma$ then
  - $plim(\theta_n + \gamma_n) = \theta + \gamma + plim(\theta_n \gamma_n) = \theta\gamma + plim(\theta_n / \gamma_n) = \theta/\gamma$ if $\gamma \neq 0$
- Also hold for random vectors/ matrices

2.3.3.2 Convergence in Distribution

- As $n \to \infty$, the distribution of a random variable may converge towards another ("fixed") distribution.
- The random variable X_n with CDF $F_n(x)$ converges in distribution to a random variable X with CDF $F(X)$ if

$$\lim_{n\to\infty} |F_n(x) - F(x)| = 0$$

at all points of continuity of $F(X)$

Notation F(x) is the limiting distribution of X_n or $X_n \to^d X$

- E(X) is the limiting mean (asymptotic mean)
- Var(X) is the limiting variance (asymptotic variance)

Note

$$E(X) \neq \lim_{n\to\infty} E(X_n) Avar(X_n) \neq \lim_{n\to\infty} Var(X_n)$$

Properties of Convergence in Distribution

- Continuous Mapping Theorem: for a continuous function g(.), if $X_n \to^d g(X)$ then $g(X_n) \to^d g(X)$
- if $Y_n \to^d c$, then
  - $X_n + Y_n \to^d X + c$
  - $Y_n X_n \to^d cX$
  - $X_n Y_n \to^d X/c$ if $c \neq 0$
- also hold for random vectors/matrices

2.3.3.3 Summary

Properties of Convergence

Probability	Distribution
Slutsky's Theorem: for a continuous function g(.), if $plim(\theta_n) = \theta$ then $plim(g(\theta_n)) = g(\theta)$ if $\gamma_n \to^p \gamma$ then $plim(\theta_n + \gamma_n) = \theta + \gamma$ $plim(\theta_n \gamma_n) = \theta\gamma$ $plim(\theta_n/\gamma_n) = \theta/\gamma$ if $\gamma \neq 0$	Continuous Mapping Theorem: for a continuous function g(.), if $X_n \to^d g(X)$ then $g(X_n) \to^d g(X)$ if $Y_n \to^d c$, then $X_n + Y_n \to^d X + c$ $Y_n X_n \to^d cX$ $X_n Y_n \to^d X/c$ if $c \neq 0$

Convergence in Probability is stronger than Convergence in Distribution. However, Convergence in Distribution does not guarantee Convergence in Probability

2.3.4 Sufficient Statistics

Likelihood

- describes the extent to which the sample provides support for any particular parameter value.
- Higher support corresponds to a higher value for the likelihood
- The exact value of any likelihood is **meaningless**,
- The relative value, (i.e., comparing two values of θ), is **informative**.

$$L(\theta_0; y) = P(Y = y|\theta = \theta_0) = f_Y(y; \theta_0)$$

Likelihood Ratio

$$\frac{L(\theta_0; y)}{L(\theta_1; y)}$$

Likelihood Function

For a given sample, you can create likelihoods for all possible values of θ, which is called *likelihood function*

$$L(\theta) = L(\theta; y) = f_Y(y; \theta)$$

In a sample of size n, the likelihood function takes the form of a product

$$L(\theta) = \prod_{i=1}^{n} f_i(y_i; \theta)$$

Equivalently, the log likelihood function

$$l(\theta) = \sum_{i=1}^{n} log f_i(y_i; \theta)$$

Sufficient statistics

- A statistic, T(y), is any quantity that can be calculated purely from a sample (independent of θ)
- A statistic is **sufficient** if it conveys all the available information about the parameter.

$$L(\theta; y) = c(y) L^*(\theta; T(y))$$

Nuisance parameters If we are interested in a parameter (e.g., mean). Other parameters requiring estimation (e.g., standard deviation) are **nuisance** parameters. We can replace nuisance parameters in likelihood function with their estimates to create a **profile likelihood*.

2.3.5 Parameter transformations

log-odds transformation

$$Logodds = g(\theta) = ln[\frac{\theta}{1-\theta}]$$

log transformation

2.4 Methods

Trade-off between parametric and non-parametric

2.5 Data Import/Export

Extended Manual by R

Table 2.6: Table by Rio Vignette

Format	Typical Extension	Import Package	Export Package	Install
Comma-separated data	.csv	**data.table**	**data.table**	Yes
Pipe-separated data	.psv	**data.table**	**data.table**	Yes
Tab-separated data	.tsv	**data.table**	**data.table**	Yes
CSVY (CSV + YAML metadata header)	.csvy	**data.table**	**data.table**	Yes
SAS	.sas7bdat	**haven**	**haven**	Yes
SPSS	.sav	**haven**	**haven**	Yes
SPSS (compressed)	.zsav	**haven**	**haven**	Yes
Stata	.dta	**haven**	**haven**	Yes
SAS XPORT	.xpt	**haven**	**haven**	Yes
SPSS Portable	.por	**haven**		Yes
Excel	.xls	**readxl**		Yes
Excel	.xlsx	**readxl**	**openxlsx**	Yes
R syntax	.R	**base**	**base**	Yes
Saved R objects	.RData, .rda	**base**	**base**	Yes
Serialized R objects	.rds	**base**	**base**	Yes
Epiinfo	.rec	**foreign**		Yes
Minitab	.mtp	**foreign**		Yes
Systat	.syd	**foreign**		Yes
“XBASE” database files	.dbf	**foreign**	**foreign**	Yes

Format	Typical Extension	Import Package	Export Pack
Weka Attribute-Relation File Format	.arff	**foreign**	**foreign**
Data Interchange Format	.dif	**utils**	
Fortran data	no recognized extension	**utils**	
Fixed-width format data	.fwf	**utils**	**utils**
gzip comma-separated data	.csv.gz	**utils**	**utils**
Apache Arrow (Parquet)	.parquet	**arrow**	**arrow**
EViews	.wf1	**hexView**	
Feather R/Python interchange format	.feather	**feather**	**feather**
Fast Storage	.fst	**fst**	**fst**
JSON	.json	**jsonlite**	**jsonlite**
Matlab	.mat	**rmatio**	**rmatio**
OpenDocument Spreadsheet	.ods	**readODS**	**readODS**
HTML Tables	.html	**xml2**	**xml2**
Shallow XML documents	.xml	**xml2**	**xml2**
YAML	.yml	**yaml**	**yaml**
Clipboard	default is tsv	**clipr**	**clipr**
Google Sheets	as Comma-separated data		

R limitations:

- By default, R use 1 core in CPU
- R puts data into memory (limit around 2-4 GB), while SAS uses data from files on demand
- Categorization
    - Medium-size file: within RAM limit, around 1-2 GB
    - Large file: 2-10 GB, there might be some workaround solution
    - Very large file > 10 GB, you have to use distributed or parallel computing

Solutions:

- buy more RAM
- HPC packages
    - Explicit Parallelism
    - Implicit Parallelism
    - Large Memory
    - Map/Reduce
- specify number of rows and columns, typically including command `nrow =`

- Use packages that store data differently
  - bigmemory, biganalytics, bigtabulate , synchronicity, bigalgebra, bigvideo use C++ to store matrices, but also support one class type
  - For multiple class types, use ff package
- Very Large datasets use
  - RHaddop package
  - HadoopStreaming
  - Rhipe

2.5.1 Medium size

```
library("rio")
```

To import multiple files in a directory

```
str(import_list(dir()), which = 1)
```

To export a single data file

```
export(data, "data.csv")
export(data,"data.dta")
export(data,"data.txt")
export(data,"data_cyl.rds")
export(data,"data.rdata")
export(data,"data.R")
export(data,"data.csv.zip")
export(data,"list.json")
```

To export multiple data files

```
export(list(mtcars = mtcars, iris = iris), "data_file_type") # where data_file_type should subst
```

To convert between data file types

```
# convert Stata to SPSS
convert("data.dta", "data.sav")
```

2.5.2 Large size

Use R on a cluster

- Amazon Web Service (AWS): $1/hr

Import files as chunks

```
file_in    <- file("in.csv","r")
chunk_size <- 100000 # choose the best size for you
x          <- readLines(file_in, n=chunk_size)
```

`data.table` method

```
require(data.table)
mydata = fread("in.csv", header = T)
```

`ff` package: this method does not allow you to pass connections

```
library("ff")
x <- read.csv.ffdf(
    file = "file.csv",
    nrow = 10,
    header = TRUE,
    VERBOSE = TRUE,
    first.rows = 10000,
    next.rows = 50000,
    colClasses = NA
)
```

`bigmemory` package

```
my_data <- read.big.matrix('in.csv', header = T)
```

`sqldf` package

```
library(sqldf)
my_data <- read.csv.sql('in.csv')

iris2 <- read.csv.sql("iris.csv",
    sql = "select * from file where Species = 'setosa' ")
```

```
library(RMySQL)
```

```
## Loading required package: DBI
```

`RQLite` package

- Download SQLite, pick "A bundle of command-line tools for managing SQLite database files" for Window 10
- Unzip file, and open `sqlite3.exe.`
- Type in the prompt
  - `sqlite> .cd 'C:\Users\data'` specify path to your desired directory
  - `sqlite> .open database_name.db` to open a database
  - To import the CSV file into the database
    - `sqlite> .mode csv` specify to SQLite that the next file is .csv file

        - `sqlite> .import file_name.csv datbase_name` to import the csv file to the database
    - `sqlite> .exit` After you're done, exit the sqlite program

```
library(DBI)
library(dplyr)
library("RSQLite")
setwd("")
con <- dbConnect(RSQLite::SQLite(), "data_base.db")
tbl <- tbl(con, "data_table")
tbl %>%
    filter() %>%
    select() %>%
    collect() # to actually pull the data into the workspace
dbDisconnect(con)
```

`arrow` package

```
library("arrow")
read_csv_arrow()
```

`vroom` package

```
library(vroom)
spec(vroom(file_path))
compressed <- vroom_example("mtcars.csv.zip")
vroom(compressed)
```

`data.table` package

```
s = fread("sample.csv")
```

Comparisons regarding storage space

```
test = ff::read.csv.ffdf(file = "")
object.size(test) # worst

test1 = data.table::fread(file = "")
object.size(test1) # best

test2 = readr::read_csv(""))
object.size(test2) # 2nd

test3 = vroom(file = "")
object.size(test3) # equal to read_csv
```

To work with big data, you can convert it to `csv.gz` , but since typically, R would require you to load the whole data then export it. With data greater than 10 GB, we have to do it sequentially. Even though `read.csv` is much slower than `readr::read_csv` , we still have to use it because it can pass connection, and it allows you to loop sequentially. On the other, because currently `readr::read_csv` does not have the `skip` function, and even if we can use the skip, we still have to read and skip lines in previous loop.

For example, say you `read_csv(, n_max = 100, skip =0)` and then `read_csv(, n_max = 200, skip = 100)` you actually have to read again the first 100 rows. However, `read.csv` without specifying anything, will continue at the 100 mark.

Notice, sometimes you might have error looking like this

"Error in (function (con, what, n = 1L, size = NA_integer_, signed = TRUE, : can only read from a binary connection"

then you can change it instead of `r` in the connection into `rb` . Even though an author of the package suggested that `file` should be able to recognize the appropriate form, so far I did not prevail.

2.6 Data Manipulation

```r
# load packages
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
x <- c(1, 4, 23, 4, 45)
n <- c(1, 3, 5)
g <- c("M", "M", "F")
df <- data.frame(n, g)
df
```

```
##   n g
## 1 1 M
## 2 3 M
## 3 5 F
```

```
str(df)
```

```
## 'data.frame':    3 obs. of  2 variables:
##  $ n: num  1 3 5
##  $ g: chr  "M" "M" "F"
```

```
#Similarly
df <- tibble(n, g)
df
```

```
## # A tibble: 3 x 2
##       n g
##   <dbl> <chr>
## 1     1 M
## 2     3 M
## 3     5 F
```

```
str(df)
```

```
## tibble[,2] [3 x 2] (S3: tbl_df/tbl/data.frame)
##  $ n: num [1:3] 1 3 5
##  $ g: chr [1:3] "M" "M" "F"
```

```
# list form
lst <- list(x, n, g, df)
lst
```

```
## [[1]]
## [1]  1  4 23  4 45
##
## [[2]]
## [1] 1 3 5
##
## [[3]]
## [1] "M" "M" "F"
##
## [[4]]
## # A tibble: 3 x 2
##       n g
##   <dbl> <chr>
## 1     1 M
## 2     3 M
## 3     5 F
```

```
# Or
lst2 <- list(num = x, size = n, sex = g, data = df)
lst2
```

```
## $num
## [1]  1  4 23  4 45
##
## $size
## [1] 1 3 5
##
## $sex
## [1] "M" "M" "F"
##
## $data
## # A tibble: 3 x 2
##       n g
##   <dbl> <chr>
## 1     1 M
## 2     3 M
## 3     5 F
```

```
# Or
lst3 <- list(x = c(1, 3, 5, 7),
             y = c(2, 2, 2, 4, 5, 5, 5, 6),
             z = c(22, 3, 3, 3, 5, 10))
lst3
```

```
## $x
## [1] 1 3 5 7
##
## $y
## [1] 2 2 2 4 5 5 5 6
##
## $z
## [1] 22  3  3  3  5 10
```

```
# find the means of x, y, z.

# can do one at a time
mean(lst3$x)
```

```
## [1] 4
```

```
mean(lst3$y)
```

```
## [1] 3.875
```

```
mean(lst3$z)
```

```
## [1] 7.666667
```

```
# list apply
lapply(lst3, mean)
```

```
## $x
## [1] 4
##
## $y
## [1] 3.875
##
## $z
## [1] 7.666667
```

```
# OR
sapply(lst3, mean)
```

```
##        x        y        z
## 4.000000 3.875000 7.666667
```

```
# Or, tidyverse function map()
map(lst3, mean)
```

```
## $x
## [1] 4
##
## $y
## [1] 3.875
##
## $z
## [1] 7.666667
```

```
# The tidyverse requires a modified map function called map_dbl()
map_dbl(lst3, mean)
```

```
##        x        y        z
## 4.000000 3.875000 7.666667
```

```
# Binding
dat01 <- tibble(x = 1:5, y = 5:1)
dat01
```

```
## # A tibble: 5 x 2
##       x     y
##   <int> <int>
## 1     1     5
## 2     2     4
## 3     3     3
## 4     4     2
## 5     5     1
```

```
dat02 <- tibble(x = 10:16, y = x/2)
dat02
```

```
## # A tibble: 7 x 2
##       x     y
##   <int> <dbl>
## 1    10   5
## 2    11   5.5
## 3    12   6
## 4    13   6.5
## 5    14   7
## 6    15   7.5
## 7    16   8
```

```
dat03 <- tibble(z = runif(5)) # 5 random numbers from interval (0,1)
dat03
```

```
## # A tibble: 5 x 1
##       z
##   <dbl>
## 1 0.532
## 2 0.570
## 3 0.447
## 4 0.479
## 5 0.714
```

```
# row binding
bind_rows(dat01, dat02, dat01)
```

```
## # A tibble: 17 x 2
##        x     y
##    <int> <dbl>
##  1     1   5
##  2     2   4
##  3     3   3
##  4     4   2
##  5     5   1
##  6    10   5
##  7    11   5.5
##  8    12   6
##  9    13   6.5
## 10    14   7
## 11    15   7.5
## 12    16   8
## 13     1   5
## 14     2   4
## 15     3   3
## 16     4   2
## 17     5   1
```

```
# use ".id" argument to create a new column that contains an identifier for the original data.
bind_rows(dat01, dat02, .id = "id")
```

```
## # A tibble: 12 x 3
##    id         x     y
##    <chr> <int> <dbl>
##  1 1         1   5
##  2 1         2   4
##  3 1         3   3
##  4 1         4   2
##  5 1         5   1
##  6 2        10   5
##  7 2        11   5.5
##  8 2        12   6
##  9 2        13   6.5
## 10 2        14   7
## 11 2        15   7.5
## 12 2        16   8
```

```
# with name
bind_rows("dat01" = dat01, "dat02" = dat02, .id = "id")
```

```
## # A tibble: 12 x 3
##    id         x     y
##    <chr> <int> <dbl>
##  1 dat01     1   5
##  2 dat01     2   4
##  3 dat01     3   3
##  4 dat01     4   2
##  5 dat01     5   1
##  6 dat02    10   5
##  7 dat02    11   5.5
##  8 dat02    12   6
##  9 dat02    13   6.5
## 10 dat02    14   7
## 11 dat02    15   7.5
## 12 dat02    16   8
```

```
# bind_rows() also works on lists of data frames
list01 <- list("dat01" = dat01, "dat02" = dat02)
list01
```

```
## $dat01
## # A tibble: 5 x 2
##       x     y
##   <int> <int>
## 1     1     5
## 2     2     4
## 3     3     3
## 4     4     2
## 5     5     1
##
## $dat02
## # A tibble: 7 x 2
##       x     y
##   <int> <dbl>
## 1    10   5
## 2    11   5.5
## 3    12   6
## 4    13   6.5
## 5    14   7
## 6    15   7.5
## 7    16   8
```

```
bind_rows(list01)
```

```
## # A tibble: 12 x 2
##        x     y
##    <int> <dbl>
##  1     1   5
##  2     2   4
##  3     3   3
##  4     4   2
##  5     5   1
##  6    10   5
##  7    11   5.5
##  8    12   6
##  9    13   6.5
## 10    14   7
## 11    15   7.5
## 12    16   8
```

```
bind_rows(list01, .id = "source")
```

```
## # A tibble: 12 x 3
##    source     x     y
##    <chr>  <int> <dbl>
##  1 dat01      1   5
##  2 dat01      2   4
##  3 dat01      3   3
##  4 dat01      4   2
##  5 dat01      5   1
##   6 dat02     10   5
##   7 dat02     11   5.5
##   8 dat02     12   6
##   9 dat02     13   6.5
##  10 dat02     14   7
##  11 dat02     15   7.5
##  12 dat02     16   8
```

```
# The extended example below demonstrates how this can be very handy.

# column binding
bind_cols(dat01, dat03)
```

```
## # A tibble: 5 x 3
##       x     y     z
##   <int> <int> <dbl>
## 1     1     5 0.532
## 2     2     4 0.570
## 3     3     3 0.447
## 4     4     2 0.479
## 5     5     1 0.714
```

```
# Regular expressions -----------------------------------------------------
names <- c("Ford, MS", "Jones, PhD", "Martin, Phd", "Huck, MA, MLS")

# pattern: first comma and everything after it
str_remove(names, pattern = ", [[:print:]]+")
```

```
## [1] "Ford"   "Jones"  "Martin" "Huck"
```

```
# [[:print:]]+ = one or more printable characters


# Reshaping ---------------------------------------------------------------

# Example of a wide data frame. Notice each person has multiple test scores
# that span columns.
wide <- data.frame(name=c("Clay","Garrett","Addison"),
                   test1=c(78, 93, 90),
                   test2=c(87, 91, 97),
                   test3=c(88, 99, 91))
wide
```

```
##      name test1 test2 test3
## 1    Clay    78    87    88
## 2 Garrett    93    91    99
## 3 Addison    90    97    91
```

```
# Example of a long data frame. This is the same data as above, but in long
# format. We have one row per person per test.
long <- data.frame(name=rep(c("Clay","Garrett","Addison"),each=3),
                   test=rep(1:3, 3),
                   score=c(78, 87, 88, 93, 91, 99, 90, 97, 91))
long
```

```
##      name test score
## 1    Clay    1    78
## 2    Clay    2    87
## 3    Clay    3    88
## 4 Garrett    1    93
## 5 Garrett    2    91
## 6 Garrett    3    99
## 7 Addison    1    90
## 8 Addison    2    97
## 9 Addison    3    91
```

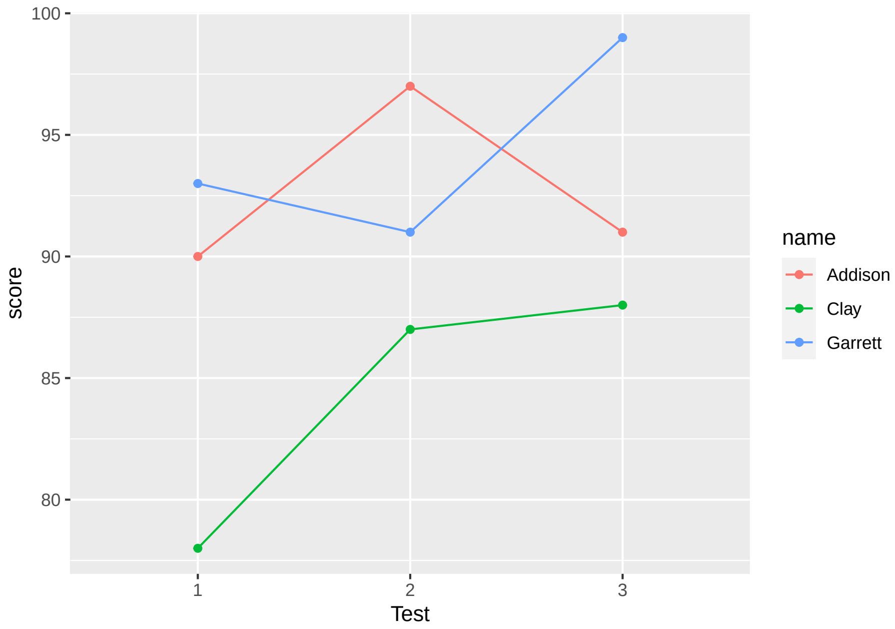
```
# mean score per student
aggregate(score ~ name, data = long, mean)
```

```
##      name    score
## 1 Addison 92.66667
## 2    Clay 84.33333
## 3 Garrett 94.33333
```

```
aggregate(score ~ test, data = long, mean)
```

```
##   test    score
## 1    1 87.00000
## 2    2 91.66667
## 3    3 92.66667
```

```
# line plot of scores over test, grouped by name
ggplot(long, aes(x = factor(test), y = score, color = name, group = name)) +
  geom_point() +
  geom_line() +
  xlab("Test")
```



```
#### reshape wide to long
pivot_longer(wide, test1:test3, names_to = "test", values_to = "score")
```

```
## # A tibble: 9 x 3
##   name    test  score
##   <chr>   <chr> <dbl>
## 1 Clay    test1    78
## 2 Clay    test2    87
## 3 Clay    test3    88
## 4 Garrett test1    93
## 5 Garrett test2    91
## 6 Garrett test3    99
## 7 Addison test1    90
## 8 Addison test2    97
## 9 Addison test3    91
```

```
# Or
pivot_longer(wide, -name, names_to = "test", values_to = "score")
```

```
## # A tibble: 9 x 3
##   name    test  score
##   <chr>   <chr> <dbl>
## 1 Clay    test1    78
## 2 Clay    test2    87
## 3 Clay    test3    88
## 4 Garrett test1    93
## 5 Garrett test2    91
## 6 Garrett test3    99
## 7 Addison test1    90
## 8 Addison test2    97
## 9 Addison test3    91
```

```
# drop "test" from the test column with names_prefix argument
pivot_longer(wide, -name, names_to = "test", values_to = "score",
             names_prefix = "test")
```

```
## # A tibble: 9 x 3
##   name    test  score
##   <chr>   <chr> <dbl>
## 1 Clay    1        78
## 2 Clay    2        87
## 3 Clay    3        88
## 4 Garrett 1        93
## 5 Garrett 2        91
## 6 Garrett 3        99
## 7 Addison 1        90
## 8 Addison 2        97
## 9 Addison 3        91
```

```
#### reshape long to wide
pivot_wider(long, name, names_from = test, values_from = score)
```

```
## # A tibble: 3 x 4
##   name      `1`   `2`   `3`
##   <chr>   <dbl> <dbl> <dbl>
## 1 Clay       78    87    88
## 2 Garrett    93    91    99
## 3 Addison    90    97    91
```

```
# using the names_prefix argument lets us prepend text to the column names.
pivot_wider(long, name, names_from = test, values_from = score,
            names_prefix = "test")
```

```
## # A tibble: 3 x 4
##   name    test1 test2 test3
##   <chr>   <dbl> <dbl> <dbl>
## 1 Clay       78    87    88
## 2 Garrett    93    91    99
## 3 Addison    90    97    91
```

The verbs of data manipulation

- select: selecting (or not selecting) columns based on their names (eg: select columns Q1 through Q25)
- slice: selecting (or not selecting) rows based on their position (eg: select

rows 1:10)

- mutate: add or derive new columns (or variables) based on existing columns (eg: create a new column that expresses measurement in cm based on existing measure in inches)
- rename: rename variables or change column names (eg: change "GraduationRate100" to "grad100")
- filter: selecting rows based on a condition (eg: all rows where gender = Male)
- arrange: ordering rows based on variable(s) numeric or alphabetical order (eg: sort in descending order of Income)
- sample: take random samples of data (eg: sample 80% of data to create a "training" set)
- summarize: condense or aggregate multiple values into single summary values (eg: calculate median income by age group)
- group_by: convert a tbl into a grouped tbl so that operations are performed "by group"; allows us to summarize data or apply verbs to data by groups (eg, by gender or treatment)
- the pipe: %>%
- Use Ctrl + Shift + M (Win) or Cmd + Shift + M (Mac) to enter in RStudio
- The pipe takes the output of a function and "pipes" into the first argument of the next function.

Part I

BASIC

Chapter 3

Descriptive Statistics

When you have an area of interest that you want to research, a problem that you want to solve, a relationship that you want to investigate, theoretical and empirical processes will help you.

Estimand is defined as "a quantity of scientific interest that can be calculated in the population and does not change its value depending on the data collection design used to measure it (i.e., it does not vary with sample size and survey design, or the number of nonrespondents, or follow-up efforts)." (Rubin, 1996)

Estimands include:

- population means
- Population variances
- correlations
- factor loading
- regression coefficients

3.1 Numerical Measures

There are differences between a population and a sample

Measures of	Category	Population	Sample
-	What is it?	Reality	A small fraction of reality (inference)
-	Characteristics described by	Parameters	Statistics
Central Tendency	Mean	$\mu = E(Y)$	$\hat{\mu} = \overline{y}$

Measures of	Category	Population	Sample
Central Tendency	Median	50-th percentile	$y_{(\frac{n+1}{2})}$
Dispersion	Variance	$\sigma^2 = var(Y)$ $= E(Y-\mu)^2$	$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \overline{y})^2$ $= \frac{1}{n-1}\sum_{i=1}^{n}(y_i^2 - n\overline{y}^2)$
Dispersion	Coefficient of Variation	$\frac{\sigma}{\mu}$	$\frac{s}{\overline{y}}$
Dispersion	Interquartile Range	difference between 25th and 75th percentiles. Robust to outliers	
Shape	Skewness Standardized 3rd central moment (unitless)	$g_1 = \frac{\mu_3}{\mu_2^{3/2}}$	$\hat{g_1} = \frac{m_3}{m_2 sqrt(m_2)}$
Shape	Central moments	$\mu = E(Y)$ $\mu_2 = \sigma^2 = E(Y-\mu)^2$ $\mu_3 = E(Y-\mu)^3$ $\mu_4 = E(Y-\mu)^4$	$m_2 = \sum_{i=1}^{n}(y_1 - \overline{y})^2/n$ $m_3 = \sum_{i=1}^{n}(y_1 - \overline{y})^3/n$
Shape	Kurtosis (peakedness and tail thickness) Standardized 4th central moment	$g_2^* = \frac{E(Y-\mu)^4}{\sigma^4}$	$\hat{g_2} = \frac{m_4}{m_2^2} - 3$

Note:

- Order Statistics: $y_{(1)}, y_{(2)}, ..., y_{(n)}$ where $y_{(1)} < y_{(2)} < ... < y_{(n)}$
- Coefficient of variation: standard deviation over mean. This metric is stable, dimensionless statistic for comparison.
- Symmetric: mean = median, skewness = 0
- Skewed right: mean > median, skewness > 0
- Skewed left: mean < median, skewness < 0
- Central moments: $\mu = E(Y)$, $\mu_2 = \sigma^2 = E(Y-\mu)^2$, $\mu_3 = E(Y-\mu)^3$, $\mu_4 = E(Y-\mu)^4$
- For normal distributions, $\mu_3 = 0$, so $g_1 = 0$
- $\hat{g_1}$ is distributed approximately as N(0,6/n) if sample is from a normal population. (valid when n > 150)

  - For large samples, inference on skewness can be based on normal tables with 95% confidence interval for g_1 as $\hat{g_1} \pm 1.96\sqrt{6/n}$
  - For small samples, special tables from Snedecor and Cochran 1989, Table A 19(i) or Monte Carlo test

Kurtosis > 0 (leptokurtic)	heavier tail	compared to a normal distribution with the same σ (e.g., t-distribution)
Kurtosis < 0 (platykurtic)	lighter tail	compared to a normal distribution with the same σ

- For a normal distribution, $g_2^* = 3$. Kurtosis is often redefined as: $g_2 = \frac{E(Y-\mu)^4}{\sigma^4} - 3$ where the 4th central moment is estimated by $m_4 = \sum_{i=1}^n (y_i - \overline{y})^4/n$
  - the asymptotic sampling distribution for $\hat{g_2}$ is approximately N(0,24/n) (with n > 1000)
  - large sample on kurtosis uses standard normal tables
  - small sample uses tables by Snedecor and Cochran, 1989, Table A 19(ii) or Geary 1936

```
data = rnorm(100)
library(e1071)
skewness(data,type = 1)
```

```
## [1] -0.04685268
```

```
kurtosis(data, type = 1)
```

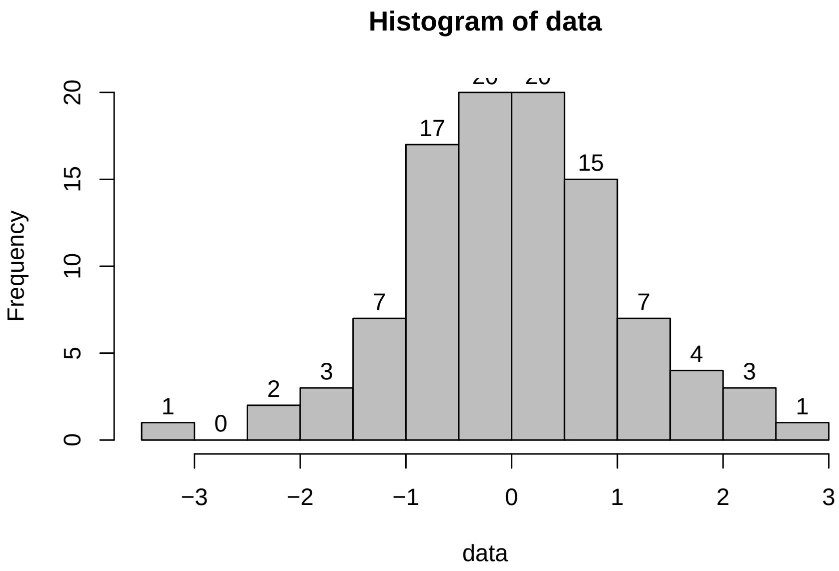
```
## [1] 0.1851826
```

3.2 Graphical Measures

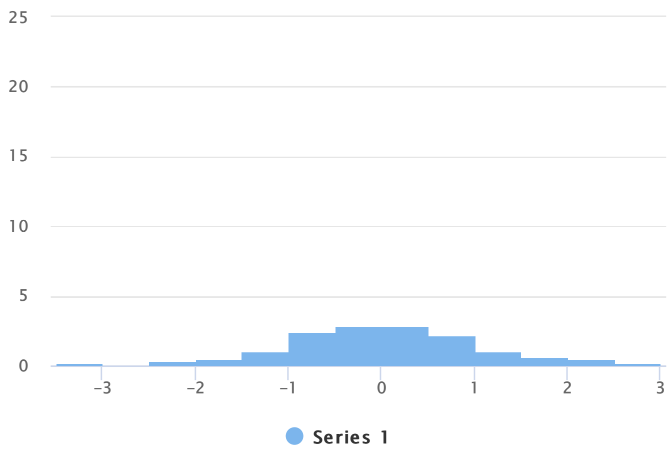
3.2.1 Shape

It's a good habit to label your graph, so others can easily follow.

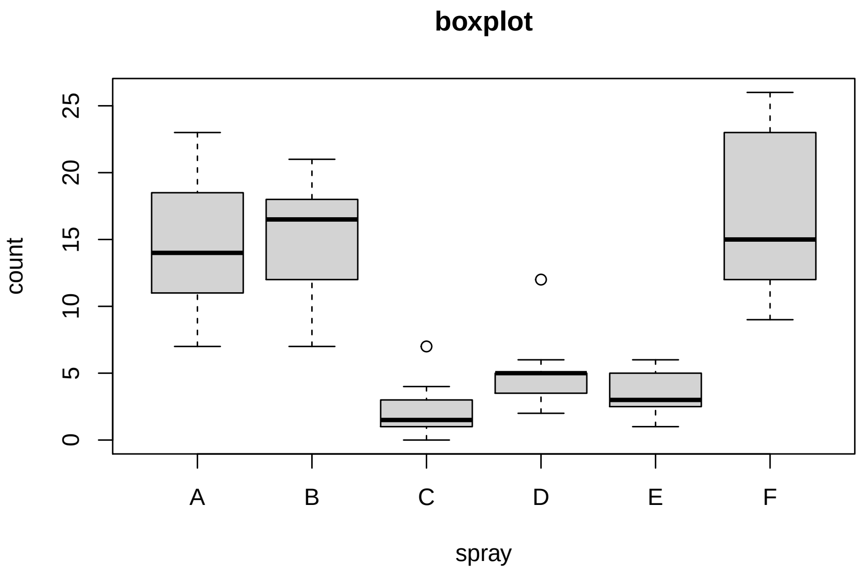
```
data = rnorm(100)

# Histogram
hist(data,labels = T,col="grey",breaks = 12)

# Interactive histogram
pacman::p_load("highcharter")
```



```
hchart(data)
```

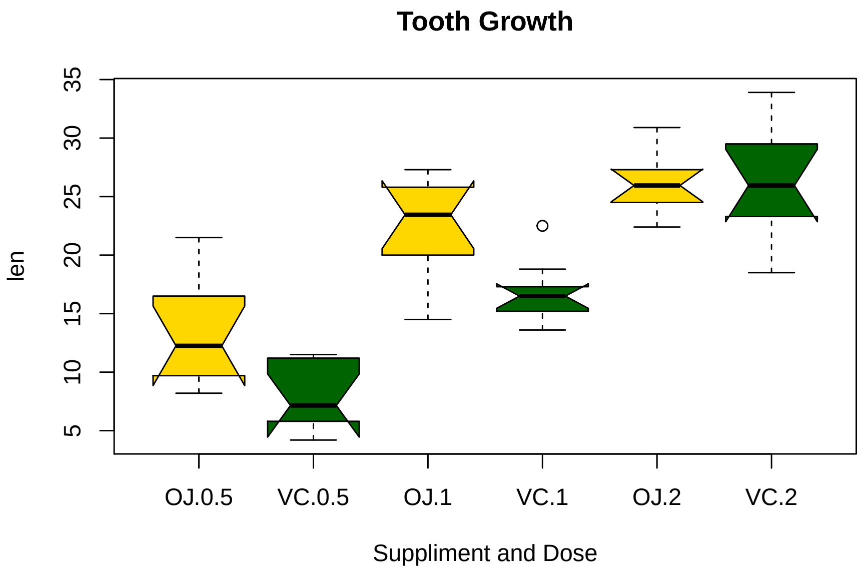


```
# Box-and-Whisker plot
boxplot(count ~ spray, data = InsectSprays,col = "lightgray",main="boxplot")
```



```
# Notched Boxplot
boxplot(len~supp*dose, data=ToothGrowth, notch=TRUE,
  col=(c("gold","darkgreen")),
  main="Tooth Growth", xlab="Suppliment and Dose")
```

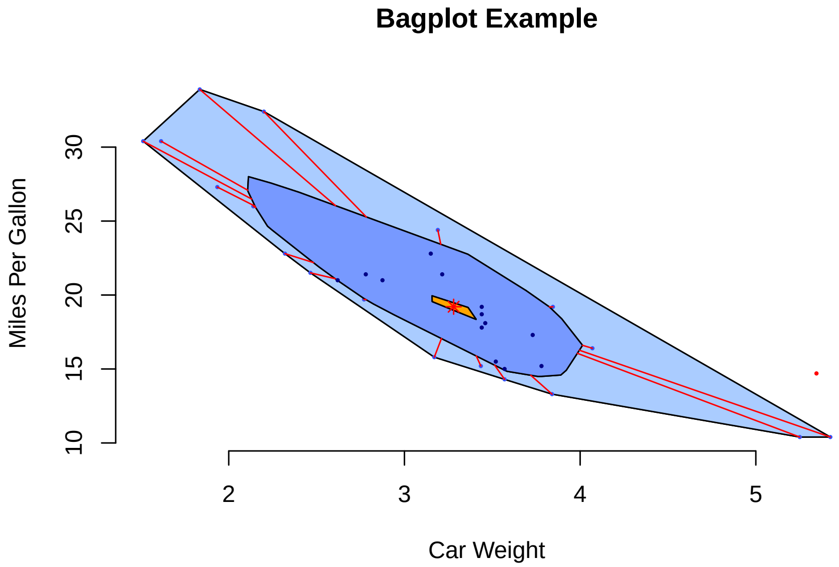
```
## Warning in bxp(list(stats = structure(c(8.2, 9.7, 12.25, 16.5, 21.5, 4.2, : some
## notches went outside hinges ('box'): maybe set notch=FALSE
```



```
# If notches differ -> medians differ

# Stem-and-Leaf Plots
stem(data)
```

```
##
##   The decimal point is at the |
##
##   -3 | 1
##   -2 | 30
##   -1 | 9765433221000
##   -0 | 999888777776555444433333221111100
##    0 | 11111222233333344455567777888888999
##    1 | 01122336668
##    2 | 1236
```

```
# Bagplot - A 2D Boxplot Extension
pacman::p_load(aplpack)
attach(mtcars)
bagplot(wt,mpg, xlab="Car Weight", ylab="Miles Per Gallon",
  main="Bagplot Example")
```



Others more advanced plots

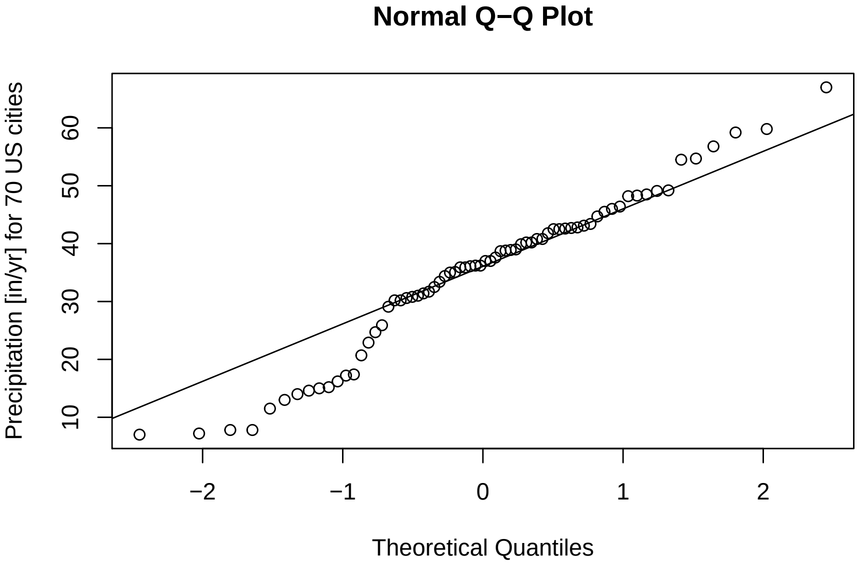
```
# boxplot.matrix()  #library("sfsmisc")
# boxplot.n()       #library("gplots")
# vioplot()         #library("vioplot")
```

3.3 Normality Assessment

Since Normal (Gaussian) distribution has many applications, we typically want/wish our data or our variable is normal. Hence, we have to assess the normality based on not only Numerical Measures but also Graphical Measures

3.3.1 Graphical Assessment

```
pacman::p_load("car")
qqnorm(precip, ylab = "Precipitation [in/yr] for 70 US cities")
qqline(precip)
```



The straight line represents the theoretical line for normally distributed data. The dots represent real empirical data that we are checking. If all the dots fall on the straight line, we can be confident that our data follow a normal distribution. If our data wiggle and deviate from the line, we should be concerned with the normality assumption.

3.3.2 Summary Statistics

Sometimes it's hard to tell whether your data follow the normal distribution by just looking at the graph. Hence, we often have to conduct statistical test to aid our decision. Common tests are

- Methods based on normal probability plot
  - Correlation Coefficient with Normal Probability Plots
  - Shapiro-Wilk Test
- Methods based on empirical cumulative distribution function
  - Anderson-Darling Test
  - Kolmogorov-Smirnov Test
  - Cramer-von Mises Test
  - Jarque–Bera Test

3.3.2.1 Methods based on normal probability plot

3.3.2.1.1 Correlation Coefficient with Normal Probability Plots (Looney and Gulledge, 1985) (Shapiro and Francia, 1972) The correlation coefficient between $y_{(i)}$ and m_i^* as given on the normal probability plot:

$$W^* = \frac{\sum_{i=1}^{n}(y_{(i)} - \bar{y})(m_i^* - 0)}{(\sum_{i=1}^{n}(y_{(i)} - \bar{y})^2 \sum_{i=1}^{n}(m_i^* - 0)^2)^{\cdot 5}}$$

where $\bar{m}^* = 0$

Pearson product moment formula for correlation:

$$\hat{p} = \frac{\sum_{i-1}^{n}(y_i - \bar{y})(x_i - \bar{x})}{(\sum_{i=1}^{n}(y_i - \bar{y})^2 \sum_{i=1}^{n}(x_i - \bar{x})^2)^{\cdot 5}}$$

- When the correlation is 1, the plot is exactly linear and normality is assumed.
- The closer the correlation is to zero, the more confident we are to reject normality
- Inference on W* needs to be based on special tables (Looney and Gulledge, 1985)

```
library("EnvStats")
```

```
##
## Attaching package: 'EnvStats'

## The following object is masked from 'package:car':
##
##     qqPlot

## The following objects are masked from 'package:e1071':
##
##     kurtosis, skewness

## The following objects are masked from 'package:stats':
##
##     predict, predict.lm

## The following object is masked from 'package:base':
##
##     print.default
```

```
gofTest(data,test="ppcc")$p.value #Probability Plot Correlation Coefficient
```

```
## [1] 0.8153719
```

3.3.2.1.2 Shapiro-Wilk Test (Shapiro and Wilk, 1965)

$$W = (\frac{\sum_{i=1}^n a_i(y_{(i)} - \bar{y})(m_i^* - 0)}{(\sum_{i=1}^n a_i^2(y_{(i)} - \bar{y})^2 \sum_{i=1}^n (m_i^* - 0)^2)^.5})^2$$

where $a_1, .., a_n$ are weights computed from the covariance matrix for the order statistics.

- Researchers typically use this test to assess normality. (n < 2000) Under normality, W is close to 1, just like W^*. Notice that the only difference between W and W* is the "weights".

```
gofTest(data,test="sw")$p.value #Shapiro-Wilk is the default.
```

```
## [1] 0.9653237
```

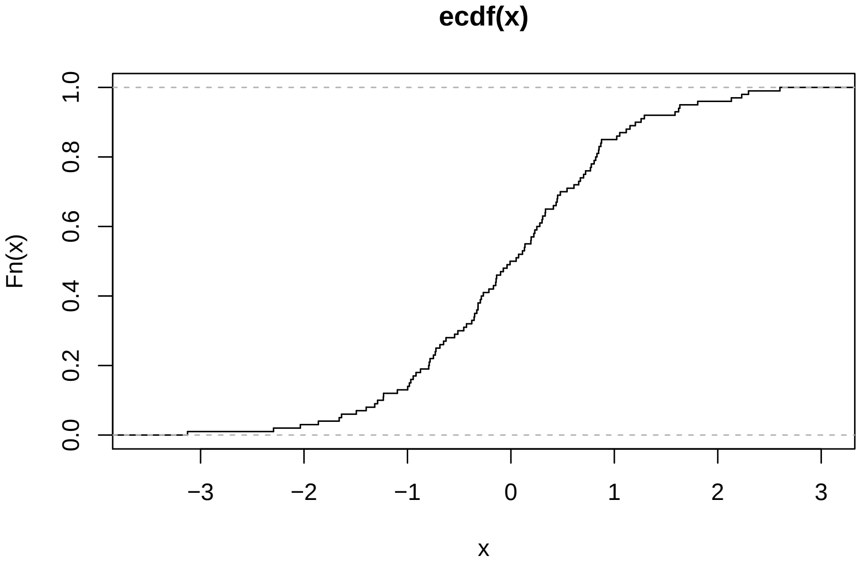
3.3.2.2 Methods based on empirical cumulative distribution function

The formula for the empirical cumulative distribution function (CDF) is:

$F_n(t)$ = estimate of probability that an observation ≤ t = (number of observation ≤ t)/n

This method requires large sample sizes. However, it can apply to distributions other than the normal (Gaussian) one.

```
# Empirical CDF hand-code
plot.ecdf(data,verticals = T, do.points=F)
```



3.3.2.2.1 Anderson-Darling Test (Anderson and Darling, 1952)

The Anderson-Darling statistic:

$$A^2 = \int_{-\infty}^{\infty} (F_n(t) = F(t))^2 \frac{dF(t)}{F(t)(1-F(t))}$$

- a weight average of squared deviations (it weights small and large values of t more)

For the normal distribution,

$A^2 = -(\sum_{i=1}^{n}(2i-1)(ln(p_i) + ln(1-p_{n+1-i}))/n - n$

where $p_i = \Phi(\frac{y_{(i)}-\bar{y}}{s})$, the probability that a standard normal variable is less than $\frac{y_{(i)}-\bar{y}}{s}$

- Reject normal assumption when A^2 is too large
- Evaluate the null hypothesis that the observations are randomly selected from a normal population based on the critical value provided by (Marsaglia and Marsaglia, 2004) and (Stephens, 1974)
- This test can be applied to other distributions:
  - Exponential
  - Logistic
  - Gumbel

- Extreme-value
- Weibull: log(Weibull) = Gumbel
- Gamma
- Logistic
- Cauchy
- von Mises
- Log-normal (two-parameter)

Consult (Stephens, 1974) for more detailed transformation and critical values.

```
gofTest(data,test="ad")$p.value #Anderson-Darling
```

```
## [1] 0.9594825
```

3.3.2.2.2 Kolmogorov-Smirnov Test

- Based on the largest absolute difference between empirical and expected cumulative distribution
- Another deviation of K-S test is Kuiper's test

```
gofTest(data,test="ks")$p.value #Komogorov-Smirnov
```

```
## Warning in ksGofTest(x = c(0.609788118656738, 1.05252871748987, 0.221995500532717,
```

```
## [1] 0.9653349
```

3.3.2.2.3 Cramer-von Mises Test

- Based on the average squared discrepancy between the empirical distribution and a given theoretical distribution. Each discrepancy is weighted equally (unlike Anderson-Darling test weights end points more heavily)

```
gofTest(data,test="cvm")$p.value #Cramer-von Mises
```

```
## [1] 0.9750525
```

3.3.2.2.4 Jarque–Bera Test (Bera and Jarque, 1981)

Based on the skewness and kurtosis to test normality.

$JB = \frac{n}{6}(S^2 + (K-3)^2/4)$ where S is the sample skewness and K is the sample kurtosis

$S = \frac{\hat{\mu_3}}{\hat{\sigma}^3} = \frac{\sum_{i=1}^n (x_i - \bar{x})^3/n}{(\sum_{i=1}^n (x_i - \bar{x})^2/n)^{\frac{3}{2}}}$

$K = \frac{\hat{\mu_4}}{\hat{\sigma}^4} = \frac{\sum_{i=1}^n (x_i - \bar{x})^4/n}{(\sum_{i=1}^n (x_i - \bar{x})^2/n)^2}$

recall $\hat{\sigma^2}$ is the estimate of the second central moment (variance) $\hat{\mu_3}$ and $\hat{\mu_4}$ are the estimates of third and fourth central moments.

If the data comes from a normal distribution, the JB statistic asymptotically has a chi-squared distribution with two degrees of freedom.

The null hypothesis is a joint hypothesis of the skewness being zero and the excess kurtosis being zero.

Chapter 4

Basic Statistical Inference

- One Sample Inference

- Two Sample Inference

- Categorical Data Analysis

- Make **inferences** (an interpretation) about the true parameter value β based on our estimator/estimate
- Test whether our underlying assumptions (about the true population parameters, random variables, or model specification) hold true.

Testing does not

- Confirm with 100% a hypothesis is true
- Confirm with 100% a hypothesis is false
- Tell you how to interpret the estimate value (Economic vs. Practical vs. Statistical Significance)

Hypothesis: Translate an objective in better understanding the results in terms of specifying a value (or sets of values) in which our population parameters should/should not lie.

- **Null hypothesis** (H_0): A statement about the population parameter that we take to be true in which we would need the data to provide substantial evidence that against it.
  - Can be either a single value (ex: $H_0 : \beta = 0$) or a set of values (ex: $H_0 : \beta_1 \geq 0$)
  - Will generally be the value you would not like the population parameter to be (subjective)
    - $H_0 : \beta_1 = 0$ means you would like to see a non-zero coefficient
    - $H_0 : \beta_1 \geq 0$ means you would like to see a negative effect
  - "Test of Significance" refers to the two-sided test: $H_0 : \beta_j = 0$

- **Alternative hypothesis** (H_a or H_1) (Research Hypothesis): All other possible values that the population parameter may be if the null hypothesis does not hold.

Type I Error

Error made when H_0 is rejected when, in fact, H_0 is true.
The probability of committing a Type I error is α (known as **level of significance** of the test)

Type I error (α): probability of rejecting H_0 when it is true.

Legal analogy: In U.S. law, a defendant is presumed to be "innocent until proven guilty".
If the null hypothesis is that a person is innocent, the Type I error is the probability that you conclude the person is guilty when he is innocent.

Type II Error

Type II error level (β): probability that you fail to reject the null hypothesis when it is false.

In the legal analogy, this is the probability that you fail to find the person guilty when he or she is guilty.

Error made when H_0 is not rejected when, in fact, H_1 is true
The probability of committing a Type II error is β (known as the **power** of the test)

Random sample of size n: A collection of n independent random variables taken from the distribution X, each with the same distribution as X.

Sample mean

$$\bar{X} = (\sum_{i=1}^{n} X_i)/n$$

Sample Median

$\tilde{x}$ = the middle observation in a sample of observation order from smallest to largest (or vice versa).

If n is odd, $\tilde{x}$ is the middle observation,
If n is even, $\tilde{x}$ is the average of the two middle observations.

Sample variance

$$S^2 = \frac{\sum_{i=1}^{n}(X_i = \bar{X})^2}{n-1} = \frac{n\sum_{i=1}^{n} X_i^2 - (\sum_{i=1}^{n} X_i)^2}{n(n-1)}$$

Sample standard deviation

$$S = \sqrt{S^2}$$

Sample proportions

$$\hat{p} = \frac{X}{n} = \frac{\text{number in the sample with trait}}{\text{sample size}}$$

$$\widehat{p_1 - p_2} = \hat{p_1} - \hat{p_2} = \frac{X_1}{n_1} - \frac{X_2}{n_2} = \frac{n_2 X_1 = n_1 X_2}{n_1 n_2}$$

Estimators
Point Estimator
$\hat{\theta}$ is a statistic used to approximate a population parameter θ

Point estimate
The numerical value assumed by $\hat{\theta}$ when evaluated for a given sample

Unbiased estimator
If $E(\hat{\theta}) = \theta$, then $\hat{\theta}$ is an unbiased estimator for θ

1. $\bar{X}$ is an unbiased estimator for μ
2. S^2 is an unbiased estimator for σ^2
3. $\hat{p}$ is an unbiased estimator for p
4. $\widehat{p_1 - P_2}$ is an unbiased estimator for $p_1 - p_2$
5. $\bar{X}_1 - \bar{X}_2$ is an unbiased estimator for $\mu_1 - \mu_2$

Note: S is a biased estimator for σ

Distribution of the sample mean

If $\bar{X}$ is the sample mean based on a random sample of size n drawn from a normal distribution X with mean μ and standard deviation σ, the $\bar{X}$ is normally distributed, with mean $\mu_{\bar{X}} = \mu$ and variance $\sigma^2_{\bar{X}} = Var(\bar{X}) = \frac{\sigma^2}{n}$. Then the **standard error of the mean** is: $\sigma_{\bar{X}} = \frac{\sigma}{\sqrt{n}}$

4.1 One Sample Inference

$Y_i \sim i.i.d. N(\mu, \sigma^2)$

i.i.d. standards for "independent and identically distributed"

Hence, we have the following model:

$Y_i = \mu + \epsilon_i$ where

- $\epsilon_i \sim^{iid} N(0, \sigma^2)$
- $E(Y_i) = \mu$
- $Var(Y_i) = \sigma^2$
- $\bar{y} \sim N(\mu, \sigma^2/n)$

4.1.1 The Mean

When σ^2 is estimated by s^2, then

$$\frac{\bar{y}-\mu}{s/\sqrt{n}} \sim t_{n-1}$$

Then, a $100(1-\alpha)\%$ confidence interval for μ is obtained from:

$$1-\alpha = P(-t_{\alpha/2;n-1} \leq \frac{\bar{y}-\mu}{s/\sqrt{n}} \leq t_{\alpha/2;n-1}) = P(\bar{y}-(t_{\alpha/2;n-1})s/\sqrt{n} \leq \mu \leq \bar{y}+(t_{\alpha/2;n-1})s/\sqrt{n})$$

And the interval is

$$\bar{y} \pm (t_{\alpha/2;n-1})s/\sqrt{n}$$

and $s/\sqrt{n}$ is the standard error of $\bar{y}$

If the experiment were repeated many times, $100(1-\alpha)\%$ of these intervals would contain μ

	Confidence Interval $100(1-\alpha)$	Sample Sizes Confidence α, Error d	Hypothesis Testing Test Statistic
When σ^2 is known, X is normal (or $n \geq 25$)	$\bar{X} \pm z_{\alpha/2}\frac{\sigma}{\sqrt{n}}$	$n \approx \frac{z^2_{\alpha/2}\sigma^2}{d^2}$	$z = \frac{\bar{X}-\mu_0}{\sigma/\sqrt{n}}$
When σ^2 is unknown, X is normal (or $n \geq 25$)	$\bar{X} \pm t_{\alpha/2}\frac{s}{\sqrt{n}}$	$n \approx \frac{z^2_{\alpha/2}s^2}{d^2}$	$t = \frac{\bar{X}-\mu_0}{s/\sqrt{n}}$

4.1.1.1 For Difference of Means ($\mu_1 - \mu_2$), Independent Samples

	$100(1-\alpha)$ Confidence Interval	Hypothesis Testing Test Statistic
When σ^2 is known	$\bar{X}_1 - \bar{X}_2 \pm z_{\alpha/2}\sqrt{\frac{\sigma_1^2}{n_1}+\frac{\sigma_2^2}{n_2}}$	$z = \frac{(\bar{X}_1-\bar{X}_2)-(\mu_1-\mu_2)_0}{\sqrt{\frac{\sigma_1^2}{n_1}+\frac{\sigma_2^2}{n_2}}}$

	$100(1-\alpha)$ Confidence Interval	Hypothesis Testing Test Statistic	
When σ^2 is unknown, Variances Assumed EQUAL	$\bar{X}_1 - \bar{X}_2 \pm t_{\alpha/2}\sqrt{s_p^2(\frac{1}{n_1} + \frac{1}{n_2})}$	$t = \frac{(\bar{X}_1-\bar{X}_2)-(\mu_1-\mu_2)_0}{\sqrt{s_p^2(\frac{1}{n_1}+\frac{1}{n_2})}}$	Pooled Variance: $s_p^2 = \frac{(n_1-1)s_1^2-(n_2-1)s_2^2}{n_1+n_2-2}$ Degrees of Freedom: $\gamma = n_1 + n_2 - 2$
When σ^2 is unknown, Variances Assumed UNEQUAL	$\bar{X}_1 - \bar{X}_2 \pm t_{\alpha/2}\sqrt{(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2})}$	$t = \frac{(\bar{X}_1-\bar{X}_2)-(\mu_1-\mu_2)_0}{\sqrt{(\frac{s_1^2}{n_1}+\frac{s_2^2}{n_2})}}$	Degrees of Freedom: $\gamma = \frac{(\frac{s_1^2}{n_1}+\frac{s_2^2}{n_2})^2}{\frac{(\frac{s_1^2}{n_1})^2}{n_1-1}+\frac{(\frac{s_2^2}{n_2})^2}{n_2-1}}$

4.1.1.2 For Difference of Means $(\mu_1-\mu_2)$, Paired Samples (D = X-Y)

$100(1-\alpha)$ **Confidence Interval**

$$\bar{D} \pm t_{\alpha/2}\frac{s_d}{\sqrt{n}}$$

Hypothesis Testing Test Statistic

$$t = \frac{\bar{D} - D_0}{s_d/\sqrt{n}}$$

4.1.1.3 Difference of Two Proprotions

Mean

$$\hat{p_1} - \hat{p_2}$$

Variance

$$\frac{p_1(1-p_1)}{n_1} + \frac{p_2(1-p_2)}{n_2}$$

$100(1-\alpha)$ **Confidence Interval**

$$\hat{p_1} - \hat{p_2} + z_{\alpha/2}\sqrt{\frac{p_1(1-p_1)}{n_1} + \frac{p_2(1-p_2)}{n_2}}$$

Sample Sizes, Confidence α, Error d
(Prior Estimate fo $\hat{p_1}, \hat{p_2}$)

$$n \approx \frac{z^2_{\alpha/2}[p_1(1-p_1)+p_2(1-p_2)]}{d^2}$$

(No Prior Estimates for $\hat{p}$)

$$n \approx \frac{z^2_{\alpha/2}}{2d^2}$$

Hypothesis Testing - Test Statistics

Null Value $(p_1 - p_2) \neq 0$

$$z = \frac{(\hat{p_1} - \hat{p_2}) - (p_1 - p_2)_0}{\sqrt{\frac{p_1(1-p_1)}{n_1} + \frac{p_2(1-p_2)}{n_2}}}$$

Null Value $(p_1 - p_2)_0 = 0$

$$z = \frac{\hat{p_1} - \hat{p_2}}{\sqrt{\hat{p}(1-\hat{p})(\frac{1}{n_1} + \frac{1}{n_2})}}$$

where

$$\hat{p} = \frac{x_1 + x_2}{n_1 + n_2} = \frac{n_1\hat{p_1} + n_2\hat{p_2}}{n_1 + n_2}$$

4.1.2 Single Variance

$$1-\alpha = P(\chi^2_{1-\alpha/2;n-1}) \leq (n-1)s^2/\sigma^2 \leq \chi^2_{\alpha/2;n-1}) = P(\frac{(n-1)s^2}{\chi^2_{\alpha/2}} \leq \sigma^2 \leq \frac{(n-1)s^2}{\chi^2_{1-\alpha/2}})$$

and a $100(1-\alpha)\%$ confidence interval for σ^2 is:

$$(\frac{(n-1)s^2}{\chi^2_{\alpha/2;n-1}}, \frac{(n-1)s^2}{\chi^2_{1-\alpha/2;n-1}})$$

Confidence limits for σ^2 are obtained by computing the positive square roots of these limits

Equivalently,

$100(1-\alpha)$ **Confidence Interval**

$$L_1 = \frac{(n-1)s^2}{\chi^2_{\alpha/2}} L_1 = \frac{(n-1)s^2}{\chi^2_{1-\alpha/2}}$$

Hypothesis Testing Test Statistic

$$\chi^2 = \frac{(n-1)s^2}{\sigma_0^2}$$

4.1.3 Single Proportion (p)

Confidence Interval $100(1-\alpha)$	Sample Sizes Confidence α, Error d (prior estimate for $\hat{p}$)	(No prior estimate for $\hat{p}$)	Hypothesis Testing Test Statistic
$\hat{p} \pm z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$	$n \approx \frac{z_{\alpha/2}^2\hat{p}(1-\hat{p})}{d^2}$	$n \approx \frac{z_{\alpha/2}^2}{4d^2}$	$z = \frac{\hat{p}-p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}}$

4.1.4 Power

Formally, power (for the test of the mean) is given by:

$$\pi(\mu) = 1 - \beta = P(\text{test rejects } H_0|\mu)$$

To evaluate the power, one needs to know the distribution of the test statistic if the null hypothesis is false.

For 1-sided z-test where $H_0 : \mu \leq \mu_0$
$H_A : \mu > 0$

The power is:

$$\begin{aligned}
\pi(\mu) &= P(\bar{y} > \mu_0 + z_\alpha \sigma/\sqrt{n}|\mu) \\
&= P(Z = \frac{\bar{y}-\mu}{\sigma/\sqrt{n}} > z_\alpha + \frac{\mu_0 - \mu}{\sigma/\sqrt{n}}|\mu) \\
&= 1 - \Phi(z_\alpha + \frac{(\mu_0 - \mu)\sqrt{n}}{\sigma}) \\
&= \Phi(-z_\alpha + \frac{(\mu - \mu_0)\sqrt{n}}{\sigma})
\end{aligned}$$

where $1 - \Phi(x) = \Phi(-x)$ since the normal pdf is symmetric

Power is correlated to the difference in $\mu - \mu_0$, sample size n, variance σ^2, and the α-level of the test (through z_α)
Equivalently, power can be increased by making α large, σ^2 smaller, or n larger.

For 2-sided z-test is:

$$\pi(\mu) = \Phi(-z_{\alpha/2} + \frac{(\mu_0 - \mu)\sqrt{n}}{\sigma}) + \Phi(-z_{\alpha/2} + \frac{(\mu - \mu_0)\sqrt{n}}{\sigma})$$

4.1.5 Sample Size

4.1.5.1 1-sided Z-test

Example: to show that the mean response μ under the treatment is higher than the mean response μ_0 without treatment (show that the treatment effect $\delta = \mu - \mu_0$ is large)

Because power is an increasing function of $\mu - \mu_0$, it is only necessary to find n that makes the power equal to $1-\beta$ at $\mu = \mu_0 + \delta$

Hence, we have

$$\pi(\mu_0 + \delta) = \Phi(-z_\alpha + \frac{\delta\sqrt{n}}{\sigma}) = 1 - \beta$$

Since $\Phi(z_\beta) = 1 - \beta$, we have

$$-z_\alpha + \frac{\delta\sqrt{n}}{\sigma} = z_\beta$$

Then n is

$$n = (\frac{(z_\alpha + z_\beta)\sigma}{\delta})^2$$

Then, we need larger samples, when

- the sample variability is large (σ is large)
- α is small (z_α is large)
- power $1-\beta$ is large (z_β is large)
- The magnitude of the effect is smaller (δ is small)

Since we don't know δ and σ. We can base σ on previous studies, pilot studies. Or, obtain an estimate of σ by anticipating the range of the observation (without outliers). divide this range by 4 and use the resulting number as an approximate estimate of σ. For normal (distribution) data, this is reasonable.

4.1.5.2 2-sided Z-test

We want to know the min n, required to guarantee $1-\beta$ power when the treatment effect $\delta = |\mu - \mu_0|$ is at least greater than 0. Since the power function for the 2-sided is increasing and symmetric in $|\mu - \mu_0|$, we only need to find n that makes the power equal to $1-\beta$ when $\mu = \mu_0 + \delta$

$$n = (\frac{(z_{\alpha/2} + z_\beta)\sigma}{\delta})^2$$

We could also use the confidence interval apporach. If we reuqire that an α-level two-cided CI for μ be

$$\bar{y} \pm D$$

where $D = z_{\alpha/2}\sigma/\sqrt{n}$ gives

$$n = (\frac{z_{\alpha/2}\sigma}{D})^2$$

(round up to the nearest integer)

```
data = rnorm(100)
t.test(data, conf.level=0.95)
```

```
##
##  One Sample t-test
##
## data:  data
## t = 0.1346, df = 99, p-value = 0.8932
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.1884123  0.2158339
## sample estimates:
##  mean of x
## 0.01371084
```

$$H_0 : \mu \geq 30 H_a : \mu < 30$$

```
t.test(data, mu=30,alternative="less")
```

```
##
##  One Sample t-test
##
## data:  data
## t = -294.37, df = 99, p-value < 2.2e-16
## alternative hypothesis: true mean is less than 30
## 95 percent confidence interval:
##       -Inf 0.1828473
## sample estimates:
##  mean of x
## 0.01371084
```

4.1.6 Note

For t-tests, the sample and power are not as easy as z-test.

$$\pi(\mu) = P(\frac{\bar{y} - \mu_0}{s/\sqrt{n}} > t_{n-1;\alpha}|\mu)$$

when $\mu > \mu_0$ (i.e., $\mu - \mu_0 = \delta$), the random variable $(\bar{y} - \mu_0)/(s/\sqrt{n})$ does not have a Student's t distribution, but rather is distributed as a non-central t-distribution with non-centrality parameter $\delta\sqrt{n}/\sigma$ and d.f. of $n-1$

- The power is an increasing function of this non-centrality parameter (note, when $\delta = 0$ the distribution is usual Student's t-distribution).
- To evaluate power, one must consider numerical procedure or use special charts

Approximate Sample Size Adjustment for t-test. We use an adjustment to the z-test determination for sample size.

Let $v = n - 1$, where n is sample size derived based on the z-test power. Then the 2-sided t-test sample size (apporximate) is given:

$$n^* = \frac{(t_{v;\alpha/2} + t_{v;\beta})^2\sigma^2}{\delta^2}$$

4.1.7 One-sample Non-parametric Methods

```
lecture.data=c(0.76, 0.82, 0.80, 0.79, 1.06, 0.83, -0.43, -0.34, 3.34, 2.33)
```

4.1.7.1 Sign Test

If we want to test $H_0 : \mu_{(0.5)} = 0; H_a : \mu_{(0.5)} > 0$ where $\mu_{(0.5)}$ is the population median. We can

(1) Count the number of observation (y_i's) that exceed 0. Denote this number by s_+, called the number of plus signs. Let $s_- = n - s_+$, which is the number of minus signs.
(2) Reject H_0 if s_+ is large or equivalently, if s_- is small.

To determine how large s_+ must be to reject H_0 at a given significance level, we need to know the distribution of the corresponding random variable S_+ under the null hypothesis, which is a binomial with p = 1/2,w hen the null is true.

To work out the null distribution using the binomial formula, we have α-level test rejects H_0 if $s_+ \geq b_{n,\alpha}$, where $b_{n,\alpha}$ is the upper α critical point of the $Bin(n, 1/2)$ distribution. Both S_+ and S_- have this same distribution ($S = S_+ = S_-$).

$$\text{p-value} = P(S \geq s_+) = \sum_{i=s_+}^{n} \binom{n}{i} (\frac{1}{2})^n$$

equivalently,

$$P(S \leq s_-) = \sum_{i=0}^{s_-} \binom{n}{i} (\frac{1}{2})^2$$

For large sample sizes, we could use the normal approximation for the binomial, in which case reject H_0 if

$$s_+ \geq n/2 + 1/2 + z_\alpha \sqrt{n/4}$$

For the 2-sided test, we use the tests statistic $s_{max} = max(s_+, s_-)$ or $s_{min} = min(s_+, s_-)$. An α-level test rejects H_0 if the p-value is $\leq \alpha$, where the p-value is computed from:

$$p - value = 2 \sum_{i=s_{max}}^{n} \binom{n}{i} (\frac{1}{2})^n = s \sum_{i=0}^{s_{min}} \binom{n}{i} (\frac{1}{2})^n$$

Equivalently, rejecting H_0 if $s_{max} \geq b_{n,\alpha/2}$

A large sample normal approximation can be used, where

$$z = \frac{s_{max} - n/2 - 1/2}{\sqrt{n/4}}$$

and reject H_0 at α if $z \geq z_{\alpha/2}$

However, treatment of 0 is problematic for this test.

- Solution 1: randomly assign 0 to the positive or negative (2 researchers might get different results).

- Solution 2: count each 0 as a contribution 1/2 toward s_+ and s_- (but then could not apply the binomial distribution)

- Solution 3: ignore 0 (reduces the power of test due to decreased sample size).

```
binom.test(sum(lecture.data > 0), length(lecture.data)) # alternative = "greater" or alternative
```

```
##
##  Exact binomial test
##
## data:  sum(lecture.data > 0) and length(lecture.data)
## number of successes = 8, number of trials = 10, p-value = 0.1094
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.4439045 0.9747893
## sample estimates:
## probability of success
##                       0.8
```

4.1.7.2 Wilcoxon Signed Rank Test

Since the Sign Test could not consider the magnitude of each observation from 0, the Wilcoxon Signed Rank Test improves by taking account the ordered magnitudes of the observation, but it will impose the requirement of symmetric to this test (while Sign Test does not)

$$H_0 : \mu_{0.5} = 0 H_a : \mu_{0.5} > 0$$

(assume no ties or same observations)

The signed rank test procedure:

1. rank order the observation y_i in terms of their absolute values. Let r_i be the rank of y_i in this ordering. Since we assume no ties, the ranks r_i are uniquely determined and are a permutation of the integers 1,2,…,n.

2. Calculate w_+, which is the sum of the ranks of the positive values, and w_-, which is the sum of the ranks of the negative values. Note that $w_+ + w_- = r_1 + r_2 + ... = 1 + 2 + ... + n = n(n+1)/2$

3. Reject H_0 if w_+ is large (or if w_- is small)

To know what is large or small with regard to w_+ and w_-, we need the distribution of W_+ and W_- when the null is true.

Since these null distributions are identical and symmetric, the p-value is $P(W \geq w_+) = P(W \leq w_-)$

An α-level test rejects the null if the p-value is $\leq \alpha$, or if $w_+ \geq w_{n,\alpha}$, where $w_{n,\alpha}$ is the upper α critical point of the null distribution of W.

This distribution of W has a special table. For large n, the distribution of W is approximately normal.

$$z = \frac{w_+ - n(n+1)/4 - 1/2}{\sqrt{n(n+1)(2n+1)/24}}$$

The test rejcets H_0 at level α if

$$w_+ \geq n(n+1)/4 + 1/2 + z_\alpha \sqrt{n(n+1)(2n+1)/24} \approx w_{n,\alpha}$$

For the 2-sided test, we use $w_{max} = max(w_+, w_-)$ or $w_{min} = min(w_+, w_-)$, with p-value given by:

$$p - value = 2P(W \geq w_{max}) = 2P(W \leq w_{min})$$

Same as Sign Test,we ignore 0. In some cases where some of the $|y_i|$'s may be tied for the same rank, we simply assign each of the tied ranks the average rank (or "midrank").

Example, if $y_1 = -1$, $y_3 = 3$ and $y_3 = -3$, and $y_4 = 5$, then $r_1 = 1$, $r_2 = r_3 = (2+3)/2 = 2.5$, $r_4 = 4$

```
wilcox.test(lecture.data) #does not use normal approximation (using the underlying W distributio
```

```
##
##  Wilcoxon signed rank exact test
##
## data:  lecture.data
## V = 52, p-value = 0.009766
## alternative hypothesis: true location is not equal to 0
```

```
wilcox.test(lecture.data,exact=F) #uses normal approximation
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  lecture.data
## V = 52, p-value = 0.01443
## alternative hypothesis: true location is not equal to 0
```

4.2 Two Sample Inference

4.2.1 Means

Suppose we have 2 sets of observations,

- $y_1, ..., y_{n_y}$
- $x_1, ..., x_{n_x}$

that are random samples from two independent populations with means μ_y and μ_x and variances σ^2_y,σ^2_x. Our goal is to compare μ_x and μ_y or $\sigma^2_y = \sigma^2_x$

4.2.1.1 Large Sample Tests

Assume that n_y and n_x are large (≥ 30). Then,

$$E(\bar{y} - \bar{x}) = \mu_y - \mu_x Var(\bar{y} - \bar{x}) = \sigma^2_y / n_y + \sigma^2_x / n_x$$

Then,

$$Z = \frac{\bar{y} - \bar{x} - (\mu_y - \mu_x)}{\sqrt{\sigma_y^2/n_y + \sigma_x^2/n_x}} \sim N(0,1)$$

(according to Central Limit Theorem). For large samples, we can replace variances by their unbiased estimators (s_y^2, s_x^2), and get the same large sample distribution.

An approximate $100(1-\alpha)\%$ CI for $\mu_y - \mu_x$ is given by:

$$\bar{y} - \bar{x} \pm z_{\alpha/2}\sqrt{s_y^2/n_y + s_x^2/n_x}$$

$$H_0 : \mu_y - \mu_x = \delta_0 H_A : \mu_y - \mu_x \neq \delta_0$$

at the α-level with the statistic:

$$z = \frac{\bar{y} - \bar{x} - \delta_0}{\sqrt{s_y^2/n_y + s_x^2/n_x}}$$

and reject H_0 if $|z| > z_{\alpha/2}$

If $\delta =$), it means that we are testing whether two means are equal.

4.2.1.2 Small Sample Tests

If the two samples are from normal distribution, iid $N(\mu_y, \sigma_y^2)$ and iid $N(\mu_x, \sigma_x^2)$ and the two samples are independent, we can do inference based on the t-distribution

Then we have 2 cases

- Equal Variance
- Unequal Variance

4.2.1.2.1 Equal variance Assumptions

- iid: so that $var(\bar{y}) = \sigma_y^2/n_y; var(\bar{x}) = \sigma_x^2/n_x$

- Independence between samples: No observation from one sample can influence any observation from the other sample, to have

$$\begin{aligned}
var(\bar{y} - \bar{x}) &= var(\bar{y}) + var\bar{x} - 2cov(\bar{y}, \bar{x}) \\
&= var(\bar{y}) + var\bar{x} \\
&= \sigma_y^2/n_y + \sigma_x^2/n_x
\end{aligned}$$

- Normality: Justifies the use of the t-distribution

Let $\sigma^2 = \sigma_y^2 = \sigma_x^2$. Then, s_y^2 and s_x^2 are both unbiased estimators of σ^2. We then can pool them.

Then the pooled variance estimate is

$$s^2 = \frac{(n_y - 1)s_y^2 + (n_x - 1)s_x^2}{(n_y - 1) + (n_x - 1)}$$

has $n_y + n_x - 2$ df.

Then the test statistic

$$T = \frac{\bar{y} - \bar{x} - (\mu_y - \mu_x)}{s\sqrt{1/n_y + 1/n_x}} \sim t_{n_y + n_x - 2}$$

$100(1-\alpha)\%$ CI for $\mu_y - \mu_x$ is

$$\bar{y} - \bar{x} \pm (t_{n_y + n_x - 2}) s\sqrt{1/n_y + 1/n_x}$$

Hypothesis testing:

$$H_0 : \mu_y - \mu_x = \delta_0 H_1 : \mu_y - \mu_x \neq \delta_0$$

we reject H_0 if $|t| > t_{n_y + n_x - 2;\alpha/2}$

4.2.1.2.2 Unequal Variance Assumptions

1. Two samples are independent
    1. Scatter plots
    2. Correlation coefficient (if normal)
2. Independence of observation in each sample
    1. Test for serial correlation

3. For each sample, homogeneity of variance
    1. Scatter plots
    2. Formal tests

4. Normality

5. Equality of variances (homogeneity of variance between samples)
    1. F-test
    2. Barlett test
    3. [Modified Levene Test]

To compare 2 normal $\sigma^2_y \neq \sigma^2_x$, we use the test statistic:

$$T = \frac{\bar{y} - \bar{x} - (\mu_y - \mu_x)}{\sqrt{s^2_y/n_y + s^2_x/n_x}}$$

In this case, T does not follow the t-distribution (its distribution depends on the ratio of the unknown variances σ^2_y, σ^2_x). In the case of small sizes, we can can approximate tests by using the Welch-Satterthwaite method (Satterthwaite, 1946). We assume T can be approximated by a t-distribution, and adjust the degrees of freedom.

Let $w_y = s^2_y/n_y$ and $w_x = s^2_x/n_x$ (the w's are the square of the respective standard errors)
Then, the degrees of freedom are

$$v = \frac{(w_y + w_x)^2}{w^2_y/(n_y - 1) + w^2_x/(n_x - 1)}$$

Since v is usually fractional, we truncate down to the nearest integer.

$100(1 - \alpha)\%$ CI for $\mu_y - \mu_x$ is

$$\bar{y} - \bar{x} \pm t_{v,\alpha/2}\sqrt{s^2_y/n_y + s^2_x/n_x}$$

Reject H_0 if $|t| > t_{v,\alpha/2}$, where

$$t = \frac{\bar{y} - \bar{x} - \delta_0}{\sqrt{s^2_y/n_y + s^2_x/n_x}}$$

4.2.2 Variances

$$F_{ndf,ddf} = \frac{s^2_1}{s^2_2}$$

where $s^2_1 > s^2_2, ndf = n_1 - 1, ddf = n_2 - 1$

4.2.2.1 F-test

Test

$$H_0 : \sigma^2_y = \sigma^2_x H_a : \sigma^2_y \neq \sigma^2_x$$

Consider the test statistic,

$$F = \frac{s_y^2}{s_x^2}$$

Reject H_0 if

- $F > f_{n_y-1,n_x-1,\alpha/2}$ or

- $F < f_{n_y-1,n_x-1,1-\alpha/2}$

Where $F > f_{n_y-1,n_x-1,\alpha/2}$ and $F < f_{n_y-1,n_x-1,1-\alpha/2}$ are the upper and lower $\alpha/2$ critical points of an F-distribution, with a $n_y - 1$ and $n_x - 1$ degrees of freedom.

Note

- This test depends heavily on the assumption Normality.

- In particular, it could give to many significant results when observations come from long-tailed distributions (i.e., positive kurtosis).

- If we cannot find support for normality, then we can use nonparametric tests such as the [Modified Levene Test]

```
data(iris)
irisVe=iris$Petal.Width[iris$Species=="versicolor"]
irisVi=iris$Petal.Width[iris$Species=="virginica"]

var.test(irisVe,irisVi)
```

```
##
##  F test to compare two variances
##
## data:  irisVe and irisVi
## F = 0.51842, num df = 49, denom df = 49, p-value = 0.02335
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.2941935 0.9135614
## sample estimates:
## ratio of variances
##          0.5184243
```

4.2.2.2 Modified Levene Test (Brown-Forsythe Test)

- considers averages of absolute deviations rather than squared deviations. Hence, less sensitive to long-tailed distributions.

- This test is still good for normal data

For each sample, we consider the absolute deviation of each observation form the median:

$$d_{y,i} = |y_i - y_{.5}| d_{x,i} = |x_i - x_{.5}|$$

Then,

$$t^*_L = \frac{\bar{d}_y - \bar{d}_x}{s\sqrt{1/n_y + 1/n_x}}$$

The pooled variance s^2 is given by:

$$s^2 = \frac{\sum_i^{n_y}(d_{y,i} - \bar{d}_y)^2 + \sum_j^{n_x}(d_{x,i} - \bar{d}_x)^2}{n_y + n_x - 2}$$

- If the error terms have constant variance and n_y and n_x are not extremely small, then $t^*_L \sim t_{n_x+n_y-2}$
- We reject the null hypothesis when $|t^*_L| > t_{n_y+n_x-2;\alpha/2}$
- This is just the two-sample t-test applied to the absolute deviations.

```
dVe=abs(irisVe-median(irisVe))
dVi=abs(irisVi-median(irisVi))
t.test(dVe,dVi,var.equal=T)
```

```
##
##  Two Sample t-test
##
## data:  dVe and dVi
## t = -2.5584, df = 98, p-value = 0.01205
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.12784786 -0.01615214
## sample estimates:
## mean of x mean of y
##     0.154     0.226
```

```
# small samples t-test
t.test(irisVe,irisVi,var.equal=F)
```

```
##
##  Welch Two Sample t-test
##
## data:  irisVe and irisVi
## t = -14.625, df = 89.043, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.7951002 -0.6048998
## sample estimates:
## mean of x mean of y
##     1.326     2.026
```

4.2.3 Power

Consider $\sigma_y^2 = \sigma_x^2 = \sigma^2$
Under the assumption of equal variances, we take size samples from both groups $(n_y = n_x = n)$

For 1-sided testing,

$$H_0 : \mu_y - \mu_x \leq 0 H_a : \mu_y - \mu_x > 0$$

α-level z-test rejects H_0 if

$$z = \frac{\bar{y} - \bar{x}}{\sigma\sqrt{2/n}} > z_\alpha$$

$$\pi(\mu_y - \mu_x) = \Phi(-z_\alpha + \frac{\mu_y - \mu_x}{\sigma}\sqrt{n/2})$$

We need sample size n that gie at least $1 - \beta$ power when $\mu_y - \mu_x = \delta$, where δ is the smallest difference that we want to see.

Power is given by:

$$\Phi(-z_\alpha + \frac{\delta}{\sigma}\sqrt{n/2}) = 1 - \beta$$

4.2.4 Sample Size

Then, the sample size is

$$n = 2(\frac{\sigma(z_\alpha + z_\beta}{\delta})^2$$

For 2-sided test, replace z_α with $z_{\alpha/2}$.
As with the one-sample case, to perform an exact 2-sample t-test sample size calculation, we must use a non-central t-distribution.

A correction that gives the approximate t-test sample size can be obtained by using the z-test n value in the formula:

$$n^* = 2(\frac{\sigma(t_{2n-2;\alpha} + t_{2n-2;\beta})}{\delta})^2$$

where we use $alpha/2$ for the two-sided test

4.2.5 Matched Pair Designs

We have two treatments

Subject	Treatment A	Treatment B	Difference
1	y_1	x_1	$d_1 = y_1 - x_1$
2	y_2	x_2	$d_2 = y_2 - x_2$
.	.	.	.
n	y_n	x_n	$d_n = y_n - x_n$

we assume $y_i \sim^{iid} N(\mu_y, \sigma_y^2)$ and $x_i \sim^{iid} N(\mu_x, \sigma_x^2)$, but since y_i and x_i are measured on the same subject, they are correlated.

Let

$$\mu_D = E(y_i - x_i) = \mu_y - \mu_x \sigma_D^2 = var(y_i - x_i) = Var(y_i) + Var(x_i) - 2cov(y_i, x_i)$$

If the matching induces **positive** correlation, then the variance of the difference of the measurements is reduced as compared to the independent case. This is the point of Matched Pair Designs. Although covariance can be negative, giving a larger variance of the difference than the independent sample case, usually the covariance is positive. This means both y_i and x_i are large for many of the same subjects, and for others, both measurement are small. (we still assume that various subjects respond independently of each other, which is necessary for the iid assumption within groups).

Let $d_i = y_i - x_i$, then

- $\bar{d} = \bar{y} - \bar{x}$ is the sample mean of the d_i
- $s_d^2 = \frac{1}{n-1}\sum_{i=1}^{n}(d_i - \bar{d})^2$ is the sample variance of the difference

Once the data are converted to differences, we are back to One Sample Inference and can use its tests and CIs.

4.2.6 Nonparametric Tests for Two Samples

For Matched Pair Designs, we can use the One-sample Non-parametric Methods.

Assume that Y and X are random variables with CDF F_y and F_x. then, Y is **stochastically** larger than X for all real number u, $P(Y > u) \geq P(X > u)$.

Equivalently, $P(Y \leq u) \leq P(X \leq u)$, which is $F_Y(u) \leq F_X(u)$, same thing as $F_Y < F_X$

If two distributions are identical, except that one is shifted relative to the other, then each of distribution can be indexed by a location parameter, say θ_y and θ_x. In this case, $Y > X$ if $\theta_y > \theta_x$

Consider the hypotheses,

$$H_0 : F_Y = F_X H_a : F_Y < F_X$$

where the alternative is an upper one-sided alternative.

- We can also consider the lower one-sided alternative

$$H_a : F_Y > F_X \text{ or} H_a : F_Y < F_X \text{ or } F_Y > F_X$$

- In this case, we don't use $H_a : F_Y \neq F_X$ as that allows arbitrary differences between the distributions, without requiring one be stochastically larger than the other.

If the distributions only differ in terms of their location parameters, we can focus hypothesis tests on the parameters (e.g., $H_0 : \theta_y = \theta_x$ vs. $\theta_y > \theta_x$)

We have 2 equivalent nonparametric tests that consider the hypothesis mentioned above

1. Wilcoxon rank test

2. Mann-Whitney U test

4.2.6.1 Wilcoxon rank test

1. Combine all $n = n_y + n_x$ observations and rank them in ascending order.

2. Sum the ranks of the y's and x's separately. Let w_y and w_x be these sums. ($w_y + w_x = 1 + 2 + ... + n = n(n+1)/2$)

3. Reject H_0 if w_y is large (equivalently, w_x is small)

Under H_0, any arrangement of the y's and x's is equally likely to occur, and there are $(n_y + n_x)!/(n_y!n_x!)$ possible arrangements.

- Technically, for each arrangement we can compute the values of w_y and w_x, and thus generate the distribution of the statistic under the null hypothesis.

- This could lead to computationally intensive.

```
wilcox.test(irisVe,irisVi,alternative="two.sided",conf.level=0.95, exact=F,correct=T)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  irisVe and irisVi
## W = 49, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

4.2.6.2 Mann-Whitney U test

The Mann-Whitney test is computed as follows:

1. Compare each y_i wiht each x_i.
   Let u_y be the number of pairs in which $y_i > x_i$ Let u_x be the number of pairs in which $y_i < x_i$. (assume there are no ties). There are $n_y n_x$ such comparisons and $u_y + u_x = n_y n_x$.

2. Reject H_0 if u_y is large (or u_x is small)

Mann-Whitney U test and Wilcoxon rank test are related:

$$u_y = w_y - n_y(n_y + 1)/2 u_x = w_x - n_x(n_x + 1)/2$$

An α-level test rejects H_0 if $u_y \geq u_{n_y,n_x,\alpha}$, where $u_{n_y,n_x,\alpha}$ is the upper α critical point of the null distribution of the random variable, U.

The p-value is defined to be $P(Y \geq u_y) = P(U \leq u_x)$. One advantage of Mann-Whitney U test is that we can use either u_y or u_x to carry out the test.

For large n_y and n_x, the null distribution of U can be well approximated by a normal distribution with mean $E(U) = n_y n_x/2$ and variance $var(U) = n_y n_x(n+1)/12$. A large sample z-test can be based on the statistic:

$$z = \frac{u_y - n_y n_x/2 - 1/2}{\sqrt{n_y n_x(n+1)/12}}$$

The test rejects H_0 at level α if $z \geq z_\alpha$ or if $u_y \geq u_{n_y,n_x,\alpha}$ where

$$u_{n_y,n_x,\alpha} \approx n_y n_x/2 + 1/2 + z_\alpha \sqrt{n_y n_x(n+1)/12}$$

For the 2-sided test, we use the test statistic $u_{max} = max(u_y, u_x)$ and $u_{min} = min(u_y, u_x)$ and p-value is given by

$$p - value = 2P(U \geq u_{max}) = 2P(U \leq u_{min})$$

Since we assume there are no ties (when $y_i = x_j$), we count 1/2 towards both u_y and u_x. Even though the sampling distribution is not the same, but large sample approximation is still reasonable,

4.3 Categorical Data Analysis

Categorical Data Analysis when we have categorical outcomes

- Nominal variables: no logical ordering (e.g., sex)

- Ordinal variables: logical order, but relative distances between values are not clear (e.g., small, medium, large)

The distribution of one variable changes when the level (or values) of the other variable change. The row percentages are different in each column.

4.3.1 Inferences for Small Samples

The approximate tests based on the asymptotic normality of $\hat{p}_1 - \hat{p}_2$ do not apply for small samples.

Using **Fisher's Exact Test** to evaluate $H_0 : p_1 = p_2$

- Assume X_1 and X_2 are independent Binomial

- Let x_1 and x_2 be the corresponding observed values.
- Let $n = n_1 + n_2$ be the total sample size
- $m = x_1 + x_2$ be the observed number of successes.

- By assuming that m (total successes) is fixed, and conditioning on this value, one can show that the conditional distribution of the number of successes from sample 1 is Hypergeometric

- If we want to test $H_0 : p_1 = p_2$ and $H_a : p_1 \neq p_2$, we have

$$Z^2 = (\frac{\hat{p}_1 - \hat{p}_2}{\sqrt{\hat{p}(1-\hat{p})(1/n_1 + 1/n_2)}})^2 \sim \chi^2_{1,\alpha}$$

where $\chi^2_{1,\alpha}$ is the upper α percentage point for the central Chi-squared with one d.f.

This extends to the contingency table setting: whether the observed frequencies are equal to those expected under a null hypothesis of no association.

4.3.2 Test of Association

Pearson Chi-square test statistic is

$$\chi^2 = \sum_{\text{all categories}} \frac{(observed - epxected)^2}{expected}$$

Comparison of proportions for several independent surveys or experiments

	Experiment 1	Experiment 2	...	Experiment k
Number of successes	x_1	x_2	...	x_k
Number of failures	$n_1 - x_1$	$n_2 - x_2$	...	$n_k - x_k$
	n_1	n_2	...	n_k

$H_0 : p_1 = p_2 = ... = p_k$ vs. the alternative that the null is not true (at least one pair are not equal).

We estimate the common value of the probability of success on a single trial assuming H_0 is true:

$$\hat{p} = \frac{x_1 + x_2 + ... + x_k}{n_1 + n_2 + ... + n_k}$$

we use table of expected counts when H_0 is true:

success	$n_1\hat{p}$	$n_2\hat{p}$	...	$n_k\hat{p}$
failure	$n_1(1-\hat{p})$	$n_2(1-\hat{p})$	...	$n_k(1-\hat{p})$
	n_1	n_2	...	n_k

$$\chi^2 = \sum_{\text{all cells in table}} \frac{(observed - expected)^2}{expected}$$

with k-1 degrees of freedom

4.3.2.1 Two-way Count Data

	1	2	...	j	...	c	Row Total
1	n_{11}	n_{12}	...	n_{1j}	...	n_{1c}	$n_{1.}$
2	n_{21}	n_{22}	...	n_{2j}	...	n_{2c}	$n_{2.}$
.	.	.	.	.	.	.	.
r	n_{r1}	n_{r2}	...	n_{rj}	...	n_{rc}	$n_{r.}$
Column Total	$n_{.1}$	$n_{.2}$	...	$n_{.j}$	...	$n_{.c}$	n

Design 1
total sample size fixed n = constant (e.g., survey on job satisfaction and income); both row and column totals are random variables

Design 2
Fix the sample size in each group (in each row) (e.g., Drug treatments success or failure); fixed number of participants for each treatment; independent random samples from the two row populations.

These different sampling designs imply two different probability models.

4.3.2.2 Total Sample Size Fixed

Design 1

random sample of size n drawn from a single population, and sample units are cross-classified into r row categories and c column

This results in an r x c table of observed counts

$n_{ij} = 1, ..., r; j = 1, ..., c$

Let p_{ij} be the probability of classification into cell (i,j) and $\sum_{i=1}^{r}\sum_{j=1}^{c} p_{ij} = 1$.
Let N_{ij} be the random variable corresponding to n_{ij}
The joint distribution of the N_{ij} is multinomial with unknown parameters p_{ij}

Denote the row variable by X and column variable by Y, then $p_{ij} = P(X = i, Y = j)$ and $p_{i.} = P(X = i)$ and $p_{.j} = P(Y = j)$ are the marginal probabilities. The null hypothesis that X and Y are statistically independent (i.e., no association) is just:

$$H_0 : p_{ij} = P(X = i, Y = j) = P(X = i)P(Y = j) = p_{i.}p_{.j} H_a : p_{ij} \neq p_{i.}p_{.j}$$

for all i,j.

4.3.2.3 Row Total Fixed

Design 2

Random samples of sizes $n_1, ..., n_r$ are drawn independently from $r \geq 2$ row populations. In this case, the 2-way table row totals are $n_{i.} = n_i$ for $i = 1, ..., r$.

The counts from each row are modeled by independent multinomial distributions.

X is fixed, Y is observed.

Then, p_{ij} represent conditional probabilities $p_{ij} = P(Y = j|X = i)$

The null hypothesis is the probability of response j is the same, regardless of the row population (i.e., no association):

$H_0 : p_{ij} = P(Y = j|X = i) = p_j$for all i,j =1,2,...,cor $H_0 : (p_{i1}, p_{i2}, ..., p_{ic}) = (p_1, p_2, ..., p_c)$ for all i$H_a : (p$

Although the hypotheses to be tested are different for two sampling designs, **The chi-square test is identical**

We have estimated expected frequencies:

$$\hat{e}_{ij} = \frac{n_{i.} n_{.j}}{n}$$

The Chi-square statistic is

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(n_{ij} - \hat{e}_{ij})^2}{\hat{e}_{ij}} \sim \chi_{(r-1)(c-1)}$$

α-level test rejects H_0 if $\chi^2 > \chi^2_{(r-1)(c-1),\alpha}$

4.3.2.4 Pearson Chi-square Test

- Determine whether an association exists

- Sometimes, H_0 represents the model whose validity is to be tested. Contrast this with the conventional formulation of H_0 as the hypothesis that is to be disproved. The goal in this case is not to disprove the model, but to see whether data are consistent with the model and if deviation can be attributed to chance.

- These tests do not measure the strength of an association.

- These tests depend on and reflect the sample size - double the sample size by copying each observation, double the χ^2 statistic even thought the strength of the association does not change.

- The Pearson Chi-square Test is not appropriate when more than about 20% of the cells have an expected cell frequency of less than 5 (large-sample p-values not appropriate).

- When the sample size is small the exact p-values can be calculated (this is prohibitive for large samples); calculation of the exact p-values assumes that the column totals and row totals are fixed.

```
july.x=480
july.n=1000
sept.x=704
sept.n=1600
```

$$H_0 : p_J = 0.5 H_a : p_J < 0.5$$

```
prop.test(x=july.x,n=july.n,p=0.5,alternative="less",correct=F)
```

```
##
##  1-sample proportions test without continuity correction
##
## data:  july.x out of july.n, null probability 0.5
## X-squared = 1.6, df = 1, p-value = 0.103
## alternative hypothesis: true p is less than 0.5
## 95 percent confidence interval:
##  0.0000000 0.5060055
## sample estimates:
##    p
## 0.48
```

$$H_0 : p_J = p_S H_a : p_j \neq p_S$$

```
prop.test(x=c(july.x,sept.x),n=c(july.n,sept.n),correct=F)
```

```
##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(july.x, sept.x) out of c(july.n, sept.n)
## X-squared = 3.9701, df = 1, p-value = 0.04632
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.0006247187 0.0793752813
## sample estimates:
## prop 1 prop 2
##   0.48   0.44
```

4.3.3 Ordinal Association

- An ordinal association implies that as one variable increases, the other tends to increase or decrease (depending on the nature of the association).

- For tests for variables with two or more levels, the levels must be in a logical ordering.

4.3.3.1 Mantel-Haenszel Chi-square Test

The Mantel-Haenszel Chi-square Test is more powerful for testing ordinal associations, but does not test for the strength of the association.

This test is presented in the case where one has a series of 2 x 2 tables that examine the same effects under different conditions (If there are K such tables, we have 2 x 2 x K table)

In stratum k, given the marginal totals $(n_{.1k}, n_{.2k}, n_{1.k}, n_{2.k})$, the sampling model for cell counts is the Hypergeometric (knowing n_{11k} determines $(n_{12k}, n_{21k}, n_{22k})$, given the marginal totals)

Assuming conditional independence, the Hypergeometric mean and variance of n_{11k} are

$$m_{11k} = E(n_{11k}) = \frac{n_{1.k}n_{.1k}}{n_{..k}} var(n_{11k}) = \frac{n_{1.k}n_{2.k}n_{.1k}n_{.2k}}{n^2_{..k}(n_{..k}-1)}$$

To test conditional independence, Mantel and Haenszel proposed

$$M^2 = \frac{(|\sum_k n_{11k} - \sum_k m_{11k}| - .5)^2}{\sum_k var(n_{11k})} \sim \chi^2_1$$

This method can be extended to general I x J x K tables.

(2 x 2 x 3) table

```
Bron=array(c(20, 9, 382, 214, 10, 7, 172, 120, 12, 6, 327, 183), dim = c(2, 2, 3), dimn
margin.table(Bron,c(1,2))
```

```
##            Bronchitis
## Particulate Yes  No
##        High  42 881
##        Low   22 517
```

```
# assess whether the relationship between Bronchitis by Particulate Level varies by Ag
library(samplesizeCMH)
marginal_table=margin.table(Bron,c(1,2))
odds.ratio(marginal_table)
```

```
## [1] 1.120318
```

```
#  whether these odds vary by age. The conditional odds can be calculated using the or
apply(Bron,3,odds.ratio)
```

```
##     15-24     25-39       40+
## 1.2449098 0.9966777 1.1192661
```

```
# Mantel-Haenszel Test
mantelhaen.test(Bron,correct=T)
```

```
##
##  Mantel-Haenszel chi-squared test with continuity correction
##
```

```
## data:  Bron
## Mantel-Haenszel X-squared = 0.11442, df = 1, p-value = 0.7352
## alternative hypothesis: true common odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.6693022 1.9265813
## sample estimates:
## common odds ratio
##          1.135546
```

4.3.3.1.1 McNemar's Test special case of Mantel-Haenszel Chi-square Test

```
vote=cbind(c(682,22),c(86,810))
mcnemar.test(vote,correct=T)
```

```
##
##  McNemar's Chi-squared test with continuity correction
##
## data:  vote
## McNemar's chi-squared = 36.75, df = 1, p-value = 1.343e-09
```

4.3.3.2 Spearman Rank Correlation

To test for the strength of association between two ordinally scaled variables, we can use Spearman Rank Correlation statistic

Let X and Y be two random variables measured on an ordinal scale. Consider n pairs of observations (x_i, y_i), i = 1,…,n

The Spearman Rank Correlation coefficient (denoted by r_S is calculated using the Pearson correlation formula, but based on the ranks of x_i and y_i).

Spearman Rank Correlation be calculated

1. Assign ranks to x_i's and y_i's separately. Let $u_i = rank(x_i)$ and $v_i = rank(y_i)$

2. Calculate r_S using the formula for the Pearson correlation coefficient, but applied to the ranks:

$$r_S = \frac{\sum_{i=1}^{n}(u_i - \bar{u})(v_i - \bar{v})}{\sqrt{(\sum_{i=1}^{n}(u_i - \bar{u})^2)(\sum_{i=1}^{n}(v_i - \bar{v})^2)}}$$

r_S ranges between -1 and +1 , with

- $r_S = -1$ if there is a perfect negative monotone association
- $r_S = +1$ if there is a perfect positive monotone association between X and Y.

To test

H_0 : X and Y independent
H_a: X and Y positively associated

For large n (e.g., $n \geq 10$),

$$r_S \sim N(0, 1/(n-1))$$

Then,

$$Z = r_s\sqrt{n-1} \sim N(0,1)$$

Part II

REGRESSION

Chapter 5

Linear Regression

Estimator Desirable Properties

1. Unbiased

2. Consistency

- $plim\hat{\beta_n} = \beta$
- based on the law of large numbers, we can derive consistency
- More observations means more precise, closer to the true value.

3. Efficiency

- Minimum variance in comparison to another estimator.
  - OLS is BlUE (best linear unbiased estimator) means that OLS is the most efficient among the class of linear unbiased estimator Gauss-Markov Theorem
  - If we have correct distributional assumptions, then the Maximum Likelihood is asymptotically efficient among consistent estimators.

5.1 Ordinary Least Squares

The most fundamental model in statistics or econometric is a OLS linear regression. OLS = Maximum likelihood when the error term is assumed to be normally distributed.

5.1.1 Simple Regression (Basic Model)

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

- Y_i: response (dependent) variable at i-th observation

- β_0, β_1: regression parameters for intercept and slope.
- X_i: known constant (independent or predictor variable) for i-th observation
- ϵ_i: random error term

$$E(\epsilon_i) = 0 var(\epsilon_i) = \sigma^2$$

$$cov(\epsilon_i, \epsilon_j) = 0 \text{ for all } i \neq j$$

Y_i is random since ϵ_i is:

$$\begin{aligned} E(Y_i) &= E(\beta_0 + \beta_1 X_i + \epsilon_i) \\ &= E(\beta_0) + E(\beta_1 X_i) + E(\epsilon) \\ &= \beta_0 + \beta_1 X_i \end{aligned}$$

$$\begin{aligned} var(Y_i) &= var(\beta_0 + \beta_1 X_i + \epsilon_i) \\ &= var(\epsilon_i) \\ &= \sigma^2 \end{aligned}$$

Since $cov(\epsilon_i, \epsilon_j) = 0$ (uncorrelated), the outcome in any one trail has no effect on the outcome of any other. Hence, Y_i, Y_j are uncorrelated as well (conditioned on the X's)

Note
Least Squares does not require a distributional assumption

5.1.1.1 Estimation

Deviation of Y_i from its expected value:

$$Y_i - E(Y_i) = Y_i - (\beta_0 + \beta_1 X_i)$$

Consider the sum of the square of such deviations:

$$Q = \sum_{i=1}^{n} (Y_i - \beta_0 - \beta_1 X_i)^2$$

$$b_1 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^{n}(X_i - \bar{X})^2} b_0 = \frac{1}{n}(\sum_{i=1}^{n} Y_i - b_1 \sum_{i=1}^{n} X_i) = \bar{Y} - b_1 \bar{X}$$

5.1.1.2 Properties of Least Least Estimators

$$E(b_1) = \beta_1 E(b_0) = E(\bar{Y}) - \bar{X}\beta_1 E(\bar{Y}) = \beta_0 + \beta_1 \bar{X} E(b_0) = \beta_0 var(b_1) = \frac{\sigma^2}{\sum_{i=1}^{n}(X_i - \bar{X})^2} var(b_0) = \sigma^2(\frac{1}{n} + \frac{\bar{X}^2}{\sum(X_i - \bar{X})^2}$$

$var(b_1)$ approaches 0 as more measurements are taken at more X_i values (unless X_i is at its mean value)
$var(b_0)$ approaches 0 as n increases when the X_i values are judiciously selected.

Mean Square Error

$$MSE = \frac{SSE}{n-2} = \frac{\sum_{i=1}^{n} e_i^2}{n-2} = \frac{\sum(Y_i - \hat{Y}_i)^2}{n-2}$$

Unbiased estimator of MSE:

$$E(MSE) = \sigma^2$$

$$s^2(b_1) = \widehat{var(b_1)} = \frac{MSE}{\sum_{i=1}^{n}(X_i - \bar{X})^2} s^2(b_0) = \widehat{var(b_0)} = MSE(\frac{1}{n} + \frac{\bar{X}^2}{\sum_{i=1}^{n}(X_i - \bar{X})^2})$$

$$E(s^2(b_1)) = var(b_1) E(s^2(b_0)) = var(b_0)$$

5.1.1.3 Residuals

$$e_i = Y_i - \hat{Y} = Y_i - (b_0 + b_1 X_i)$$

- e_i is an estimate of $\epsilon_i = Y_i - E(Y_i)$
- ϵ_i is always unknown since we don't know the true β_0, β_1

$$\sum_{i=1}^{n} e_i = 0 \sum_{i=1}^{n} X_i e_i = 0$$

5.1.1.4 Inference

Normality Assumption

- Least Squares estimation does not require assumptions of normality.
- However, to do inference on the parameters, we need distributional assumptions.

- Inference on β_0, β_1 and Y_h are not extremely sensitive to moderate departures from normality, especially if the sample size is large

- Inference on Y_{pred} is very sensitive to the normality assumptions.

Normal Error Regression Model

$$Y_i \sim N(\beta_0 + \beta_1 X_i, \sigma^2)$$

5.1.1.4.1 β_1 Under the normal error model,

$$b_1 \sim N(\beta_1, \frac{\sigma^2}{\sum_{i=1}^{n}(X_i - \bar{X})^2})$$

A linear combination of independent normal random variable is normally distributed

Hence,

$$\frac{b_1 - \beta_1}{s(b_1)} \sim t_{n-2}$$

A $(1-\alpha)100\%$ confidence interval for β_1 is

$$b_1 \pm t_{t-\alpha/2;n-2} s(b_1)$$

5.1.1.4.2 β_0 Under the normal error model, the sampling distribution for b_0 is

$$b_0 \sim N(\beta_0, \sigma^2(\frac{1}{n} + \frac{\bar{X}^2}{\sum_{i=1}^{n}(X_i - \bar{X})^2}))$$

Hence,

$$\frac{b_0 - \beta_0}{s(b_0)} \sim t_{n-2}$$

A $(1-\alpha)100\%$ confidence interval for β_0 is

$$b_0 \pm t_{1-\alpha/2;n-2} s(b_0)$$

5.1.1.4.3 Mean Response Let X_h denote the level of X for which we wish to estimate the mean response

- We denote the mean response when $X = X_h$ by $E(Y_h)$
- A point estimator of $E(Y_h)$ is $\hat{Y}_h$:

$$\hat{Y}_h = b_0 + b_1 X_h$$

Note

$$E(\bar{Y}_h) = E(b_0 + b_1 X_h) = \beta_0 + \beta_1 X_h = E(Y_h)$$

(unbiased estimator)

$$\begin{aligned}
var(\hat{Y}_h) &= var(b_0 + b_1 X_h) \\
&= var(\hat{Y} + b_1(X_h - \bar{X})) \\
&= var(\bar{Y}) + (X_h - \bar{X})^2 var(b_1) + 2(X_h - \bar{X}) cov(\bar{Y}, b_1) \\
&= \frac{\sigma^2}{n} + (X_h - \bar{X})^2 \frac{\sigma^2}{\sum (X_i - \bar{X})^2} \\
&= \sigma^2 (\frac{1}{n} + \frac{(X_h - \bar{X})^2}{\sum_{i=1}^n (X_i - \bar{X})^2})
\end{aligned}$$

Since $cov(\bar{Y}, b_1) = 0$ due to the iid assumption on ϵ_i

An estimate of this variance is

$$s^2(\hat{Y}_h) = MSE(\frac{1}{n} + \frac{(X_h - \bar{X})^2}{\sum_{i=1}^n (X_i - \bar{X})^2})$$

the sampling distribution for the mean response is

$$\hat{Y}_h \sim N(E(Y_h), var(\hat{Y}_h)) \frac{\hat{Y}_h - E(Y_h)}{s(\hat{Y}_h)} \sim t_{n-2}$$

A $100(1-\alpha)\%$ CI for $E(Y_h)$ is

$$\hat{Y}_h \pm t_{1-\alpha/2; n-2} s(\hat{Y}_h)$$

5.1.1.4.4 Prediction of a new observation Regarding the Mean Response, we are interested in estimating **mean** of the distribution of Y given a certain X.

Now, we want to **predict** an individual outcome for the distribution of Y at a given X. We call Y_{pred}

Estimation of mean response versus prediction of a new observation:

- the point estimates are the same in both cases: $\hat{Y}_{pred} = \hat{Y}_h$
- It is the variance of the prediction that is different; hence, prediction intervals are different than confidence intervals. The prediction variance must consider:
  - Variation in the mean of the distribution of Y
  - variation within the distribution of Y

We want to predict: mean response + error

$$\beta_0 + \beta_1 X_h + \epsilon$$

Since $E(\epsilon) = 0$, use the least squares predictor:

$$\hat{Y}_h = b_0 + b_1 X_h$$

The variance of the predictor is

$$\begin{aligned} var(b_0 + b_1 X_h + \epsilon) &= var(b_0 + b_1 X_h) + var(\epsilon) \\ &= \sigma^2 (\frac{1}{n} + \frac{(X_h - \bar{X})^2}{\sum_{i=1}^n (X_i - \bar{X})^2}) + \sigma^2 \\ &= \sigma^2 (1 + \frac{1}{n} + \frac{(X_h - \bar{X})^2}{\sum_{i=1}^n (X_i - \bar{X})^2}) \end{aligned}$$

An estimate of the variance is given by

$$s^2(pred) = MSE(1 + \frac{1}{n} + \frac{(X_h - \bar{X})^2}{\sum_{i=1}^n (X_i - \bar{X})^2}) \frac{Y_{pred} - \hat{Y}_h}{s(pred)} \sim t_{n-2}$$

$100(1-\alpha)\%$ prediction interval is

$$\bar{Y}_h \pm t_{1-\alpha/2; n-2} s(pred)$$

The prediction interval is very sensitive to the distributional assumption on the errors, ϵ

5.1.1.4.5 Confidence Band

We want to know the confidence interval for the entire regression line, so we can draw conclusions about any and all mean response fo the entire regression line $E(Y) = \beta_0 + \beta_1 X$ rather than for a given response Y

Working-Hotelling Confidence Band

For a given X_h, this band is

$$\hat{Y}_h \pm W s(\hat{Y}_h)$$

where $W^2 = 2F_{1-\alpha;2,n-2}$, which is just 2 times the F-stat with 2 and n-2 degrees of freedom

- the interval width will change with each X_h (since $s(\hat{Y}_h)$ changes)
- the boundary values for this confidence band will always define a hyperbole containing the regression line
- will be smallest at $X = \bar{X}$

5.1.1.5 ANOVA

Partitioning the Total Sum of Squares: Consider the corrected Total sum of squres:

$$SSTO = \sum_{i=1}^{n} (Y_i - \bar{Y})^2$$

Measures the overall dispersion in the response variable
We use the term corrected because we correct for mean, the uncorrected total sum of squares is given by $\sum Y_i^2$

use $\hat{Y}_i = b_0 + b_1 X_i$ to estimate the conditional mean for Y at X_i

$$\begin{aligned}
\sum_{i=1}^{n} (Y_i - \bar{Y})^2 &= \sum_{i=1}^{n} (Y_i - \hat{Y}_i + \hat{Y}_i - \bar{Y})^2 \\
&= \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 + \sum_{i=1}^{n} (\hat{Y}_i - \bar{Y})^2 + 2 \sum_{i=1}^{n} (Y_i - \hat{Y}_i)(\hat{Y}_i - \bar{Y}) \\
&= \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 + \sum_{i=1}^{n} (\bar{Y}_i - \bar{Y})^2 \\
STTO &= SSE + SSR
\end{aligned}$$

where SSR is the regression sum of squares, which measures how the conditional mean varies about a central value.

The cross-product term in the decomposition is 0:

$$\begin{aligned}
\sum_{i=1}^{n}(Y_i-\hat{Y}_i)(\hat{Y}_i-\bar{Y}) &= \sum_{i=1}^{n}(Y_i-\bar{Y}-b_1(X_i-\bar{X}))(\bar{Y}+b_1(X_i-\bar{X})-\bar{Y}) \\
&= b_1\sum_{i=1}^{n}(Y_i-\bar{Y})(X_i-\bar{X})-b_1^2\sum_{i=1}^{n}(X_i-\bar{X})^2 \\
&= b_1\frac{\sum_{i=1}^{n}(Y_i-\bar{Y})(X_i-\bar{X})}{\sum_{i=1}^{n}(X_i-\bar{X})^2}\sum_{i=1}^{n}(X_i-\bar{X})^2-b_1^2\sum_{i=1}^{n}(X_i-\bar{X})^2 \\
&= b_1^2\sum_{i=1}^{n}(X_i-\bar{X})^2-b_1^2\sum_{i=1}^{n}(X_i-\bar{X})^2 \\
&= 0
\end{aligned}$$

$$SSTO = SSR + SSE$$
$$(n-1d.f) = (1d.f.) + (n-2d.f.)$$

Source of Variation	Sum of Squares	df	Mean Square	F
Regression (model)	SSR	1	MSR = SSR/df	MSR/MSE
Error	SSE	n-2	MSE = SSE/df	
Total (Corrected)	SSTO	n-1		

$$E(MSE) = \sigma^2 E(MSR) = \sigma^2 + \beta_1^2\sum_{i=1}^{n}(X_i-\bar{X})^2$$

- If $\beta_1 = 0$, then these two expected values are the same
- if $\beta_1 \neq 0$ then E(MSR) will be larger than E(MSE)

which means the ratio of these two quantities, we can infer something about β_1

Distribution theory tells us that if $\epsilon_i \sim iidN(0,\sigma^2)$ and assuming $H_0 : \beta_1 = 0$ is true,

$$\frac{MSE}{\sigma^2} \sim \chi^2_{n-2}\frac{MSR}{\sigma^2} \sim \chi^2_1 \text{ if } \beta_1 = 0$$

where these two chi-square random variables are independent.

Since the ratio of 2 independent chi-square random variable follows an F distribution, we consider:

$$F = \frac{MSR}{MSE} \sim F_{1,n-2}$$

when $\beta_1 = 0$. Thus, we reject $H_0 : \beta_1 = 0$ (or $E(Y_i) = \text{constant}$) at α if

$$F > F_{1-\alpha;1,n-2}$$

this is the only null hypothesis that can be tested with this approach.

Coefficient of Determination

$$R^2 = \frac{SSR}{SSTO} = 1 - \frac{SSE}{SSTO}$$

where $0 \leq R^2 \leq 1$

Interpretation: The proportionate reduction of the total variation in Y after fitting a linear model in X.
It is not really correct to say that R^2 is the "variation in Y explained by X".

R^2 is related to the correlation coefficient between Y and X:

$$R^2 = (r)^2$$

where $r = corr(x, y)$ is an estimate of the Pearson correlation coefficient. Also, note

$$b_1 = (\frac{\sum_{i=1}^n (Y_i - \bar{Y})^2}{\sum_{i=1}^n (X_i - \bar{X})^2})^{1/2} r = \frac{s_y}{s_x} r$$

Lack of Fit

$Y_{11}, Y_{21}, ..., Y_{n_1,1}$: n_1 repeat obs at X_1

$Y_{1c}, Y_{2c}, ..., Y_{n_c,c}$: n_c repeat obs at X_c

So, there are c distinct X values.

Let $\bar{Y}_j$ be the mean over replicates for X_j

Partition the Error Sum of Squares:

$$\begin{aligned}
\sum_i \sum_j (Y_{ij} - \hat{Y}_{ij})^2 &= \sum_i \sum_j (Y_{ij} - \bar{Y}_j + \bar{Y}_j + \hat{Y}_{ij})^2 \\
&= \sum_i \sum_j (Y_{ij} - \bar{Y}_j)^2 + \sum_i \sum_j (\bar{Y}_j - \hat{Y}_{ij})^2 + \text{cross product term} \\
&= \sum_i \sum_j (Y_{ij} - \bar{Y}_j)^2 + \sum_j n_j (\bar{Y}_j - \hat{Y}_{ij})^2 \\
SSE &= SSPE + SSLF
\end{aligned}$$

- SSPE: "pure error sum of squares" has n-c degrees of freedom since we need to estimate c means

- SSLF: "lack of fit sum of squares" has c - 2 degrees of freedom (the number of unique X values - number of parameters used to specify the conditional mean regression model)

$$MSPE = \frac{SSPE}{df_{pe}} = \frac{SSPE}{n-c} MSLF = \frac{SSLF}{df_{lf}} = \frac{SSLF}{c-2}$$

The **F-test for Lack-of-Fit** tests

$$H_0 : Y_{ij} = \beta_0 + \beta_1 X_i + \epsilon_{ij}, \epsilon_{ij} \sim iidN(0,\sigma^2) H_a : Y_{ij} = \alpha_0 + \alpha_1 X_i + f(X_i, Z_1, ...) + \epsilon^*_{ij}, \epsilon^*_{ij} \sim iidN(0,\sigma^2)$$

$E(MSPE) = \sigma^2$ under either H_0, H_a

$E(MSLF) = \sigma^2 + \frac{\sum n_j (f(X_i, ...))^2}{n-2}$ in general and

$E(MSLF) = \sigma^2$ when H_0 is true

We reject H_0 (i.e., the model $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$ is not adequate) if

$$F = \frac{MSLF}{MSPE} > F_{1-\alpha; c-2, n-c}$$

Failing to reject H_0 does not imply that $H_0 : Y_{ij} = \beta_0 + \beta_1 X_i + \epsilon_{ij}$ is exactly true, but it suggests that this model may provide a reasonable approximation to the true model.

Source of Variation	Sum of Squares	df	Mean Square	F
Regression	SSR	1	MSR	MSR / MSE
Error	SSE	n-2	MSE	
Lack of fit	SSLF	c-2	MSLF	MSLF / MSPE
Pure Error	SSPE	n-c	MSPE	
Total(Corrected)	SSTO	n-1		

Repeat observations have an effect on R^2:

- It is impossible for R^2 to attain 1 when repeat obs. exist (SSE can't be 0)

- The maximum R^2 attainable in this situation:

$$R^2_{max} = \frac{SSTo - SSPE}{SSTO}$$

- Not all levels of X need have repeat observations.

- Typically, when H_0 is appropriate, one still uses MSE as the estimate for σ^2 rather than MSPE, Since MSE has more degrees of freedom, sometimes people will pool these estimates.

Joint Inference
The confidence coefficient for both β_0 and β_1 considered simultaneously is $\leq \alpha$

Let

- $\bar{A}_1$ be the event that the first interval covers β_0

- $\bar{A}_2$ be the event that teh second interval covers β_1

$$P(\bar{A}_1) = 1 - \alpha P(\bar{A}_2) = 1 - \alpha$$

The probability that both $\bar{A}_1$ and $\bar{A}_2$

$$\begin{aligned} P(\bar{A}_1 \cap \bar{A}_2) &= 1 - P(\bar{A}_1 \cup \bar{A}_2) \\ &= 1 - P(A_1) - P(A_2) + P(A_1 \cap A_2) \\ &\geq 1 - P(A_1) - P(A_2) \\ &= 1 - 2\alpha \end{aligned}$$

If β_0 and β_1 have separate 95% confidence intervals, the joint (family) confidence coefficient is at least $1-2(0.05) = 0.9$. This is called a **Bonferroni Inequality** We could use a procedure in which we obtained $1 - \alpha/2$ confidence intervals for the two regression parameters separately, then the joint (Bonferroni) family confidence coefficient would be at least $1 - \alpha$

The $1 - \alpha$ joint Bonferroni confidence interval for β_0 and β_1 is given by calculating:

$$b_0 \pm Bs(b_0) b_1 \pm Bs(b_1)$$

where $B = t_{1-\alpha/4;n-2}$

Interpretation: If repeated samples were taken and the joint $(1 - \alpha)$ intervals for β_0 and β_1 were obtained, $(1 - \alpha)100\%$ of the joint intervals would contain the true pair (β_0, β_1). That is, in $\alpha \times 100\%$ of the samples, one or both intervals would not contain the true value.

- The Bonferroni interval is **conservative**. It is a lower bound and the joint intervals will tend to be correct more than $(1 - \alpha)100\%$ of the time (lower power). People usually consider a larger α for the Bonferroni joint tests (e.g, $\alpha = 0.1$)

- The Bonferroni procedure extends to testing more than 2 parameters. Say we are interested in testing $\beta_0, \beta_1, ..., \beta_{g-1}$ (g parameters to test). Then, the joint Bonferroni interval is obtained by calculating the $(1 - \alpha/g)$ 100% level interval for each separately.

- For example, if $\alpha = 0.05$ and $g = 10$, each individual test is done at the $1 - \frac{.05}{10}$ level. For 2-sided intervals, this corresponds to using $t_{1-\frac{0.05}{2(10)};n-p}$ in the CI formula. This procedure works best if g is relatively small, otherwise the intervals for each individual parameter are very wide and teh test is way too conservative.

- b_0, b_1 are usually correlated (negatively if $\bar{X} > 0$ and positively if $\bar{X} < 0$)

- Other multiple comparison procedures are available.

5.1.1.6 Assumptions

- Linearity of the regression function

- Error terms have constant variance

- Error terms are independent

- No outliers

- Error terms are normally distributed

- No Omitted variables

5.1.1.7 Diagnostics

Constant Variance
Plot residuals vs. X

Outliers
plot residuals vs. X
box plots
stem-leaf plots
scatter plots

we could use standardize the residuals to have unit variance. These standardized residuals are called studentized residuals:

$$r_i = \frac{e_i - \bar{e}}{s(e_i)} = \frac{e_i}{s(e_i)}$$

A simplified standardization procedure gives semi-studentized residuals:

$$e_i^* = \frac{e_i - \bar{e}}{\sqrt{MSE}} = \frac{e_i}{\sqrt{MSE}}$$

Non-independent of Error Terms
plot residuals vs. time

Residuals e_i are not independent random variables because they involve the fitted values $\hat{Y}_i$, which are based on the same fitted regression function.

If the sample size is large, the dependency among e_i is relatively unimportant.

To detect non-independence, it helps to plot the residual for the i-th response vs. the (i-1)-th

Non-normality of Error Terms

to detect non-normality (distribution plots of residuals, box plots of residuals, stem-leaf plots of residuals, normal probability plots of residuals)

- Need relatively large sample sizes.

- Other types of departure affect the distribution of the residuals (wrong regression function, non-constant error variance,…)

5.1.1.7.1 Objective Tests of Model Assumptions

- Normality
  - Use Methods based on empirical cumulative distribution function to test on residuals.

- Constancy of error variance
  - Brown-Forsythe Test (Modified Levene Test)

  - Breusch-Pagan Test (Cook-Weisberg Test)

5.1.1.8 Remeidal Measures

If the simple linear regression is not appropriate, one can:

- more complicated models

- transformations on X and/or Y (may not be “optimal” results)

Remedial measures based on deviations:

- Non-linearity:
  - Transformations

  - more complicated models

- Non-constant error variance:
  - Weighted Least Squares
  - Transformations
- Correlated errors:
  - serially correlated error models (times series)
- Non-normality:
  -
- Additional variables: multiple regression.
- Outliers:
  - Robust estimation.

5.1.1.8.1 Transformations use transformations of one or both variables before performing the regression analysis.
The properties of least-squares estimates apply to the transformed regression, not the original variable.

If we transform the Y variable and perform regression to get:

$$g(Y_i) = b_0 + b_1 X_i$$

Transform back:

$$\hat{Y}_i = g^{-1}(b_0 + b_1 X_i)$$

$\hat{Y}_i$ will be biased. we can correct this bias.

Box-Cox Family Transformations

$$Y' = Y^{\lambda}$$

where λ is a parameter to be determined from the data.

λ	Y'
2	Y^2
0.5	$\sqrt{Y}$
0	$ln(Y)$
-0.5	$1/\sqrt{Y}$
-1	$1/Y$

To pick λ, we can do estimation by:

- trial and error
- maximum likelihood
- numerical search

Variance Stabilizing Transformations

A general method for finding a variance stabilizing transformation, when the standard deviation is a function of the mean, is the **delta method** - an application of a Taylor series expansion.

$$\sigma = \sqrt{var(Y)} = f(\mu)$$

where $\mu = E(Y)$ and $f(\mu)$ is some smooth function of the mean.

COnsider the transformation h(Y). Expand this function in a Taylor series about μ. Then,

$$h(Y) = h(\mu) + h'(\mu)(Y - \mu) + \text{small terms}$$

we want to select the function h(.) so that the variance of h(Y) is nearly constant for all values of $\mu = E(Y)$:

$$\begin{aligned} const &= var(h(Y)) \\ &= var(h(\mu) + h'(\mu)(Y - \mu)) \\ &= (h'(\mu))^2 var(Y - \mu) \\ &= (h'(\mu))^2 var(Y) \\ &= (h'(\mu))^2 (f(\mu))^2 \end{aligned}$$

we must have,

$$h'(\mu) \propto \frac{1}{f(\mu)}$$

then,

$$h(\mu) = \int \frac{1}{f(\mu)} d\mu$$

Example: For the Poisson distribution: $\sigma^2 = var(Y) = E(Y) = \mu$

Then,

$$\sigma = f(\mu) = \sqrt{mu} h'(\mu) \propto \frac{1}{\mu} = \mu^{-.5}$$

Then, the variance stabilizing transformation is:

$$h(\mu) = \int \mu^{-.5} d\mu = \frac{1}{2}\sqrt{\mu}$$

hence, $\sqrt{Y}$ is used as the variance stabilizing transformation.

If we don't know $f(\mu)$

1. Trial and error. Look at residuals plots

2. Ask researchers about previous studies or find published results on similar experiments and determine what transformation was used.

3. If you have multiple observations Y_{ij} at the same X values, compute $\bar{Y}_i$ and s_i and plot them
   If $s_i \propto \bar{Y}_i^\lambda$ then consider $s_i = a\bar{Y}_i^\lambda$ or $ln(s_i) = ln(a) + \lambda ln(\bar{Y}_i)$. So regression the natural log of s_i on the natural log of $\bar{Y}_i$ gives $\hat{a}$ and $\hat{\lambda}$ and suggests the form of $f(\mu)$ If we don't have multiple obs, might still be able to "group" the observations to get $\bar{Y}_i$ and s_i.

Transformation	Situation	Comments
$\sqrt{Y}$	$var(\epsilon_i) = kE(Y_i)$	counts from Poisson dist
$\sqrt{Y} + \sqrt{Y+1}$	$var(\epsilon_i) = kE(Y_i)$	small counts or zeroes
$log(Y)$	$var(\epsilon_i) = k(E(Y_i))^2$	positive integers with wide range
$log(Y+1)$	$var(\epsilon_i) = k(E(Y_i))^2$	some counts zero
1/Y	$var(\epsilon_i) = k(E(Y_i))^4$	most responses near zero, others large
$arcsin(\sqrt{Y})$	$var(\epsilon_i) = kE(Y_i)(1 - E(Y_i))$	data are binomial proportions or %

5.1.2 Multiple Linear Regression

Geometry of Least Squares

$$\begin{aligned}\hat{\mathbf{y}} &= \mathbf{Xb}\\ &= \mathbf{X}(\mathbf{X}'\mathbf{X})^{-\mathbf{1}}\mathbf{X}'\mathbf{y}\\ &= \mathbf{Hy}\end{aligned}$$

sometimes H is denoted as P.

H is the projection operator.

$$\hat{\mathbf{y}} = \mathbf{Hy}$$

is the projection of y onto the linear space spanned by the columns of X (model space). The dimension of the model space is the rank of X.

Facts:

1. H is symmetric (i.e., H = H')

2. HH = H

$$\begin{aligned}\mathbf{HH} &= \mathbf{X}(\mathbf{X}'\mathbf{X})^{-\mathbf{1}}\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-\mathbf{1}}\mathbf{X}' \\ &= \mathbf{X}(\mathbf{X}'\mathbf{X})^{-\mathbf{1}}\mathbf{I}\mathbf{X}' \\ &= \mathbf{X}(\mathbf{X}'\mathbf{X})^{-\mathbf{1}}\mathbf{X}'\end{aligned}$$

3. H is an n x n matrix with rank(H) = rank(X)

4. $(\mathbf{I}-\mathbf{H}) = \mathbf{I}-\mathbf{X}(\mathbf{X}'\mathbf{X})^{-\mathbf{1}}\mathbf{X}'$ is also a projection operator. It projects onto the n - k dimensional space that is orthogonal to the k dimensional space spanned by the columns of X
5. $\mathbf{H}(\mathbf{I}-\mathbf{H}) = (\mathbf{I}-\mathbf{H})\mathbf{H} = \mathbf{0}$

Partition of uncorrected total sum of squares:

$$\begin{aligned}\mathbf{y}'\mathbf{y} &= \hat{\mathbf{y}}'\hat{\mathbf{y}} + \mathbf{e}'\mathbf{e} \\ &= (\mathbf{Hy})'(\mathbf{Hy}) + ((\mathbf{I}-\mathbf{H})\mathbf{y})'((\mathbf{I}-\mathbf{H})\mathbf{y}) \\ &= \mathbf{y}'\mathbf{H}'\mathbf{Hy} + \mathbf{y}'(\mathbf{I}-\mathbf{H})'(\mathbf{I}-\mathbf{H})\mathbf{y} \\ &= \mathbf{y}'\mathbf{Hy} + \mathbf{y}'(\mathbf{I}-\mathbf{H})\mathbf{y}\end{aligned}$$

or partition for the corrected total sum of squares:

$$\mathbf{y}'(\mathbf{I}-\mathbf{H_1})\mathbf{y} = \mathbf{y}'(\mathbf{H}-\mathbf{H_1})\mathbf{y} + \mathbf{y}'(\mathbf{I}-\mathbf{H})\mathbf{y}$$

where $H_1 = \frac{1}{n}J = 1'(1'1)1$

Source	SS	df	MS	F
Regression	$SSR = \mathbf{y}'(\mathbf{H}-\frac{\mathbf{1}}{\mathbf{n}}\mathbf{J})\mathbf{y}$	p - 1	SSR/(p-1)	MSR /MSE
Error	$SSE = \mathbf{y}'(\mathbf{I}-\mathbf{H})\mathbf{y}$	n - p	SSE /(n-p)	
Total	$\mathbf{y}'\mathbf{y}-\mathbf{y}'\mathbf{Jy}/\mathbf{n}$	n -1		

Equivalently, we can express

$$\mathbf{Y} = \mathbf{X}\hat{\ } + (\mathbf{Y}-\mathbf{X}\hat{\ })$$

where

- $\hat{\mathbf{Y}} = \mathbf{X}\hat{}$ = sum of a vector of fitted values

- $\mathbf{e} = (\mathbf{Y} - \mathbf{X}\hat{}\ )$ = residual
- $\mathbf{Y}$ is the n x 1 vector in a n-dimensional space R^n

- $\mathbf{X}$ is an n x p full rank matrix. and its columns generate a p-dimensional subspace of R^n. Hence, any estimator $\mathbf{X}\hat{}$ is also in this subspace.

We choose least squares estimator that minimize the distance between $\mathbf{Y}$ and $\mathbf{X}\hat{}$, which is the **orthogonal projection** of $\mathbf{Y}$ onto $\mathbf{X}$.

$$\begin{aligned}
||\mathbf{Y} - \mathbf{X}\hat{\beta}||^2 &= ||\mathbf{Y} - \mathbf{X}\hat{}\ ||^2 + ||\mathbf{X}\hat{}\ ||^2 \\
&= (\mathbf{Y} - \mathbf{X}\hat{}\ )'(\mathbf{Y} - \mathbf{X}\hat{}\ ) + (\mathbf{X}\hat{}\ )'(\mathbf{X}\hat{}\ ) \\
&= (\mathbf{Y} - \mathbf{X}\hat{}\ )'\mathbf{Y} - (\mathbf{Y} - \mathbf{X}\hat{}\ )'\mathbf{X}\hat{}\ +\hat{}\ '\mathbf{X}'\mathbf{X}\hat{} \\
&= (\mathbf{Y} - \mathbf{X}\hat{}\ )'\mathbf{Y} +\hat{}\ '\mathbf{X}'\mathbf{X}(\mathbf{X}\mathbf{X}')^{-\mathbf{1}}\mathbf{X}'\mathbf{Y} \\
&= \mathbf{Y}'\mathbf{Y} -\hat{}\ '\mathbf{X}'\mathbf{Y} +\hat{}\ '\mathbf{X}'\mathbf{Y}
\end{aligned}$$

where the norm of a (p x 1) vector $\mathbf{a}$ is defined by:

$$||\mathbf{a}|| = \sqrt{\mathbf{a'a}} = \sqrt{\sum_{i=1}^{p} a_i^2}$$

Coefficient of Multiple Determination

$$R^2 = \frac{SSR}{SSTO} = 1 - \frac{SSE}{SSTO}$$

Adjusted Coefficient of Multiple Determination

$$R_a^2 = 1 - \frac{SSE/(n-p)}{SSTO/(n-1)} = 1 - \frac{(n-1)SSE}{(n-p)SSTO}$$

Sequential and Partial Sums of Squares:

In a regression model with coefficients $\beta = (\beta_0, \beta_1, ..., \beta_{p-1})'$, we denote the uncorrected and corrected SS by

$$SSM = SS(\beta_0, \beta_1, ..., \beta_{p-1})SSM_m = SS(\beta_0, \beta_1, ..., \beta_{p-1}|\beta_0)$$

There are 2 decompositions of SSM_m:

- **Sequential SS**: (not unique -depends on order, also referred to as Type I SS, and is the default of anova() in R)

$$SSM_m = SS(\beta_1|\beta_0) + SS(\beta_2|\beta_0, \beta_1) + ... + SS(\beta_{p-1}|\beta_0, ..., \beta_{p-2})$$

- **Partial SS: (use more in practice - contribution of each given all of the others)

$$SSM_m = SS(\beta_1|\beta_0, \beta_2, ..., \beta_{p-1}) + ... + SS(\beta_{p-1}|\beta_0, \beta_1, ..., \beta_{p-2})$$

5.1.3 OLS Assumptions

- A1 Linearity
- A2 Full rank
- A3 Exogeneity of Independent Variables
- A4 Homoskedasticity
- A5 Data Generation (random Sampling)
- A6 Normal Distribution

5.1.3.1 A1 Linearity

$$A1 : y = \mathbf{x}\beta + \epsilon \tag{5.1}$$

Not restrictive

- x can be nonlinear transformation including interactions, natural log, quadratic

With A3 (Exogeneity of Independent), linearity can be restrictive

5.1.3.1.1 Log Model

Model	Form	Interpretation of β	In words
Level-Level	$y = \beta_0 + \beta_1 x + \epsilon$	$\Delta y = \beta_1 \Delta x$	A unit change in x will result in β_1 unit change in y
Log-Level	$ln(y) = \beta_0 + \beta_1 x + epsilon$	$\%\Delta y = 100\beta_1 \Delta x$	A unit change in x result in 100 β_1 % change in y
Level-Log	$y = beta_0 + \beta_1 ln(x) + \epsilon$	$\Delta y = (\beta_1/100)\%\Delta x$	One percent change in x result in $\beta_1/100$ units change in y
Log-Log	$ln(y) = \beta_0 + \beta_1 ln(x) + \epsilon$	$\%\Delta y = \beta_1 \%\Delta x$	One percent change in x result in β_1 percent change in y

5.1.3.1.2 Higher Orders $y = \beta_0 + x_1\beta_1 + x_1^2\beta_2 + \epsilon$

$$\frac{\partial y}{\partial x_1} = \beta_1 + 2x_1\beta_2$$

- The effect of x_1 on y depends on the level of x_1
- The partial effect at the average = $\beta_1 + 2E(x_1)\beta_2$
- Average Partial Effect = $E(\beta_1 + 2x_1\beta_2)$

5.1.3.1.3 Interactions $y = \beta_0 + x_1\beta_1 + x_2\beta_2 + x_1x_2\beta_3 + \epsilon$

- β_1 is the average effect on y for a unit change in x_1 when $x_2 = 0$
- $\beta_1 + x_2\beta_3$ is the partial effect of x_1 on y which depends on the level of x_2

5.1.3.2 A2 Full rank

$$A2 : rank(E(x'x)) = k \tag{5.2}$$

also known as **identification condition**

- columns of **x** cannot be written as a linear function of the other columns
- which ensures that each parameter is unique and exists in the population regression equation

5.1.3.3 A3 Exogeneity of Independent Variables

$$A3 : E[\epsilon|x_1, x_2, ..., x_k] = E[\epsilon|\mathbf{x}] = 0 \tag{5.3}$$

strict exogeneity

- also known as **mean independence** check back on Correlation and Independence
- by the Law of Iterated Expectations $E(\epsilon) = 0$, which can be satisfied by always including an intercept.
- independent variables do not carry information for prediction of ϵ
- A3 implies $E(y|x) = x\beta$, which means the conditional mean function must be a linear function of x A1 Linearity

5.1.3.3.1 A3a Weaker Exogeneity Assumption

Exogeneity of Independent variables

A3a: $E(\mathbf{x_i'}\epsilon_i) = 0$

- x_i is **uncorrelated** with ϵ_i Correlation and Independence
- Weaker than **mean independence** A3
  - A3 implies A3a, not the reverse
  - No causality interpretations
  - Cannot test the difference

5.1.3.4 A4 Homoskedasticity

$$A4 : Var(\epsilon|x) = Var(\epsilon) = \sigma^2 \tag{5.4}$$

* Variation in the disturbance to be the same over the independent variables

5.1.3.5 A5 Data Generation (random Sampling)

$$A5 : y_i, x_{i1}, ..., x_{ik-1} : i = 1, ..., n \tag{5.5}$$

is a random sample

- random sample mean samples are independent and identically distributed (iid) from a joint distribution of $(y, \mathbf{x})$
- with A3 and A4, we have
  - **Strict Exogeneity**: $E(\epsilon_i|x_1, ..., x_n) = 0$. independent variables do not carry information for prediction of ϵ
  - **Non-autocorrelation**: $E(\epsilon_i \epsilon_j|x_1, ..., x_n) = 0$ The error term is uncorrelated across the draws conditional on the independent variables $\rightarrow A4 : Var(\epsilon|\mathbf{X}) = Var(\epsilon) = \sigma^2 I_n$
- In times series and spatial settings, A5 is less likely to hold.

5.1.3.5.1 A5a A stochastic process $\{x_t\}_{t=1}^T$ is **stationary** if for every collection fo time indices $\{t_1, t_2, ..., t_m\}$, the joint distribution of

$$x_{t_1}, x_{t_2}, ..., x_{t_m}$$

is the same as the joint distribution of

$$x_{t_1+h}, x_{t_2+h}, ..., x_{t_m+h}$$

for any $h \geq 1$

- The joint distribution for the first ten observation is the same for the next ten, etc.
- Independent draws automatically satisfies this

A stochastic process $\{x_t\}_{t=1}^T$ is **weakly stationary** if x_t and x_{t+h} are "almost independent" as h increases without bounds.
* two observation that are very far apart should be "almost independent"

Common Weakly Dependent Processes

1. Moving Average process of order 1 (MA(1))

MA(1) means that there is only one period lag.

$$y_t = u_t + \alpha_1 u_{t-1} E(y_t) = E(u_t) + \alpha_1 E(u_{t-1}) = 0 Var(y_t) = var(u_t) + \alpha_1 var(u_{t-1}) = \sigma^2 + \alpha_1^2 \sigma^2 = \sigma^2(1 + \alpha_1^2)$$

where u_t is drawn iid over t with variance σ^2

An increase in the absolute value of α_1 increases the variance

When the MA(1) process can be **inverted** ($|\alpha| < 1$ then

$$u_t = y_t - \alpha_1 u_{t-1}$$

called the autoregressive representation (express current observation in term of past observation).

We can expand it to more than 1 lag, then we have MA(q) process

$$y_t = u_t + \alpha_1 u_{t-1} + ... + \alpha_q u_{t-q}$$

where $u_t \sim WN(0, \sigma^2)$

- Covariance stationary: irrespective of the value of the parameters.
- Invertibility when $\alpha < 1$
- The conditional mean of MA(q) depends on the q lags (long-term memory).
- In MA(q), all autorcorrealtions beyond q are 0.

$$\begin{aligned} Cov(y_t, y_{t-1}) &= Cov(u_t + \alpha_1 u_{t-1}, u_{t-1} + \alpha_1 u_{t-2}) \\ &= \alpha_1 var(u_{t-1}) \\ &= \alpha_1 \sigma^2 \end{aligned}$$

$$\begin{aligned} Cov(y_t, y_{t-2}) &= Cov(u_t + \alpha_1 u_{t-1}, u_{t-2} + \alpha_1 u_{t-3}) \\ &= 0 \end{aligned}$$

An MA models a linear relationship between teh dependent variable and teh current and past values of a stochastic term.

2. Auto regressive process of order 1 (AR(1))

$$y_t = \rho y_{t-1} + u_t, |\rho| < 1$$

where u_t is drawn iid over t with variance σ^2

$$\begin{aligned} Cov(y_t, y_{t-1}) &= Cov(\rho y_{t-1} + u - t, y_{t-1}) \\ &= \rho Var(y_{t-1}) \\ &= \rho \frac{\sigma^2}{1-\rho^2} \end{aligned}$$

$$Cov(y_t, y_{t-h}) = \rho^h \frac{\sigma^2}{1-\rho^2}$$

Stationarity: in the continuum of t, the distribution of each t is the same

$$E(y_t) = E(y_{t-1}) = ... = E(y_0) y_1 = \rho y_0 + u_1$$

where the initial observation $y_0 = 0$

Assume $E(y_t) = 0$

$$y_t = \rho^t y_{t-t} + \rho^{t-1} u_1 + \rho^{t-2} u_2 + ... + \rho u_{t-1} + u_t = \rho^t y_0 + \rho^{t-1} u_1 + \rho^{t-2} u_2 + ... + \rho u_{t-1} + u_t$$

Hence, y_t is the weighted of all of the u_t time observations before. y will be correlated with all the previous observations as well as future observations.

$$Var(y_t) = Var(\rho y_{t-1} + u_t) = \rho^2 Var(y_{t-1}) + Var(u_t) + 2\rho Cov(y_{t-1} u_t) = \rho^2 Var(y_{t-1}) + \sigma^2$$

Hence,

$$Var(y_t) = \frac{\sigma^2}{1 - \rho^2}$$

to have Variance constantly over time, then $\rho \neq 1$ or -1.

Then stationarity requires $\rho \neq 1$ or -1. weakly dependent process $|\rho| < 1$

To estimate the AR(1) process, we use **Yule-Walker Equation**

$$y_t = \epsilon_t + \phi y_{t-1} y_t y_{t-\tau} = \epsilon_t y_{t-\tau} + \phi y_{t-1} y_{t-\tau}$$

For $\tau \geq 1$, we have

$$\gamma\tau = \phi\gamma(\tau - 1)\rho_t = \phi^t$$

when you generalize to pth order autoregressive process, AR(p):

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + \epsilon_t$$

AR(p) process is **covariance stationary**, and decay in autocorrelations.

When we combine MA(q) and AR(p), we have ARMA(p,q) process, where you can see seasonality. For example, ARMA(1,1)

$$y_t = \phi y_{t-1} + \epsilon_t + \alpha\epsilon_{t-1}$$

Random Walk process

$$y_t = y_0 + \sum_{s=1}^{t} u_t$$

- not stationary : when $y_0 = 0$ then $E(y_t) = 0$, but $Var(y_t) = t\sigma^2$. Further along in the spectrum, the variance will be larger
- not weakly dependent: $Cov(\sum_{s=1}^{t} u_s, \sum_{s=1}^{t-h} u_s) = (t-h)\sigma^2$. So the covariance (fixed) is not diminishing as h increases

$$Assumption A5a : \{y_t, x_{t1}, .., x_{tk-1}\}$$

where $t = 1, ..., T$ are **stationary and weakly dependent processes**.

Alternative Weak Law, Central Limit Theorem
If z_t is a weakly dependent stationary process with a finite first absolute moment and $E(z_t) = \mu$, then

$$T^{-1} \sum_{t=1}^{T} z_t \to^p \mu$$

If additional regulatory conditions hold (Greene, 1990), then

$$\sqrt{T}(\bar{z} - \mu) \to^d N(0, B)$$

where $B = Var(z_t) + 2\sum_{h=1}^{\infty} Cov(z_t, z_{t-h})$

5.1.3.6 A6 Normal Distribution

$$A6 : \epsilon|\mathbf{x} \sim N(0, \sigma^2 I_n) \tag{5.6}$$

The error term is normally distributed

From A1-A3, we have **identification** (also known as **Orthogonality Condition**) of the population parameter β

$$\begin{aligned}
y &= x\beta + \epsilon && \text{A1} \\
x'y &= x'x\beta + x'\epsilon \\
E(x'y) &= E(x'x)\beta + E(x'\epsilon) \\
E(x'y) &= E(x'x)\beta && \text{A3} \\
[E(x'x)]^{-1}E(x'y) &= [E(x'x)]^{-1}E(x'x)\beta && \text{A2} \\
[E(x'x)]^{-1}E(x'y) &= \beta
\end{aligned}$$

β is the row vector of parameters that produces the best predictor of y we choose the min of γ :

$$\underset{\gamma}{\text{argmin}} E((y - x\gamma)^2)$$

First Order Condition

$$\begin{aligned}
\frac{\partial((y - x\gamma)^2)}{\partial\gamma} &= 0 \\
-2E(x'(y - x\gamma)) &= 0 \\
E(x'y) - E(x'x\gamma) &= 0 \\
E(x'y) &= E(x'x)\gamma \\
(E(x'x))^{-1}E(x'y) &= \gamma
\end{aligned}$$

Second Order Condition

$$\begin{aligned}
\frac{\partial^2 E((y - x\gamma)^2)}{} &= 0 \\
E(\frac{\partial(y - x\partial)^2)}{\partial\gamma\partial\gamma'}) &= 2E(x'x)
\end{aligned}$$

If A3 holds, then $2E(x'x)$ is PSD → minimum

5.1.4 Theorems

5.1.4.1 Frisch-Waugh-Lovell Theorem

$$\mathbf{y} = \mathbf{X} \ + \ = \mathbf{X_1}_{\mathbf{1}} + \mathbf{X_2}_{\mathbf{2}} +$$

Equivalently,

$$\left(\begin{array}{cc} X_1'X_1 & X_1'X_2 \\ X_2'X_1 & X_2'X_2 \end{array} \right) \left(\begin{array}{c} \hat{\beta}_1 \\ \hat{\beta}_2 \end{array} \right) = \left(\begin{array}{c} X_1'y \\ X_2'y \end{array} \right)$$

Hence,

$$\hat{}_{\mathbf{1}} = (\mathbf{X_1'X_1})^{-\mathbf{1}}\mathbf{X_1'y} - (\mathbf{X_1'X_1})^{-\mathbf{1}}\mathbf{X_1'X_2}\hat{}_{\mathbf{2}}$$

1. Betas from the multiple regression are not the same as the betas from each of the individual simple regression
2. Different set of X will affect all the coefficient estimates.
3. If $X_1'X_2 = 0$ or $\hat{\beta}_2 = 0$, then 1 and 2 do not hold.

5.1.4.2 Gauss-Markov Theorem

For a linear regression model

$$\mathbf{y} = \mathbf{X} \ +$$

Under A1, A2, A3, A4, OLS estimator defined as

$$\hat{\beta} = (\mathbf{X'X})^{-\mathbf{1}}\mathbf{X'y}$$

is the minimum variance linear (in y) unbiased estimator of β

Let $\tilde{\beta} = \mathbf{Cy}$, be another linear estimator where $\mathbf{C}$ is k x n and only function of X), then for it be unbiased,

$$\begin{aligned} E(\tilde{\beta}|\mathbf{X}) &= E(\mathbf{Cy}|\mathbf{X}) \\ &= E(\mathbf{CX} \ + \mathbf{C} \ |\mathbf{X}) \\ &= \mathbf{CX} \end{aligned}$$

which equals the true parameter β only if $\mathbf{CX} = \mathbf{I}$
Equivalently, $\tilde{\beta} = \beta + \mathbf{C}\epsilon$ and the variance of the estimator is $Var(\tilde{\beta}|\mathbf{X}) = \sigma^2\mathbf{CC'}$

To show minimum variance,

$$\begin{aligned} &= \sigma^2(\mathbf{C} - (\mathbf{X'X})^{-\mathbf{1}}\mathbf{X'})(\mathbf{C} - (\mathbf{X'X})^{-\mathbf{1}}\mathbf{X'})' \\ &= \sigma^2(\mathbf{CC'} - \mathbf{CX}(\mathbf{X'X})^{-\mathbf{1}}) - (\mathbf{X'X})^{-\mathbf{1}}\mathbf{X'C} + (\mathbf{X'X})^{-\mathbf{1}}\mathbf{X'X}(\mathbf{X'X})^{-\mathbf{1}}) \\ &= \sigma^2(\mathbf{CC'} - (\mathbf{X'X})^{-\mathbf{1}} - (\mathbf{X'X})^{-\mathbf{1}} + (\mathbf{X'X})^{-\mathbf{1}}) \\ &= \sigma^2\mathbf{CC'} - \sigma^2(\mathbf{X'X})^{-1} \\ &= Var(\tilde{\beta}|\mathbf{X}) - Var(\hat{\beta}|\mathbf{X}) \end{aligned}$$

Hierarchy of OLS Assumptions

Identification Data Description	Unbiasedness Consistency	Gauss-Markov (BLUE) Asymptotic Inference (z and Chi-squared)	Classical LM (BUE) Small-sample Inference (t and F)
Variation in X	Variation in X Random Sampling Linearity in Parameters Zero Conditional Mean	Variation in X Random Sampling Linearity in Parameters Zero Conditional Mean	Variation in X Random Sampling Linearity in Parameters Zero Conditional Mean

Identification Data Description	Unbiasedness Consistency	Gauss-Markov (BLUE) Asymptotic Inference (z and Chi-squared)	Classical LM (BUE) Small-sample Inference (t and F)
		Homoskedasticity	Homoskedasticity Normality of Errors

5.1.5 Variable Selection

depends on

- Objectives or goals
- Previously acquired expertise
- availability of data
- availability of computer software

Let P - 1 be the number of possible X variables

5.1.5.1 Mallows's C_p Statistic

(Mallows, 1973, Technometrics, 15, 661-675)

A measure of the predictive ability of a fitted model
Let $\hat{Y}_{ip}$ be the predicted value of Y_i using the model with p parameters. The total standardized mean square error of prediction is:

$$\Gamma_p = \frac{\sum_{i=1}^n E(\hat{Y}_{ip} - E(Y_i))^2}{\sigma^2} = \frac{\sum_{i=1}^n [E(\hat{Y}_{ip}) - E(Y_i)]^2 + \sum_{i=1}^n var(\hat{Y}_{ip})}{\sigma^2}$$

the first term in the numerator is the (bias)^2 term and the 2nd term is the prediction variance term.

- bias term decreases as more variables are added to the model.
- if we assume the full model (p=P) is the true model, then $E(\hat{Y}_{ip}) - E(Y_i) = 0$ and the bias is 0.
- Prediction variance increase as more variables are added to the model $\sum var(\hat{Y}_{ip}) = p\sigma^2$

- thus, a tradeoff between bias and variance terns is achieved by minimizing Γ_p.

- Since Γ_p is unknown (due to β). we use an estimate: $C_p = \frac{SSE_p}{\hat{\sigma^2}} - (n-2p)$ which is an unbiased estimate of Γ_p

- As more variables are added to the model, the SSE_p decreases but 2p increases. where $\hat{\sigma^2} = MSE(X_1, .., X_{P-1})$ the MSE with all possible X variables in the model.

- when there is no bias then $E(C_p) \approx p$. Thus, good models have C_p close to p.

- Prediction: consider models with $C_p \leq p$

- Parameter estimation: consider models with $C_p \leq 2p - (P-1)$. Fewer variables should be eliminated from the model to avoid excess bias in the estimates.

5.1.5.2 Akaike Information Criterion (AIC)

$$AUC = nln(\frac{SSE_p}{n}) + 2p$$

- increasing p (number of parameters) leads first-term decreases, and second-term increases.
- We want model with small values of AIC. If the AIC increases when a parameter is added to the model, that parameter is not needed.

- AIC represents a tradeoff between precision of fit against the number of parameters used.

5.1.5.3 Bayes (or Schwarz) Information Criterion

$$BIC = n\ln(\frac{SSE_p}{n}) + (\ln n)p$$

The coefficient in front of p tends to penalize more heavily models with a larger number of parameters (as compared to AIC).

5.1.5.4 Prrediciton Error Sum of Squares (PRESS)

$$PRESS_p = \sum_{i=1}^{n} (Y_i - \hat{Y}_{i(i)})^2$$

where $\hat{Y}_{i(i)}$ is the prediction of the i-th response when the i-th observation is not used, obtained for the model with p parameters.

- evaluates the predictive ability of a postulated model by omitting one observation at a time.
- we want small $PRESS_p$ values
- It can be computationally intensive when you have large p.

5.1.5.5 Best Subsets Algorithm

- "leap and bounds" algorithm of (Furnival and Wilson, 1974) combines comparison of SSE for different subset models with control over the sequence in which the subset regression are computed.
- Guarantees finding the best m subset regressions within each subset size with less computational burden than all possible subsets.

```
library("leaps")
regsubsets()
```

5.1.5.6 Stepwise Selection Procedures

The **forward stepwise** procedure:

- finds a plausible subset sequentially.
- at each step, a variable is added or deleted.
- criterion for adding or deleting is based on SSE, R^2, T, or F-statistic.

Note:

- Instead of using exact F-values, computer packages usually specify the equivalent "significance" level. For example, SLE is the "significance" level to enter, and SLS is the "significance" level to stay. The SLE and SLS are guides rather than true tests of significance.
- The choice of SLE and SLS represents a balancing of opposing tendencies. Use of large SLE values tends to result in too many predictor variables; models with small SLE tend to be under-specified resulting in σ^2 being badly overestimated.
- As for choice of SLE, can choose between 0.05 and 0.5.
- If SLE > SLS then a cycling pattern may occur. Although most computer packages can detect can stop when it happens. A quick fix: SLS = SLE /2 (Bendel and Afifi, 1977).

- If SLE < SLS then the procedure is conservative and may lead variables with low contribution to be retained.

- Order of variable entry does not matter.

Automated Selection Procedures:

- Forward selection: Same idea as forward stepwise except it doesn't test if variables should be dropped once enter. (not as good as forward stepwise).

- Backward Elimination: begin with all variables and identifies the one with the smallest F-value to be dropped.

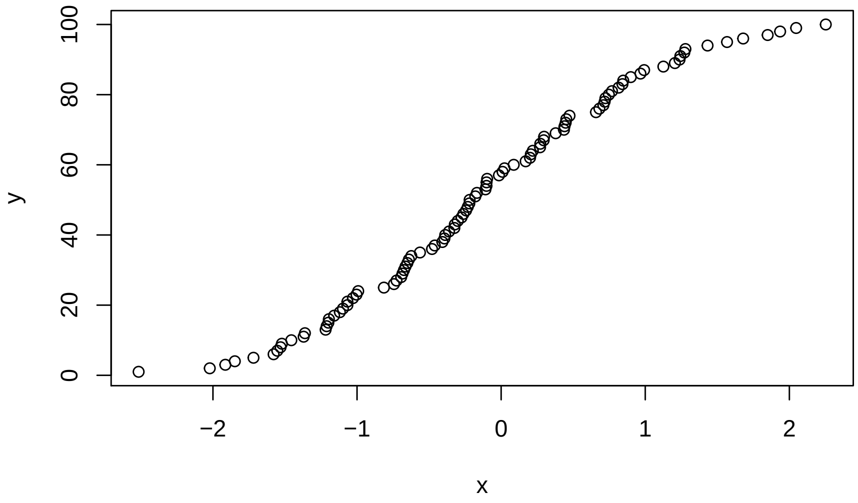
5.1.6 Diagnostics

5.1.6.1 Normality of errors

could use Methods based on normal probability plot or Methods based on empirical cumulative distribution function

or plots such as

```
y = 1:100
x = rnorm(100)
qqplot(x,y)
```



5.1.6.2 Influential observations/outliers

5.1.6.2.1 Hat matrix

$$\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-\mathbf{1}}$$

where $\hat{\mathbf{Y}} = \mathbf{HY}, \mathbf{e} = (\mathbf{I} - \mathbf{H})\mathbf{Y}$ and $var(\mathbf{e}) = \sigma^2(\mathbf{I} - \mathbf{H})$

- $\sigma^2(e_i) = \sigma^2(1 - h_{ii})$, where h_{ii} is the i-th element of the main diagonal of $\mathbf{H}$ (must be between 0 and 1).
- $\sum_{i=1}^{n} h_{ii} = p$
- $cov(e_i, e_j) = -h_{ii}\sigma^2$ where $i \neq j$
- Estimate: $s^2(e_i) = MSE(1 - h_{ii})$
- Estimate: $\hat{cov}(e_i, e_j) = -h_{ij}(MSE)$; if model assumption are correct, this covariance is very small for large data sets.
- If $\mathbf{x}_i = [1X_{i,1}...X_{i,p-1}]'$ (the vector of X-values for a given response), then $h_{ii} = \mathbf{x_i'}(\mathbf{X}'\mathbf{X})^{-\mathbf{1}}\mathbf{x_i}$ (depends on relative positions of the design points $X_{i,1}, ..., X_{i,p-1}$)

5.1.6.2.2 Studentized Residuals

$$r_i = \frac{e_i}{s(e_i)} r_i \sim N(0,1)$$

where $s(e_i) = \sqrt{MSE(1 - h_{ii})}$. r_i is called the studentized residual or standardized residual.

- you can use the semi-studentized residual before, $e_i^* = e_i\sqrt{MSE}$. This doesn't take into account the different variances for each e_i.

We would want to see the model without a particular value. You delete the i-th case, fit the regression to the remaining n-1 cases, get estimated responses for the i-th case, $\hat{Y}_{i(i)}$, and find the difference, called the **deleted residual**:

$$d_i = Y_i - \hat{Y}_{i(i)} = \frac{e_i}{1 - h_{ii}}$$

we don't need to recompute the regression model for each case

As h_{ii} increases, d_i increases.

$$s^2(d_i) = \frac{MSE_{(i)}}{1 - h_{ii}}$$

where $MSE_{(i)}$ is the mean square error when the i-th case is omitted.

Let

$$t_i = \frac{d_i}{s(d_i)} = \frac{e_i}{\sqrt{MSE_{(i)}(1-h_{ii})}}$$

be the **studentized deleted residual**, which follows a t-distribution with n-p-1 df.

$$(n-p)MSE = (n-p-1)MSE_{(i)} + \frac{e_i^2}{1-h_{ii}}$$

hence, we do not need to fit regressions for each case and

$$t_i = e_i(\frac{n-p-1}{SSE(1-h_{ii})-e_i^2})^{1/2}$$

The outlying Y-observations are those cases whose studentized deleted residuals are large in absolute value. If there are many residuals to consider, a Bonferroni critical value can be can ($t_{1-\alpha/2n;n-p-1}$)

Outlying X Observations

Recall, $0 \leq h_{ii} \leq 1$ and $\sum_{i=1}^{n} h_{ii} = p$ (the total number of parameters)

A large h_{ii} indicates that the i-th case is distant from the center of all X observations (the **leverage** of the i-th case). That is, a large value suggests that the observation exercises substantial leverage in determining the fitted value $\hat{Y}_i$

We have $\hat{\mathbf{Y}} = \mathbf{HY}$, a linear combination of Y-values; h_{ii} is the weight of the observation Y_i; so h_{ii} measures the role of the X values in determining how important Y_i is in affecting the $\hat{Y}_i$.

Large h_{ii} implies $var(e_i)$ is small, so larger h_{ii} implies that $\hat{Y}_i$ is close to Y_i

- small data sets: $h_{ii} > .5$ suggests “large”.
- large data sets: $h_{ii} > \frac{2p}{n}$ is ”large.

Using the hat matrix to identify extrapolation:

- Let $\mathbf{x_{new}}$ be a vector containing the X values for which an inference about a mean response or a new observation is to be made.
- Let $\mathbf{X}$ be the data design matrix used to fit the data. Then, if $h_{new,new} = \mathbf{x_{new}}(\mathbf{X'X})^{-\mathbf{1}}\mathbf{x_{new}}$ is within the range of leverage values (h_{ii}) for cases in the data set, no extrapolation is involved; otherwise; extrapolation is indicated.

Identifying Influential Cases:

by influential we mean that exclusion of an observation causes major changes int he fitted regression. (not all outliers are influential)

- Influence on Single Fitted Values: DFFITS
- Influence on All Fitted Values: Cook's D
- Influence on the Regression Coefficients: DFBETAS

5.1.6.2.3 DFFITS

Influence on Single Fitted Values: DFFITS

$$(DFFITS)_i = \frac{\hat{Y}_i - \hat{Y}_{i(i)}}{\sqrt{MSE_{(i)}h_{ii}}} = t_i(\frac{h_{ii}}{1-h_{ii}})^{1/2}$$

- the standardized difference between the i-th fitted value with all observations and with the i-th case removed.
- studentized deleted residual multiplied by a factor that is a function fo the i-th leverage value.
- influence if:
  - small to medium data sets: $|DFFITS| > 1$
  - large data sets: $|DFFITS| > 2\sqrt{p/n}$

5.1.6.2.4 Cook's D

Influence on All Fitted Values: Cook's D

$$D_i = \frac{\sum_{j=1}^n (\hat{Y}_j - \hat{Y}_{j(i)})^2}{p(MSE)} = \frac{e_i^2}{p(MSE)}(\frac{h_{ii}}{(1-h_{ii})^2})$$

gives the influence of i-th case on all fitted values.

If e_i increases or h_{ii} increases, then D_i increases.

D_i is a percentile of an $F_{(p,n-p)}$ distribution. If the percentile is greater than .5(50%) then the i-th case has major influence. In practice, if $D_i > 4/n$, then the i-th case has major influence.

5.1.6.2.5 DFBETAS

Influence on the Regression Coefficients: DFBETAS

$$(DFBETAS)_{k(i)} = \frac{b_k - b_{k(i)}}{\sqrt{MSE_{(i)}c_{kk}}}$$

for $k = 0, ..., p-1$ and c_{kk} si the k-th diagonal element of $\mathbf{X'X}^{-1}$

Influence of the i-th case on each regression coefficient b_k (k=0,..,p-1) is the difference between the estimated regression coefficients based on all n cases and the regression coefficients obtained when the i-th case is omitted ($b_{k(i)}$)

- small data sets: $|DFBETA| > 1$
- large data sets: $|DFBETA| > 2\sqrt{n}$
- Sign of DFBETA inculcates whether inclusion of a case leads to an increase or a decrease in estimates of the regression coefficient.

5.1.6.3 Collinearity

Multicollinearity refers to correlation among explanatory variables.

- large changes in the estimated regression coefficient when a predictor variable is added or deleted, or when an observation is altered or deleted.
- noninsignificant results in individual tests on regression coefficients for important predictor variables.
- estimated regression coefficients with an algebraic sign that is the opposite of that expected from theoretical consideration or prior experience.
- large coefficients of simple correlation between pairs of predictor variables in the correlation matrix.
- wide confidence intervals for the regression coefficients representing important predictor variables.

When some of X variables are so highly correlated that the inverse $(X'X)^{-1}$ does not exist or is very computationally unstable.

Correlated Predictor Variables: if some X variables are "perfectly" correlated, the system is undetermined and there are an infinite number of models that fit the data. That is, if $X'X$ is singular, then $(X'X)^{-1}$ doesn't exist. Then,

- parameters cannot be interpreted ($\mathbf{b} = (\mathbf{X'X})^{\mathbf{-1}}\mathbf{X'y}$)
- sampling variability is infinite ($\mathbf{s^2}(\mathbf{b}) = \mathbf{MSE}(\mathbf{X'X})^{\mathbf{-1}}$)

5.1.6.3.1 VIFs Let R^2_k be the coefficient of multiple determination when X_k is regressed on the p - 2 other X variables in the model. Then,

$$VIF_k = \frac{1}{1 - R^2_k}$$

- large values indicate that a near collinearity is causing the variance of b_k to be inflated, $var(b_k) \propto \sigma^2(VIF_k)$
- typically, $VIF_k > 10$ indicates a collinearity problem that could result in poor parameters estimates.
- the mean of all VIF's provide an estimate of the ratio of the true multicollinearity to a model where the X variables are uncorrelated
- serious multicollinearity if $avg(VIF) >> 1$

5.1.6.3.2 Condition Number Condition Number

spectral decomposition

$$\mathbf{X'X} = \sum_{i=1}^{p} \lambda_i \mathbf{u_i u_i'}$$

where λ_i is the eigenvalue and $\mathbf{u}_i$ is the eigenvector. $\lambda_1 > ... > \lambda_p$ and the eigenvecotrs are orthogonal:

$$\begin{cases} \mathbf{u_i'u_j} = 0 & \text{for } i \neq j \\ 1 & \text{for } i = j \end{cases}$$

The condition number is then

$$k = \sqrt{\frac{\lambda_{max}}{\lambda_{min}}}$$

- values $k > 30$ are cause for concern
- values $30 < k < 100$ imply moderate dependencies.
- values $k > 100$ imply strong collinearity

Condition index

$$\delta_i = \sqrt{\frac{\lambda_{max}}{\lambda_i}}$$

where i = 1,…,p

we can find the proportion of the total variance associated with the k-th regression coefficient and the i-th eigenmode:

$$\frac{u_{ik}^2/\lambda_i}{\sum_j (u_{jk}^2/\lambda_j)}$$

These variance proportions can be helpful for identifying serious conllienarity

- the condition index must be large
- the variance proportions must be large (>,5) for at least two regression coefficients.

5.1.6.4 Constancy of Error Variance

5.1.6.4.1 Brown-Forsythe Test (Modified Levene Test)

- Does not depend on normality
- Applicable when error variance increases or decreases with X
- relatively large sample size needed (so we can ignore dependency between residuals)
- Split residuals into 2 groups ($e_{i1}, i = 1, ..., n_1; e_{i2}, j = 1, ..., n_2$)
- Let $d_{i1} = |e_{i1} - \tilde{e}_1|$ where $\tilde{e}_1$ is the median of group 1.
- Let $d_{j2} = |e_{j2} - \tilde{e}_2|$.
- Then, a 2-sample t-test:

  $$t_L = \frac{\bar{d}_1 - \bar{d}_2}{s\sqrt{1/n_1 + 1/n_2}}$$

  where

  $$s^2 = \frac{\sum_i (d_{i1} - \bar{d}_1)^2 + \sum_j (d_{j2} - \bar{d}_2)^2}{n-2}$$

  If $|t_L| > t_{1-\alpha/2;n-2}$ conclude the error variance is not constant.

5.1.6.4.2 Breusch-Pagan Test (Cook-Weisberg Test) Assume the error terms are independent and normally distributed, and

$$\sigma_i^2 = \gamma_0 + \gamma_1 X_i$$

Constant error variance corresponds to $\gamma_1 = 0$, i.e., test

- $H_0 : \gamma_1 = 0$

- $H_1 : \gamma_1 \neq 0$

by regressing the squared residuals on X in the usual manner. Obtain the regression sum of squares from this: SSR* (the SSR from the regression of e_i^2 on X_i). Then, define

$$X^2_{BP} = \frac{SSR*/2}{(SSE/n)^2}$$

where SSE is the error sum of squares from the regression of Y on X.

If $H_0 : \gamma_1 = 0$ holds and n is reasonably large, X^2_{BP} follows approximately the χ^2 distribution with 1 d.f. We reject H_0 (Homogeneous variance) if $X^2_{BP} > \chi^2_{1-\alpha;1}$

5.1.6.5 Independence

5.1.6.5.1 Plots

5.1.6.5.2 Durbin-Watson

5.1.6.5.3 Time-series

5.1.6.5.4 Spatial Statistics

5.1.7 Model Validation

- split data into 2 groups: training (model building) sample and validation (prediction) sample.

- the model MSE will tend to underestimate the inherent variability in making future predictions. to consider actual predictive ability, consider mean squared prediction error (MSPE):

$$MSPE = \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{n^*}$$

  - where Y_i is the known value of the response variable in the i-th validation case.
  - $\hat{Y}_i$ is the predicted value based on a model fit with the training data set.
  - n^* is the number of cases in the validation set.

- we want MSPE to be close to MSE (in which MSE is not biased); so look at the the ratio MSPE / MSE (closer to 1, the better).

5.1.8 Finite Sample Properties

- n is fixed
- **Bias** On average, how close is our estimate to the true value
  - $Bias = E(\hat{\beta}) - \beta$ where β is the true parameter value and $\hat{\beta}$ is the estimator for β
  - An estimator is **unbiased** when
    - $Bias = E(\hat{\beta}) - \beta = 0$ or $E(\hat{\beta}) = \beta$
    - means that the estimator will produce estimates that are, on average, equal to the value it it trying to estimate
- **Distribution of an estimator**: An estimator is a function of random variables (data)
- **Standard Deviation**: the spread of the estimator.

OLS

Under A1 A2 A3, OLS is unbiased

$$
\begin{aligned}
E(\hat{\beta}) &= E((\mathbf{X'X})^{-\mathbf{1}}\mathbf{X'y}) && \text{A2} \\
&= E((\mathbf{X'X})^{-\mathbf{1}}\mathbf{X'}(\mathbf{X}\ \ +\ \ )) && \text{A1} \\
&= E((\mathbf{X'X})^{-\mathbf{1}}\mathbf{X'X}\ \ + (\mathbf{X'X})^{-\mathbf{1}}\mathbf{X'}\ ) \\
&= E(\beta + (\mathbf{X'X})^{-\mathbf{1}}\mathbf{X'}\ ) \\
&= \beta + E((\mathbf{X'X^{-1}}\ )) \\
&= \beta + E(E((\mathbf{X'X})^{-\mathbf{1}}\mathbf{X'}\ |\mathbf{X})) && \text{LIE} \\
&= \beta + E((\mathbf{X'X})^{-\mathbf{1}}\mathbf{X'}E(\ |\mathbf{X})) \\
&= \beta + E((\mathbf{X'X})^{-\mathbf{1}}\mathbf{X'}0)) && \text{A3} \\
&= \beta
\end{aligned}
$$

where LIE stands for Law of Iterated Expectation

If A3 does not hold, then OLS will be **biased**

From **Frisch-Waugh-Lovell Theorem**, if we have the omitted variable $\hat{\beta}_2 \neq 0$ and $\mathbf{X_1'X_2} \neq 0$, then the omitted variable will cause OLS estimator to be biased.

Under A1 A2 A3 A4, we have the conditional variance of the OLS estimator as follows]

$$
\begin{aligned}
Var(\hat{\beta}|\mathbf{X}) &= Var(\beta + (\mathbf{X'X})^{-\mathbf{1}}\mathbf{X'}\ |\mathbf{X}) && \text{A1-A2} \\
&= Var((\mathbf{X'X})^{-\mathbf{1}}\mathbf{X'}\ |\mathbf{X}) \\
&= \mathbf{X'X^{-1}X'}Var(\epsilon|\mathbf{X})\mathbf{X}(\mathbf{X'X})^{-\mathbf{1}} \\
&= \mathbf{X'X^{-1}X'}\sigma^2 I\mathbf{X}(\mathbf{X'X})^{-\mathbf{1}} && \text{A4} \\
&= \sigma^2\mathbf{X'X^{-1}X'}I\mathbf{X}(\mathbf{X'X})^{-\mathbf{1}} \\
&= \sigma^2(\mathbf{X'X})^{-\mathbf{1}}
\end{aligned}
$$

Sources of variation

1. $\sigma^2 = Var(\epsilon_i|\mathbf{X})$
    - The amount of unexplained variation ϵ_i is large relative to the explained $\mathbf{x_i}$ variation
2. "Small" $Var(x_{i1}), Var(x_{i1}), ..$
    - Not a lot of variation in $\mathbf{X}$ (no information)
    - small sample size
3. "Strong" correlation between the explanatory variables
    - x_{i1} is highly correlated with a linear combination of 1, x_{i2}, x_{i3}, …
    - include many irrelevant variables will contribute to this.
    - If x_1 is perfectly determined in the regression → **Perfect Collinearity** → A2 is violated.
    - If x_1 is highly correlated with a linear combination of other variables, then we have **Multicollinearity**

5.1.8.1 Check for Multicollinearity

Variance Inflation Factor (VIF) Rule of thumb $VIF \geq 10$ is large

$$VIF = \frac{1}{1 - R_1^2}$$

5.1.8.2 Standard Errors

- $Var(\hat{\beta}|\mathbf{X}) = \sigma^2(\mathbf{X'X})^{-\mathbf{1}}$ is the variance of the estimate $\hat{\beta}$
- **Standard Errors** are estimators/estimates of the standard deviation (square root of the variance) of the estimator $\hat{\beta}$
- Under A1-A5, then we can estimate $\sigma^2 = Var(\epsilon^2|\mathbf{X})$ the standard errors as

$$s^2 = \frac{1}{n-k}\sum_{i=1}^{n} e_i^2 = \frac{1}{n-k}SSR$$

- degrees of freedom adjustment: because $e_i \neq \epsilon_i$ and are estimated using k estimates for β, we lose degrees of freedom in our variance estimate.
- $s = \sqrt{s^2}$ is a biased estimator for the standard deviation ([Jensen's Inequality])

Standard Errors for $\hat{\beta}$

$$SE(\hat{\beta}_{j-1}) = s\sqrt{[(\mathbf{X'X})^{-1}]_{jj}} = \frac{s}{\sqrt{SST_{j-1}(1 - R_{j-1}^2)}}$$

where SST_{j-1} and R_{j-1}^2 from the following regression

x_{j-1} on 1, x_1,… x_{j-2},x_j,x_{j+1}, …, x_{k-1}

Summary of Finite Sample Properties

- Under A1-A3: OLS is unbiased
- Under A1-A4: The variance of the OLS estimator is $Var(\hat{\beta}|\mathbf{X}) = \sigma^2(\mathbf{X'X})^{\mathbf{-1}}$
- Under A1-A4, A6: OLS estimator $\hat{\beta} \sim N(\beta, \sigma^2(\mathbf{X'X})^{\mathbf{-1}})$
- Under A1-A4, Gauss-Markov Theorem holds → OLS is BLUE
- Under A1-A5, the above standard errors are unbiased estimator of standard deviation for $\hat{\beta}$

5.1.9 Large Sample Properties

- let $n \to \infty$
- A perspective that allows us to evaluate the "quality" of estimators when finite sample properties are not informative, or impossible to compute
- consistency, asymptotic distribution, asymptotic variance

Motivation

- Finite Sample Properties need strong assumption A1 A3 A4 A6
- Other estimation such as GLS, MLE need to be analyzed using Large Sample Properties

Let $\mu(\mathbf{X}) = E(y|\mathbf{X})$ be the **Conditional Expectation Function**

- $\mu(\mathbf{X})$ is the minimum mean squared predictor (over all possible functions)

$$minE((y - f(\mathbf{X}))^2)$$

under A1 and A3,

$$\mu(\mathbf{X}) = \mathbf{X}\beta$$

Then the **linear projection**

$$L(y|1, \mathbf{X}) = \gamma_0 + \mathbf{X}Var(X)^{-1}Cov(X, Y)$$

where $\mathbf{X}Var(X)^{-1}Cov(X, Y) = \gamma$

is the minimum mean squared linear approximation to be conditional mean function

$$(\gamma_0, \gamma) = argminE((E(y|\mathbf{X}) - (a + \mathbf{Xb})^2)$$

- OLS is always **consistent** for the linear projection, but not necessarily unbiased.
- Linear projection has no causal interpretation

- Linear projection does not depend on assumption A1 and A3

Evaluating an estimator using large sample properties:

- Consistency: measure of centrality
- Limiting Distribution: the shape of the scaled estimator as the sample size increases
- Asymptotic variance: spread of the estimator with regards to its limiting distribution.

An estimator $\hat{\theta}$ is consistent for θ if $\hat{\theta}_n \to^p \theta$

- As n increases, the estimator converges to the population parameter value.
- Unbiased does not imply consistency and consistency does not imply unbiased.

Based on Weak Law of Large Numbers

$$
\begin{aligned}
\hat{\beta} &= (\mathbf{X'X})^{-\mathbf{1}}\mathbf{X'y} \\
&= (\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{x_i'x_i})^{-\mathbf{1}} \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{x_i'y_i} \\
&= (n^{-1}\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{x_i'x_i})^{-\mathbf{1}} n^{-1}\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{x_i'y_i}
\end{aligned}
$$

$$
\begin{aligned}
plim(\hat{\beta}) &= plim((n^{-1}\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{x_i'x_i})^{-\mathbf{1}} n^{-1}\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{x_i'y_i}) \\
&= plim((n^{-1}\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{x_i'x_i})^{-\mathbf{1}}) plim(n^{-1}\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{x_i'y_i}) \\
&= (plim(n^{-1}\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{x_i'x_i})^{-\mathbf{1}}) plim(n^{-1}\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{x_i'y_i}) \text{ due to A2, A5} \\
&= E(\mathbf{x_i'x_i})^{-1}E(\mathbf{x_i'y_i})
\end{aligned}
$$

$$
E(\mathbf{x_i'x_i})^{-1}E(\mathbf{x_i'y_i}) = \beta + E(\mathbf{x_i'x_i})^{-1}E(\mathbf{x_i'}\ _{\mathbf{i}})
$$

Under A1, A2, A3a, A5 OLS is consistent, but not guarantee unbiased.

Under A1, A2, A3a, A5, and $\mathbf{x_i'x_i}$ has finite first and second moments (CLT), $Var(\mathbf{x_i'}\epsilon_i) = \mathbf{B}$

- $(n^{-1}\sum_{i=1}^{n} \mathbf{x_i'x_i})^{-1} \to^p (E(\mathbf{x_i'x_i}))^{-1}$
- $\sqrt{n}(n^{-1}\sum_{i=1}^{n} \mathbf{x_i'}\epsilon_i) \to^d N(0, \mathbf{B})$

$$\sqrt{n}(\hat{\beta}-\beta) = (n^{-1}\sum_{i=1}^{n}\mathbf{x_i'x_i})^{-1}\sqrt{n}(n^{-1}\sum_{i=1}^{n}\mathbf{x_i'x_i}) \to^d N(0,\Sigma)$$

where $\Sigma = (E(\mathbf{x_i'x_i}))^{-1}\mathbf{B}(E(\mathbf{x_i'x_i}))^{-1}$

- holds under A3a
- Do not need A4 and A6 to apply CLT
  - If A4 does not hold, then $\mathbf{B} = Var(\mathbf{x_i'}\epsilon_i) = \sigma^2 E(x_i'x_i)$ which means $\Sigma = \sigma^2(E(\mathbf{x_i'x_i}))^{-1}$, use standard errors

Heteroskedasticity can be from

- Limited dependent variable
- Dependent variables with large/skewed ranges

Solving Asymptotic Variance

$$\begin{aligned}
\Sigma &= (E(\mathbf{x_i'x_i}))^{-1}\mathbf{B}(E(\mathbf{x_i'x_i}))^{-1} \\
&= (E(\mathbf{x_i'x_i}))^{-1}Var(\mathbf{x_i'}\epsilon_i)(E(\mathbf{x_i'x_i}))^{-1} \\
&= (E(\mathbf{x_i'x_i}))^{-1}E[(\mathbf{x_i'}\epsilon_i - 0)(\mathbf{x_i'}\epsilon_i - 0)](E(\mathbf{x_i'x_i}))^{-1} && \text{A3a} \\
&= (E(\mathbf{x_i'x_i}))^{-1}E[E(\,\mathbf{^2_i}|\mathbf{x_i})\mathbf{x_i'x_i}](E(\mathbf{x_i'x_i}))^{-1} && \text{LIE} \\
&= (E(\mathbf{x_i'x_i}))^{-1}\sigma^2 E(\mathbf{x_i'x_i})(E(\mathbf{x_i'x_i}))^{-1} && \text{A4} \\
&= \sigma^2(E(\mathbf{x_i'x_i}))
\end{aligned}$$

Under A1, A2, A3a, A4, A5:

$$\sqrt{n}(\hat{\beta}-\beta) \to^d N(0,\sigma^2(E(\mathbf{x_i'x_i}))^{-1})$$

- The Asymptotic variance is approximation for the variance in the scaled random variable for $\sqrt{n}(\hat{\beta}-\beta)$ when n is large.
- use $Avar(\sqrt{n}(\hat{\beta}-\beta))/n$ as an approximation for finite sample variance for large n:

$$Avar(\sqrt{n}(\hat{\beta}-\beta)) \approx Var(\sqrt{n}(\hat{\beta}-\beta))Avar(\sqrt{n}(\hat{\beta}-\beta))/n \approx Var(\sqrt{n}(\hat{\beta}-\beta))/n = Var(\hat{\beta})$$

- Avar(.) does not behave the same way as Var(.)

$$Avar(\sqrt{n}(\hat{\beta}-\beta))/n \neq Avar(\sqrt{n}(\hat{\beta}-\beta)/\sqrt{n}) \neq Avar(\hat{\beta})$$

In Finite Sample Properties, we calculate standard errors as an estimate for the conditional standard deviation:

$$SE_{fs}(\hat{\beta}_{j-1}) = \sqrt{\hat{Var}(\hat{\beta}_{j-1}|\mathbf{X})} = \sqrt{s^2[(\mathbf{X'X})^{-1}]_{jj}}$$

In Large Sample Properties, we calculate standard errors as an estimate for the square root of asymptotic variance

$$SE_{ls}(\hat{\beta}_{j-1}) = \sqrt{A\hat{var}(\sqrt{n}\hat{\beta}_{j-1})/n} = \sqrt{s^2[(\mathbf{X'X})^{-1}]_{jj}}$$

Hence, the standard error estimator is the same for finite sample and large sample. * Same estimator, but conceptually estimating two different things. * Valid under weaker assumptions: the assumptions needed to produce a consistent estimator for the finite sample conditional variance (A1-A5) are stronger than those needed to produce a consistent estimator for the asymptotic variance (A1,A2,A3a,A4,A5)

Suppose that $y_1, ..., y_n$ are a random sample from some population with mean μ and variance-covariance matrix Σ

- $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ is a consistent estimator for μ
- $S = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})(y_i - \bar{y})'$ is a consistent estimator for Σ.
- Multivariate Central limit Theorem: Similar to the univariate case, $\sqrt{n}(\bar{y} - \mu) \sim N_p(0, \Sigma)$, when n is large relative to p (e.g., $n \geq 25p$). Equivalently, $\bar{y} \sim N_p(\mu, \Sigma/n)$.
- Wald's Theorem: $n(\bar{y} - \mu)'S^{-1}(\bar{y} - \mu) \sim \chi^2_{(p)}$ when n is large relative to p.

5.1.10 Application

5.2 Feasible Generalized Least Squares

Motivation for a more efficient estimator

- Gauss-Markov Theorem holds under A1-A4
- A4: $Var(\epsilon|\mathbf{X}) = \sigma^2 I_n$
  - Heteroskedasticity: $Var(\epsilon_i|\mathbf{X}) \neq \sigma^2 I_n$
  - Serial Correlation: $Cov(\epsilon_i, \epsilon_j|\mathbf{X}) \neq 0$
- Without A4, how can we know which unbiased estimator is the most efficient?

Original (unweighted) model:

$$\mathbf{y} = \mathbf{X} \quad +$$

Suppose A1-A3 hold, but A4 does not hold,

$$\mathbf{Var}(\ |\mathbf{X}) = \ \neq {}^{\mathbf{2}}\mathbf{I_n}$$

We will try to use OLS to estimate the transformed (weighted) model

$$\mathbf{wy} = \mathbf{wX} \; + \mathbf{w}$$

We need to choose **w** so that

$$\mathbf{w'w} = \;^{\mathbf{-1}}$$

then **w** (full-rank matrix) is the **Cholesky decomposition** of $^{\mathbf{-1}}$ (full-rank matrix)

In other words, **w** is the squared root of Ω (squared root version in matrix)

$$\Omega = var(\epsilon|X)\Omega^{-1} = var(\epsilon|X)^{-1}$$

Then, the transformed equation (IGLS) will have the following properties.

$$\begin{aligned} \hat{}_{\mathbf{IGLS}} &= (\mathbf{X'w'wX})^{-\mathbf{1}}\mathbf{X'w'wy} \\ &= (\mathbf{X'}\;^{-\mathbf{1}}\mathbf{X})^{-\mathbf{1}}\mathbf{X'}\;^{-\mathbf{1}}\mathbf{y} \\ &= \; + \mathbf{X'}\;^{-\mathbf{1}}\mathbf{X'}\;^{-\mathbf{1}} \end{aligned}$$

Since A1-A3 hold for the unweighted model

$$\begin{aligned} \mathbf{E}(\hat{}_{\mathbf{IGLS}}|\mathbf{X}) &= E(\; + (\mathbf{X'}\;^{-\mathbf{1}}\mathbf{X'}\;^{-\mathbf{1}}\;)|X) \\ &= \; + \mathbf{E}(\mathbf{X'}\;^{-\mathbf{1}}\mathbf{X'}\;^{-\mathbf{1}}\;)|\mathbf{X}) \\ &= \; + \mathbf{X'}\;^{-\mathbf{1}}\mathbf{X'}\;^{-\mathbf{1}}\mathbf{E}(\;|\mathbf{X}) \qquad \text{since A3} : E(\epsilon|X) = 0 \\ &= \end{aligned}$$

→ IGLS estimator is unbiased

$$\begin{aligned} \mathbf{Var}(\mathbf{w}\;|\mathbf{X}) &= \mathbf{wVar}(\;|\mathbf{X})\mathbf{w'} \\ &= \mathbf{w}\;\mathbf{w'} \\ &= \mathbf{w}(\mathbf{w'w})^{-\mathbf{1}}\mathbf{w'} \qquad \text{since w is a full-rank matrix} \\ &= \mathbf{ww}^{-\mathbf{1}}(\mathbf{w'})^{-\mathbf{1}}\mathbf{w'} \\ &= \mathbf{I_n} \end{aligned}$$

→ A4 holds for the transformed (weighted) equation

Then, the variance for the estimator is

$$\begin{aligned}
Var(\hat{\beta}_{IGLS}|\mathbf{X}) &= \mathbf{Var}(\ \ + (\mathbf{X}'\ {}^{-\mathbf{1}}\mathbf{X})^{-\mathbf{1}}\mathbf{X}'\ {}^{-\mathbf{1}}\ |\mathbf{X}) \\
&= \mathbf{Var}((\mathbf{X}'\ {}^{-\mathbf{1}}\mathbf{X})^{-\mathbf{1}}\mathbf{X}'\ {}^{-\mathbf{1}}\ |\mathbf{X}) \\
&= (\mathbf{X}'\ {}^{-\mathbf{1}}\mathbf{X})^{-\mathbf{1}}\mathbf{X}'\ {}^{-\mathbf{1}}\mathbf{Var}(\ |\mathbf{X})\ {}^{-\mathbf{1}}\mathbf{X}(\mathbf{X}'\ {}^{-\mathbf{1}}\mathbf{X})^{-\mathbf{1}} \quad \text{because A4 holds} \\
&= (\mathbf{X}'\ {}^{-\mathbf{1}}\mathbf{X})^{-\mathbf{1}}\mathbf{X}'\ {}^{-\mathbf{1}}\ \ {}^{-\mathbf{1}}\ {}^{-\mathbf{1}}\mathbf{X}(\mathbf{X}'\ {}^{-\mathbf{1}}\mathbf{X})^{-\mathbf{1}} \\
&= (\mathbf{X}'\ {}^{-\mathbf{1}}\mathbf{X})^{-\mathbf{1}}
\end{aligned}$$

Let $A = (\mathbf{X}'\mathbf{X})^{-\mathbf{1}}\mathbf{X}' - (\mathbf{X}'\ {}^{-\mathbf{1}}\mathbf{X})\mathbf{X}'\ {}^{-\mathbf{1}}$ then

$$Var(\hat{\beta}_{OLS}|X) - Var(\hat{\beta}_{IGLS}|X) = A\Omega A'$$

And Ω is Positive Semi Definite, then $A\Omega A'$ also PSD, then IGLS is more efficient

The name **Infeasible** comes from the fact that it is impossible to compute this estimator.

$$\mathbf{w} = \begin{pmatrix} w_{11} & 0 & 0 & ... & 0 \\ w_{21} & w_{22} & 0 & ... & 0 \\ w_{31} & w_{32} & w_{33} & ... & ... \\ w_{n1} & w_{n2} & w_{n3} & ... & w_{nn} \end{pmatrix}$$

With $n(n+1)/2$ number of elements and n observations → infeasible to estimate. (number of equation $>$ data)

Hence, we need to make assumption on Ω to make it feasible to estimate $\mathbf{w}$:

1. Heteroskedasticity : multiplicative exponential model
2. AR(1)
3. Cluster

5.2.1 Heteroskedasticity

$$\begin{aligned}
Var(\epsilon_i|x_i) &= E(\epsilon^2|x_i) \neq \sigma^2 \\
&= h(x_i) = \sigma_i^2(\text{variance of the error term is a function of x})
\end{aligned} \tag{5.7}$$

For our model,

$$y_i = x_i\beta + \epsilon_i(1/\sigma_i)y_i = (1/\sigma_i)x_i\beta + (1/\sigma_i)\epsilon_i$$

then, from (5.7)

$$\begin{aligned}
Var((1/\sigma_i)\epsilon_i|X) &= (1/\sigma_i^2)Var(\epsilon_i|X) \\
&= (1/\sigma_i^2)\sigma_i^2 \\
&= 1
\end{aligned}$$

then the weight matrix $\mathbf{w}$ in the matrix equation

$$\mathbf{wy} = \mathbf{wX} \ + \mathbf{w}$$

$$\mathbf{w} = \begin{pmatrix} 1/\sigma_1 & 0 & 0 & ... & 0 \\ 0 & 1/\sigma_2 & 0 & ... & 0 \\ 0 & 0 & 1/\sigma_3 & ... & . \\ . & . & . & . & 0 \\ 0 & 0 & . & . & 1/\sigma_n \end{pmatrix}$$

Infeasible Weighted Least Squares

1. Assume we know σ_i^2 (Infeasible)
2. The IWLS estimator is obtained as the least squared estimated for the following weighted equation

$$(1/\sigma_i)y_i = (1/\sigma_i)\mathbf{x}_i\beta + (1/\sigma_i)\epsilon_i$$

- Usual standard errors for the weighted equation are valid if $Var(\epsilon|\mathbf{X}) = \sigma_i^2$
- If $Var(\epsilon|\mathbf{X}) \neq \sigma_i^2$ then heteroskedastic robust standard errors are valid.

Problem: We do not know $\sigma_i^2 = Var(\epsilon_i|\mathbf{x_i}) = E(\epsilon_i^2|\mathbf{x}_i)$

- One observation ϵ_i cannot estimate a sample variance estimate σ_i^2
  - Model ϵ_i^2 as reasonable (strictly positive) function of x_i and independent error v_i (strictly positive)

$$\epsilon_i^2 = v_i exp(\mathbf{x_i} \ )$$

Then we can apply a log transformation to recover a linear in parameters model,

$$ln(\epsilon_i^2) = \mathbf{x_i} \ + ln(v_i)$$

where $ln(v_i)$ is independent $\mathbf{x}_i$

We do not observe ϵ_i * OLS residual (e_i) as an approximate

5.2.2 Serial Correlation

$$Cov(\epsilon_i, \epsilon_j|\mathbf{X}) \neq 0$$

Under covariance stationary,

$$Cov(\epsilon_i, \epsilon_j|\mathbf{X}) = Cov(\epsilon_i, \epsilon_{i+h}|\mathbf{x_i}, \mathbf{x_{i+h}}) = \gamma_h$$

And the variance covariance matrix is

$$Var(\epsilon|\mathbf{X}) = \Omega = \begin{pmatrix} \sigma^2 & \gamma_1 & \gamma_2 & \cdots & \gamma_{n-1} \\ \gamma_1 & \sigma^2 & \gamma_1 & \cdots & \gamma_{n-2} \\ \gamma_2 & \gamma_1 & \sigma^2 & \cdots & \cdots \\ . & . & . & . & \gamma_1 \\ \gamma_{n-1} & \gamma_{n-2} & . & \gamma_1 & \sigma^2 \end{pmatrix}$$

There n parameters to estimate - need some sort fo structure to reduce number of parameters to estimate.

- Time Series
  - Effect of inflation and deficit on Treasury BIll interest rates
- Cross-sectional
  - Clustering

5.2.2.1 AR(1)

$$y_t = \beta_0 + x_t\beta_1 + \epsilon_t \epsilon_t = \rho\epsilon_{t-1} + u_t$$

and the variance covariance matrix is

$$Var(\epsilon|\mathbf{X}) = \frac{\sigma_u^2}{1-\rho} \begin{pmatrix} 1 & \rho & \rho^2 & \cdots & \rho^{n-1} \\ \rho & 1 & \rho & \cdots & \rho^{n-2} \\ \rho^2 & \rho & 1 & . & . \\ . & . & . & . & \rho \\ \rho^{n-1} & \rho^{n-2} & . & \rho & 1 \end{pmatrix}$$

Hence, there is only 1 parameter to estimate: ρ

- Under A1, A2, A3a, A5a, OLS is consistent and asymptotically normal
- Use Newey West Standard Errors for valid inference.
- Apply Infeasible Cochrane Orcutt (as if we knew ρ)
- Because

$$u_t = \epsilon_t - \rho\epsilon_{t-1}$$

satisfies A3, A4, A5 we'd like to to transform the above equation to one that has u_t as the error.

$$\begin{aligned} y_t - \rho y_{t-1} &= (\beta_0 + x\beta_1 + \epsilon_t) - \rho(\beta_0 + x_{t-1}\beta_1 + \epsilon_{t-1}) \\ &= (1-\rho)\beta_0 + (x_t - \rho x_{t-1})\beta_1 + u_t \end{aligned}$$

5.2.2.1.1 Infeasible Cochrane Orcutt

1. Assume that we know ρ (Infeasible)
2. The ICO estimator is obtained as the least squared estimated for the following weighted first difference equation

$$y_t - \rho y_{t-1} = (1-\rho)\beta_0 + (x_t - \rho x_{t-1})\beta_1 + u_t$$

- Usual standard errors for the weighted first difference equation are valid if the errors truly follow an AR(1) process
- If the serial correlation is generated from a more complex dynamic process then Newey-West HAC standard errors are valid

Problem We do not know ρ

- ρ is the correlation between ϵ_t and ϵ_{t-1}: estimate using OLS residuals (e_i) as proxy

$$\hat{\rho} = \frac{\sum_{t=1}^{T} e_t e_{t-1}}{\sum_{t=1}^{T} e_t^2}$$

which can be obtained from the OLS regression of

$$e_t = \rho e_{t-1} + u_t$$

where we suppress the intercept.

- We are losing an observation
  - By taking the first difference we are dropping the first observation

$$y_1 = \beta_0 + x_1 \beta_1 + \epsilon_1$$

+ Feasiable Prais Winsten Transformation applies the Infeasible Cochrane Orcutt but includes a weighted version of the first observation

$$(\sqrt{1-\rho^2})y_1 = \beta_0 + (\sqrt{1-\rho^2})x_1\beta_1 + (\sqrt{1-\rho^2})\epsilon_1$$

5.2.2.2 Cluster

$$y_{gi} = \mathbf{x}_{gi}\beta + \epsilon_{gi}$$

$$Cov(\epsilon_{gi}, \epsilon_{hj}) \begin{cases} = 0 & \text{for } g \neq h \text{ and any pair (i,j)} \\ \neq 0 & \text{for any (i,j) pair} \end{cases}$$

Intra-group Correlation
Each individual in a single group may be correlated but independent across groups.

- A4 is violated. usual standard errors for OLS are valid.
- Use **cluster robust standard errors** for OLS.

Suppose there are 3 groups with different n

$$Var(\epsilon|\mathbf{X}) = \Omega = \begin{pmatrix} \sigma^2 & \delta_{12}^1 & \delta_{13}^1 & 0 & 0 & 0 \\ \delta_{12}^1 & \sigma^2 & \delta_{23}^1 & 0 & 0 & 0 \\ \delta_{13}^1 & \delta_{23}^1 & \sigma^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2 & \delta_{12}^2 & 0 \\ 0 & 0 & 0 & \delta_{12}^2 & \sigma^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma^2 \end{pmatrix}$$

where $Cov(\epsilon_{gi}, \epsilon_{gj}) = \delta_{ij}^g$ and $Cov(\epsilon_{gi}, \epsilon_{hj}) = 0$ for any i and j

Infeasible Generalized Least Squares (Cluster)

1. Assume that σ^2 and δ_{ij}^g are known, plug into Ω and solve for the inverse Ω^{-1} (infeasible)
2. The Infeasible Generalized Least Squares Estimator is

$$\hat{\beta}_{IGLS} = (\mathbf{X}' \, {}^{-\mathbf{1}}\mathbf{X})^{-\mathbf{1}}\mathbf{X}' \, {}^{-\mathbf{1}}\mathbf{y}$$

Problem * We do not know σ^2 and δ_{ij}^g + Can make assumptions about data generating process that is causing the clustering behavior. - Will give structure to $Cov(\epsilon_{gi}, \epsilon_{gj}) = \delta_{ij}^g$ which makes it feasible to estimate - if the assumptions are wrong then we should use cluster robust standard errors.

Solution Assume **group level random effects** specification in the error

$$y_{gi} = \mathbf{g}_i\beta + c_g + u_{gi}Var(c_g|\mathbf{x}_i) = \sigma_c^2 Var(u_{gi}|\mathbf{x}_i) = \sigma_u^2$$

where c_g and u_{gi} are independent of each other, and mean independent of $\mathbf{x}_i$

- c_g captures the common group shocks (independent across groups)
- u_{gi} captures the individual shocks (independent across individuals and groups)

Then the error variance is

$$Var(\epsilon|\mathbf{X}) = \Omega = \begin{pmatrix} \sigma_c^2 + \sigma_u^2 & \sigma_c^2 & \sigma_c^2 & 0 & 0 & 0 \\ \sigma_c^2 & \sigma^2 + \sigma_u^2 & \sigma_c^2 & 0 & 0 & 0 \\ \sigma_c^2 & \sigma_c^2 & \sigma^2 + \sigma_u^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2 + \sigma_u^2 & \sigma_c^2 & 0 \\ 0 & 0 & 0 & \sigma_c^2 & \sigma^2 + \sigma_u^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma^2 + \sigma_u^2 \end{pmatrix}$$

Use Feasible group level Random Effects

5.3 Weighted Least Squares

1. Estimate the following equation using OLS

$$y_i = \mathbf{x}_i\beta + \epsilon_i$$

and obtain the residuals $e_i = y_i - \mathbf{x}_i\hat{\beta}$

2. Transform the residual and estimate the following by OLS,

$$ln(e_i^2) = \mathbf{x}_i\gamma + ln(v_i)$$

and obtain the predicted values $g_i = \mathbf{x}_i\hat{\gamma}$

3. The weights will be the untransformed predicted outcome,

$$\hat{\sigma}_i = \sqrt{exp(g_i)}$$

4. The FWLS (Feasible WLS) estimator is obtained as the least squared estimated for the following weighted equation

$$(1/\hat{\sigma}_i)y_i = (1/\hat{\sigma}_i)\mathbf{x}_i\beta + (1/\hat{\sigma}_i)\epsilon_i$$

Properties of the FWLS

- The infeasible WLS estimator is unbiased under A1-A3 for the unweighted equation.
- The FWLS estimator is NOT an unbiased estimator.
- The FWLS estimator is consistent under A1, A2, (for the unweighted equation), A5, and $E(\mathbf{x}_i'\epsilon_i/\sigma_i^2) = 0$
  - A3a is not sufficient for the above equation
  - A3 is sufficient for the above equation.
- The FWLS estimator is asymptotically more efficient than OLS if the errors have multiplicative exponential heteroskedasticity.
  - If the errors are truly multiplicative exponential heteroskedasticity, then usual standard errors are valid
  - If we believe that there may be some mis-specification with the **multiplicative exponential model**, then we should report heteroskedastic robust standard errors.

5.4 Generalized Least Squares

Consider

$$\mathbf{y} = \mathbf{X} \ +$$

where,

$$var(\epsilon) = \mathbf{G} = \begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ . & . & . & . \\ g_{n1} & . & . & g_{nn} \end{pmatrix}$$

The variances are heterogeneous, and the errors are correlated.

$$\hat{\mathbf{b}}_{\mathbf{G}} = (\mathbf{X}'\mathbf{G}^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{G}^{-1}\mathbf{Y}$$

if we know G, we can estimate **b** just like OLS. However, we do not know G. Hence, we model the structure of G.

5.5 Feasiable Prais Winsten

Weighting Matrix

$$\mathbf{w} = \begin{pmatrix} \sqrt{1-\hat{\rho}^2} & 0 & 0 & ... & 0 \\ -\hat{\rho} & 1 & 0 & ... & 0 \\ 0 & -\hat{\rho} & 1 & & . \\ . & . & . & . & 0 \\ 0 & . & 0 & -\hat{\rho} & 1 \end{pmatrix}$$

1. Estimate the following equation using OLS

$$y_t = \mathbf{x}_t\beta + \epsilon_t$$

and obtain the residuals $e_t = y_t - \mathbf{x}_t\hat{\beta}$

2. Estimate the correlation coefficient for the AR(1) process by estimating the following by OLS (without no intercept)

$$e_t = \rho e_{t-1} + u_t$$

3. Transform the outcome and independent variables **wy** and **wX** respectively (weight matrix as stated).
4. The FPW estimator is obtained as the least squared estimated for the following weighted equation

$$\mathbf{wy} = \mathbf{wX} \;\; + \mathbf{w}$$

Properties of Feasiable Prais Winsten Estimator

- The Infeasible PW estimator is under A1-A3 for the unweighted equation

- The FPW estimator is biased
- The FPW is consistent under A1 A2 A5 and

$$E((\mathbf{x_t} - \ \mathbf{x_{t-1}})')(\epsilon_t - \rho\epsilon_{t-1}) = 0$$

+ A3a is not sufficient for the above equation + A3 is sufficient for the above equation

- The FPW estimator is asymptotically more efficient than OLS if the errors are truly generated as AR(1) process
  - If the errors are truly generated as AR(1) process then usual standard errors are valid
  - If we are concerned that there may be a more complex dependence structure of heteroskedasticity, then we use Newey West Standard Errors

5.6 Feasible group level Random Effects

1. Estimate the following equation using OLS

$$y_{gi} = \mathbf{x}_{gi}\beta + \epsilon_{gi}$$

and obtain the residuals $e_{gi} = y_{gi} - \mathbf{x}_{gi}\hat{\beta}$ 2. Estimate the variance using the usual $s^2 estimator

$$s^2 = \frac{1}{n-k}\sum_{i=1}^{n} e_i^2$$

as an estimator for $\sigma_c^2 + \sigma_u^2$ and estimate the within group correlation,

$$\hat{\sigma}_c^2 = \frac{1}{G}\sum_{g=1}^{G}(\frac{1}{\sum_{i=1}^{n_g-1} i}\sum_{i \neq j}\sum_{j}^{n_g} e_{gi}e_{gj})$$

and plug in the estimates to obtain $\hat{\Omega}$

3. The feasible group level RE estimator is obtained as

$$\hat{\beta} = (\mathbf{X\hat{}\ ^{-1}X)^{-1}X\hat{}\ ^{-1}y}$$

Properties of the Feasible group level Random Effects Estimator

- The infeasible group RE estimator is a linear estimator and is unbiased under A1-A3 for the unweighted equation

  - A3 requires $E(\epsilon_{gi}|\mathbf{x}_i) = E(c_g|\mathbf{x}_i) + (u_{gi}|\mathbf{x}_i) = 0$ so we generally assume $E(c_g|\mathbf{x}_i) + (u_{gi}|\mathbf{x}_i) = 0$. The assumption $E(c_g|\mathbf{x}_i) = 0$ is generally called **random effects assumption**
- The Feasible group level Random Effects is biased
- The Feasible group level Random Effects is consistent under A1-A3a, and A5a for the unweighted equation.
  - A3a requires $E(\mathbf{x}_i'\epsilon_{gi}) = E(\mathbf{x}_i'c_g) + (\mathbf{x}_i'u_{gi}) = 0$
- The Feasible group level Random Effects estimator is asymptotically more efficient than OLS if the errors follow the random effects specification
  - If the errors do follow the random effects specification than the usual standard errors are consistent
  - If there might be a more complex dependence structure or heteroskedasticity, then we need cluster robust standard errors.

5.7 Ridge Regression

When we have the Collinearity problem, we could use the Ridge regression.

The main problem with multicollinearity is that $\mathbf{X}'\mathbf{X}$ is "ill-conditioned". The idea for ridge regression: adding a constant to the diagonal of $\mathbf{X}'\mathbf{X}$ improves the conditioning

$$\mathbf{X}'\mathbf{X} + c\mathbf{I}(c > 0)$$

The choice of c is hard. The estimator

$$\mathbf{b}^R = (\mathbf{X}'\mathbf{X} + c\mathbf{I})^{-1}\mathbf{X}'\mathbf{y}$$

is **biased**.

- it has smaller variance than the OLS estimator; as c increases, the bias increases but the variance decreases.

- always exists some value of c for which the ridge regression estimator has a smaller total MSE than the OLS

- the optimal c varies with application and data set.

- to find the "optimal" c we could use "ridge trace".

we plot the values of the p - 1 parameter estimates for different values of c, simultaneously.

- typically, as c increases toward 1 the coefficients decreases to 0.

- the values of the VIF tend to decrease rapidly as c gets bigger than 0. The VIF values begin to change slowly as c approaches 1.
- then we can examine the ridge trace and VIF values and chooses the smallest value of c where the regression coefficients first become stable in the ridge trace and the VIF values have become sufficiently small (which is very subjective).
- typically, this procedure is applied to the standardized regression model.

5.8 Principal Component Regression

This also addresses the problem of multicollinearity

5.9 Robust Regression

- To address the problem of influential cases.
- can be used when a known functional form is to be fitted, and when the errors are not normal due to a few outlying cases.

5.9.1 Least Absolute Residuals (LAR) Regression

also known as minimum L_1-norm regression.

$$L_1 = \sum_{i=1}^{n} |Y_i - (\beta_0 + \beta_1 X_{i1} + .. + \beta_{p-1} X_{i,p-1})$$

which is not sensitive to outliers and inadequacies of the model specification.

5.9.2 Least Median of Squares (LMS) Regression

$$median\{[Y_i - (\beta_0 - \beta_1 X_{i1} + ... + \beta_{p-1} X_{i,p-1})]^2\}$$

5.9.3 Iteratively Reweighted Least Squares (IRLS) Robust Regression

- uses Weighted Least Squares to lessen the influence of outliers.
- the weights w_i are inversely proportional to how far an outlying case is (e.g., based on the residual)
- the weights are revised iteratively until a robust fit

Process:

Step 1: Choose a weight function for weighting the cases.
Step 2: obtain starting weights for all cases.
Step 3: Use the starting weights in WLS and obtain the residuals from the fitted regression function.
Step 4: use the residuals in Step 3 to obtain revised weights.
Step 5: continue until convergence.

Note:

- If you don't know the form of the regression function, consider using non-parametric regression (e.g., locally weighted regression, regression trees, projection pursuit, neural networks, smoothing splines, loess, wavelets).
- could use to detect outliers or confirm OLS.

5.10 Maximum Likelihood

Premise: find values of the parameters that maximize the probability of observing the data In other words, we try to maximize the value of theta in the likelihood function

$$L(\theta) = \prod_{i=1}^{n} f(y_i|\theta)$$

$f(y|\theta)$ is the probability density of observing a single value of Y given some value of θ $f(y|\theta)$ can be specify as various type of distributions. You can review back section Distributions. For example If y is a dichotomous variable, then

$$L(\theta) = \prod_{i=1}^{n} \theta^{y_i}(1-\theta)^{1-y_i}$$

$\hat{\theta}$ is the Maximum Likelihood estimate if $L(\hat{\theta}) > L(\theta_0)$ for all values of θ_0 in the parameter space.

5.10.1 Motivation for MLE

Suppose we know the conditional distribution of y given x:

$$f_{Y|X}(y,x;\theta)$$

where θ is the unknown parameter of distribution. Sometimes we are only concerned with the unconditional distribution $f_Y(y;\theta)$

Then given a sample of iid data, we can calculate the joint distribution of the entire sample,

$$f_{Y_1,\ldots,Y_n|X_1,\ldots,X_n(y_1,\ldots y_n,x_1,\ldots,x_n;\theta)} = \prod_{i=1}^{n} f_{Y|X}(y_i, x_i; \theta)$$

The joint distribution evaluated at the sample is the likelihood (probability) that we observed this particular sample (depends on θ)

Idea for MLE: Given a sample, we choose our estimates of the parameters that gives the highest likelihood (probability) of observing our particular sample

$$max_\theta \prod_{i=1}^{n} f_{Y|X}(y_i, x_i; \theta)$$

Equivalently,

$$max_\theta \prod_{i=1}^{n} ln(f_{Y|X}(y_i, x_i; \theta))$$

Solving for the Maximum Likelihood Estimator

1. Solve First Order Condition

$$\frac{\partial}{\partial \theta} \sum_{i=1}^{n} ln(f_{Y|X}(y_i, x_i; \hat{\theta}_{MLE})) = 0$$

where $\hat{\theta}_{MLE}$ is defined.

2. Evaluate Second Order Condition

$$\frac{\partial^2}{\partial \theta^2} \sum_{i=1}^{n} ln(f_{Y|X}(y_i, x_i; \hat{\theta}_{MLE})) < 0$$

where the above condition ensures we can solve for a maximum

Examples:
Unconditional Poisson Distribution: Number of products ordered on Amazon within an hour, number of website visits a day for a political campaign.

Exponential Distribution: Length of time until an earthquake occurs, length of time a car battery lasts.

$$f_{Y|X}(y, x; \theta) = exp(-y/x\theta)/x\theta f_{Y_1,..Y_n|X_1,\ldots,X_n(y_1,\ldots,y_n,x_1,\ldots,x_n;\theta)} = \prod_{i=1}^{n} exp(-y_i/x_i\theta)/x_i\theta$$

5.10.2 Assumption

- **High Level Regulatory Assumptions** is the sufficient condition used to show large sample properties
  - Hence, for each MLE, we will need to either assume or verify if the regulatory assumptiosn holds.
- observations are independent and have the same density function.
- Under multivariate normal assumption, ML yields consistent estimates of the means and the covariance matrix for multivariate distribution with finite fourth moments (Little and Smith, 1987)

To find the MLE, we usually differentiate the **log-likelihood** function and set it equal to 0.

$$\frac{d}{d\theta} l(\theta) = 0$$

This is the **score** equation

Our confidence in the MLE is quantified by the "pointedness" of the log-likelihood

$$I_O(\theta) = \frac{d^2}{d\theta^2} l(\theta) = 0$$

called the **observed information**

while

$$I(\theta) = E[I_O(\theta; Y)]$$

is the expected information. (also known as Fisher Information). which we base our variance of the estimator.

$$V(\hat{\Theta}) \approx I(\theta)^{-1}$$

Consistency of MLE
Suppose that y_i and x_i are iid drawn from the true conditional pdf $f_{Y|X}(y_i, x_i; \theta_0)$. If the following regulatory assumptions hold,

R1: If $\theta \neq \theta_0$ then $f_{Y|X}(y_i, x_i; \theta) \neq f_{Y|X}(y_i, x_i; \theta_0)$
R2: The set Θ that contains the true parameters θ_0 is compact
R3: The log-likelihood $ln(f_{Y|X}(y_i, x_i; \theta_0))$ is continuous at each θ with probability 1
R4: $E(sup_{\theta \in \Theta} |ln(f_{Y|X}(y_i, x_i; \theta_0))|)$

then the MLE estimator is consistent,

$$\hat{\theta}_{MLE} \to^p \theta_0$$

Asymptotic Normality of MLE

Suppose that y_1 and x_i are iid drawn from the true conditional pdf $f_{Y|X}(y_i, x_i; \theta)$. If R1-R4 and the following hold

R5: θ_0 is in the interior of the set Θ
R6: $f_{Y|X}(y_i, x_i; \theta)$ is twice continuously differentiable in θ and $f_{Y|X}(y_i, x_i; \theta) > 0$ for a neighborhood $N \in \Theta$ around θ_0
R7: $\int sup_{\theta \in N} ||\partial f_{Y|X}(y_i, x_i; \theta) \partial \theta|| d(y,x) < \infty$, $\int sup_{\theta \in N} ||\partial^2 f_{Y|X}(y_i, x_i; \theta)/\partial\theta\partial\theta'|| d(y,x) < \infty$ and $E(sup_{\theta \in N} ||\partial^2 ln(f_{Y|X}(y_i, x_i; \theta))/\partial\theta\partial\theta'||) < \infty$
R8: The information matrix $I(\theta_0) = Var(\partial f_{Y|X}(y, x_i; \theta_0)/\partial\theta)$ exists and is non-singular

then the MLE estimator is asymptotically normal,

$$\sqrt{n}(\hat{\theta}_{MLE} - \theta_0) \to^d N(0, I(\theta_0)^{-1})$$

5.10.3 Properties

(EJD et al., 1998)

(1) Consistent: estimates are approximately unbiased in large samples
(2) Asymptotically efficient: approximately smaller standard errors compared to other estimator
(3) Asymptotically normal: with repeated sampling, the estimates will have an approximately normal distribution. Suppose that $\hat{\theta}_n$ is the MLE for θ based on n independent observations. then $\hat{\theta}_n \sim N(\theta, H^{-1})$.
+ where H is called the Fisher information matrix. It contains the expected values of the second partial derivatives of the log-likelihood function. The (i.j)th element of H is $-E(\frac{\partial^2 l(\theta)}{\partial \theta_i \partial \theta_j})$
+ We can estimate H by finding the form determined above, and evaluating it at $\theta = \hat{\theta}_n$
(4) Invariance: MLE for $g(\theta) = g(\hat{\theta})$ for any function g(.)

$$\hat{\Theta} \approx^d (\theta, \widehat{I(\theta)^{-1}}))$$

Explicit vs Implicit MLE

- If we solve the score equation to get an expression of MLE, then it's called **explicit**
- If there is no closed form for MLE, and we need some algorithms to derive its expression, it's called **implicit**

Large Sample Property of MLE

Implicit in these theorems is the assumption that we know what the conditional distribution,

$$f_{Y|X}(y_i, x_i; \theta_0)$$

but just do now know the exact parameter value.

- Any Distributional mis-specification will result in inconsistent parameter estimates.

- Quasi-MLE: Particular settings/ assumption that allow for certain types of distributional mis-specification (Ex: as long as the distribution is part of particular class or satisfies a particular assumption, then estimating with a wrong distribution will not lead to inconsistent parameter estimates).

- non-parametric/ Semi-parametric estimation: no or very little distributional assumption are made. (hard to implement, derive properties, and interpret)

The asymptotic variance of the MLE achieves the **Cramer-Rao Lower Bound**

- The **Cramer-Rao Lower Bound** is a lower brand for the asymptotic variance of a consistent and asymptotically normally distributed estimator.

- If an estimator achieves the lower bound then it is the most efficient estimator.

The maximum Likelihood estimator (assuming the distribution is correctly specified and R1-R8 hold) is the most efficient consistent and asymptotically normal estimator.
* most efficient among ALL consistent estimators (not limited to unbiased or linear estimators).

Note

- ML is better choice for binary, strictly positive, count, or inherent heteroskedasticity than linear model.
- ML will assume that we know the conditional distribution of the outcome, and derive an estimator using that information.
  - Adds an assumption that we know the distribution (which is similar to A6 Normal Distribution in linear model)
  - will produce a more efficient estimator.

5.10.4 Compare to OLS

MLE is not a cure for most of OLS problems:

- To do joint inference in MLE, we typically use log-likelihood calculation, instead of F-score

- Functional form affects estimation of MLE and OLS.

- Perfect Collinearity/Multicollinearity: highly correlated are likely to yield large standard errors.

- Endogeneity (Omitted variables bias, Simultaneous equations): Like OLS, MLE is also biased against this problem

5.10.5 Application

Other applications of MLE

- Corner Solution
  - Ex: hours worked, donations to charity

  - Estimate with Tobit

- Non-negative count
  - Ex: Numbers of arrest, Number of cigarettes smoked a day

  - Estimate with Poisson regression

- Multinomial Choice
  - Ex: Demand for cars, votes for primary election
  - Estimate with mutinomial probit or logit

- Ordinal Choice
  - Ex: Levels of Happiness, Levels of Income

  - Ordered Probit

Model for binary Response
A binary variable will have a Bernoulli distribution:

$$f_Y(y_i;p) = p^{y_i}(1-p)^{(1-y_i)}$$

where p is the probability of success. The conditional distribution is:

$$f_{Y|X}(y_i, x_i; p(.)) = p(x_i)^{y_i}(1-p(x_i))^{(1-y_i)}$$

So choose $p(x_i)$ to be a reasonable function of x_i and unknown parameters θ

We can use **latent variable model** as probability functions

$$y_i = 1\{y_i^* > 0\} y_i^* = x_i\beta - \epsilon_i$$

- y_i^* is a latent variable (unobserved) that is not well-defined in terms of units/magnitudes

- ϵ_i is a mean 0 unobserved random variable.

We can rewrite the mdoel without the latent variable,

$$y_i = 1\{x_i beta > \epsilon_i\}$$

Then the probability function,

$$p(x_i) = P(y_i = 1|x_i) = P(x_i\beta > \epsilon_i|x_i) = F_{\epsilon|X}(x_i\beta|x_i)$$

then we need to choose a conditional distribution for ϵ_i. Hence, we can make additional strong independence assumption

ϵ_i is independent of x_i

Then the probability function is simply,

$$p(x_i) = F_\epsilon(x_i\beta)$$

The probability function is also the conditional expectation function,

$$E(y_i|x_i) = P(y_i = 1|x_i) = F_\epsilon(x_i\beta)$$

so we allow the conditional expectation function to be non-linear.

Common distributional assumption

1. **Probit**: Assume ϵ_i is standard normally distributed, then $F_\epsilon(.) = \Phi(.)$ is the standard normal CDF.

2. **Logit**: Assume ϵ_i is standard logistically distributed, then $F_\epsilon(.) = \Lambda(.)$ is the standard normal CDF.

Step to derive

1. Choose a distribution (normal or logistic) and plug into the following log likelihood,

$$ln(f_{Y|X}(y_i, x_i; \beta)) = y_i ln(F_\epsilon(x_i\beta)) + (1 - y_i) ln(1 - F_\epsilon(x_i\beta))$$

2. Solve the MLE by finding the Maximum of

$$\hat{\beta}_{MLE} = argmax \sum_{i=1}^{n} ln(f_{Y|X}(y_i, x_i; \beta))$$

Properties of the Probit and Logit Estimators

- Probit or Logit is consistent and asymptotically normal if

  - [A2][] holds: $E(x_i'x_i)$ exists and is non-singular
  - [A5][] (or A5a) holds: {y_i,x_i} are iid (or stationary and weakly dependent).
  - Distributional assumptions on ϵ_i hold: Normal/Logistic and independent of x_i
- Under the same assumptions, Probit or Logit is also asymptotically efficient with asymptotic variance,

$$I(\beta_0)^{-1} = [E(\frac{(f_\epsilon(x_i\beta_0))^2}{F_\epsilon(x_i\beta_0)(1-F_\epsilon(x_i\beta_0))}x_i'x_i)]^{-1}$$

where $F_\epsilon(x_i\beta_0)$ is the probability density function (derivative of the CDF)

5.10.5.1 Interpretation

β is the average response in the latent variable associated with a change in x_i

- Magnitudes do not have meaning
- Direction does have meaning

The **partial effect** for a Non-linear binary response model

$$E(y_i|x_i) = F_\epsilon(x_i\beta)PE(x_{ij}) = \frac{\partial E(y_i|x_i)}{\partial x_{ij}} = f_\epsilon(x_i\beta)\beta_j$$

- The partial effect is the coefficient parameter β_j multiplied by a scaling factor $f_\epsilon(x_i\beta)$
- The scaling factor depends on x_i so the partial effect changes depending on what x_i is

Single value for the partial effect

- **Partial Effect at the Average (PEA)** is the partial effect for an average individual

$$f_\epsilon(\bar{x}\hat{\beta})\hat{\beta}_j$$

- **Average Partial Effect (APE)** is the average of all partial effect for each individual.

$$\frac{1}{n}\sum_{i=1}^{n} f_\epsilon(x_i\hat{\beta})\hat{\beta}_j$$

In the linear model, APE = PEA.
In a non-linear model (e.g., binary response), APE $\neq$ PEA

Chapter 6

Non-linear Regression

Definition: models in which the derivatives of the mean function with respect to the parameters depend on one or more of the parameters.

To approximate data, we can approximate the function

- by a high-order polynomial
- by a linear model (e.g., a Taylor expansion around X's)
- a collection of locally linear models or basis function

but it would not easy to interpret, or not enough data, or can't interpret them globally.

intrinsically nonlinear models:

$$Y_i = f(\mathbf{x_i}; \ ) + \epsilon_i$$

where $f(\mathbf{x_i}; \ )$ is a nonlinear function relating $E(Y_i)$ to the independent variables x_i

- $\mathbf{x}_i$ is a k x 1 vector of independent variables (fixed).
- is a p x 1 vector of parameters.
- ϵ_is are iid variables mean 0 and variance σ^2. (sometimes it's normal).

6.1 Inference

Since $Y_i = f(\mathbf{x}_i, \theta) + \epsilon_i$, where $\epsilon_i \sim iid(0, \sigma^2)$. We can obtain $\hat{\theta}$ by minimizing $\sum_{i=1}^{n}(Y_i - f(x_i, \theta))^2$ and estimate $s^2 = \hat{\sigma}^2_\epsilon = \frac{\sum_{i=1}^{n}(Y_i - f(x_i, \theta))^2}{n-p}$

6.1.1 Linear Function of the Parameters

If we assume $\epsilon_i \sim N(0, \sigma^2)$, then

$$\hat{\theta} \sim AN(\ ,\sigma^2[\mathbf{F}(\theta)'\mathbf{F}(\theta)]^{-1})$$

where An = asymptotic normality

Asymptotic means we have enough data to make inference (As your sample size increases, this becomes more and more accurate (to the true value)).

Since we want to do inference on linear combinations of parameters or contrasts.

If we have $\ = (\theta_0, \theta_1, \theta_2)'$ and we want to look at $\theta_1 - \theta_2$; we can define vector $\mathbf{a} = (0, 1, -1)'$, consider inference for $\mathbf{a}'$

Rules for expectation and variance of a fixed vector $\mathbf{a}$ and random vector $\mathbf{Z}$;

$$E(\mathbf{a}'\mathbf{Z}) = \mathbf{a}'E(\mathbf{Z})var(\mathbf{a}'\mathbf{Z}) = \mathbf{a}'var(\mathbf{Z})\mathbf{a}$$

Then,

$$\mathbf{a}\hat{\ } \sim AN(\mathbf{a}'\ ,\sigma^2\mathbf{a}'[\mathbf{F}(\ )'\mathbf{F}(\ )]^{-\mathbf{1}}\mathbf{a})$$

and $\mathbf{a}\hat{\ }$ is asymptotically independent of s^2 (to order 1/n) then

$$\frac{\mathbf{a}\hat{\ } - \mathbf{a}'}{s(\mathbf{a}'[\mathbf{F}(\ )'\mathbf{F}(\ )]^{-\mathbf{1}}\mathbf{a})^{1/2}} \sim t_{n-p}$$

to construct $100(1-\alpha)\%$ confidence interval for $\mathbf{a}'$

$$\mathbf{a}'\ \pm t_{(1-\alpha/2,n-p)}s(\mathbf{a}'[\mathbf{F}(\ )'\mathbf{F}(\ )]^{-\mathbf{1}}\mathbf{a})^{1/2}$$

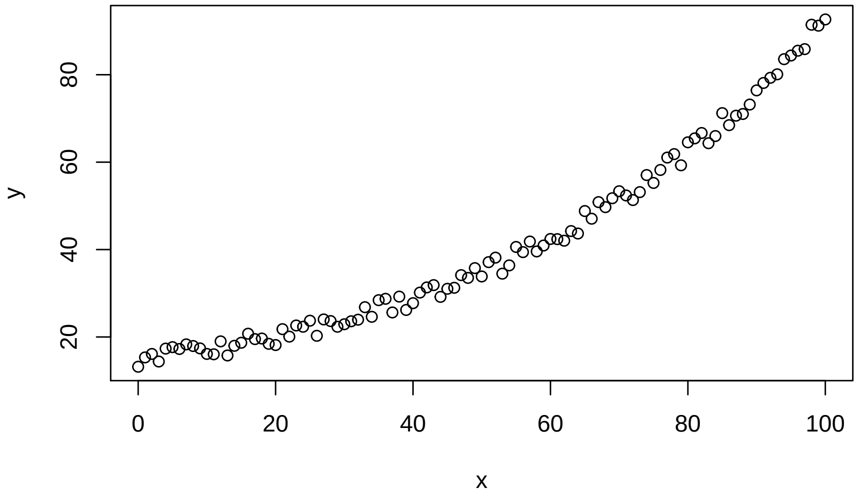
Suppose $\mathbf{a}' = (0, ..., j, ..., 0)$. Then, a confidence interval for the jth element of is

$$\hat{\theta}_j \pm t_{(1-\alpha/2,n-p)}s\sqrt{\hat{c}^j}$$

where $\hat{c}^j$ is the jth diagonal element of $[\mathbf{F}(\hat{\ })'\mathbf{F}(\hat{\ })]^{-1}$

```
#set a seed value
set.seed(23)

#Generate x as 100 integers using seq function
x<-seq(0,100,1)

#Generate y as a*e^(bx)+c
y<-runif(1,0,20)*exp(runif(1,0.005,0.075)*x)+runif(101,0,5)
```

```
# visualize
plot(x,y)
```



```
#define our data frame
datf = data.frame(x,y)

#define our model function
mod =function(a,b,x) a*exp(b*x)
```

In this example, we can get the starting values by using linearized version of the function $\log y = \log a + bx$. Then, we can fit a linear regression to this and use our estimates as starting values

```
#get starting values by linearizing
lin_mod=lm(log(y)~x,data=datf)

#convert the a parameter back from the log scale; b is ok
astrt = exp(as.numeric(lin_mod$coef[1]))
bstrt = as.numeric(lin_mod$coef[2])
print(c(astrt,bstrt))
```

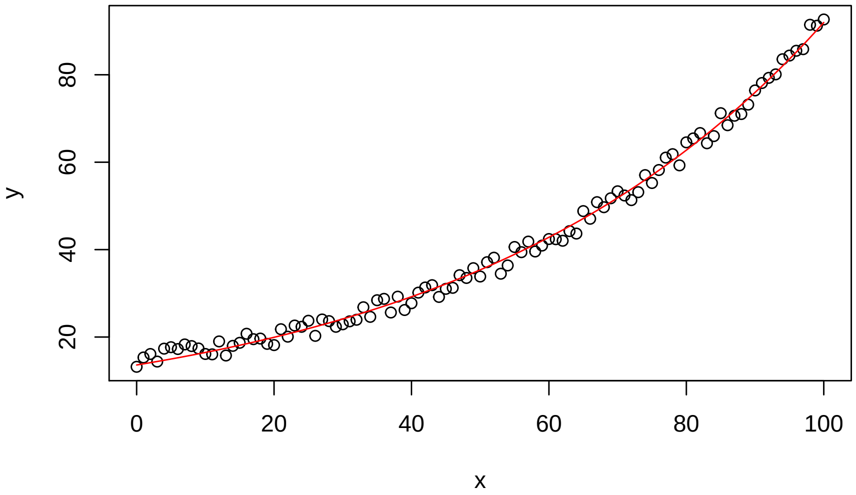
```
## [1] 14.07964761  0.01855635
```

with `nls`, we can fit the nonlinear model via least squares

```
nlin_mod = nls(y ~ mod(a, b, x),
               start = list(a = astrt, b = bstrt),
                data = datf)

#look at model fit summary
summary(nlin_mod)
```

```
##
## Formula: y ~ mod(a, b, x)
##
## Parameters:
##    Estimate Std. Error t value Pr(>|t|)
## a 13.603909   0.165390   82.25   <2e-16 ***
## b  0.019110   0.000153  124.90   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.542 on 99 degrees of freedom
##
## Number of iterations to convergence: 3
## Achieved convergence tolerance: 7.006e-07
```

```
#add prediction to plot
plot(x, y)
lines(x, predict(nlin_mod), col = "red")
```



6.1.2 Nonlinear

Suppose that $h(\theta)$ is a nonlinear function of the parameters. We can use Taylor series about θ

$$h(\hat{\theta}) \approx h(\theta) + \mathbf{h}'[\hat{\theta} - \theta]$$

where $\mathbf{h} = (\frac{\partial h}{\partial \theta_1}, ..., \frac{\partial h}{\partial \theta_p})'$

with

$$E(\hat{\theta}) \approx \theta var(\hat{\theta}) \approx \sigma^2[\mathbf{F}(\ )'\mathbf{F}(\ )]^{-1}E(h(\hat{\theta})) \approx h(\theta)var(h(\hat{\theta})) \approx \sigma^2\mathbf{h'}[\mathbf{F}(\ )'\mathbf{F}(\ )]^{\mathbf{-1}}\mathbf{h}$$

Thus,

$$h(\hat{\theta}) \sim AN(h(\theta), \sigma^2\mathbf{h'}[\mathbf{F}(\ )'\mathbf{F}(\ )]^{\mathbf{-1}}\mathbf{h})$$

and an approximate $100(1-\alpha)\%$ confidence interval for $h(\theta)$ is

$$h(\hat{\theta}) \pm t_{(1-\alpha/2;n-p)}s(\mathbf{h'}[\mathbf{F}(\ )'\mathbf{F}(\ )]^{\mathbf{-1}}\mathbf{h})^{1/2}$$

where $\mathbf{h}$ and $\mathbf{F}(\theta)$ are evaluated at $\hat{\theta}$

Regarding **prediction interval** for Y at $x = x_0$

$$Y_0 = f(x_0;\theta) + \epsilon_0, \epsilon_0 \sim N(0,\sigma^2)\hat{Y}_0 = f(x_0,\hat{\theta})$$

As $n \to \infty$, $\hat{\theta} \to \theta$, so we

$$f(x_0,\hat{\theta}) \approx f(x_0,\theta) + \mathbf{f}_0(\ )'[\hat{\theta} - \theta]$$

where

$$f_0(\theta) = (\frac{\partial f(x_0,\theta)}{\partial \theta_1}, .., \frac{\partial f(x_0,\theta)}{\partial \theta_p})'$$

(note: this $f_0(\theta)$ is different from $f(\theta)$).

$$Y_0 - \hat{Y}_0 \approx Y_0 - f(x_0,\theta) - f_0(\theta)'[\hat{\theta} - \theta] = \epsilon_0 - f_0(\theta)'[\hat{\theta} - \theta]$$

$$E(Y_0 - \hat{Y}_0) \approx E(\epsilon_0)E(\hat{\theta} - \theta) = 0var(Y_0 - \hat{Y}_0) \approx var(\epsilon_0 - (\mathbf{f_0}(\ )'\hat{\ } - \ ])) = \sigma^2 + \sigma^2\mathbf{f_0}(\ )'[\mathbf{F}(\ )'\mathbf{F}(\ )]^{\mathbf{-1}}\mathbf{f_0}(\ ) = \sigma^2(1 + \mathbf{f_0}(\ )'[\mathbf{F}($$

Hence, combining

$$Y_0 - \hat{Y}_0 \sim AN(0, \sigma^2(1 + \mathbf{f_0}(\ )'[\mathbf{F}(\ )'\mathbf{F}(\ )]^{-\mathbf{1}}\mathbf{f_0}(\ )))$$

Note:

Confidence intervals for the mean response Y_i (which is different from prediction intervals) can be obtained similarly.

6.2 Non-linear Least Squares

- The LS estimate of θ, $\hat{\theta}$ is the set of parameters that minimizes the residual sum of squares:

$$S(\hat{\theta}) = SSE(\hat{\theta}) = \sum_{i=1}^{n}\{Y_i - f(\mathbf{x_i}; \hat{\theta})\}^2$$

- to obtain the solution, we can consider the partial derivatives of $S(\theta)$ with respect to each θ_j and set them to 0, which gives a system of p equations. Each normal equation is

$$\frac{\partial S(\theta)}{\partial \theta_j} = -2\sum_{i=1}^{n}\{Y_i - f(\mathbf{x}_i; \theta)\}[\frac{\partial(\mathbf{x}_i; \theta)}{\partial \theta_j}] = 0$$

- but we can't obtain a solution directly/analytically for this equation.

Numerical Solutions

- Grid search
  - A "grid" of possible parameter values and see which one minimize the residual sum of squares.
  - finer grid = greater accuracy
  - could be inefficient, and hard when p is large.
- Gauss-Newton Algorithm
  - we have an initial estimate of θ denoted as $\hat{\theta}^{(0)}$
  - use a Taylor expansions of $f(\mathbf{x}_i; \theta)$ as a function of θ about the point $\hat{\theta}^{(0)}$

$$\begin{aligned} Y_i &= f(x_i; \theta) + \epsilon_i \\ &= f(x_i; \theta) + \sum_{j=1}^{p}\{\frac{\partial f(x_i; \theta)}{\partial \theta_j}\}_{\theta = \hat{\theta}^{(0)}}(\theta_j - \hat{\theta}^{(0)}) + \text{remainder} + \epsilon_i \end{aligned}$$

Equivalently,

In matrix notation,

$$\mathbf{Y} = \left[\begin{array}{c} Y_1 \\ . \\ Y_n \end{array}\right]$$

$$\mathbf{f}(\hat{\theta}^{(0)}) = \left[\begin{array}{c} f(\mathbf{x_1}, \hat{}^{(0)}) \\ . \\ f(\mathbf{x_n}, \hat{}^{(0)}) \end{array}\right]$$

$$= \left[\begin{array}{c} \epsilon_1 \\ . \\ \epsilon_n \end{array}\right]$$

$$\mathbf{F}(\hat{\theta}^{(0)}) = \left[\begin{array}{ccc} \frac{\partial f(x_1,)}{\partial \theta_1} & \dots & \frac{\partial f(x_1,)}{\partial \theta_p} \\ . & . & . \\ \frac{\partial f(x_n,)}{\partial \theta_1} & \dots & \frac{\partial f(x_n,)}{\partial \theta_p} \end{array}\right]_{\theta = \hat{\theta}^{(0)}}$$

Hence,

$$\mathbf{Y} = \mathbf{f}(\hat{\theta}^{(0)}) + \mathbf{F}(\hat{\theta}^{(0)})(\theta - \hat{\theta}^{(0)}) + \epsilon + \text{remainder}$$

where we assume that the remainder is small and the error term is only assumed to be iid with mean 0 and variance σ^2.

We can rewrite the above equation as

$$\mathbf{Y} - \mathbf{f}(\hat{\theta}^{(0)}) \approx \mathbf{F}(\hat{\theta}^{(0)})(\theta - \hat{\theta}^{(0)}) + \epsilon$$

where it is in the form of linear model. After we solve for $(\theta - \hat{\theta}^{(0)})$ and let it equal to $\hat{\delta}^{(1)}$
Then we new estimate is given by adding the Gauss increment adjustment to the initial estimate $\hat{\theta}^{(1)} = \hat{\theta}^{(0)} + \hat{\delta}^{(1)}$
We can repeat this process.

Gauss-Newton Algorithm Steps:

1. initial estimate $\hat{\theta}^{(0)}$, set j = 0
2. Taylor series expansion and calculate $\mathbf{f}(\hat{\theta}^{(j)})$ and $\mathbf{F}(\hat{\theta}^{(j)})$
3. Use OLS to get $\hat{\delta}^{(j+1)}$
4. get the new estimate $\hat{\theta}^{(j+1)}$, return to step 2
5. continue until "convergence"
6. With the final parameter estimate $\hat{\theta}$, we can estimate σ^2 if $\epsilon \sim (\mathbf{0}, \sigma^2 \mathbf{I})$ by

$$\hat{\sigma}^2 = \frac{1}{n-p}(\mathbf{Y} - \mathbf{f}(x;\hat{\theta}))'(\mathbf{Y} - \mathbf{f}(x;\hat{\theta}))$$

Criteria for convergence

1. Minor change in the objective function (SSE = residual sum of squares)

$$\frac{|SSE(\hat{\theta}^{(j+1)}) - SSE(\hat{\theta}^{(j)})|}{SSE(\hat{\theta}^{(j)})} < \gamma_1$$

2. Minor change in the parameter estimates

$$|\hat{\theta}^{(j+1)} - \hat{\theta}^{(j)}| < \gamma_2$$

3. "residual projection" criterion of (Bates and Watts, 1981)

6.2.1 Alternative of Gauss-Newton Algorithm

6.2.1.1 Gauss-Newton Algorithm

Normal equations:

$$\frac{\partial SSE(\theta)}{\partial \theta} = 2\mathbf{F}(\theta)'[\mathbf{Y} - \mathbf{f}(\theta)]$$

$$\begin{aligned}
\hat{\theta}^{(j+1)} &= \hat{\theta}^{(j)} + \hat{\delta}^{(j+1)} \\
&= \hat{\theta}^{(j)} + [\mathbf{F}((\hat{\theta})^{(j)})'\mathbf{F}(\hat{\theta}^{(j)})]^{-1}\mathbf{F}(\hat{\theta})^{(j)} \\
&= \hat{\theta}^{(j)} - \frac{1}{2}[\mathbf{F}(\hat{\theta}^{(j)})'\mathbf{F}(\hat{\theta}^{(j)})]^{-1}\frac{\partial SSE(\hat{\theta}^{(j)})}{\partial \theta}
\end{aligned}$$

where

- $\frac{\partial SSE(\hat{\theta}^{(j)})}{\partial \theta}$ is a gradient vecotr (points in the direction in which the SSE increases most rapidly). This path is known as steepest ascent.

- $[\mathbf{F}(\hat{\theta}^{(j)})'\mathbf{F}(\hat{\theta}^{(j)})]^{-1}$ indicates how far to move

- $-1/2$: indicator of the direction of steepest descent.

6.2.1.2 Modified Gauss-Newton Algorithm

To avoid overstepping (the local min), we can use the modified Gauss-Newton Algorithm. We define a new proposal for θ

$$\hat{\theta}^{(j+1)} = \hat{\theta}^{(j)} + \alpha_j \hat{\delta}^{(j+1)}, 0 < \alpha_j < 1$$

where

- α_j (called the "learning rate"): is used to modify the step length.

We could also have $\alpha * 1/2$, but typically it is assumed to be absorbed into the learning rate.

A way to choose α_j, we can use **step halving**

$$\hat{\theta}^{(j+1)} = \hat{\theta}^{(j)} + \frac{1}{2^k}\hat{\delta}^{(j+1)}$$

where

- k is the smallest non-negative integer such that

$$SSE(\hat{\theta}^{(j)} + \frac{1}{2^k}\hat{\delta}^{(j+1)}) < SSE(\hat{\theta}^{(j)})$$

  which means we try $\hat{\delta}^{(j+1)}$, then $\hat{\delta}^{(j+1)}/2$, $\hat{\delta}^{(j+1)}/4$, etc.

The most general form of the convergence algorithm is

$$\hat{\theta}^{(j+1)} = \hat{\theta}^{(j)} - \alpha_j \mathbf{A}_j \frac{\partial Q(\hat{\theta}^{(j)})}{\partial \theta}$$

where

- $\mathbf{A}_j$ is a positive definite matrix
- α_j is the learning rate
- $\frac{\partial Q(\hat{\theta}^{(j)})}{\partial \theta}$is the gradient based on some objective function Q (a function of θ), which is typically the SSE in nonlinear regression applications (e.g., cross-entropy for classification).

Refer back to the **Modified Gauss-Newton Algorithm**, we can see it is in this form

$$\hat{\theta}^{(j+1)} = \hat{\theta}^{(j)} - \alpha_j [\mathbf{F}(\hat{\theta}^{(j)})' \mathbf{F}(\hat{\theta}^{(j)})]^{-1} \frac{\partial SSE(\hat{\theta}^{(j)})}{\partial \theta}$$

where Q = SSE, $[\mathbf{F}(\hat{\theta}^{(j)})' \mathbf{F}(\hat{\theta}^{(j)})]^{-1} = \mathbf{A}$

6.2.1.3 Steepest Descent

(also known just "gradient descent")

$$\hat{\theta}^{(j+1)} = \hat{\theta}^{(j)} - \alpha_j \mathbf{I}_{p \times p} \frac{\partial \mathbf{Q}(\hat{\theta}^{(j)})}{\partial \theta}$$

- slow to converge, moves rapidly initially.
- could be use for starting values

6.2.1.4 Levenberg -Marquardt

$$\hat{\theta}^{(j+1)} = \hat{\theta}^{(j)} - \alpha_j [\mathbf{F}(\hat{\theta}^{(j)})' \mathbf{F}(\hat{\theta}^{(j)}) + \tau \mathbf{I}_{p \times p}] \frac{\partial \mathbf{Q}(\hat{\theta}^{(j)})}{\partial \theta}$$

which is a compromise between the Gauss-Newton Algorithm and the Steepest Descent.

- best when $\mathbf{F}(\hat{\theta}^{(j)})'\mathbf{F}(\hat{\theta}^{(j)})$ is nearly singular ($\mathbf{F}(\hat{\theta}^{(j)})$ isn't of full rank)
- similar to ridge regression
- If $SSE(\hat{\theta}^{(j+1)}) < SSE(\hat{\theta}^{(j)})$, then $\tau = \tau/10$ for the next iteration. Otherwise, $\tau = 10\tau$

6.2.1.5 Newton-Raphson

$$\hat{\theta}^{(j+1)} = \hat{\theta}^{(j)} - \alpha_j [\frac{\partial^2 Q(\hat{\theta}^{(j)})}{\partial \theta \partial \theta'}]^{-1} \frac{\partial \mathbf{Q}(\hat{\theta}^{(j)})}{\partial \theta}$$

The **Hessian matrix** can be rewritten as:

$$\frac{\partial^2 Q(\hat{\theta}^{(j)})}{\partial \theta \partial \theta'} = 2\mathbf{F}((\hat{\theta})^{(j)})'\mathbf{F}(\hat{\theta}^{(j)}) - 2\sum_{i=1}^{n}[Y_i - f(x_i;\theta)]\frac{\partial^2 f(x_i;\theta)}{\partial \theta \partial \theta'}$$

which contains the same term that Gauss-Newton Algorithm, combined with one containing the second partial derivatives of f(). (methods that require the second derivatives of the objective function are known as “second-order methods”.) However, the last term $\frac{\partial^2 f(x_i;\theta)}{\partial \theta \partial \theta'}$ can sometimes be nonsingular.

6.2.1.6 Quasi-Newton

update θ according to

$$\hat{\theta}^{(j+1)} = \hat{\theta}^{(j)} - \alpha_j \mathbf{H}_j^{-1} \frac{\partial \mathbf{Q}(\hat{\theta}^{(j)})}{\partial \theta}$$

where H_j is a symmetric positive definite approximation to the Hessian, which gets closer as $j \to \infty$.

- $\mathbf{H}_j$ is computed iteratively
- AMong first-order methods(where only first derivatives are required), this method performs best.

6.2.1.7 Derivative Free Methods

- **secant Method**: like Gauss-Newton Algorithm, but calculates the derivatives numerically from past iterations.

- **Simplex Methods**

- **Genetic Algorithm**

- **Differential Evolution Algorithms**

- **Particle Swarm Optimization**

- **Ant Colony Optimization**

6.2.2 Practical Considerations

To converge, algorithm need good initial estimates.

- Starting values:

  - Prior or theoretical info

  - A grid search or a graph of $SSE(\theta)$

  - could also use OLS to get starting values.

  - Model interpretation: if you have some idea regarding the form of the objective function, then you can try to guess the initial value.

  - Expected Value Parameterization

- Constrained Parameters: (constraints on parameters like $\theta_i > a, a < \theta_i < b$)

  - fit the model first to see if the converged parameter estimates satisfy the constraints.
  - if they dont' satisfy, then try re-parameterizing

6.2.2.1 Failure to converge

- $SSE(\theta)$ may be "flat" in a neighborhood of the minimum.

- You can try different or "better" starting values.

- Might suggest the model is too complex for the data, might consider simpler model.

6.2.2.2 Convergence to a Local Minimum

- Linear least squares has the property that $SSE(\theta) = (\mathbf{Y} - \mathbf{X}\ )'(\mathbf{Y} - \mathbf{X}\ )$, which is quadratic and has a unique minimum (or maximum).
- Nonlinear east squares need not have a unique minimum
- Using different starting values can help
- If the dimension of θ is low, graph $SSE(\theta)$ as a function of θ_i
- Different algorithm can help (e.g., genetic algorithm, particle swarm)

To converge, algorithms need good initial estimates.

- Starting values:
  - prior or theoretical info
  - A grid search or a graph
  - OLS estimates as starting values
  - Model interpretation
  - Expected Value Parameterization
- Constrained Parameters:
  - try the model without the constraints first.
  - If the resulted parameter estimates does not satisfy the constraint, try re-parameterizing

```
# Grid search
#choose grid of a and b values
aseq = seq(10,18,.2)
bseq = seq(.001,.075,.001)

na = length(aseq)
nb = length(bseq)
SSout = matrix(0,na*nb,3) #matrix to save output
cnt = 0
for (k in 1:na){
   for (j in 1:nb){
      cnt = cnt+1
      ypred = mod(aseq[k],bseq[j],x) #evaluate model w/ these parms
      ss = sum((y-ypred)^2)  #this is our SSE objective function
      #save values of a, b, and SSE
      SSout[cnt,1]=aseq[k]
      SSout[cnt,2]=bseq[j]
      SSout[cnt,3]=ss
   }
}
#find minimum SSE and associated a,b values
mn_indx = which.min(SSout[,3])
astrt = SSout[mn_indx,1]
bstrt = SSout[mn_indx,2]
```

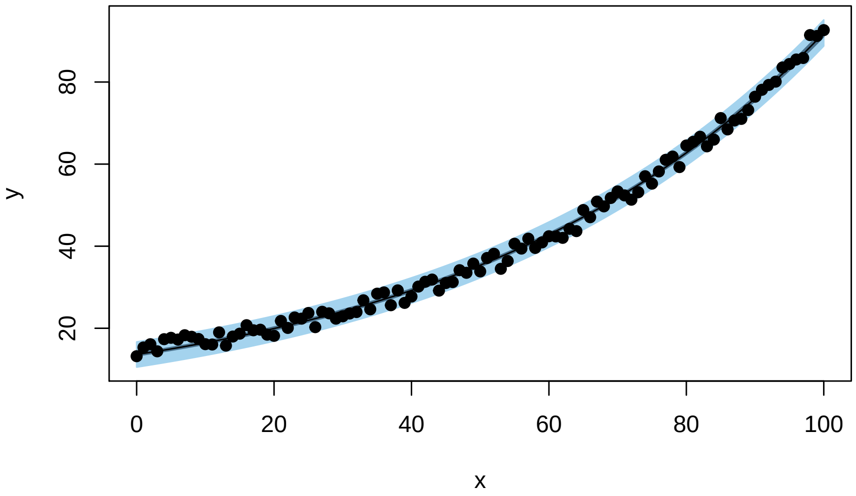
```
#now, run nls function with these starting values
nlin_modG=nls(y~mod(a,b,x),start=list(a=astrt,b=bstrt))

nlin_modG
```

```
## Nonlinear regression model
##   model: y ~ mod(a, b, x)
##    data: parent.frame()
##        a        b
## 13.60391  0.01911
##  residual sum-of-squares: 235.5
##
## Number of iterations to convergence: 3
## Achieved convergence tolerance: 2.293e-07
```

```
# Note, the package `nls_multstart` will allow you to do a grid search without programming your o
```

For prediction interval

```
plotFit(
  nlin_modG,
  interval = "both",
  pch = 19,
  shade = TRUE,
  col.conf = "skyblue4",
  col.pred = "lightskyblue2",
  data = datf
)
```



Based on the forms of your function, you can also have programmed starting values from `nls` function (e.e.g, logistic growth, asymptotic regression, etc).

```
apropos("^SS")
```

```
##  [1] "ss"           "SSasymp"      "SSasympOff"  "SSasympOrig" "SSbiexp"
##  [6] "SSD"          "SSfol"        "SSfpl"       "SSgompertz"  "SSlogis"
## [11] "SSmicmen"     "SSout"        "SSweibull"
```

For example, a logistic growth model:

$$P = \frac{K}{1+exp(P_0 + rt)} + \epsilon$$

where

- P = population at time t
- K = carrying capacity
- r = population growth rate

but in `R` you have slight different parameterization:

$$P = \frac{asym}{1+exp(\frac{xmid-t}{scal})}$$

where

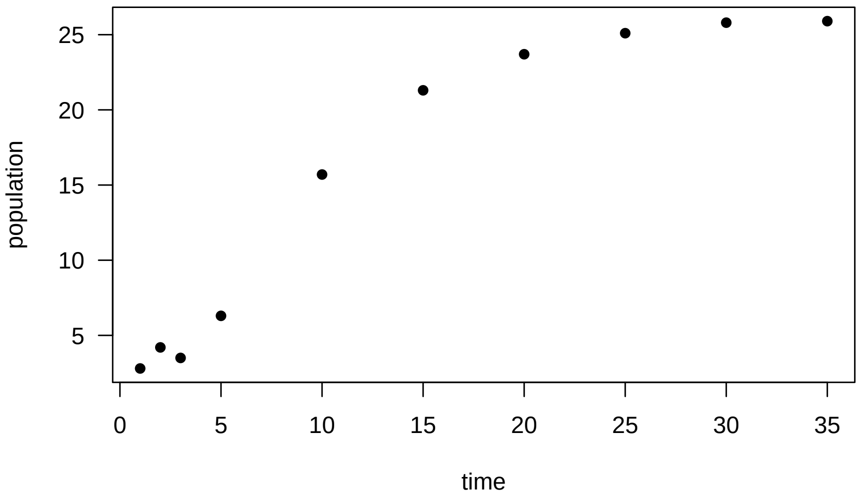
- asym = carrying capacity

- xmid = the x value at the inflection point of the curve
- scal = scaling parameter.

Hence, you have

- K = asym
- r = -1/scal
- $P_0 = -rxmid$

```
# simulated data
time <- c(1, 2, 3, 5, 10, 15, 20, 25, 30, 35)
population <- c(2.8, 4.2, 3.5, 6.3, 15.7, 21.3, 23.7, 25.1, 25.8, 25.9)
plot(time, population, las = 1, pch = 16)
```



```
# model fitting
logisticModelSS <- nls(population ~ SSlogis(time, Asym, xmid, scal))
summary(logisticModelSS)
```

```
##
## Formula: population ~ SSlogis(time, Asym, xmid, scal)
##
## Parameters:
##      Estimate Std. Error t value Pr(>|t|)
## Asym  25.5029     0.3666   69.56 3.34e-11 ***
## xmid   8.7347     0.3007   29.05 1.48e-08 ***
## scal   3.6353     0.2186   16.63 6.96e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6528 on 7 degrees of freedom
##
## Number of iterations to convergence: 1
## Achieved convergence tolerance: 1.908e-06
```

```
coef(logisticModelSS)
```

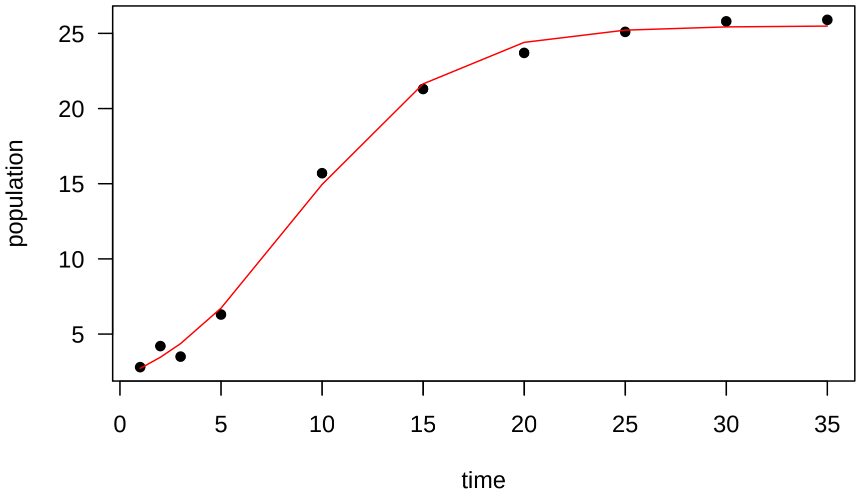
```
##      Asym      xmid      scal
## 25.502890  8.734698  3.635333
```

Other parameterization

```
#convert to other parameterization
Ks = as.numeric(coef(logisticModelSS)[1])
rs = -1/as.numeric(coef(logisticModelSS)[3])
Pos = - rs * as.numeric(coef(logisticModelSS)[2])
#let's refit with these parameters
logisticModel <- nls(population ~ K / (1 + exp(Po + r * time)),start=list(Po=Pos,r=rs,
summary(logisticModel)
```

```
##
## Formula: population ~ K/(1 + exp(Po + r * time))
##
## Parameters:
##    Estimate Std. Error t value Pr(>|t|)
## Po  2.40272    0.12702   18.92 2.87e-07 ***
## r  -0.27508    0.01654  -16.63 6.96e-07 ***
## K  25.50289    0.36665   69.56 3.34e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6528 on 7 degrees of freedom
##
## Number of iterations to convergence: 0
## Achieved convergence tolerance: 1.924e-06
```

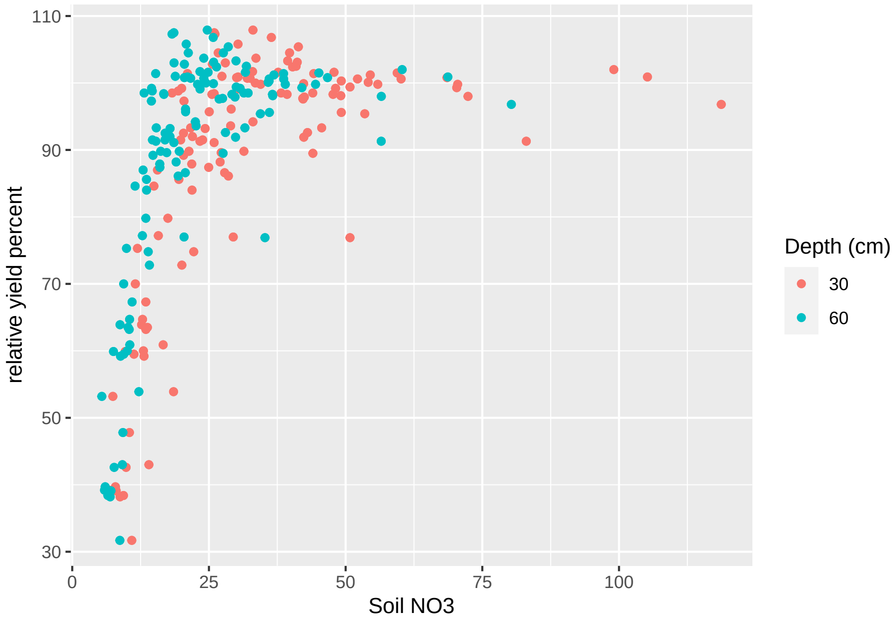
```
#note: initial values =  solution (highly unusual, but ok)
plot(time, population, las = 1, pch = 16)
lines(time, predict(logisticModel), col = "red")
```



If can also define your own self-starting fucntion if your models are uncommon (built in `nls`)

Example is based on (Schabenberger and Pierce, 2001)

```r
#Load data
dat <- read.table("images/dat.txt", header = T)
# plot
dat.plot <-
  ggplot(dat) + geom_point(aes(
    x = no3,
    y = ryp,
    color = as.factor(depth)
  )) +
  labs(color = 'Depth (cm)') + xlab('Soil NO3') + ylab('relative yield percent')
dat.plot
```



The suggested model (known as plateau model) is

$$E(Y_{ij}) = (\beta_{0j} + \beta_{1j}N_{ij})I_{N_{ij}\le\alpha_j} + (\beta_{0j} + \beta_{1j}\alpha_j)I_{N_{ij}>\alpha_j}$$

where

- N is an observation
- i is a particular observation
- j = 1,2 corresponding to depths (30,60)

```
#First define model as a function
nonlinModel <- function(predictor,b0,b1,alpha){
  ifelse(predictor<=alpha,
         b0+b1*predictor, #if observation less than cutoff simple linear model
         b0+b1*alpha) #otherwise flat line
}
```

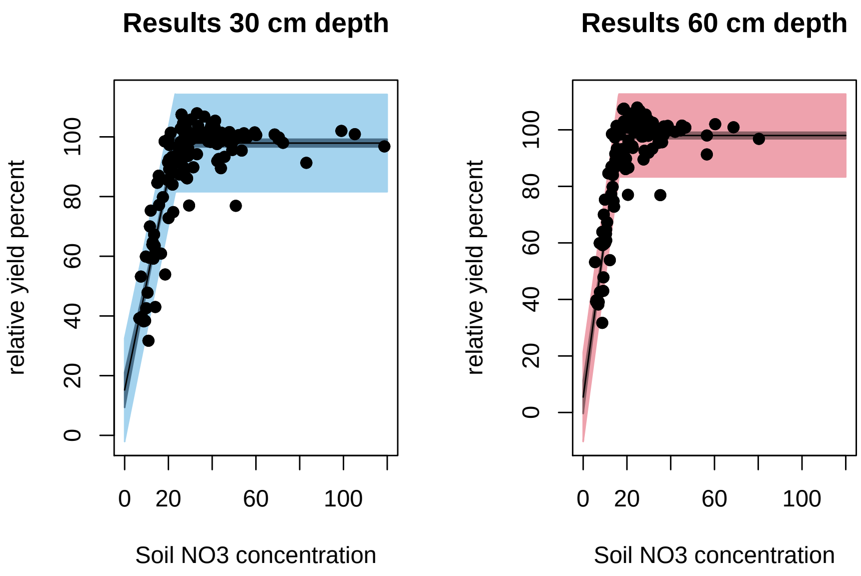
define `selfStart` function. Because we defined our model to be linear in the first part and then plateau (remain constant) we can use the first half of our predictors (sorted by increasing value) to get an initial estimate for the slope and intercept of the model, and the last predictor value (alpha) can be the starting value for the plateau parameter.

```
nonlinModelInit <- function(mCall,LHS,data){
  #sort data by increasing predictor value -
  #done so we can just use the low level no3 conc to fit a simple model
  xy <- sortedXyData(mCall[['predictor']],LHS,data)
  n <- nrow(xy)
  #For the first half of the data a simple linear model is fit
  lmFit <- lm(xy[1:(n/2),'y']~xy[1:(n/2),'x'])
  b0 <- coef(lmFit)[1]
  b1 <- coef(lmFit)[2]
  #for the cut off to the flat part select the last x value used in creating linear model
  alpha <- xy[(n/2),'x']
  value <- c(b0,b1,alpha)
  names(value) <- mCall[c('b0','b1','alpha')]
  value
}
```

combine model and custom function to calculate starting values.

```
SS_nonlinModel <- selfStart(nonlinModel,nonlinModelInit,c('b0','b1','alpha'))
```

```
#Above code defined model and selfStart now just need to call it for each of the depths
sep30_nls <-
  nls(ryp ~ SS_nonlinModel(predictor = no3, b0, b1, alpha), data = dat[dat$depth ==
                                                                      30, ])

sep60_nls <-
  nls(ryp ~ SS_nonlinModel(predictor = no3, b0, b1, alpha), data = dat[dat$depth ==
                                                                      60, ])

par(mfrow = c(1, 2))
plotFit(
  sep30_nls,
  interval = "both",
  pch = 19,
  shade = TRUE,
  col.conf = "skyblue4",
  col.pred = "lightskyblue2",
  data = dat[dat$depth == 30, ],
  main = 'Results 30 cm depth',
  ylab = 'relative yield percent',
  xlab = 'Soil NO3 concentration',
  xlim = c(0, 120)
)
plotFit(
  sep60_nls,
  interval = "both",
  pch = 19,
  shade = TRUE,
  col.conf = "lightpink4",
  col.pred = "lightpink2",
  data = dat[dat$depth == 60, ],
  main = 'Results 60 cm depth',
  ylab = 'relative yield percent',
  xlab = 'Soil NO3 concentration',
  xlim = c(0, 120)
)
```



```
summary(sep30_nls)
```

```
##
## Formula: ryp ~ SS_nonlinModel(predictor = no3, b0, b1, alpha)
##
## Parameters:
##       Estimate Std. Error t value Pr(>|t|)
## b0     15.1943     2.9781   5.102 6.89e-07 ***
## b1      3.5760     0.1853  19.297  < 2e-16 ***
## alpha  23.1324     0.5098  45.373  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.258 on 237 degrees of freedom
##
## Number of iterations to convergence: 6
## Achieved convergence tolerance: 3.608e-09
```

```
summary(sep60_nls)
```

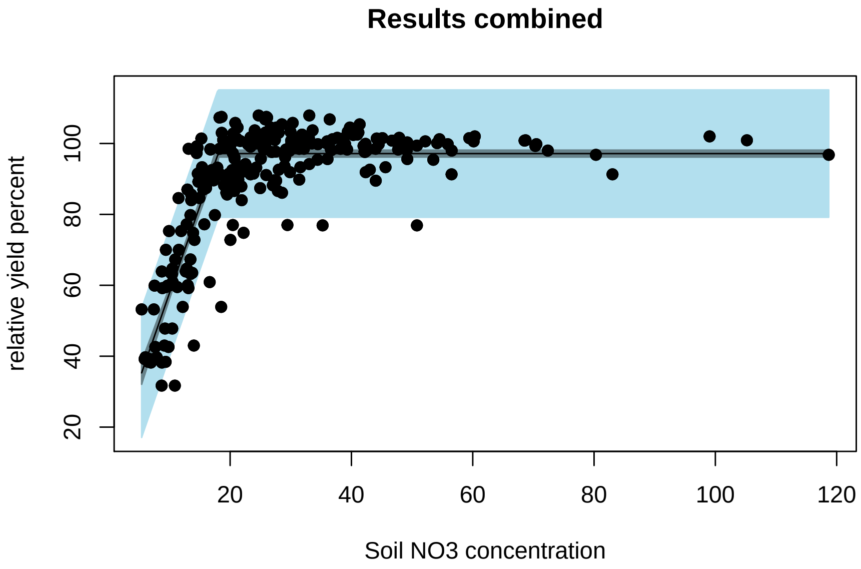
```
##
## Formula: ryp ~ SS_nonlinModel(predictor = no3, b0, b1, alpha)
##
## Parameters:
##       Estimate Std. Error t value Pr(>|t|)
## b0      5.4519     2.9785    1.83   0.0684 .
## b1      5.6820     0.2529   22.46   <2e-16 ***
## alpha  16.2863     0.2818   57.80   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.427 on 237 degrees of freedom
##
## Number of iterations to convergence: 5
## Achieved convergence tolerance: 8.571e-09
```

Instead of modeling the depths model separately we model them together - so there is a common slope, intercept, and plateau.

```
red_nls <-
  nls(ryp ~ SS_nonlinModel(predictor = no3, b0, b1, alpha), data = dat)

summary(red_nls)
```

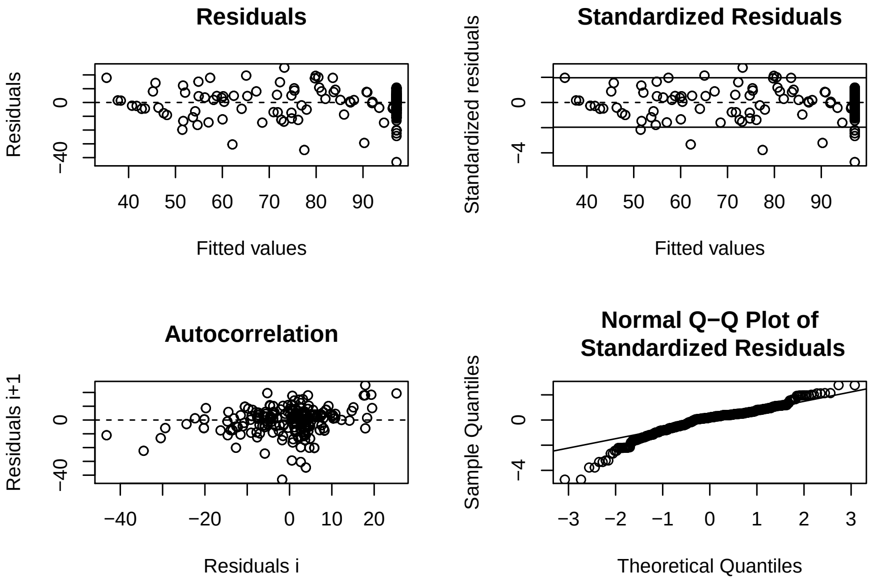
```
##
## Formula: ryp ~ SS_nonlinModel(predictor = no3, b0, b1, alpha)
##
## Parameters:
##       Estimate Std. Error t value Pr(>|t|)
## b0      8.7901     2.7688   3.175   0.0016 **
## b1      4.8995     0.2207  22.203   <2e-16 ***
## alpha  18.0333     0.3242  55.630   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.13 on 477 degrees of freedom
##
## Number of iterations to convergence: 7
## Achieved convergence tolerance: 7.126e-09
```

```
par(mfrow = c(1, 1))
plotFit(
  red_nls,
  interval = "both",
  pch = 19,
  shade = TRUE,
  col.conf = "lightblue4",
  col.pred = "lightblue2",
  data = dat,
  main = 'Results combined',
  ylab = 'relative yield percent',
  xlab = 'Soil NO3 concentration'
)
```



Examine residual values for the combined model.

```
library(nlstools)
#using nlstools nlsResiduals function to get some quick residual plots
#can also use test.nlsResiduals(resid)
# https://www.rdocumentation.org/packages/nlstools/versions/1.0-2
resid <- nlsResiduals(red_nls)
plot(resid)
```



can we test whether the parameters for the two soil depth fits are significantly different? To know if the combined model is appropriate, we consider a parameterization where we let the parameters for the 60cm model be equal to the parameters from the 30cm model plus some increment:

$$\beta_{02} = \beta_{01} + d_0 \beta_{12} = \beta_{11} + d_1 \alpha_2 = \alpha_1 + d_a$$

We can implement this in the following function:

```
nonlinModelF <- function(predictor,soildep,b01,b11,a1,d0,d1,da){
   b02 = b01 + d0 #make 60cm parms = 30cm parms + increment
   b12 = b11 + d1
   a2 = a1 + da

   y1 = ifelse(predictor<=a1,
         b01+b11*predictor, #if observation less than cutoff simple linear model
         b01+b11*a1) #otherwise flat line
   y2 = ifelse(predictor<=a2,
               b02+b12*predictor,
               b02+b12*a2)
   y =  y1*(soildep == 30) + y2*(soildep == 60)  #combine models
   return(y)
}
```

Starting values are easy now because we fit each model individually.

```
Soil_full=nls(ryp~nonlinModelF(predictor=no3,soildep=depth,b01,b11,a1,d0,d1,da),
              data=dat,
              start=list(b01=15.2,b11=3.58,a1=23.13,d0=-9.74,d1=2.11,da=-6.85))

summary(Soil_full)
```

```
##
## Formula: ryp ~ nonlinModelF(predictor = no3, soildep = depth, b01, b11,
##     a1, d0, d1, da)
##
## Parameters:
##     Estimate Std. Error t value Pr(>|t|)
## b01  15.1943     2.8322   5.365 1.27e-07 ***
## b11   3.5760     0.1762  20.291  < 2e-16 ***
## a1   23.1324     0.4848  47.711  < 2e-16 ***
## d0   -9.7424     4.2357  -2.300   0.0219 *
## d1    2.1060     0.3203   6.575 1.29e-10 ***
## da   -6.8461     0.5691 -12.030  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.854 on 474 degrees of freedom
##
## Number of iterations to convergence: 1
## Achieved convergence tolerance: 3.742e-06
```

So, the increment parameters, d_1,d_2,d_a are all significantly different from 0, suggesting that we should have two models here.

6.2.3 Model/Estiamtion Adequcy

(Bates and Watts, 1980) assess nonlinearity in terms of 2 components of curvature:

- **Intrinsic nonlinearity**: the degree of bending and twisting in $f(\theta)$; our estimation approach assumes that hte true function is relatively flat (planar) in the neighborhood fo $\hat{\theta}$, which would not be true if $f()$ has a lot of "bending" int he neighborhood of $\hat{\theta}$ (independent of parameterizaiton)
    - If bad, the distribution of residuals will be seriously distorted
    - slow to converge
    - difficult to identify (could use this function `rms.curve`)
    - Solution:
        - could use higher order Taylor expansions estimation
        - Bayesian method

- **Parameter effects nonlinearity**: degree to which curvature (nonlinearity) is affected by choice of θ (data dependent; dependent on parameterization)
    - leads to problems with inferecne on $\hat{\theta}$
    - `rms.curve` in `MASS` can identify
    - bootstrap-based inference can also be used
    - Solution: try to reparaemterize.

```
#check parameter effects and intrinsic curvature

modD = deriv3(~ a*exp(b*x), c("a","b"),function(a,b,x) NULL)

nlin_modD=nls(y~modD(a,b,x),start=list(a=astrt,b=bstrt),data=datf)

rms.curv(nlin_modD)
```

```
## Parameter effects: c^theta x sqrt(F) = 0.0626
##         Intrinsic: c^iota  x sqrt(F) = 0.0062
```

In linear model, we have Linear Regression, we have goodness of fit measure as R^2:

$$R^2 = \frac{SSR}{SSTO} = 1 - \frac{SSE}{SSTO} = \frac{\sum_{i=1}^n (\hat{Y}_i - \bar{Y})^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2} = 1 - \frac{\sum_{i=1}^n (Y_i - \hat{Y})^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2}$$

but not valid in the nonlinear case because the error sum of squares and model sum of squares do not add to the total corrected sum of squares

$$SSR + SSE \neq SST$$

but we can use pseudo-R^2:

$$R^2_{pseudo} = 1 - \frac{\sum_{i=1}^n (Y_i - \hat{Y})^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2}$$

But we can't interpret this as the proportion of variability explained by the model. We should use as a relative comparison of different models.

Residual Plots: standardize, similar to OLS. useful when the intrinsic curvature is small:

The studentized residuals

$$r_i = \frac{e_i}{s\sqrt{1-\hat{c}_i}}$$

where $\hat{c}_i$is the i-th diagonal of $\hat{\mathbf{H}} = \mathbf{F}(\hat{}\ )[\mathbf{F}(\hat{}\ )'\mathbf{F}(\hat{}\ )]^{-\mathbf{1}}\mathbf{F}(\hat{}\ )'$

We could have problems of

- Collinearity: the condition number of $[\mathbf{F}(\hat{}\ )'\mathbf{F}(\hat{}\ )]^{-\mathbf{1}}$ should be less than 30. Follow (Magel and Hertsgaard, 1987); reparameterize if possible

- Leverage: Like OLS, but consider $\hat{\mathbf{H}} = \mathbf{F}(\hat{}\ )[\mathbf{F}(\hat{}\ )'\mathbf{F}(\hat{}\ )]^{-\mathbf{1}}\mathbf{F}(\hat{}\ )'$ (also known as "tangent plant hat matrix") (Laurent and Cook, 1992)

- Heterogeneous Errors: weighted Non-linear Least Squares

- Correlated Errors:

    - Generalized Nonlinear Least Squares
    - Nonlinear Mixed Models
    - Bayesian methods

6.2.4 Application

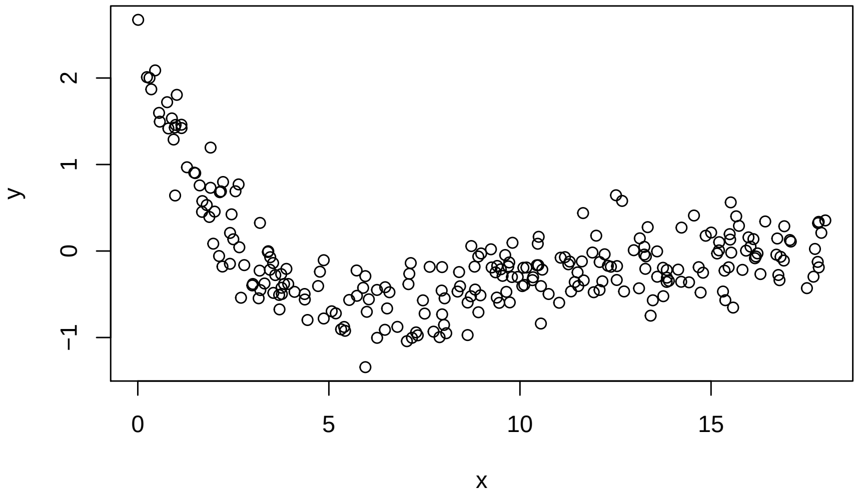
$$y_i = \frac{\theta_0 + \theta_1 x_i}{1 + \theta_2 \exp(0.4 x_i)} + \epsilon_i$$

where $i = 1, .., n$

Get the starting values

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

## The following object is masked from 'package:kableExtra':
##
##     group_rows

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
plot(my_data)
```



We notice that $Y_{max} = \theta_0 + \theta_1 x_i$ in which we can find x_i from data

```
max(my_data$y)
```

```
## [1] 2.6722
```

```
my_data$x[which.max(my_data$y)]
```

```
## [1] 0.0094
```

hence, x = 0.0094 when y = 2.6722 when we have the first equation as

$$2.6722 = \theta_0 + 0.0094\theta_1\theta_0 + 0.0094\theta_1 + 0\theta_2 = 2.6722$$

Secondly, we notice that we can obtain the "average" of y when

$$1 + \theta_2 exp(0.4x) = 2$$

then we can find this average numbers of x and y

```
mean(my_data$y) #find mean y
```

```
## [1] -0.0747864
```

```
my_data$y[which.min(abs(my_data$y-(mean(my_data$y))))] # find y closest to its mean
```

```
## [1] -0.0773
```

```
my_data$x[which.min(abs(my_data$y-(mean(my_data$y))))] #find x closest to the mean y
```

```
## [1] 11.0648
```

we have the second equation

$$1+\theta_2 exp(0.4*11.0648)=20\theta_1+0\theta_1+83.58967\theta_2=1$$

Thirdly, we can plug in the value of x closest to 1 to find the value of y

```
my_data$x[which.min(abs(my_data$x-1))] # find value of x closet to 1
```

```
## [1] 0.9895
```

```
match(my_data$x[which.min(abs(my_data$x-1))], my_data$x) # find index of x closest to
```

```
## [1] 14
```

```
my_data$y[match(my_data$x[which.min(abs(my_data$x-1))], my_data$x)]# find y value
```

```
## [1] 1.4577
```

hence we have

$$1.457=\frac{\theta_0+\theta_1*0.9895}{1+\theta_2 exp(0.4*0.9895)}1.457+2.164479*\theta_2=\theta_0+\theta_1*0.9895\theta_0+\theta_1*0.9895-2.164479*\theta_2=1.457$$

with 3 equations, we can solve them to get the starting value for $\theta_0,\theta_1,\theta_2$

$$\theta_0+0.0094\theta_1+0\theta_2=2.67220\theta_1+0\theta_1+83.58967\theta_2=1\theta_0+\theta_1*0.9895-2.164479*\theta_2=1.457$$

```
library(matlib)
A = matrix(c(0,0.0094, 0, 0,0, 83.58967, 1, 0.9895, - 2.164479), nrow = 3, ncol = 3, b
b = c(2.6722,1,1.457 )
showEqn(A, b)
```

```
## 0*x1 + 0.0094*x2        + 0*x3  =  2.6722
## 0*x1      + 0*x2 + 83.58967*x3  =       1
## 1*x1 + 0.9895*x2 - 2.164479*x3  =   1.457
```

```
Solve(A, b, fractions = F)
```

```
## x1      =  -279.80879739
##   x2    =   284.27659574
##     x3  =      0.0119632
```

Construct manually Gauss-Newton Algorithm

```
#starting value
theta_0_strt = -279.80879739
theta_1_strt =  284.27659574
theta_2_strt = 0.0119632

#model
mod_4 = function(theta_0,theta_1,theta_2,x){
    (theta_0 + theta_1*x)/(1+ theta_2*exp(0.4*x))
}

#define a function
f_4 = expression((theta_0 + theta_1*x)/(1+ theta_2*exp(0.4*x)))

#take the first derivative
df_4.d_theta_0=D(f_4,'theta_0')

df_4.d_theta_1=D(f_4,'theta_1')

df_4.d_theta_2=D(f_4,'theta_2')

# save the result of all iterations
theta_vec = matrix(c(theta_0_strt,theta_1_strt,theta_2_strt))
delta= matrix(NA, nrow=3,ncol = 1)

f_theta = as.matrix(eval(f_4,list(x=my_data$x,theta_0 = theta_vec[1,1],theta_1 = theta_vec[2,1],t

i = 1

repeat {
    F_theta_0 = as.matrix(cbind(
        eval(
            df_4.d_theta_0,
            list(
                x = my_data$x,
                theta_0 = theta_vec[1, i],
                theta_1 = theta_vec[2, i],
                theta_2 = theta_vec[3, i]
            )
        ),
        eval(
            df_4.d_theta_1,
            list(
                x = my_data$x,
                theta_0 = theta_vec[1, i],
        theta_1 = theta_vec[2, i],
        theta_2 = theta_vec[3, i]
      )
    ),
    eval(
      df_4.d_theta_2,
      list(
        x = my_data$x,
        theta_0 = theta_vec[1, i],
        theta_1 = theta_vec[2, i],
        theta_2 = theta_vec[3, i]
      )
    )
  ))
  delta[, i] = (solve(t(F_theta_0) %*% F_theta_0)) %*% t(F_theta_0) %*% (my_data$y -
  theta_vec = cbind(theta_vec, matrix(NA, nrow = 3, ncol = 1))
  theta_vec[, i+1] = theta_vec[, i] + delta[, i]
  i = i + 1

  f_theta = cbind(f_theta, as.matrix(eval(
    f_4,
    list(
      x = my_data$x,
      theta_0 = theta_vec[1, i],
      theta_1 = theta_vec[2, i],
      theta_2 = theta_vec[3, i]
    )
  )))
  delta = cbind(delta, matrix(NA, nrow = 3, ncol = 1))

  #convergence criteria based on SSE
  if (abs(sum((my_data$y - f_theta[,i])^2)-sum((my_data$y - f_theta[,i-1])^2))/(sum(
    break
  }
}
delta
```

```
##               [,1]        [,2]        [,3]       [,4]       [,5]       [,6]
## [1,]  2.811840e+02 -0.03929013  0.43160654  0.6904856  0.6746748  0.4056460
## [2,] -2.846545e+02  0.03198446 -0.16403964 -0.2895487 -0.2933345 -0.1734087
## [3,] -1.804567e-05  0.01530258  0.05137285  0.1183271  0.1613129  0.1160404
##             [,7] [,8]
## [1,]  0.09517681   NA
## [2,] -0.03928239   NA
## [3,]  0.03004911   NA
```

```
theta_vec
```

```
##               [,1]        [,2]        [,3]        [,4]       [,5]       [,6]
## [1,] -279.8087974  1.37521388  1.33592375  1.76753029  2.4580158  3.1326907
## [2,]  284.2765957 -0.37788712 -0.34590266 -0.50994230 -0.7994910 -1.0928255
## [3,]    0.0119632  0.01194515  0.02724773  0.07862059  0.1969477  0.3582607
##             [,7]       [,8]
## [1,]  3.5383367  3.6335135
## [2,] -1.2662342 -1.3055166
## [3,]  0.4743011  0.5043502
```

```
head(f_theta)
```

```
##            [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]
## [1,] -273.8482 1.355410 1.297194 1.633802 2.046023 2.296554 2.389041 2.404144
## [2,] -209.0859 1.268192 1.216738 1.514575 1.863098 2.059505 2.126009 2.135969
## [3,] -190.3323 1.242916 1.193433 1.480136 1.810629 1.992095 2.051603 2.060202
## [4,] -177.1891 1.225196 1.177099 1.456024 1.774000 1.945197 1.999945 2.007625
## [5,] -148.5872 1.186618 1.141549 1.403631 1.694715 1.844154 1.888953 1.894730
## [6,] -119.9585 1.147980 1.105961 1.351301 1.615968 1.744450 1.779859 1.783866
```

```
# estimate sigma^2

sigma2 = 1 / (nrow(my_data) - 3) * (t(my_data$y - (f_theta[, ncol(f_theta)]))) %*%
    (my_data$y - (f_theta[, ncol(f_theta)])) # p = 3
sigma2
```

```
##           [,1]
## [1,] 0.0801686
```

After 8 iterations, my function has converged. And objective function value at convergence is

```
sum((my_data$y - f_theta[,i])^2)
```

```
## [1] 19.80165
```

and the parameters of θs are

```
theta_vec[,ncol(theta_vec)]
```

```
## [1]  3.6335135 -1.3055166  0.5043502
```

and the asymptotic variance covariance matrix is

```
as.numeric(sigma2)*as.matrix(solve(crossprod(F_theta_0)))
```

```
##             [,1]        [,2]        [,3]
## [1,]  0.11552571 -0.04817428  0.02685848
## [2,] -0.04817428  0.02100861 -0.01158212
## [3,]  0.02685848 -0.01158212  0.00703916
```

Issue that I encounter in this problem was that it was very sensitive to starting values. when I tried the value of 1 for all θs, I have vastly different parameter estimates. Then, I try to use the model interpretation to try to find reasonable starting values.

Check with predefined function in nls

```
nlin_4 = nls(y ~ mod_4(theta_0,theta_1, theta_2, x), start = list(theta_0=-279.80879739
nlin_4
```

```
## Nonlinear regression model
##   model: y ~ mod_4(theta_0, theta_1, theta_2, x)
##    data: my_data
## theta_0 theta_1 theta_2
##  3.6359 -1.3064  0.5053
##  residual sum-of-squares: 19.8
##
## Number of iterations to convergence: 9
## Achieved convergence tolerance: 2.294e-07
```

Chapter 7

Generalized Linear Models

Even though we call it generalized linear model, it is still under the paradigm of non-linear regression, because the form of the regression model is non-linear. The name generalized linear model derived from the fact that we have $\mathbf{x_i'}$ (which is linear form) in the model.

7.1 Logistic Regression

$$p_i = f(\mathbf{x}_i;\beta) = \frac{exp(\mathbf{x_i'}\ )}{1 + exp(\mathbf{x_i'}\ )}$$

Equivalently,

$$logit(p_i) = log(\frac{p_i}{1+p_i}) = \mathbf{x_i'}$$

where $\frac{p_i}{1+p_i}$ is the **odds**.

In this form, the model is specified such that **a function of the mean response is linear**. Hence, **Generalized Linear Models**

The likelihood function

$$L(p_i) = \prod_{i=1}^{n} p_i^{Y_i}(1-p_i)^{1-Y_i}$$

where $p_i = \frac{\mathbf{x_i'}}{1+\mathbf{x_i'}}$ and $1 - p_i = (1 + exp(\mathbf{x_i'}\ ))^{-1}$

Hence, our objective function is

$$Q(\beta) = log(L(\beta)) = \sum_{i=1}^{n} Y_i \mathbf{x_i'} \; - \sum_{i=1}^{n} log(1 + exp(\mathbf{x_i'} \;))$$

we could maximize this function numerically using the optimization method above, which allows us to find numerical MLE for $\hat{\beta}$. Then we can use the standard asymptotic properties of MLEs to make inference.

Property of MLEs is that parameters are asymptotically unbiased with sample variance-covariance matrix given by the **inverse Fisher information matrix**

$$\hat{\beta} \dot{\sim} AN(\beta, [\mathbf{I}(\beta)]^{-1})$$

where the **Fisher Information matrix**, $\mathbf{I}(\beta)$ is

$$\begin{aligned}
\mathbf{I}(\beta) &= E[\frac{\partial \log(L(\beta))}{\partial(\beta)} \frac{\partial \log(L(\beta))}{\partial \beta'}] \\
&= E[(\frac{\partial \log(L(\beta))}{\partial \beta_i} \frac{\partial \log(L(\beta))}{\partial \beta_j})_{ij}]
\end{aligned}$$

Under **regularity conditions**, this is equivalent to the negative of the expected value of the Hessian Matrix

$$\begin{aligned}
\mathbf{I}(\beta) &= -E[\frac{\partial^2 \log(L(\beta))}{\partial \beta \partial \beta'}] \\
&= -E[(\frac{\partial^2 \log(L(\beta))}{\partial \beta_i \partial \beta_j})_{ij}]
\end{aligned}$$

Example:

$$x_i'\beta = \beta_0 + \beta_1 x_i$$

$$-\frac{\partial^2 \ln(L(\beta))}{\partial \beta_0^2} = \sum_{i=1}^{n} \frac{\exp(x_i'\beta)}{1+\exp(x_i'\beta)} - [\frac{\exp(x_i'\beta)}{1+\exp(x_i'\beta)}]^2 = \sum_{i=1}^{n} p_i(1-p_i) - \frac{\partial^2 \ln(L(\beta))}{\partial \beta_1^2} = \sum_{i=1}^{n} \frac{x_i^2 \exp(x_i'\beta)}{1+\exp(x_i'\beta)} - [\frac{x}{1}$$

Hence,

$$\mathbf{I}(\beta) = \begin{bmatrix} \sum_i p_i(1-p_i) & \sum_i x_i p_i(1-p_i) \\ \sum_i x_i p_i(1-p_i) & \sum_i x_i^2 p_i(1-p_i) \end{bmatrix}$$

Inference

Likelihood Ratio Tests

To formulate the test, let $\beta = [\beta_1', \beta_2']'$. If you are interested in testing a hypothesis about β_1, then we leave β_2 unspecified (called **nuisance parameters**). β_1 and β_2 can either a **vector** or **scalar**, or β_2 can be null.

Example: $H_0 : \beta_1 = \beta_{1,0}$ (where $\beta_{1,0}$ is specified) and $\hat{\beta}_{2,0}$ be the MLE of β_2 under the restriction that $\beta_1 = \beta_{1,0}$. The likelihood ratio test statistic is

$$-2 \log \Lambda = -2[\log(L(\beta_{1,0}, \hat{\beta}_{2,0})) - \log(L(\hat{\beta}_1, \hat{\beta}_2))]$$

where

- the first term is the value fo the likelihood for the fitted restricted model
- the second term is the likelihood value of the fitted unrestricted model

Under the null,

$$-2 \log \Lambda \sim \chi^2_v$$

where v is the dimension of β_1

We reject the null when $-2 \log \Lambda > \chi^2_{v,1-\alpha}$

Wald Statistics

Based on

$$\hat{\beta} \sim AN(\beta, [\mathbf{I}(\beta)^{-1}])$$

$$H_0 : \mathbf{L}\hat{\beta} = 0$$

where **L** is a q x p matrix with q linearly independent rows. Then

$$W = (\mathbf{L}\hat{\ })'(\mathbf{L}[\mathbf{I}(\hat{\ })]^{\mathbf{-1}}\mathbf{L}')^{-1}(\mathbf{L}\hat{\ })$$

under the null hypothesis

Confidence interval

$$\hat{\beta}_i \pm 1.96\hat{s}^2_{ii}$$

where $\hat{s}^2_{ii}$ is the i-th diagonal of $[\mathbf{I}(\hat{\ })]^{-1}$

If you have

- large sample size, the likelihood ratio and Wald tests have similar results.
- small sample size, the likelihood ratio test is better.

Logistic Regression: Interpretation of β

For single regressor, the model is

$$logit\{\hat{p}_{x_i}\} \equiv logit(\hat{p}_i) = \log(\frac{\hat{p}_i}{1-\hat{p}_i}) = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

When $x = x_i + 1$

$$logit\{\hat{p}_{x_i+1}\} = \hat{\beta}_0 + \hat{\beta}(x_i + 1) = logit\{\hat{p}_{x_i}\} + \hat{\beta}_1$$

Then,

$$logit\{\hat{p}_{x_i+1}\} - logit\{\hat{p}_{x_i}\} = log\{odds[\hat{p}_{x_i+1}]\} - log\{odds[\hat{p}_{x_i}]\} = log(\frac{odds[\hat{p}_{x_i+1}]}{odds[\hat{p}_{x_i}]}) = \hat{\beta}_1$$

and

$$exp(\hat{\beta}_1) = \frac{odds[\hat{p}_{x_i+1}]}{odds[\hat{p}_{x_i}]}$$

the estimated **odds ratio**

the estimated odds ratio, when there is a difference of c units in the regressor x, is $exp(c\hat{\beta}_1)$. When there are multiple covariates, $exp(\hat{\beta}_k)$ is the estimated odds ratio for the variable x_k, assuming that all of the other variables are held constant.

Inference on the Mean Response

Let $x_h = (1, x_{h1}, ..., x_{h,p-1})'$. Then

$$\hat{p}_h = \frac{exp(\mathbf{x}_\mathbf{h}'\hat{\ })}{1 + exp(\mathbf{x}_\mathbf{h}'\hat{\ })}$$

and $s^2(\hat{p}_h) = \mathbf{x}_\mathbf{h}'[\mathbf{I}(\hat{\ })]^{-\mathbf{1}}\mathbf{x}_\mathbf{h}$

For new observation, we can have a cutoff point to decide whether y = 0 or 1.

7.1.1 Application

```r
library(kableExtra)
library(dplyr)
library(pscl)
```

```
## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis
```

```r
library(ggplot2)
library(faraway)
```

```
## Registered S3 methods overwritten by 'lme4':
##   method                          from
##   cooks.distance.influence.merMod car
##   influence.merMod                car
##   dfbeta.influence.merMod         car
##   dfbetas.influence.merMod        car

##
## Attaching package: 'faraway'

## The following object is masked from 'package:investr':
##
##     beetle
```

```r
library(nnet)
library(agridat)
library(nlstools)
```

Logistic Regression

$x \sim Unif(-0.5, 2.5)$. Then $\eta = 0.5 + 0.75x$

```r
set.seed(23) #set seed for reproducibility
x <- runif(1000,min = -0.5,max = 2.5)
eta1 <- 0.5 + 0.75*x
```

Passing η's into the inverse-logit function, we get

$$p = \frac{\exp(\eta)}{1 + \exp(\eta)}$$

where $p \in [0, 1]$

Then, we generate $y \sim Bernoulli(p)$

```
p <- exp(eta1)/(1+exp(eta1))
y <- rbinom(1000,1,p)
BinData <- data.frame(X = x, Y = y)
```

Model Fit

```
Logistic_Model <- glm(formula = Y ~ X,
                      family = binomial, # family = specifies the response distributio
                      data = BinData)
summary(Logistic_Model)
```

```
##
## Call:
## glm(formula = Y ~ X, family = binomial, data = BinData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2317   0.4153   0.5574   0.7922   1.1469
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.46205    0.10201   4.530 5.91e-06 ***
## X            0.78527    0.09296   8.447  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1106.7  on 999  degrees of freedom
## Residual deviance: 1027.4  on 998  degrees of freedom
## AIC: 1031.4
##
## Number of Fisher Scoring iterations: 4
```

```
nlstools::confint2(Logistic_Model)
```

```
##                 2.5 %    97.5 %
## (Intercept) 0.2618709 0.6622204
## X           0.6028433 0.9676934
```

```
OddsRatio <- coef(Logistic_Model) %>% exp
OddsRatio
```

```
## (Intercept)           X
##    1.587318    2.192995
```

Based on the odds ratio, when

- $x = 0$, the odds of success of 1.59

- $x = 1$, the odds of success increase by a factor of 2.19 (i.e., 119.29% increase).

Deviance Tests

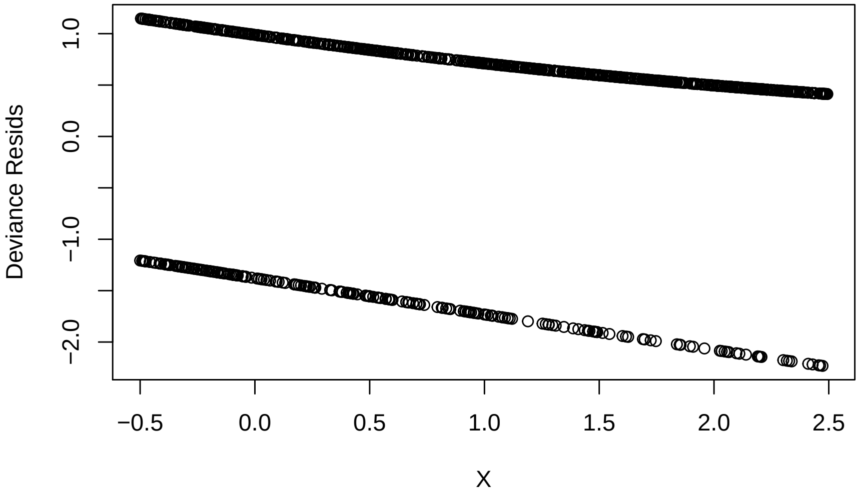
H_0 : No variables are related to the response (i.e., model with just the intercept)H_1 : at least one variable is related

```
Test_Dev = Logistic_Model$null.deviance - Logistic_Model$deviance
p_val_dev <- 1-pchisq(q = Test_Dev, df = 1)
```

Since we see the p-value of 0, we reject the null that no variables are related to the response

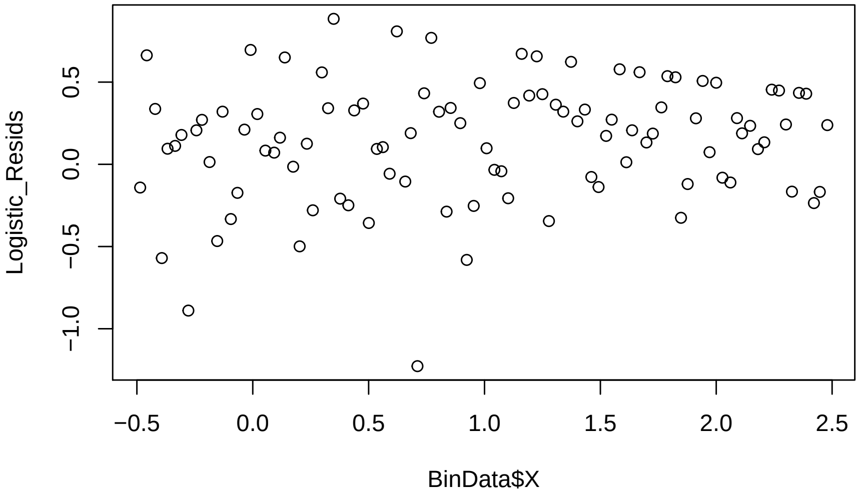
Deviance residuals

```
Logistic_Resids <- residuals(Logistic_Model, type = "deviance")
plot(
  y = Logistic_Resids,
  x = BinData$X,
  xlab = 'X',
  ylab = 'Deviance Resids'
)
```



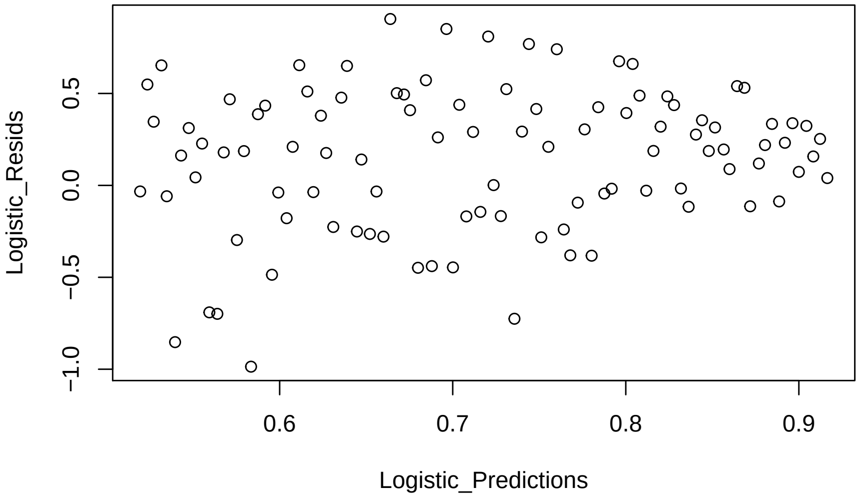
However, this plot is not informative. Hence, we can can see the residudals plots that are grouped into bins based on prediction values.

```
plot_bin <- function(Y,
                     X,
                     bins = 100,
                     return.DF = FALSE) {
  Y_Name <- deparse(substitute(Y))
  X_Name <- deparse(substitute(X))
  Binned_Plot <- data.frame(Plot_Y = Y, Plot_X = X)
  Binned_Plot$bin <-
    cut(Binned_Plot$Plot_X, breaks = bins) %>% as.numeric
  Binned_Plot_summary <- Binned_Plot %>%
    group_by(bin) %>%
    summarise(
      Y_ave = mean(Plot_Y),
      X_ave = mean(Plot_X),
      Count = n()
    ) %>% as.data.frame
  plot(
    y = Binned_Plot_summary$Y_ave,
    x = Binned_Plot_summary$X_ave,
    ylab = Y_Name,
    xlab = X_Name
  )
  if (return.DF)
    return(Binned_Plot_summary)
}
plot_bin(Y = Logistic_Resids,
         X = BinData$X,
         bins = 100)
```



We can also see the predicted value against the residuals.

```
Logistic_Predictions <- predict(Logistic_Model, type = "response")
plot_bin(Y = Logistic_Resids, X = Logistic_Predictions, bins = 100)
```

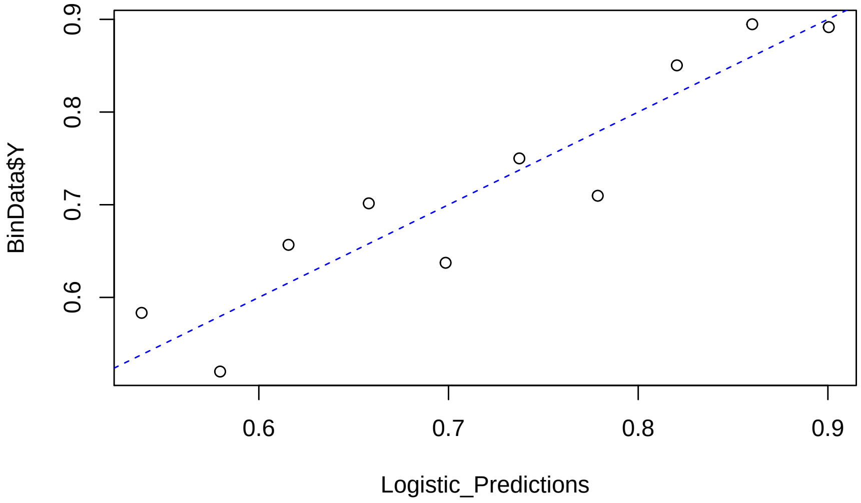


We can also look at a binned plot of the logistic prediction versus the true category

```
NumBins <- 10
Binned_Data <- plot_bin(
  Y = BinData$Y,
  X = Logistic_Predictions,
  bins = NumBins,
  return.DF = TRUE
)
Binned_Data
```

```
##    bin     Y_ave     X_ave Count
## 1    1 0.5833333 0.5382095    72
## 2    2 0.5200000 0.5795887    75
## 3    3 0.6567164 0.6156540    67
## 4    4 0.7014925 0.6579674    67
## 5    5 0.6373626 0.6984765    91
## 6    6 0.7500000 0.7373341    72
## 7    7 0.7096774 0.7786747    93
## 8    8 0.8503937 0.8203819   127
## 9    9 0.8947368 0.8601232   133
## 10  10 0.8916256 0.9004734   203
```

```
abline(0, 1, lty = 2, col = 'blue')
```



Formal deviance test

Hosmer-Lemeshow test

Null hypothesis: the observed events match the expected evens

$$X^2_{HL} = \sum_{j=1}^{J} \frac{(y_j - m_j \hat{p}_j)^2}{m_j \hat{p}_j (1 - \hat{p}_j)}$$

where

- within the j-th bin, y_j is the number of successes
- m_j = number of observations
- $\hat{p}_j$ = predicted probability

Under the null hypothesis, $X^2_{HLL} \sim \chi^2_{J-1}$

```
HL_BinVals <-
  (Binned_Data$Count * Binned_Data$Y_ave - Binned_Data$Count * Binned_Data$X_ave) ^
  2 /
  Binned_Data$Count * Binned_Data$X_ave * (1 - Binned_Data$X_ave)
HLpval <-
  pchisq(q = sum(HL_BinVals),
         df = NumBins,
         lower.tail = FALSE)
HLpval
```

```
## [1] 0.9999989
```

Since p-value = 0.99, we do not reject the null hypothesis (i.e., the model is fitting well).

7.2 Probit Regression

$$E(Y_i) = p_i = \Phi(\mathbf{x_i'})$$

where $\Phi()$ is the CDF of a N(0,1) random variable.

Other models (e..g, t–distribution; log-log; I complimentary log-log)

We let $Y_i = 1$ success, $Y_i = 0$ no success. We assume $Y \sim Ber$ and $p_i = P(Y_i = 1)$, the success probability. We cosnider a logistic regression with the response function $logit(p_i) = x_i'\beta$

Confusion matrix

	Predicted	
Truth	1	0
1	True Positive (TP)	False Negative (FN)
0	False Positive (FP)	True Negative (TN)

Sensitivity: ability to identify positive results

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

Specificity: ability to identify negative results

$$\text{Specificity} = \frac{TN}{TN + FP}$$

False positive rate: Type I error (1- specificity)

$$\text{False Positive Rate} = \frac{FP}{TN + FP}$$

False Negative Rate: Type II error (1-sensitivity)

$$\text{False Negative Rate} = \frac{FN}{TP + FN}$$

	Predicted	
Truth	1	0
1	Sensitivity	False Negative Rate
0	False Positive Rate	Specificity

7.3 Binomial Regression

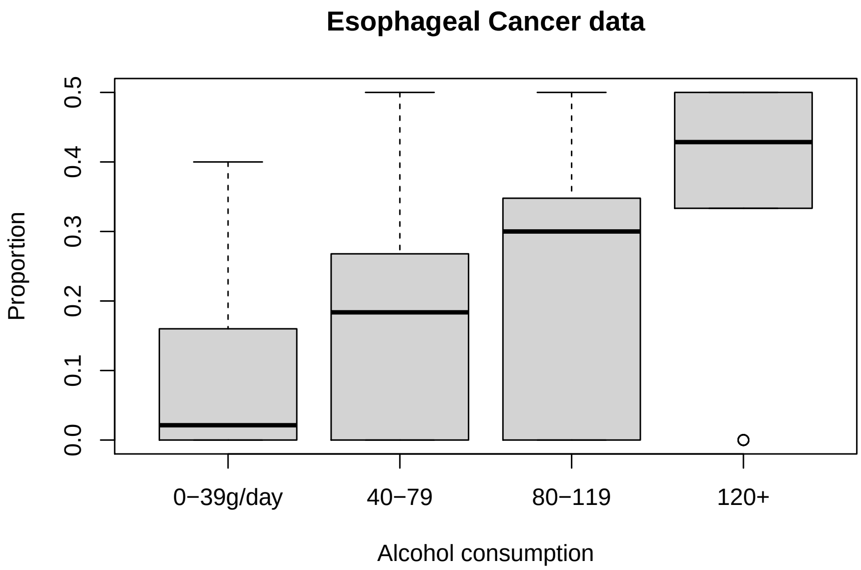
Binomial

Here, cancer case = successes, and control case = failures.

```r
data("esoph")
head(esoph, n = 3)
```

```
##   agegp     alcgp    tobgp ncases ncontrols
## 1 25-34 0-39g/day 0-9g/day      0        40
## 2 25-34 0-39g/day    10-19      0        10
## 3 25-34 0-39g/day    20-29      0         6
```

```r
plot(
  esoph$ncases / (esoph$ncases + esoph$ncontrols) ~ esoph$alcgp,
  ylab = "Proportion",
  xlab = 'Alcohol consumption',
  main = 'Esophageal Cancer data'
)
```



```
class(esoph$agegp) <- "factor"
class(esoph$alcgp) <- "factor"
class(esoph$tobgp) <- "factor"
```

```
#  only the alcohol consumption as a predictor
model <- glm(cbind(ncases, ncontrols) ~ alcgp, data = esoph, family = binomial)
summary(model)
```

```
##
## Call:
## glm(formula = cbind(ncases, ncontrols) ~ alcgp, family = binomial,
##     data = esoph)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.6629  -1.0478  -0.0081   0.6307   3.0296
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.6610     0.1921 -13.854  < 2e-16 ***
## alcgp40-79    1.1064     0.2303   4.804 1.56e-06 ***
## alcgp80-119   1.6656     0.2525   6.597 4.20e-11 ***
## alcgp120+     2.2630     0.2721   8.317  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 227.24  on 87  degrees of freedom
## Residual deviance: 138.79  on 84  degrees of freedom
## AIC: 294.27
##
## Number of Fisher Scoring iterations: 5
```

```
#Coefficient Odds
coefficients(model) %>% exp
```

```
## (Intercept)  alcgp40-79 alcgp80-119   alcgp120+
##  0.06987952  3.02331229  5.28860570  9.61142563
```

```
deviance(model)/df.residual(model)
```

```
## [1] 1.652253
```

```
model$aic
```

```
## [1] 294.27
```

```
# alcohol consumption and age as predictors
better_model <- glm(cbind(ncases, ncontrols) ~ agegp + alcgp, data = esoph, family = bi
summary(better_model)
```

```
##
## Call:
## glm(formula = cbind(ncases, ncontrols) ~ agegp + alcgp, family = binomial,
##     data = esoph)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8979  -0.5592  -0.1995   0.5029   2.6250
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -5.6180     1.0217  -5.499 3.82e-08 ***
## agegp35-44    1.5376     1.0646   1.444 0.148669
## agegp45-54    2.9470     1.0217   2.884 0.003922 **
## agegp55-64    3.3116     1.0172   3.255 0.001132 **
## agegp65-74    3.5774     1.0209   3.504 0.000458 ***
## agegp75+      3.5858     1.0620   3.377 0.000734 ***
## alcgp40-79    1.1392     0.2367   4.814 1.48e-06 ***
## alcgp80-119   1.4951     0.2600   5.749 8.97e-09 ***
## alcgp120+     2.2228     0.2843   7.820 5.29e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 227.241  on 87  degrees of freedom
## Residual deviance:  64.572  on 79  degrees of freedom
## AIC: 230.05
##
## Number of Fisher Scoring iterations: 6
```

```
better_model$aic #smaller AIC is better
```

```
## [1] 230.0526
```

```
coefficients(better_model) %>% exp
```

```
##  (Intercept)   agegp35-44   agegp45-54   agegp55-64   agegp65-74     agegp75+
##  0.003631855  4.653273722 19.047899816 27.428640745 35.780787582 36.082010052
##   alcgp40-79  alcgp80-119    alcgp120+
##  3.124334222  4.459579378  9.233256747
```

```
pchisq(
    q = model$deviance - better_model$deviance,
    df = model$df.residual - better_model$df.residual,
    lower = FALSE
)
```

```
## [1] 1.354906e-14
```

```
# specify link function as probit
Prob_better_model <- glm(
    cbind(ncases, ncontrols) ~ agegp + alcgp,
    data = esoph,
    family = binomial(link = probit)
)
summary(Prob_better_model)
```

```
##
## Call:
## glm(formula = cbind(ncases, ncontrols) ~ agegp + alcgp, family = binomial(link = probit),
##     data = esoph)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8676  -0.5938  -0.1802   0.4852   2.6056
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.9800     0.4291  -6.945 3.79e-12 ***
## agegp35-44    0.6991     0.4491   1.557 0.119520
## agegp45-54    1.4212     0.4292   3.311 0.000929 ***
## agegp55-64    1.6512     0.4262   3.874 0.000107 ***
## agegp65-74    1.8039     0.4297   4.198 2.69e-05 ***
## agegp75+      1.8025     0.4613   3.908 9.32e-05 ***
## alcgp40-79    0.6224     0.1247   4.990 6.03e-07 ***
## alcgp80-119   0.8256     0.1418   5.823 5.80e-09 ***
## alcgp120+     1.2839     0.1596   8.043 8.77e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 227.241  on 87  degrees of freedom
## Residual deviance:  61.938  on 79  degrees of freedom
## AIC: 227.42
##
## Number of Fisher Scoring iterations: 6
```

7.4 Poisson Regression

From the Poisson distribution

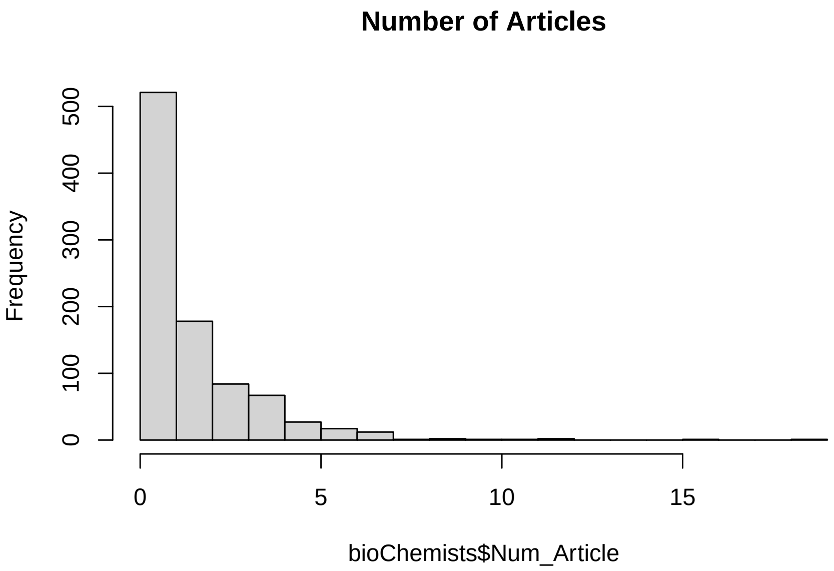
$$f(Y_i) = \frac{\mu_i^{Y_i} exp(-\mu_i)}{Y_i!}, Y_i = 0, 1, ..E(Y_i) = \mu_i var(Y_i) = \mu_i$$

which is a natural distribution for counts. We can see that the variance is a function of the mean. If we let $\mu_i = f(\mathbf{x_i};)$, it would be similar to Logistic Regression since we can choose $f()$ as $\mu_i = \mathbf{x_i'}$, $\mu_i = \exp(\mathbf{x_i'}), \mu_i = \log(\mathbf{x_i'})$

7.4.1 Application

Count Data and Poisson regression

```
data(bioChemists, package = "pscl")
bioChemists <- bioChemists %>%
    rename(
        Num_Article = art, #articles in last 3 years of PhD
        Sex = fem, #coded 1 if female
        Married = mar, #coded 1 if married
        Num_Kid5 = kid5, #number of childeren under age 6
        PhD_Quality = phd, #prestige of PhD program
        Num_MentArticle = ment #articles by mentor in last 3 years
    )
hist(bioChemists$Num_Article, breaks = 25, main = 'Number of Articles')
```



```
Poisson_Mod <- glm(Num_Article ~ ., family=poisson, bioChemists)
summary(Poisson_Mod)
```

```
##
## Call:
## glm(formula = Num_Article ~ ., family = poisson, data = bioChemists)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.5672  -1.5398  -0.3660   0.5722   5.4467
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       0.304617   0.102981   2.958   0.0031 **
## SexWomen         -0.224594   0.054613  -4.112 3.92e-05 ***
## MarriedMarried    0.155243   0.061374   2.529   0.0114 *
## Num_Kid5         -0.184883   0.040127  -4.607 4.08e-06 ***
## PhD_Quality       0.012823   0.026397   0.486   0.6271
## Num_MentArticle   0.025543   0.002006  12.733  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1817.4  on 914  degrees of freedom
## Residual deviance: 1634.4  on 909  degrees of freedom
## AIC: 3314.1
##
## Number of Fisher Scoring iterations: 5
```

Residual of 1634 with 909 df isn't great.

We see Pearson χ^2

```
Predicted_Means <- predict(Poisson_Mod,type = "response")
X2 <- sum((bioChemists$Num_Article - Predicted_Means)^2/Predicted_Means)
X2
```

```
## [1] 1662.547
```

```
pchisq(X2,Poisson_Mod$df.residual, lower.tail = FALSE)
```

```
## [1] 7.849882e-47
```

With interaction terms, there are some improvements

```
Poisson_Mod_All2way <- glm(Num_Article ~ .^2, family=poisson, bioChemists)
Poisson_Mod_All3way <- glm(Num_Article ~ .^3, family=poisson, bioChemists)
```

Consider the $\hat{\phi} = \frac{\text{deviance}}{df}$

```
Poisson_Mod$deviance / Poisson_Mod$df.residual
```

```
## [1] 1.797988
```

This is evidence for over-dispersion. Likely cause is missing variables. And remedies could either be to include more variables or consider random effects.

A quick fix is to force the Poisson Regression to include this value of ϕ, and this model is called "Quasi-Poisson".

```
phi_hat = Poisson_Mod$deviance/Poisson_Mod$df.residual
summary(Poisson_Mod,dispersion = phi_hat)
```

```
##
## Call:
## glm(formula = Num_Article ~ ., family = poisson, data = bioChemists)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.5672  -1.5398  -0.3660   0.5722   5.4467
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        0.30462    0.13809   2.206  0.02739 *
## SexWomen          -0.22459    0.07323  -3.067  0.00216 **
## MarriedMarried     0.15524    0.08230   1.886  0.05924 .
## Num_Kid5        -0.18488    0.05381  -3.436  0.00059 ***
## PhD_Quality      0.01282    0.03540   0.362  0.71715
## Num_MentArticle  0.02554    0.00269   9.496  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1.797988)
##
##     Null deviance: 1817.4  on 914  degrees of freedom
## Residual deviance: 1634.4  on 909  degrees of freedom
## AIC: 3314.1
##
## Number of Fisher Scoring iterations: 5
```

Or directly rerun the model as

```
quasiPoisson_Mod <- glm(Num_Article ~ ., family=quasipoisson, bioChemists)
```

Quasi-Poisson is not recommended, but Negative Binomial Regression that has an extra parameter to account for over-dispersion is.

7.5 Negative Binomial Regression

```
library(MASS)
NegBinom_Mod <- MASS::glm.nb(Num_Article ~ .,bioChemists)
summary(NegBinom_Mod)
```

```
##
## Call:
## MASS::glm.nb(formula = Num_Article ~ ., data = bioChemists, init.theta = 2.264387695,
##     link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1678  -1.3617  -0.2806   0.4476   3.4524
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      0.256144   0.137348   1.865 0.062191 .
## SexWomen        -0.216418   0.072636  -2.979 0.002887 **
## MarriedMarried   0.150489   0.082097   1.833 0.066791 .
## Num_Kid5        -0.176415   0.052813  -3.340 0.000837 ***
## PhD_Quality      0.015271   0.035873   0.426 0.670326
## Num_MentArticle  0.029082   0.003214   9.048  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for Negative Binomial(2.2644) family taken to be 1)
##
##     Null deviance: 1109.0  on 914  degrees of freedom
## Residual deviance: 1004.3  on 909  degrees of freedom
## AIC: 3135.9
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  2.264
##           Std. Err.:  0.271
##
##  2 x log-likelihood:  -3121.917
```

We can see the dispersion is 2.264 with SE $= 0.271$, which is significantly different from 1, indicating overdispersion. Check Over-Dispersion for more detail

7.6 Multinomial

If we have more than two categories or groups that we want to model relative to covariates (e.g., we have observations $i = 1, ..., n$ and groups/ covariates $j = 1, 2, ..., J$), multinomial is our candidate model

Let

- p_{ij} be the probability that the i-th observation belongs to the j-th group
- Y_{ij} be the number of observations for individual i in group j; An individual will have observations $Y_{i1}, Y_{i2}, ...Y_{iJ}$
- assume the probability of observing this response is given by a multinomial distribution in terms of probabilities p_{ij}, where $\sum_{j=1}^J p_{ij} = 1$. For interpretation, we have a baseline category $p_{i1} = 1 - \sum_{j=2}^J p_{ij}$

The link between the mean response (probability) p_{ij} and a linear function of the covariates

$$\eta_{ij} = \mathbf{x'_i}_{\mathbf{j}} = \log \frac{p_{ij}}{p_{i1}}, j = 2, .., J$$

We compare p_{ij} to the baseline p_{i1}, suggesting

$$p_{ij} = \frac{\exp(\eta_{ij})}{1 + \sum_{i=2}^J \exp(\eta_{ij})}$$

which is known as **multinomial logistic** model.

```
library(faraway)
library(dplyr)
data(nes96, package="faraway")
head(nes96,3)
```

```
##   popul TVnews selfLR ClinLR DoleLR     PID age  educ   income    vote
## 1     0      7 extCon extLib    Con  strRep  36    HS $3Kminus    Dole
## 2   190      1 sliLib sliLib sliCon weakDem  20  Coll $3Kminus Clinton
## 3    31      7    Lib    Lib    Con weakDem  24 BAdeg $3Kminus Clinton
```

We try to understand their political strength

```
table(nes96$PID)
```

```
##
##  strDem weakDem  indDem  indind  indRep weakRep  strRep
##     200     180     108      37      94     150     175
```

```
nes96$Political_Strength <- NA
nes96$Political_Strength[nes96$PID %in% c("strDem", "strRep")] <-
  "Strong"
nes96$Political_Strength[nes96$PID %in% c("weakDem", "weakRep")] <-
  "Weak"
nes96$Political_Strength[nes96$PID %in% c("indDem", "indind", "indRep")] <-
  "Neutral"
nes96 %>% group_by(Political_Strength) %>% summarise(Count = n())
```

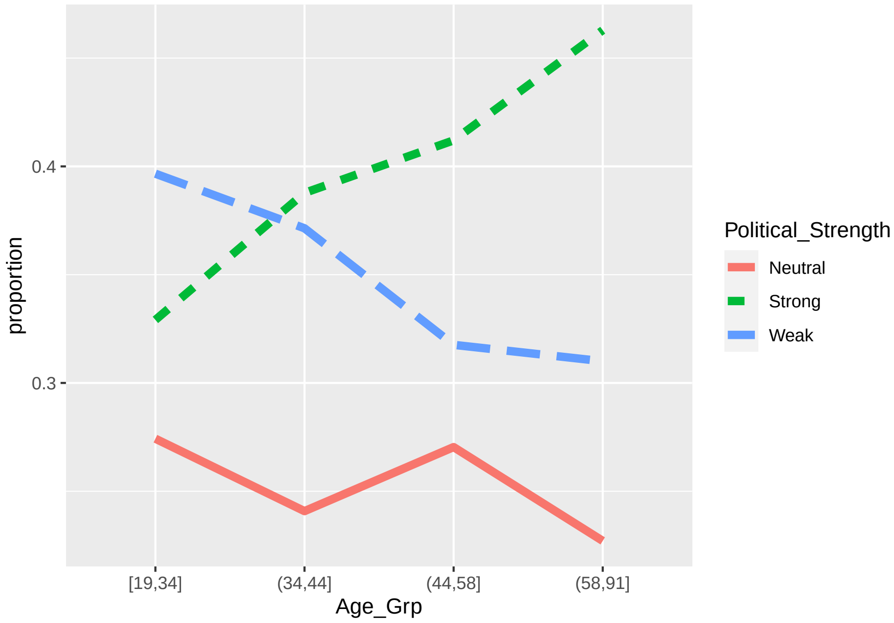
```
## # A tibble: 3 x 2
##   Political_Strength Count
##   <chr>              <int>
## 1 Neutral              239
## 2 Strong               375
## 3 Weak                 330
```

visualize the political strength variable

```
library(ggplot2)
Plot_DF <- nes96 %>%
  mutate(Age_Grp = cut_number(age, 4)) %>%
  group_by(Age_Grp, Political_Strength) %>%
  summarise(count = n()) %>%
  group_by(Age_Grp) %>%
  mutate(etotal = sum(count), proportion = count / etotal)
```

```
## `summarise()` has grouped output by 'Age_Grp'. You can override using the `.groups` argument.
```

```
Age_Plot <- ggplot(
  Plot_DF,
  aes(
    x = Age_Grp,
    y = proportion,
    group = Political_Strength,
    linetype = Political_Strength,
    color = Political_Strength
  )
) +
  geom_line(size = 2)
Age_Plot
```



Fit the multinomial logistic model:

model political strength as a function of age and education

```
library(nnet)
Multinomial_Model <-
    multinom(Political_Strength ~ age + educ, nes96, trace = F)
summary(Multinomial_Model)
```

```
## Call:
## multinom(formula = Political_Strength ~ age + educ, data = nes96,
##     trace = F)
##
## Coefficients:
##        (Intercept)         age     educ.L     educ.Q     educ.C      educ^4
## Strong -0.08788729  0.010700364 -0.1098951 -0.2016197 -0.1757739 -0.02116307
## Weak    0.51976285 -0.004868771 -0.1431104 -0.2405395 -0.2411795  0.18353634
##            educ^5     educ^6
## Strong -0.1664377 -0.1359449
## Weak   -0.1489030 -0.2173144
##
## Std. Errors:
##        (Intercept)         age    educ.L    educ.Q    educ.C    educ^4
## Strong   0.3017034 0.005280743 0.4586041 0.4318830 0.3628837 0.2964776
## Weak     0.3097923 0.005537561 0.4920736 0.4616446 0.3881003 0.3169149
##           educ^5    educ^6
## Strong 0.2515012 0.2166774
## Weak   0.2643747 0.2199186
##
## Residual Deviance: 2024.596
## AIC: 2056.596
```

Alternatively, stepwise model selection based AIC

```
Multinomial_Step <- step(Multinomial_Model,trace = 0)
```

```
## trying - age
## trying - educ
## trying - age
```

```
Multinomial_Step
```

```
## Call:
## multinom(formula = Political_Strength ~ age, data = nes96, trace = F)
##
## Coefficients:
##        (Intercept)          age
## Strong -0.01988977  0.009832916
## Weak    0.59497046 -0.005954348
##
## Residual Deviance: 2030.756
## AIC: 2038.756
```

compare the best model to the full model based on deviance

```
pchisq(q = deviance(Multinomial_Step) - deviance(Multinomial_Model),
df = Multinomial_Model$edf-Multinomial_Step$edf,lower=F)
```

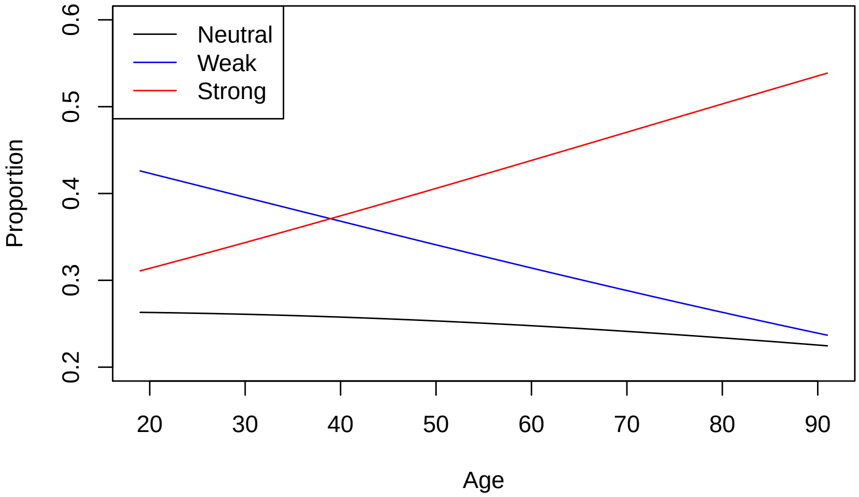
```
## [1] 0.9078172
```

We see no significant difference

Plot of the fitted model

```
PlotData <- data.frame(age = seq(from = 19, to = 91))
Preds <-
```

```
  PlotData %>% bind_cols(data.frame(predict(
    object = Multinomial_Step,
    PlotData, type = "probs"
  )))
plot(
  x = Preds$age,
  y = Preds$Neutral,
  type = "l",
  ylim = c(0.2, 0.6),
  col = "black",
  ylab = "Proportion",
  xlab = "Age"
)
lines(x = Preds$age,
      y = Preds$Weak,
      col = "blue")
lines(x = Preds$age,
      y = Preds$Strong,
      col = "red")
legend(
  'topleft',
  legend = c('Neutral', 'Weak', 'Strong'),
  col = c('black', 'blue', 'red'),
  lty = 1
)
```



```
predict(Multinomial_Step,data.frame(age = 34)) # predicted result (categoriy of political strengt
```

```
## [1] Weak
## Levels: Neutral Strong Weak
```

```
predict(Multinomial_Step,data.frame(age = c(34,35)),type="probs") # predicted result of the proba
```

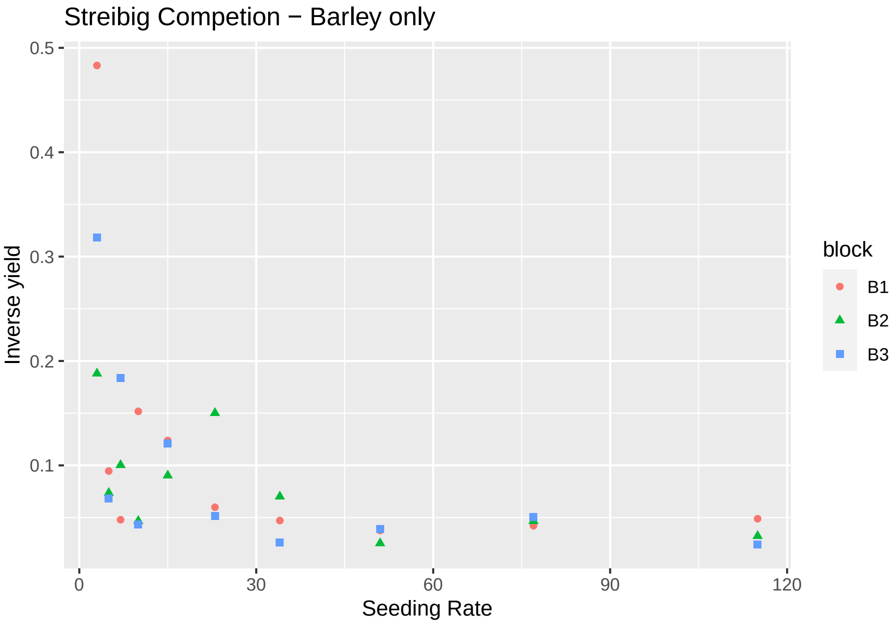
```
##     Neutral    Strong      Weak
## 1 0.2597275 0.3556910 0.3845815
## 2 0.2594080 0.3587639 0.3818281
```

If categories are ordered (i.e., ordinal data), we must use another approach (still multinomial, but use cumulative probabilities).

Another example

```
library(agridat)
dat <- agridat::streibig.competition
# See Schaberger and Pierce, pages 370+
# Consider only the mono-species barley data (no competition from Sinapis)
gammaDat <- subset(dat, sseeds < 1)
gammaDat <-
  transform(gammaDat,
            x = bseeds,
            y = bdwt,
            block = factor(block))
# Inverse yield looks like it will be a good fit for Gamma's inverse link
```

```
ggplot(gammaDat, aes(x = x, y = 1 / y)) + geom_point(aes(color = block, shape =
                                                           block)) +
  xlab('Seeding Rate') + ylab('Inverse yield') + ggtitle('Streibig Competion - Barley
```



$$Y \sim Gamma$$

because Gamma is non-negative as opposed to Normal. The canonical Gamma link function is the inverse (or reciprocal) link

$$\eta_{ij} = \beta_{0j} + \beta_{1j}x_{ij} + \beta_2 x_{ij}^2 Y_{ij} = \eta_{ij}^{-1}$$

The linear predictor is a quadratic model fit to each of the j-th blocks. A different model (not fitted) could be one with common slopes: glm(y $\sim$ x + I(xˆ2),...)

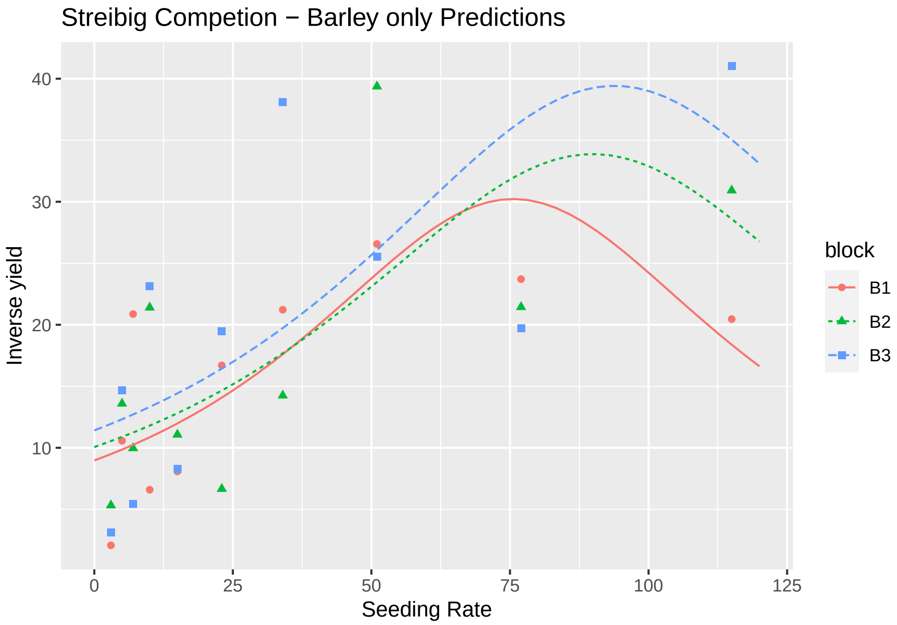
```
# linear predictor is quadratic, with separate intercept and slope per block
m1 <- glm(y ~ block + block*x + block*I(x^2), data=gammaDat,family=Gamma(link="inverse
summary(m1)
```

```
##
## Call:
## glm(formula = y ~ block + block * x + block * I(x^2), family = Gamma(link = "invers
##     data = gammaDat)
##
## Deviance Residuals:
```

```
##      Min        1Q    Median        3Q       Max
## -1.21708  -0.44148   0.02479   0.17999   0.80745
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.115e-01  2.870e-02   3.886 0.000854 ***
## blockB2        -1.208e-02  3.880e-02  -0.311 0.758630
## blockB3        -2.386e-02  3.683e-02  -0.648 0.524029
## x              -2.075e-03  1.099e-03  -1.888 0.072884 .
## I(x^2)          1.372e-05  9.109e-06   1.506 0.146849
## blockB2:x       5.198e-04  1.468e-03   0.354 0.726814
## blockB3:x       7.475e-04  1.393e-03   0.537 0.597103
## blockB2:I(x^2) -5.076e-06  1.184e-05  -0.429 0.672475
## blockB3:I(x^2) -6.651e-06  1.123e-05  -0.592 0.560012
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.3232083)
##
##     Null deviance: 13.1677  on 29  degrees of freedom
## Residual deviance:  7.8605  on 21  degrees of freedom
## AIC: 225.32
##
## Number of Fisher Scoring iterations: 5
```

For predict new value of x

```
newdf <-
  expand.grid(x = seq(0, 120, length = 50), block = factor(c('B1', 'B2', 'B3')))
newdf$pred <- predict(m1, new = newdf, type = 'response')
ggplot(gammaDat, aes(x = x, y = y)) + geom_point(aes(color = block, shape =
                                                       block)) +
  xlab('Seeding Rate') + ylab('Inverse yield') + ggtitle('Streibig Competion - Barley only Predic
  geom_line(data = newdf, aes(
    x = x,
    y = pred,
    color = block,
    linetype = block
  ))
```



7.7 Generalization

We can see that Poisson regression looks similar to logistic regression. Hence, we can generalize to a class of modeling. Thanks to (Nelder and Wedderburn, 1972), we have the **generalized linear models** (GLMs). Estimation is generalize in these models.

Exponential Family
The theory of GLMs is developed for data with distribution given y the **exponential family**.
The form of the data distribution that is useful for GLMs is

$$f(y;\theta,\phi) = \exp(\frac{\theta y - b(\theta)}{a(\phi)} + c(y,\phi))$$

where

- θ is called the natural parameter
- ϕ is called the dispersion parameter

Note:

This family includes the Gamma, Normal, Poisson, and other. For all parameterization of the exponential family, check this link

Example

if we have $Y \sim N(\mu, \sigma^2)$

$$
\begin{aligned}
f(y;\mu,\sigma^2) &= \frac{1}{(2\pi\sigma^2)^{1/2}}\exp(-\frac{1}{2\sigma^2}(y-\mu)^2) \\
&= \exp(-\frac{1}{2\sigma^2}(y^2-2y\mu+\mu^2)-\frac{1}{2}\log(2\pi\sigma^2)) \\
&= \exp(\frac{y\mu-\mu^2/2}{\sigma^2}-\frac{y^2}{2\sigma^2}-\frac{1}{2}\log(2\pi\sigma^2)) \\
&= \exp(\frac{\theta y-b(\theta)}{a(\phi)}+c(y,\phi))
\end{aligned}
$$

where

- $\theta = \mu$
- $b(\theta) = \frac{\mu^2}{2}$
- $a(\phi) = \sigma^2 = \phi$
- $c(y,\phi) = -\frac{1}{2}(\frac{y^2}{\phi} + \log(2\pi\sigma^2))$

Properties of GLM exponential families

1. $E(Y) = b'(\theta)$ where $b'(\theta) = \frac{\partial b(\theta)}{\partial \theta}$ (here `'` is "prime", not transpose)
2. $var(Y) = a(\phi)b''(\theta) = a(\phi)V(\mu)$.
   - $V(\mu)$ is the *variance function*; however, it is only the variance in the case that $a(\phi) = 1$
3. If $a(), b(), c()$ are identifiable, we will derive expected value and variance of Y.

Example

Normal distribution

$$
b'(\theta) = \frac{\partial b(\mu^2/2)}{\partial \mu} = \mu V(\mu) = \frac{\partial^2(\mu^2/2)}{\partial \mu^2} = 1 \to var(Y) = a(\phi) = \sigma^2
$$

Poisson distribution

$$
\begin{aligned}
f(y,\theta,\phi) &= \frac{\mu^y\exp(-\mu)}{y!} \\
&= \exp(y\log(\mu)-\mu-\log(y!)) \\
&= \exp(y\theta-\exp(\theta)-\log(y!))
\end{aligned}
$$

where

- $\theta = \log(\mu)$
- $a(\phi) = 1$

- $b(\theta) = \exp(\theta)$
- $c(y, \phi) = \log(y!)$

Hence,

$$E(Y) = \frac{\partial b(\theta)}{\partial \theta} = \exp(\theta) = \mu var(Y) = \frac{\partial^2 b(\theta)}{\partial \theta^2} = \mu$$

Since $\mu = E(Y) = b'(\theta)$

In GLM, we take some monotone function (typically nonlinear) of μ to be linear in the set of covariates

$$g(\mu) = g(b'(\theta)) = \mathbf{x}'$$

Equivalently,

$$\mu = g^{-1}(\mathbf{x}'\ )$$

where $g(.)$ is the **link function** since it links mean response ($\mu = E(Y)$) and a linear expression of the covariates

Some people use $\eta = \mathbf{x}'$ where $\eta =$ the "linear predictor"

GLM is composed of 2 components

The **random component**:

- is the distribution chosen to model the response variables $Y_1, ..., Y_n$
- is specified by the choice fo $a(), b(), c()$ in the exponential form
- Notation:
  - Assume that there are n **independent** response variables $Y_1, ..., Y_n$ with densities

    $$f(y_i; \theta_i, \phi) = \exp(\frac{\theta_i y_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi))$$

    notice each observation might have different densities
  - Assume that ϕ is constant for all $i = 1, ..., n$, but θ_i will vary. $\mu_i = E(Y_i)$ for all i.

The **systematic component**

- is the portion of the model that gives the relation between μ and the covariates $\mathbf{x}$
- consists of 2 parts:
  - the *link* function, $g(.)$

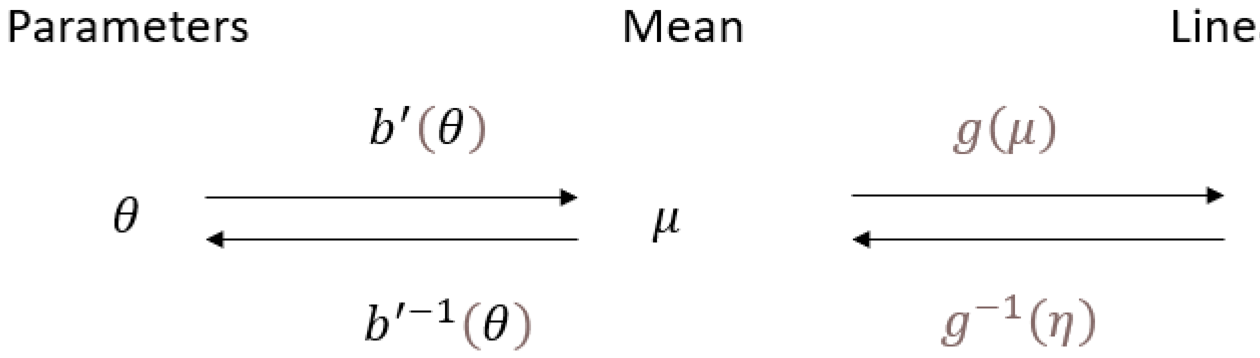
  - the *linear predictor*, $\eta = \mathbf{x}'$

- Notation:
  - assume $g(\mu_i) = \mathbf{x}' \quad = \eta_i$ where $\quad = (\beta_1, ..., \beta_p)'$
  - The parameters to be estimated are $\beta_1, ...\beta_p, \phi$

The Canonical Link

To choose $g(.)$, we can use **canonical link function** (Remember: Canonical link is just a special case of the link function)

If the link function $g(.)$ is such $g(\mu_i) = \eta_i = \theta_i$, the natural parameter, then $g(.)$ is the canonical link.



- $b(\theta)$ = cumulant moment generating function
- $g(\mu)$ is the link function, which relates the linear predictor to the mean and is required to be monotone increasing, continuously differentiable and invertible.

Equivalently, we can think of canonical link function as

$$\gamma^{-1} \circ g^{-1} = I$$

which is the identity. Hence,

$$\theta = \eta$$

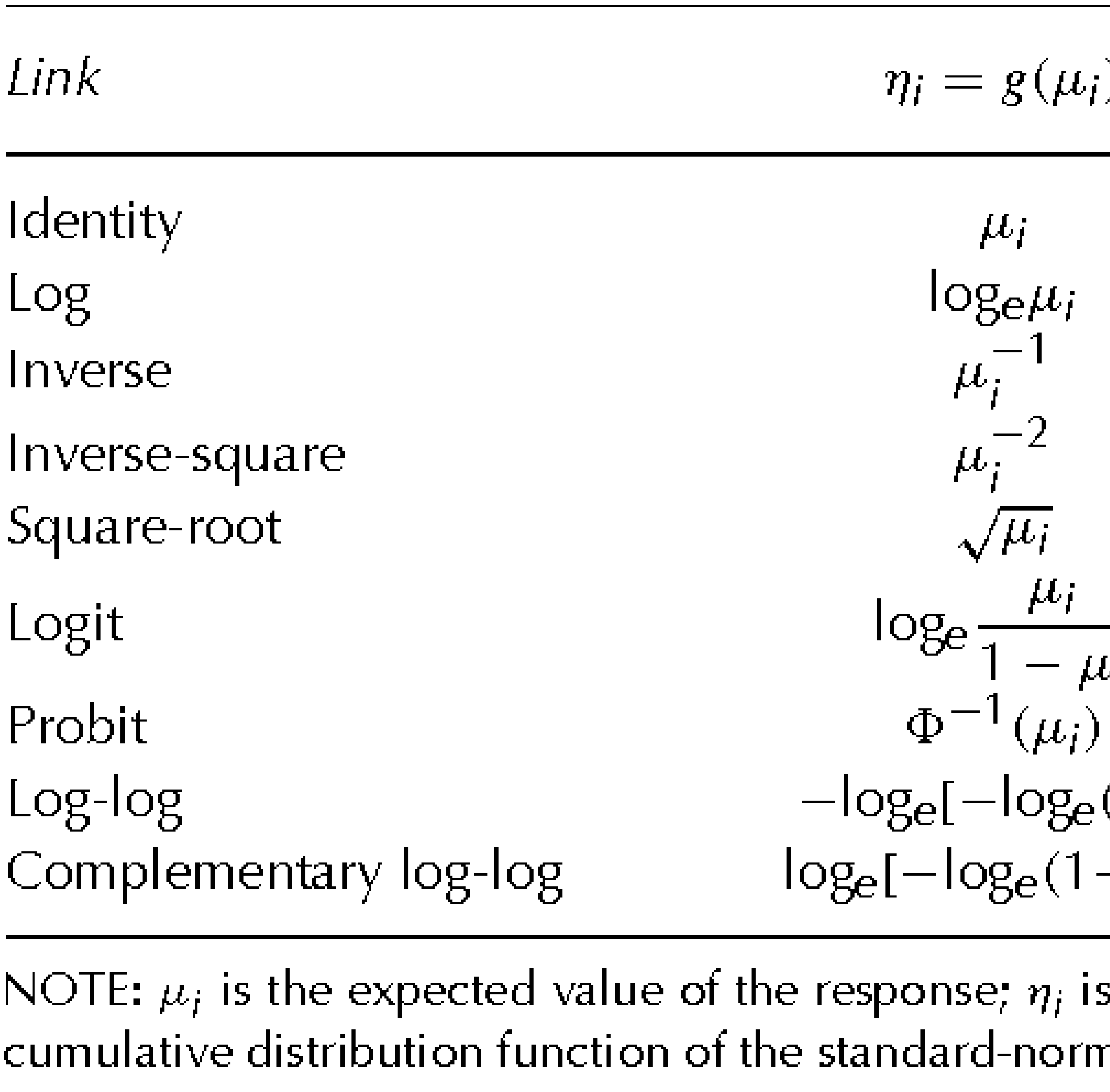
The inverse link

$g^{-1}(.)$ is also known as the mean function, take linear predictor output (ranging from $-\infty$ to ∞) and transform it into a different scale.

- **Exponential**: converts $\mathbf{X}$ into a curve that is restricted between 0 and ∞ (which you can see that is useful in case you want to convert a linear predictor into a non-negative value). $\lambda = \exp(y) = \mathbf{X}$
- **Inverse Logit** (also known as logistic): converts $\mathbf{X}$ into a curve that is restricted between 0 and 1, which is useful in case you want to convert a linear predictor to a probability. $\theta = \frac{1}{1+\exp(-y)} = \frac{1}{1+\exp(-\mathbf{X})}$
  - $y =$ linear predictor value
  - $\theta =$ transformed value

The **identity link** is that

$$\eta_i = g(\mu_i) = \mu_i \mu_i = g^{-1}(\eta_i) = \eta_i$$



Link	$\eta_i = g(\mu_i$
Identity	μ_i
Log	$\log_e \mu_i$
Inverse	μ_i^{-1}
Inverse-square	μ_i^{-2}
Square-root	$\sqrt{\mu_i}$
Logit	$\log_e \frac{\mu_i}{1 - \mu}$
Probit	$\Phi^{-1}(\mu_i)$
Log-log	$-\log_e[-\log_e($
Complementary log-log	$\log_e[-\log_e(1-$

NOTE: μ_i is the expected value of the response; η_i is cumulative distribution function of the standard-norm

Table 15.1 Generalized Linear Models 15.1 the Structure of Generalized Linear Models

More example on the link functions and their inverses can be found on page 380

Example

Normal random component

- Mean Response: $\mu_i = \theta_i$
- Canonical Link: $g(\mu_i) = \mu_i$ (the identity link)

Binomial random component

- Mean Response: $\mu_i = \frac{n_i \exp(\theta)}{1+\exp(\theta_i)}$ and $\theta(\mu_i) = \log(\frac{p_i}{1-p_i}) = \log(\frac{\mu_i}{n_i - \mu_i})$
- Canonical link: $g(\mu_i) = \log(\frac{\mu_i}{n_i - \mu_i})$ (logit link)

Poisson random component

- Mean Response: $\mu_i = \exp(\theta_i)$
- Canonical Link: $g(\mu_i) = \log(\mu_i)$

Gamma random component:

- Mean response: $\mu_i = -\frac{1}{\theta_i}$ and $\theta(\mu_i) = -\mu_i^{-1}$
- Canonical Link: $g(\mu\_i) = -\frac{1}{\mu_i}$

Inverse Gaussian random

- Canonical Link: $g(\mu_i) = \frac{1}{\mu_i^2}$

7.7.1 Estimation

- MLE for parameters of the **systematic component (**β)
- Unification of derivation and computation (thanks to the exponential forms)
- No unification for estimation of the dispersion parameter (ϕ)

7.7.1.1 Estimation of β

We have

$$f(y_i;\theta_i,\phi) = \exp(\frac{\theta_i y_i - b(\theta_i)}{a(\phi)} + c(y_i,\phi)) E(Y_i) = \mu_i = b'(\theta) var(Y_i) = b''(\theta)a(\phi) = V(\mu_i)a(\phi) g(\mu_i) = \mathbf{x}_i'\beta =$$

If the log-likelihood for a single observation is $l_i(\beta,\phi)$. The log-likelihood for all n observations is

$$
\begin{aligned}
l(\beta, \phi) &= \sum_{i=1}^{n} l_i(\beta, \phi) \\
&= \sum_{i=1}^{n} (\frac{\theta_i y_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi))
\end{aligned}
$$

Using MLE to find β, we use the chain rule to get the derivatives

$$
\begin{aligned}
\frac{\partial l_i(\beta, \phi)}{\partial \beta_j} &= \frac{\partial l_i(\beta, \phi)}{\partial \theta_i} \times \frac{\partial \theta_i}{\partial \mu_i} \times \frac{\partial \mu_i}{\partial \eta_i} \times \frac{\partial \eta_i}{\partial \beta_j} \\
&= \sum_{i=1}^{n} (\frac{y_i - \mu_i}{a(\phi)} \times \frac{1}{V(\mu_i)} \times \frac{\partial \mu_i}{\partial \eta_i} \times x_{ij})
\end{aligned}
$$

If we let

$$
w_i \equiv ((\frac{\partial \eta_i}{\partial \mu_i})^2 V(\mu_i))^{-1}
$$

Then,

$$
\frac{\partial l_i(\beta, \phi)}{\partial \beta_j} = \sum_{i=1}^{n} (\frac{y_i \mu_i}{a(\phi)} \times w_i \times \frac{\partial \eta_i}{\partial \mu_i} \times x_{ij})
$$

We can also get the second derivatives using the chain rule.

Example:

For the Newton-Raphson algorithm, we need

$$
-E(\frac{\partial^2 l(\beta, \phi)}{\partial \beta_j \partial \beta_k})
$$

where (j, k)th element of the **Fisher information matrix $\mathbf{I}(\beta)$**

Hence,

$$
-E(\frac{\partial^2 l(\beta, \phi)}{\partial \beta_j \partial \beta_k}) = \sum_{i=1}^{n} \frac{w_i}{a(\phi)} x_{ij} x_{ik}
$$

for the (j,k)th element

If Bernoulli model with logit link function (which is the canonical link)

$$b(\theta) = \log(1{+}\exp(\theta)) = \log(1{+}\exp(\mathbf{x}'\ ))a(\phi) = 1c(y_i,\phi) = 0E(Y) = b'(\theta) = \frac{\exp(\theta)}{1+\exp(\theta)} = \mu = p\eta = g(\mu)$$

For Y_i, i = 1,.., the log-likelihood is

$$l_i(\beta,\phi) = \frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i,\phi) = y_i\mathbf{x}_i'\beta - \log(1+\exp(\mathbf{x}'\ ))$$

Additionally,

$$V(\mu_i) = \mu_i(1-\mu_i) = p_i(1-p_i)\frac{\partial\mu_i}{\partial\eta_i} = p_i(1-p_i)$$

Hence,

$$\begin{aligned}
\frac{\partial l(\beta,\phi)}{\partial\beta_j} &= \sum_{i=1}^{n}[\frac{y_i-\mu_i}{a(\phi)} \times \frac{1}{V(\mu_i)} \times \frac{\partial\mu_i}{\partial\eta_i} \times x_{ij}] \\
&= \sum_{i=1}^{n}(y_i-p_i) \times \frac{1}{p_i(1-p_i)} \times p_i(1-p_i) \times x_{ij} \\
&= \sum_{i=1}^{n}(y_i-p_i)x_{ij} \\
&= \sum_{i=1}^{n}(y_i - \frac{\exp(\mathbf{x_i'}\ )}{1+\exp(\mathbf{x_i'}\ )})x_{ij}
\end{aligned}$$

then

$$w_i = ((\frac{\partial\eta_i}{\partial\mu_i})^2V(\mu_i))^{-1} = p_i(1-p_i)$$

$$\mathbf{I}_{jk}(\ ) = \sum_{i=1}^{n}\frac{w_i}{a(\phi)}x_{ij}x_{ik} = \sum_{i=1}^{n}p_i(1-p_i)x_{ij}x_{ik}$$

The **Fisher-scoring** algorithm for the MLE of   is

$$\begin{pmatrix}\beta_1\\ \beta_2\\ .\\ .\\ .\\ \beta_p\end{pmatrix}^{(m+1)} = \begin{pmatrix}\beta_1\\ \beta_2\\ .\\ .\\ .\\ \beta_p\end{pmatrix}^{(m)} + \mathbf{I}^{-1}(\ )\begin{pmatrix}\frac{\partial l(\beta,\phi)}{\partial\beta_1}\\ \frac{\partial l(\beta,\phi)}{\partial\beta_2}\\ .\\ .\\ .\\ \frac{\partial l(\beta,\phi)}{\partial\beta_p}\end{pmatrix}|_{\beta=\beta^{(m)}}$$

Similar to Newton-Raphson expect the matrix of second derivatives by the expected value of the second derivative matrix.

In matrix notation,

$$\begin{aligned}\frac{\partial l}{\partial \beta} &= \frac{1}{a(\phi)}\mathbf{X}'\mathbf{W}\,(\mathbf{y}- \quad) \\ &= \frac{1}{a(\phi)}\mathbf{F}'\mathbf{V}^{\mathbf{-1}}(\mathbf{y}- \quad)\end{aligned}$$

$$\mathbf{I}(\beta) = \frac{1}{a(\phi)}\mathbf{X}'\mathbf{W}\mathbf{X} = \frac{1}{a(\phi)}\mathbf{F}'\mathbf{V}^{-\mathbf{1}}\mathbf{F}$$

where

- **X** is an n x p matrix of covariates
- **W** is an n x n diagonal matrix with (i,i)th element given by w_i
- an n x n diagonal matrix with (i,i)th element given by $\frac{\partial \eta_i}{\partial \mu_i}$
- **F** = – an n x p matrix with ith row $\frac{\partial \mu_i}{\partial \beta} = (\frac{\partial \mu_i}{\partial \eta_i})\mathbf{x}'_i$
- **V** an n x n diagonal matrix with (i,i)th element given by $V(\mu_i)$

Setting the derivative of the log-likelihood equal to 0, ML estimating equations are

$$\mathbf{F}'\mathbf{V}^{-\mathbf{1}}\mathbf{y} = \mathbf{F}'\mathbf{V}^{-\mathbf{1}}$$

where all components of this equation expect y depends on the parameters β

Special Cases

If one has a canonical link, the estimating equations reduce to

$$\mathbf{X}'\mathbf{y} = \mathbf{X}'$$

If one has an identity link, then

$$\mathbf{X}'\mathbf{V}^{-\mathbf{1}}\mathbf{y} = \mathbf{X}'\mathbf{V}^{-\mathbf{1}}\mathbf{X}$$

which gives the generalized least squares estimator

Generally, we can rewrite the Fisher-scoring algorithm as

$$\beta^{(m+1)} = \beta^{(m)} + (\hat{\mathbf{F}}'\hat{\mathbf{V}}^{-\mathbf{1}}\hat{\mathbf{F}})^{-\mathbf{1}}\hat{\mathbf{F}}'\hat{\mathbf{V}}^{-\mathbf{1}}(\mathbf{y}-\hat{\ }\ )$$

Since $\hat{F}, \hat{V}, \hat{\mu}$ depend on β, we evaluate at $\beta^{(m)}$

From starting values $\beta^{(0)}$, we can iterate until convergence.

Notes:

- if $a(\phi)$ is a constant or of the form $m_i\phi$ with known m_i, then ϕ cancels.

7.7.1.2 Estimation of ϕ

2 approaches:

1. MLE

$$\frac{\partial l_i}{\partial \phi} = \frac{(\theta_i y_i - b(\theta_i)a'(\phi))}{a^2(\phi)} + \frac{\partial c(y_i,\phi)}{\partial \phi}$$

the MLE of ϕ solves

$$\frac{a^2(\phi)}{a'(\phi)} \sum_{i=1}^n \frac{\partial c(y_i,\phi)}{\partial \phi} = \sum_{i=1}^n (\theta_i y_i - b(\theta_i))$$

- Situation others than normal error case, expression for $\frac{\partial c(y,\phi)}{\partial \phi}$ are not simple
- Even for the canonical link and $a(\phi)$ constant, there is no nice general expression for $-E(\frac{\partial^2 l}{\partial \phi^2})$, so the unification GLMs provide for estimation of β breaks down for ϕ

2. Moment Estimation ("Bias Corrected χ^2")

   - The MLE is not conventional approach to estimation of ϕ in GLMS.
   - For the exponential family $var(Y) = V(\mu)a(\phi)$. This implies

   $$a(\phi) = \frac{var(Y)}{V(\mu)} = \frac{E(Y-\mu)^2}{V(\mu)} a(\hat{\phi}) = \frac{1}{n-p} \sum_{i=1}^n \frac{(y_i - \hat{\mu}_i)^2}{V(\hat{\mu})}$$

   where p is the dimension of β
   - GLM with canonical link function $g(.) = (b'(.))^{-1}$

   $$g(\mu) = \theta = \eta = \mathbf{x}'\ \mu = g^{-1}(\eta) = b'(\eta)$$

   - so the method estimator for $a(\phi) = \phi$ is

   $$\hat{\phi} = \frac{1}{n-p} \sum_{i=1}^n \frac{(y_i - g^{-1}(\hat{\eta}_i))^2}{V(g^{-1}(\hat{\eta}_i))}$$

7.7.2 Inference

We have

$$v\hat{a}r(\beta) = a(\phi)(\hat{\mathbf{F}}'\hat{\mathbf{V}}\hat{\mathbf{F}})^{-1}$$

where

- $\mathbf{V}$ is an n x n diagonal matrix with diagonal elements given by $V(\mu_i)$
- $\mathbf{F}$ is an n x p matrix given by $\mathbf{F} = \frac{\partial \mu}{\partial \beta}$
- Both $\mathbf{V}, \mathbf{F}$ are dependent on the mean μ, and thus β. Hence, their estimates $(\hat{\mathbf{V}}, \hat{\mathbf{F}})$ depend on $\hat{\beta}$.

$$H_0 : \mathbf{L} \quad = \mathbf{d}$$

where $\mathbf{L}$ is a q x p matrix with a **Wald** test

$$W = (\mathbf{L}\hat{} \; - \mathbf{d})'(\mathbf{a}(\;)\mathbf{L}(\hat{\mathbf{F}}'\hat{\mathbf{V}}^{-\mathbf{1}}\hat{\mathbf{F}})\mathbf{L}')^{-\mathbf{1}}(\mathbf{L}\hat{} \; - \mathbf{d})$$

which follows χ^2_q distribution (asymptotically), where q is the rank of $\mathbf{L}$

In the simple case $H_0 : \beta_j = 0$ gives $W = \frac{\hat{\beta}^2_j}{v\hat{a}r(\hat{\beta}_j)} \sim \chi^2_1$ asymptotically

Likelihood ratio test

$$\Lambda = 2(l(\hat{\beta}_f) - l(\hat{\beta}_r)) \sim \chi^2_q$$

where

- q is the number of constraints used to fit the reduced model $\hat{\beta}_r$, and $\hat{\beta}_r$ is the fit under the full model.

Wald test is easier to implement, but likelihood ratio test is better (especially for small samples).

7.7.3 Deviance

Deviance is necessary for goodness of fit, inference and for alternative estimation of the dispersion parameter. We define and consider Deviance from a likelihood ratio perspective.

- Assume that ϕ is known. Let $\tilde{\theta}$ denote the full and $\hat{\theta}$ denote the reduced model MLEs. Then, the likelihood ratio (2 times the difference in log-likelihoods) is
$$2\sum_{i=1}^{n} \frac{y_i(\tilde{\theta}_i - \hat{\theta}_i) - b(\tilde{\theta}_i) + b(\hat{\theta}_i)}{a_i(\phi)}$$

- For exponential families, $\mu = E(y) = b'(\theta)$, so the natural parameter is a function of $\mu : \theta = \theta(\mu) = b'^{-1}(\mu)$, and the likelihood ratio turns into

$$2\sum_{i=1}^{m} \frac{y_i\{\theta(\tilde{\mu}_i - \theta(\hat{\mu}_i)\} - b(\theta(\tilde{\mu}_i)) + b(\theta(\hat{\mu}_i))}{a_i(\phi)}$$

- Comparing a fitted model to "the fullest possible model", which is the **saturated model**: $\tilde{\mu}_i = y_i$, i = 1,..,n. If $\tilde{\theta}_i^* = \theta(y_i), \hat{\theta}_i^* = \theta(\hat{\mu})$, the likelihood ratio is

$$2\sum_{i=1}^{n} \frac{y_i(\tilde{\theta}_i^* - \hat{\theta}_i^* + b(\hat{\theta}_i^*))}{a_i(\phi)}$$

- (McCullagh and Nelder, 2019) specify $a(\phi) = \phi$, then the likelihood ratio can be written as

$$D^*(\mathbf{y}, \hat{}) = \frac{2}{\phi}\sum_{i=1}^{n}\{y_i(\tilde{\theta}_i^* - \hat{\theta}_i^*) - b(\tilde{\theta}_i^*) + b(\hat{\theta}_i^*)\}$$

  where

- $D^*(\mathbf{y}, \hat{}) =$ **scaled deviance**
- $D(\mathbf{y}, \hat{}) = \phi D^*(\mathbf{y}, \hat{}) =$ **deviance**

Note:

- in some random component distributions, we can write $a_i(\phi) = \phi m_i$, where
  - m_i is some known scalar that may change with the observations. THen, the scaled deviance components are divided by m_i:

$$D^*(\mathbf{y}, \hat{}) \equiv 2\sum_{i=1}^{n}\{y_i(\tilde{\theta}_i^* - \hat{\theta}_i^*) - b(\tilde{\theta}_i^*) + b(\hat{\theta}_i^*)\}/(\phi m_i)$$

- $D^*(\mathbf{y}, \hat{}) = \sum_{i=1}^{n} d_i$m where d_i is the deviance contribution from the ith observation.
- D is used in model selection
- D^* is used in goodness of fit tests (as it is a likelihood ratio statistic).

$$D^*(\mathbf{y}, \hat{}) = 2\{l(\mathbf{y}, \tilde{}) - l(\mathbf{y}, \hat{})\}$$

- d_i are used to form **deviance residuals**

Example:

Normal

We have

$$\theta = \mu\phi = \sigma^2 b(\theta) = \frac{1}{2}\theta^2 a(\phi) = \phi$$

Hence,

$$\tilde{\theta}_i = y_i \hat{\theta}_i = \hat{\mu}_i = g^{-1}(\hat{\eta}_i)$$

And

$$\begin{aligned} D &= 2\sum_{1=1}^{n} Y_i^2 - y_i\hat{\mu}_i - \frac{1}{2}y_i^2 + \frac{1}{2}\hat{\mu}_i^2 \\ &= \sum_{i=1}^{n} y_i^2 - 2y_i\hat{\mu}_i + \hat{\mu}_i^2 \\ &= \sum_{i=1}^{n} (y_i - \hat{\mu}_i)^2 \end{aligned}$$

which is the **residual sum of squares**

Poisson

$$f(y) = \exp\{y\log(\mu) - \mu - \log(y!)\}\theta = \log(\mu) b(\theta) = \exp(\theta) a(\phi) = 1\tilde{\theta}_i = \log(y_i)\hat{\theta}_i = \log(\hat{\mu}_i)\hat{\mu}_i = g^{-1}(\hat{\eta}_i)$$

Then,

$$\begin{aligned} D &= 2\sum_{i=1}^{n} y_i\log(y_i) - y_i\log(\hat{\mu}_i) - y_i + \hat{\mu}_i \\ &= 2\sum_{i=1}^{n} y_i\log(\frac{y_i}{\hat{\mu}_i}) - (y_i - \hat{\mu}_i) \end{aligned}$$

and

$$d_i = 2\{y_i\log(\frac{y_i}{\hat{\mu}}) - (y_i - \hat{\mu}_i)\}$$

7.7.3.1 Analysis of Deviance

The difference in deviance between a reduced and full model, where q is the difference in the number of free parameters, has an asymptotic χ^2_q. The likelihood ratio test

$$D^*(\mathbf{y};\hat{\ }_{\mathbf{r}}) - D^*(\mathbf{y};\hat{\ }_{\mathbf{f}}) = 2\{l(\mathbf{y};\hat{\ }_{\mathbf{f}}) - l(\mathbf{y};\hat{\ }_{\mathbf{r}})\}$$

this comparison of models is **Analysis of Deviance**. GLM uses this analysis for model selection.

An estimation of ϕ is

$$\hat{\phi} = \frac{D(\mathbf{y};\hat{\ })}{n-p}$$

where p = number of parameters fit.

Excessive use of χ^2 test could be problematic since it is asymptotic (McCullagh and Nelder, 2019)

7.7.3.2 Deviance Residuals

We have $D = \sum_{i=1}^n d_i$. Then, we define **deviance residuals**

$$r_{D_i} = \text{sign}(y_i - \hat{\mu}_i)\sqrt{d_i}$$

Standardized version of deviance residuals is

$$r_{s,i} = \frac{y_i - \hat{\mu}}{\hat{\sigma}(1-h_{ii})^{1/2}}$$

Let $\mathbf{H^{GLM}} = \mathbf{W^{1/2}X(X'WX)^{-1}X'W^{-1/2}}$, where $\mathbf{W}$ is an n x n diagonal matrix with (i,i)th element given by w_i (see Estimation of β). Then Standardized deviance residuals is equivalently

$$r_{s,D_i} = \frac{r_{D_i}}{\{\hat{\phi}(1-h_{ii}^{glm}\}^{1/2}}$$

where h_{ii}^{glm} is the ith diagonal of $\mathbf{H}^{GLM}$

7.7.3.3 Pearson Chi-square Residuals

Another χ^2 statistic is **Pearson** χ^2 statistics: (assume $m_i = 1$)

$$X^2 = \sum_{i=1}^{n} \frac{(y_i - \hat{\mu}_i)^2}{V(\hat{\mu}_i)}$$

where $\hat{\mu}_i$ is the fitted mean response fo the model of interest.

The **Scaled Pearson** χ^2 statistic is given by $\frac{X^2}{\phi} \sim \chi^2_{n-p}$ where p is the number of parameters estimated. Hence, the **Pearson** χ^2 residuals are

$$X_i^2 = \frac{(y_i - \hat{\mu}_i)^2}{V(\hat{\mu}_i)}$$

If we have the following assumptions:

- Independent samples

- No over-dispersion: If $\phi = 1$, $\frac{D(\mathbf{y};\hat{})}{n-p}$ and $\frac{X^2}{n-p}$ have a value substantially larger 1 indicates **improperly specified model** or **overdispersion**

- Multiple groups

then $\frac{X^2}{\phi}$ and $D^*(\mathbf{y};\hat{})$ both follow χ^2_{n-p}

7.7.4 Diagnostic Plots

- Standardized residual Plots:
  - plot(r_{s,D_i}, $\hat{\mu}_i$) or plot(r_{s,D_i}, $T(\hat{\mu}_i)$) where $T(\hat{\mu}_i)$ is transformation($\hat{\mu}_i$) called **constant information scale**:
  - plot(r_{s,D_i}, $\hat{\eta}_i$)

Random Component	$T(\hat{\mu}_i)$
Normal	$\hat{\mu}$
Poisson	$2\sqrt{\hat{\mu}}$
Binomial	$2\sin^{-1}(\sqrt{\hat{\mu}})$
Gamma	$2\log(\hat{\mu})$
Inverse Gaussian	$-2\hat{\mu}^{-1/2}$

- If we see:
  - Trend, it means we might have a wrong link function, or choice of scale

  - Systematic change in range of residuals with a change in $T(\hat{\mu})$ (incorrect random component) (systematic $\neq$ random)

- plot($|r_{D_i}|, \hat{\mu}_i$) to check **Variance Function**.

7.7.5 Goodness of Fit

To assess goodness of fit, we can use

- Deviance

- Pearson Chi-square Residuals

In nested model, we could use likelihood-based information measures:

$$AIC = -2l(\hat{\ }) + 2pAICC = -2l(\hat{\ }) + 2p(\frac{n}{n-p-1})BIC = 2l(\hat{\mu}) + p\log(n)$$

where

- $l(\hat{\mu})$ is the log-likelihood evaluated at the parameter estimates
- p is the number of parameters
- n is the number of observations.

Note: you have to use the same data with the same model (i.e., same link function, same random underlying random distribution). but you can have different number of parameters.

Even though statisticians try to come up with measures that are similar to R^2, in practice, it is not so appropriate. For example, they comapre the log-likelihood of the fitted model against the that of a model with jsut the intercept:

$$R^2_p = 1 - \frac{l(\hat{\mu})}{l(\hat{\mu}_0)}$$

For certain specific random components such as binary response model, we have **rescaled generalized R^2:

$$\bar{R}^2 = \frac{R^2_*}{\max(R^2_*)} = \frac{1 - \exp\{-\frac{2}{n}(l(\hat{\mu}) - l(\hat{\mu}_0)\}}{1 - \exp\{\frac{2}{n}l(\hat{\mu}_0)\}}$$

7.7.6 Over-Dispersion

Random Components	$var(Y)$	$V(\mu)$
Binomial	$var(Y) = n\mu(1-\mu)$	$V(\mu) = \phi n\mu(1-\mu)$ where $m_i = n$
Poisson	$var(Y) = \mu$	$V(\mu) = \phi\mu$

In both cases $\phi = 1$. Recall $b''(\theta) = V(\mu)$ check Estimation of ϕ.

If we find

- $\phi > 1$: over-dispersion (i.e., too much variation for an independent binomial or Poisson distribution).
- $\phi < 1$: under-dispersion (i.e., too little variation for an independent binomial or Poisson distribution).

If we have either over or under-dispersion, it means we might have unspecified random component, we could

- Select a different random component distribution that can accommodate over or under-dispersion (e.g., negative binomial, Conway-Maxwell Poisson)
- use Generalized Linear Mixed Models to handle random effects in generalized linear models.

Chapter 8

Linear Mixed Models

8.1 Dependent Data

Forms of dependent data:

- Multivariate measurements on different individuals: (e.g., a person's blood pressure, fat, etc are correlated)
- Clustered measurements: (e.g., blood pressure measurements of people in the same family can be correlated).
- Repeated measurements: (e.g., measurement of cholesterol over time can be correlated) "If data are collected repeatedly on experimental material to which treatments were applied initially, the data is a repeated measure." (Schabenberger and Pierce, 2001)
- Longitudinal data: (e.g., individual's cholesterol tracked over time are correlated): "data collected repeatedly over time in an observational study are termed longitudinal." (Schabenberger and Pierce, 2001)
- Spatial data: (e.g., measurement of individuals living in the same neighborhood are correlated)

Hence, we like to account for these correlations.

Linear Mixed Model (LMM), also known as **Mixed Linear Model** has 2 components:

- **Fixed effect** (e.g, gender, age, diet, time)
- **Random effects** representing individual variation or auto correlation/spatial effects that imply **dependent (correlated) errors**

Review Two-Way Mixed Effects ANOVA

We choose to model the random subject-specific effect instead of including dummy subject covariates in our model because:

- reduction in the number of parameters to estimate
- when you do inference, it would make more sense that you can infer from a population (i.e., random effect).

LLM Motivation

In a repeated measurements analysis where Y_{ij} is the response for the i-th individual measured at the j-th time,

$i = 1, ..., N$; $j = 1, ..., n_i$

$$\mathbf{Y}_i = \left(\begin{array}{c} Y_{i1} \\ . \\ . \\ . \\ Y_{in_i} \end{array} \right)$$

is all measurements for subject i.

Stage 1: (Regression Model) how the response changes over time for the ith subject

$$\mathbf{Y_i = Z_i \, _i + \, _i}$$

where

- Z_i is an $n_i \times q$ matrix of known covariates
- β_i is an unknown q x 1 vector of subjective -specific coefficients (regression coefficients different for each subject)
- ϵ_i are the random errors (typically $\sim N(0, \sigma^2 I)$)

We notice that there are two many β to estimate here. Hence, this is the motivation for the second stage

Stage 2: (Parameter Model)

$$_{\mathbf{i}} = \mathbf{K_i} \; + \mathbf{b_i}$$

where

- K_i is a q x p matrix of known covariates
- β is a p x 1 vector of unknown parameter
- $\mathbf{b}_i$ are independent $N(0, D)$ random variables

This model explain the observed variability between subjects with respect to the subject-specific regression coefficients, β_i. We model our different coefficient (β_i) with respect to β.

Example:

Stage 1:

$$Y_{ij} = \beta_{1i} + \beta_{2i}t_{ij} + \epsilon_{ij}$$

where

- $j = 1, .., n_i$

In the matrix notation,

$$\mathbf{Y_i} = \begin{pmatrix} Y_{i1} \\ . \\ Y_{in_i} \end{pmatrix}$$

$$\mathbf{Z}_i = \begin{pmatrix} 1 & t_{i1} \\ . & . \\ 1 & t_{in_i} \end{pmatrix}$$

$$\beta_i = \begin{pmatrix} \beta_{1i} \\ \beta_{2i} \end{pmatrix}$$

$$\epsilon_i = \begin{pmatrix} \epsilon_{i1} \\ . \\ \epsilon_{in_i} \end{pmatrix}$$

Thus,

$$\mathbf{Y_i} = \mathbf{Z_i}\ _{\mathbf{i}} + \ _{\mathbf{i}}$$

Stage 2:

$$\beta_{1i} = \beta_0 + b_{1i}\beta_{2i} = \beta_1 L_i + \beta_2 H_i + \beta_3 C_i + b_{2i}$$

where L_i, H_i, C_i are indicator variables defined to 1 as the subject falls into different categories.

Subject specific intercepts do not depend upon treatment, with β_0 (the average response at the start of treatment), and $\beta_1, \beta_2, \beta_3$ (the average time effects for each of three treatment groups).

$$\mathbf{K}_i = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & L_i & H_i & C_i \end{pmatrix} \beta = (\beta_0, \beta_1, \beta_2, \beta_3)'\mathbf{b}_i = \begin{pmatrix} b_{1i} \\ b_{2i} \end{pmatrix} \beta_i = \mathbf{K_i}\ \ + \mathbf{b_i}$$

To get $\hat{\beta}$, we can fit the model sequentially:

1. Estimate $\hat{\beta}_i$ in the first stage
2. Estimate $\hat{\beta}$ in the second stage by replacing β_i with $\hat{\beta}_i$

However, problems arise from this method:

- information is lost by summarizing the vector $\mathbf{Y}_i$ solely by $\hat{\beta}_i$
- we need to account for variability when replacing β_i with its estimate
- different subjects might have different number of observations.

To address these problems, we can use **Linear Mixed Model (Laird and Ware, 1982)**

Substituting stage 2 into stage 1:

$$\mathbf{Y}_i = \mathbf{Z}_i \mathbf{K}_i \beta + \mathbf{Z}_i \mathbf{b}_i + {}_i$$

Let $\mathbf{X}_i = \mathbf{Z}_i \mathbf{K}_i$ be an $n_i \times p$ matrix . Then, the LMM is

$$\mathbf{Y}_i = \mathbf{X}_i \beta + \mathbf{Z}_i \mathbf{b}_i + {}_i$$

where

- $i = 1, .., N$
- β are the fixed effects, which are common to all subjects
- $\mathbf{b}_i$ are the subject specific random effects. $\mathbf{b}_i \sim N_q(\mathbf{0}, \mathbf{D})$
- ${}_i \sim N_{n_i}(\mathbf{0}, {}_\mathbf{i})$
- $\mathbf{b}_i$ and ϵ_i are independent
- $\mathbf{Z}_{i(n_i \times q)}$ and $\mathbf{X}_{i(n_i \times p)}$ are matrices of known covariates.

Equivalently, in the hierarchical form, we call **conditional** or **hierarchical** formulation of the linear mixed model

$$\mathbf{Y}_i | \mathbf{b}_i \sim N(\mathbf{X}_i \beta + \mathbf{Z}_i \mathbf{b}_i, {}_i) \mathbf{b}_i \sim N(\mathbf{0}, \mathbf{D})$$

for $i = 1, .., N$. denote the respective functions by $f(\mathbf{Y}_i | \mathbf{b}_i)$ and $f(\mathbf{b}_i)$

In general,

$$f(A, B) = f(A|B) f(B) f(A) = \int f(A, B) dB = \int f(A|B) f(B) dB$$

In the LMM, the marginal density of $\mathbf{Y}_i$ is

$$f(\mathbf{Y}_i) = \int f(\mathbf{Y}_i | \mathbf{b}_i) f(\mathbf{b}_i) d\mathbf{b}_i$$

which can be shown

$$\mathbf{Y}_i \sim N(\mathbf{X_i} \ , \mathbf{Z_i D Z_i'} + \ _{\mathbf{i}})$$

This is the **marginal** formulation of the linear mixed model

Notes:

We no longer have $Z_i b_i$ in the mean, but add error in the variance (marginal dependence in Y). kinda of averaging out the common effect. Technically, we shouldn't call it averaging the error b (adding it to the variance covariance matrix), it should be called adding random effect

Continue with our example

$$Y_{ij} = (\beta_0 + b_{1i}) + (\beta_1 L_i + \beta_2 H_i + \beta_3 C_i + b_{2i}) t_{ij} + \epsilon_{ij}$$

for each treatment group

$$Y_{ik} = \begin{cases} \beta_0 + b_{1i} + (\beta_1 + \ b_{2i}) t_{ij} + \epsilon_{ij} & L \\ \beta_0 + b_{1i} + (\beta_2 + \ b_{2i}) t_{ij} + \epsilon_{ij} & H \\ \beta_0 + b_{1i} + (\beta_3 + \ b_{2i}) t_{ij} + \epsilon_{ij} & C \end{cases}$$

- Intercepts and slopes are all subject specific
- Different treatment groups have different slops, but the same intercept.

In the hierarchical model form

$$\mathbf{Y}_i | \mathbf{b}_i \sim N(\mathbf{X}_i \beta + \mathbf{Z}_i \mathbf{b}_i, \ _i) \mathbf{b}_i \sim N(\mathbf{0}, \mathbf{D})$$

X will be in the form of

$$\mathbf{X}_i = \mathbf{Z}_i \mathbf{K}_i = \begin{bmatrix} 1 & t_{i1} \\ 1 & t_{i2} \\ . & . \\ 1 & t_{in_i} \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & L_i & H_i & C_i \end{bmatrix} = \begin{bmatrix} 1 & t_{i1} L_i & t_{i1} H_i & T_{i1} C_i \\ 1 & t_{i2} L_i & t_{i2} H_i & T_{i2} C_i \\ . & . & . & . \\ 1 & t_{in_i} L_i & t_{in_i} H_i & T_{in_i} C_i \end{bmatrix}$$

$$\beta = (\beta_0, \beta_1, \beta_2, \beta_3)' \mathbf{b}_i = \begin{pmatrix} b_{1i} \\ b_{2i} \end{pmatrix}, D = \begin{pmatrix} d_{11} & d_{12} \\ d_{12} & d_{22} \end{pmatrix}$$

Assuming $\ _i = \sigma^2 \mathbf{I}_{n_i}$, which is called **conditional independence**, meaning the response on subject i are independent conditional on $\mathbf{b}_i$ and β

In the marginal model form

$$Y_{ij} = \beta_0 + \beta_1 L_i t_{ij} + \beta_2 H_i t_{ij} + \beta_3 C_i t_{ij} + \eta_{ij}$$

where $\eta_i \sim N(\mathbf{0}, \mathbf{Z}_i\mathbf{D}\mathbf{Z}_i' + \ _i)$

Equivalently,

$$\mathbf{Y_i} \sim \mathbf{N}(\mathbf{X_i}\ , \mathbf{Z_i}\mathbf{D}\mathbf{Z_i'} + \ _\mathbf{i})$$

In this case that $n_i = 2$

$$\mathbf{Z_i}\mathbf{D}\mathbf{Z_i'} = \begin{pmatrix} 1 & t_{i1} \\ 1 & t_{i2} \end{pmatrix} \begin{pmatrix} d_{11} & d_{12} \\ d_{12} & d_{22} \end{pmatrix} \begin{pmatrix} 1 & 1 \\ t_{i1} & t_{i2} \end{pmatrix} = \Bigg(\begin{matrix} d_{11} + 2d_{12}t_{i1} + d_{22}t_{i1}^2 & d_{11} + d_{12}(t_{i1} + t \\ d_{11} + d_{12}(t_{i1} + t_{i2}) + d_{22}t_{i1}t_{i2} & d_{11} + 2d_{12}t_{i} \end{matrix}$$

$$var(Y_{i1}) = d_{11} + 2d_{12}t_{i1} + d_{22}t_{i1}^2 + \sigma^2$$

On top of correlation in the errors, the marginal implies that the variance function of the response is quadratic over time, with positive curvature d_{22}

8.1.1 Random-Intercepts Model

If we remove the random slopes,

- the assumption is that all variability in subject-specific slopes can be attributed to treatment differences
- the model is random-intercepts model. This has subject specific intercepts, but the same slopes within each treatment group.

$$\mathbf{Y}_i|b_i \sim N(\mathbf{X}_i\beta + 1b_i, \Sigma_i)b_i \sim N(0, d_{11})$$

The marginal model is then ($_i = \sigma^2\mathbf{I}$)

$$\mathbf{Y}_i \sim N(\mathbf{X}_i\beta, 11'd_{11} + \sigma^2\mathbf{I})$$

The marginal covariance matrix is

$$cov(\mathbf{Y}_i) = 11'd_{11} + \sigma^2 I = \begin{pmatrix} d_{11} + \sigma^2 & d_{11} & ... & d_{11} \\ d_{11} & d_{11} + \sigma^2 & d_{11} & ... \\ . & . & . & . \\ d_{11} & ... & ... & d_{11} + \sigma^2 \end{pmatrix}$$

the associated correlation matrix is

$$corr(\mathbf{Y}_i) = \begin{pmatrix} 1 & \rho & ... & \rho \\ \rho & 1 & \rho & ... \\ . & . & . & . \\ \rho & ... & ... & 1 \end{pmatrix}$$

where $\rho \equiv \frac{d_{11}}{d_{11}+\sigma^2}$

Thu, we have

- constant variance over time
- equal, positive correlation between any two measurements from the same subject
- a covariance structure that is called **compound symmetry**, and ρ is called the **intra-class correlation**
- that when ρ is large, the **inter-subject variability** (d_{11}) is large relative to the intra-subject variability (σ^2)

8.1.2 Covariance Models

If the conditional independence assumption, ($_{\mathbf{i}} = \ {}^{\mathbf{2}}\mathbf{I}_{\mathbf{n_i}}$). Consider, $\epsilon_i = \epsilon_{(1)i} + \epsilon_{(2)i}$, where

- $\epsilon_{(1)i}$ is a "serial correlation" component. That is, part of the individual's profile is a response to time-varying stochastic processes.
- $\epsilon_{(2)i}$ is the measurement error component, and is independent of $\epsilon_{(1)i}$

Then

$$\mathbf{Y_i} = \mathbf{X_i} \quad + \mathbf{Z_i}\mathbf{b_i} + \quad _{\mathbf{(1)i}} + \quad _{\mathbf{(2)i}}$$

where

- $\mathbf{b_i} \sim N(\mathbf{0}, \mathbf{D})$
- $\epsilon_{(2)i} \sim N(\mathbf{0}, \ {}^{\mathbf{2}}\mathbf{I}_{\mathbf{n_i}})$
- $\epsilon_{(1)i} \sim N(\mathbf{0}, \ {}^{\mathbf{2}}\mathbf{H_i})$
- $\mathbf{b}_i$ and ϵ_i are mutually independent

To model the structure of the $n_i \times n_i$ correlation (or covariance) matrix $\mathbf{H}_i$. Let the (j,k)th element of $\mathbf{H}_i$ be $h_{ijk} = g(t_{ij}t_{ik})$. that is a function of the times t_{ij} and t_{ik} , which is assumed to be some function of the "distance' between the times.

$$h_{ijk} = g(|t_{ij} - t_{ik}|)$$

for some decreasing function $g(.)$ with $g(0) = 1$ (for correlation matrices).

Examples of this type of function:

- Exponential function: $g(|t_{ij} - t_{ik}|) = \exp(-\phi|t_{ij} - t_{ik}|)$
- Gaussian function: $g(|t_{ij} - t_{ik}|) = \exp(-\phi(t_{ij} - t_{ik})^2)$

Similar structures could also be used for **D** matrix (of **b**)

Example: Autoregressive Covariance Structure

A first order Autoregressive Model (AR(1)) has the form

$$\alpha_t = \phi\alpha_{t-1} + \eta_t$$

where $\eta_t \sim iidN(0, \sigma^2_\eta)$

Then, the covariance between two observations is

$$cov(\alpha_t, \alpha_{t+h}) = \frac{\sigma^2_\eta \phi^{|h|}}{1-\phi^2}$$

for $h = 0, \pm 1, \pm 2, ...; |\phi| < 1$

Hence,

$$corr(\alpha_t, \alpha_{t+h}) = \phi^{|h|}$$

If we let $\alpha_T = (\alpha_1, ...\alpha_T)'$, then

$$corr(\alpha_T) = \begin{bmatrix} 1 & \phi^1 & \phi^2 & ... & \phi^2 \\ \phi^1 & 1 & \phi^1 & ... & \phi^{T-1} \\ \phi^2 & \phi^1 & 1 & ... & \phi^{T-2} \\ . & . & . & . & . \\ \phi^T & \phi^{T-1} & \phi^{T-2} & ... & 1 \end{bmatrix}$$

Notes:

- The correlation decreases as time lag increases
- This matrix structure is known as a **Toeplitz** structure
- More complicated covariance structures are possible, which is critical component of spatial random effects models and time series models.
- Often, we don't need both random effects **b** and $\epsilon_{(1)i}$

More in the Time Series section

8.2 Estimation

$$\mathbf{Y}_i = \mathbf{X}_i\beta + \mathbf{Z}_i\mathbf{b}_i + \epsilon_i$$

where $\beta, \mathbf{b}_i, \mathbf{D}, {}_i$ we must obtain estimation from the data

- $, \mathbf{D}, {}_i$ are unknown, but fixed, parameters, and must be estimated from the data

- $\mathbf{b}_i$ is a random variable. Thus, we can't estimate these values, but we can predict them. (i.e., you can't estimate a random thing).

If we have

- $\hat{\beta}$ as an estimator of β
- $\mathbf{b}_i$ as a predictor of $\mathbf{b}_i$

Then,

- The population average estimate of $\mathbf{Y}_i$ is $\hat{\mathbf{Y}}_i = \mathbf{X}_i\hat{\beta}$
- The subject-specific prediction is $\hat{\mathbf{Y}}_i = \mathbf{X}_i\hat{\beta} + \mathbf{Z}_i\hat{b}_i$

According to (Henderson, 1950), estimating equations known as the mixed model equations:

$$\begin{bmatrix} \hat{\beta} \\ \hat{\mathbf{b}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}' \;{}^{-1}\mathbf{X} & \mathbf{X}' \;{}^{-1}\mathbf{Z} \\ \mathbf{Z}' \;{}^{-1}\mathbf{X} & \mathbf{Z}' \;{}^{-1}\mathbf{Z} + \mathbf{B}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{X}' \;{}^{-1}\mathbf{Y} \\ \mathbf{Z}' \;{}^{-1}\mathbf{Y} \end{bmatrix}$$

where

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 \\ . \\ \mathbf{y}_N \end{bmatrix}; \mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ . \\ \mathbf{X}_N \end{bmatrix}; \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ . \\ \mathbf{b}_N \end{bmatrix}; \epsilon = \begin{bmatrix} \epsilon_1 \\ . \\ \epsilon_N \end{bmatrix} cov(\epsilon) = \;, \mathbf{Z} = \begin{bmatrix} \mathbf{Z}_1 & 0 & ... & 0 \\ 0 & \mathbf{Z}_2 & ... & 0 \\ . & . & . & . \\ 0 & 0 & ... & \mathbf{Z}_n \end{bmatrix}, \mathbf{B} = \begin{bmatrix} \mathbf{D} & 0 & ... \\ 0 & \mathbf{D} & ... \\ . & . & . \\ 0 & 0 & ... \end{bmatrix}$$

The model has the form

$$\mathbf{Y} = \mathbf{X} \;\; + \mathbf{Z}\mathbf{b} + \;\; \mathbf{Y} \sim N(\mathbf{X} \;, \mathbf{Z}\mathbf{B}\mathbf{Z}' + \;\;)$$

If $\mathbf{V} = \mathbf{Z}\mathbf{B}\mathbf{Z}' + \;\;$, then the solutions to the estimating equations can be

$$\hat{\beta} = (\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{V}^{-1}\mathbf{Y}\hat{\mathbf{b}} = \mathbf{B}\mathbf{Z}'\mathbf{V}^{-1}(\mathbf{Y} - \hat{\mathbf{X}} \;)$$

The estimate $\hat{\beta}$ is a generalized least squares estimate.

The predictor, $\hat{\mathbf{b}}$ is the best linear unbiased predictor (BLUP), for $\mathbf{b}$

$$E(\hat{\beta}) = \beta var(\hat{\beta}) = (\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-1}E(\hat{\mathbf{b}}) = 0var(\hat{\mathbf{b}} - \mathbf{b}) = \mathbf{B} - \mathbf{B}\mathbf{Z}'\mathbf{V}^{-1}\mathbf{Z}\mathbf{B} + \mathbf{B}\mathbf{Z}'\mathbf{V}^{-1}\mathbf{X}(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{V}^{-1}\mathbf{B}$$

The variance here is the variance of the prediction error (mean squared prediction error, MSPE), which is more meaningful than $var(\hat{\mathbf{b}})$, since MSPE accounts for both variance and bias in the prediction.

To derive the mixed model equations, consider

$$= \mathbf{Y} - \mathbf{X}\ \ - \mathbf{Zb}$$

Let $T = \sum_{i=1}^{N} n_i$ be the total number of observations (i.e., the length of $\mathbf{Y}, \epsilon$) and Nq the length of $\mathbf{b}$. The joint distribution of $\mathbf{b},\ $ is

$$f(\mathbf{b},\ \ ) = \frac{1}{(2\pi)^{(T+Nq)/2}} \begin{vmatrix} \mathbf{B} & 0 \\ 0 & \end{vmatrix}^{-1/2} \exp\left(-\frac{1}{2} \begin{bmatrix} \mathbf{b} \\ \mathbf{Y} - \mathbf{X}\ \ - \mathbf{Zb} \end{bmatrix}' \begin{bmatrix} \mathbf{B} & 0 \\ 0 & \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{b} \\ \mathbf{Y} - \mathbf{X}\ \ - \mathbf{Zb} \end{bmatrix} \right)$$

Maximization of $f(\mathbf{b}, \epsilon)$ with respect to $\mathbf{b}$ and β requires minimization of

$$Q = \begin{bmatrix} \mathbf{b} \\ \mathbf{Y} - \mathbf{X}\ \ - \mathbf{Zb} \end{bmatrix}' \begin{bmatrix} \mathbf{B} & 0 \\ 0 & \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{b} \\ \mathbf{Y} - \mathbf{X}\ \ - \mathbf{Zb} \end{bmatrix} = \mathbf{b}'\mathbf{B^{-1}b} + (\mathbf{Y} - \mathbf{X}\ \ - \mathbf{Zb})'\ \mathbf{^{-1}}(\mathbf{Y} - \mathbf{X}\ \ - \mathbf{Zb})$$

Setting the derivatives of Q with respect to $\mathbf{b}$ and   to zero leads to the system of equations:

$$\begin{aligned} \mathbf{X}'\ \mathbf{^{-1}X}\ \ + \mathbf{X}'\ \mathbf{^{-1}Zb} &= \mathbf{X}'\ \mathbf{^{-1}Y} \\ (\mathbf{Z}'\ \mathbf{^{-1}Z} + \mathbf{B^{-1}})\mathbf{b} + \mathbf{Z}'\ \mathbf{^{-1}X}\ \ &= \mathbf{Z}'\ \mathbf{^{-1}Y} \end{aligned}$$

Rearranging

$$\begin{bmatrix} \mathbf{X}'\ \mathbf{^{-1}X} & \mathbf{X}'\ \mathbf{^{-1}Z} \\ \mathbf{Z}'\ \mathbf{^{-1}X} & \mathbf{Z}'\ \mathbf{^{-1}Z} + \mathbf{B^{-1}} \end{bmatrix} \begin{bmatrix} \beta \\ \mathbf{b} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\ \mathbf{^{-1}Y} \\ \mathbf{Z}'\ \mathbf{^{-1}Y} \end{bmatrix}$$

Thus, the solution to the mixed model equations give:

$$\begin{bmatrix} \hat{\beta} \\ \hat{\mathbf{b}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\ \mathbf{^{-1}X} & \mathbf{X}'\ \mathbf{^{-1}Z} \\ \mathbf{Z}'\ \mathbf{^{-1}X} & \mathbf{Z}'\ \mathbf{^{-1}Z} + \mathbf{B^{-1}} \end{bmatrix} \begin{bmatrix} \mathbf{X}'\ \mathbf{^{-1}Y} \\ \mathbf{Z}'\ \mathbf{^{-1}Y} \end{bmatrix}$$

Equivalently,

Bayes' theorem

$$f(\mathbf{b}|\mathbf{Y}) = \frac{f(\mathbf{Y}|\mathbf{b})f(\mathbf{b})}{\int f(\mathbf{Y}|\mathbf{b})f(\mathbf{b})d\mathbf{b}}$$

where

- $f(\mathbf{Y}|\mathbf{b})$ is the "likelihood"

- $f(\mathbf{b})$ is the prior
- the denominator is the "normalizing constant"
- $f(\mathbf{b}|\mathbf{Y})$ is the posterior distribution

In this case

$$\mathbf{Y}|\mathbf{b} \sim N(\mathbf{X}\ \ + \mathbf{Zb},\ )\mathbf{b} \sim N(\mathbf{0}, \mathbf{B})$$

The posterior distribution has the form

$$\mathbf{b}|\mathbf{Y} \sim N(\mathbf{BZ'V^{-1}}(\mathbf{Y} - \mathbf{X}\ ), (\mathbf{Z'}\ \mathbf{^{-1}Z} + \mathbf{B^{-1}})\mathbf{^{-1}})$$

Hence, the best predictor (based on squared error loss)

$$E(\mathbf{b}|\mathbf{Y}) = \mathbf{BZ'V^{-1}}(\mathbf{Y} - \mathbf{X}\ )$$

8.2.1 Estimating V

If we have $\tilde{\mathbf{V}}$ (estimate of $\mathbf{V}$), then we can estimate:

$$\hat{\beta} = (\mathbf{X'\tilde{V}^{-1}X})^{-1}\mathbf{X'\tilde{V}^{-1}Y}\hat{\mathbf{b}} = \mathbf{BZ'\tilde{V}^{-1}}(\mathbf{Y} - \mathbf{X}\hat{\ }\ )$$

where $\mathbf{b}$ is **EBLUP** (estimated BLUP) or **empirical Bayes estimate**

Note:

- $\hat{var}(\hat{\beta})$ is a consistent estimator of $var(\hat{\beta})$ if $\tilde{\mathbf{V}}$ is a consistent estimator of $\mathbf{V}$
- However, $\hat{var}(\hat{\beta})$ is biased since the variability arises from estimating $\mathbf{V}$ is not accounted for in the estimate.
- Hence, $\hat{var}(\hat{\beta})$ underestimates the true variability

Ways to estimate $\mathbf{V}$

- Maximum Likelihood Estimation (MLE)
- Restricted Maximum Likelihood (REML)
- Estimated Generalized Least Squares
- Bayesian Hierarchical Models (BHM)

8.2.1.1 Maximum Likelihood Estimation (MLE)

Grouping unknown parameters in Σ and B under a parameter vector θ. Under MLE, $\hat{\theta}$ and $\hat{\beta}$ maximize the likelihood $\mathbf{y} \sim N(\mathbf{X}\ , \mathbf{V}(\ ))$. Synonymously, $-2\log L(\mathbf{y};\ ,\ )$:

$$-2l(\ ,\ ,\mathbf{y}) = \log|\mathbf{V}(\ )| + (\mathbf{y} - \mathbf{X}\ )'\mathbf{V}(\ )^{\mathbf{-1}}(\mathbf{y} - \mathbf{X}\ ) + N\log(2\pi)$$

- Step 1: Replace β with its maximum likelihood (where θ is known $\hat{\beta} = (\mathbf{X'V}(\ )^{\mathbf{-1}}\mathbf{X})^{\mathbf{-1}}\mathbf{X'V}(\ )^{\mathbf{-1}}\mathbf{y}$
- Step 2: Minimize the above equation with respect to θ to get the estimator $\hat{\theta}_{MLE}$
- Step 3: Substitute $\hat{\theta}_{MLE}$ back to get $\hat{\beta}_{MLE} = (\mathbf{X'V}(\ _{\mathbf{MLE}})^{\mathbf{-1}}\mathbf{X})^{\mathbf{-1}}\mathbf{X'V}(\ _{\mathbf{MLE}})^{\mathbf{-1}}\mathbf{y}$
- Step 4: Get $\hat{\mathbf{b}}_{MLE} = \mathbf{B}(\hat{\ }_{\mathbf{MLE}})\mathbf{Z'V}(\hat{\ }_{\mathbf{MLE}})^{\mathbf{-1}}(\mathbf{y} - \mathbf{X}\hat{\ }_{\mathbf{MLE}})$

Note:

- $\hat{\theta}$ are typically negatively biased due to unaccounted fixed effects being estimated, which we could try to account for.

8.2.1.2 Restricted Maximum Likelihood (REML)

REML accounts for the number of estimated mean parameters by adjusting the objective function. Specifically, the likelihood of linear combination of the elements of $\mathbf{y}$ is accounted for.

We have $\mathbf{K'y}$, where $\mathbf{K}$ is any $N \times (N - p)$ full-rank contrast matrix, which has columns orthogonal to the $\mathbf{X}$ matrix (that is $\mathbf{K'X} = 0$). Then,

$$\mathbf{K'y} \sim N(0, \mathbf{K'V}(\ )\mathbf{K})$$

where β is no longer in the distribution

We can proceed to maximize this likelihood for the contrasts to get $\hat{\theta}_{REML}$, which does not depend on the choice of $\mathbf{K}$. And $\hat{\beta}$ are based on $\hat{\theta}$

Comparison REML and MLE

- Both methods are based upon the likelihood principle, and have desired properties for the estimates:
    - consistency
    - asymptotic normality
    - efficiency
- ML estimation provides estimates for fixed effects, while REML can't
- In balanced models, REML is identical to ANOVA
- REML accounts for df for the fixed effects int eh model, which is important when $\mathbf{X}$ is large relative to the sample size
- Changing   has no effect on the REML estimates of θ
- REML is less sensitive to outliers than MLE
- MLE is better than REML regarding model comparisons (e.g., AIC or BIC)

8.2.1.3 Estimated Generalized Least Squares

MLE and REML rely upon the Gaussian assumption. To overcome this issue, EGLS uses the first and second moments.

$$\mathbf{Y}_i = \mathbf{X}_i\beta + \mathbf{Z}_i\mathbf{b}_i + \epsilon_i$$

where

- $\epsilon_i \sim (\mathbf{0}, \ _\mathbf{i})$
- $\mathbf{b}_i \sim (\mathbf{0}, \mathbf{D})$
- $cov(\epsilon_i, \mathbf{b}_i) = 0$

Then the EGLS estimator is

$$\begin{aligned}
\hat{\beta}_{GLS} &= \{\sum_{i=1}^{n} \mathbf{X_i' V_i(\ )^{-1} X_i}\}^{-1} \sum_{i=1}^{n} \mathbf{X_i' V_i(\ )^{-1} Y_i} \\
&= \{\mathbf{X'V(\ )^{-1}X}\}^{-1}\mathbf{X'V(\ )^{-1}Y}
\end{aligned}$$

depends on the first two moments

- $E(\mathbf{Y}_i) = \mathbf{X}_i\beta$
- $var(\mathbf{Y}_i) = \mathbf{V}_i$

EGLS use $\hat{\mathbf{V}}$ for $\mathbf{V}(\ )$

$$\hat{\beta}_{EGLS} = \{\mathbf{X'\hat{V}^{-1}X}\}^{-1}\mathbf{X'\hat{V}^{-1}Y}$$

Hence, the fixed effects estimators for the MLE, REML, and EGLS are of the same form, except for the estimate of $\mathbf{V}$

In case of non-iterative approach, EGLS can be appealing when $\mathbf{V}$ can be estimated without much computational burden.

8.2.1.4 Bayesian Hierarchical Models (BHM)

Joint distribution cane be decomposed hierarchically in terms of the product of conditional distributions and a marginal distribution

$$f(A,B,C) = f(A|B,C)f(B|C)f(C)$$

Applying to estimate $\mathbf{V}$

$$\begin{aligned}
f(\mathbf{Y}, \ ,\mathbf{b}, \ ) &= f(\mathbf{Y}| \ ,\mathbf{b}, \ )f(\mathbf{b}| \ , \ )f(\ | \ )f(\ ) && \text{based on probability decomposition} \\
&= f(\mathbf{Y}| \ ,\mathbf{b}, \ )f(\mathbf{b}| \ )f(\ )f(\ ) && \text{based on simplifying modeling assumptions}
\end{aligned}$$

elaborate on the second equality, if we assume conditional independence (e.g., given θ, no additional info about $\mathbf{b}$ is given by knowing β), then we can simply from the first equality

Using Bayes' rule

$$f(\ ,\mathbf{b},\ |\mathbf{Y}) \propto f(\mathbf{Y}|\ ,\mathbf{b},\ )f(\mathbf{b}|\ )f(\ )f(\ )$$

where

$$\mathbf{Y}|\ ,\mathbf{b},\ \sim \mathbf{N}(\mathbf{X}\ +\mathbf{Zb},\ (\ ))\mathbf{b}|\ \sim \mathbf{N}(\mathbf{0},\mathbf{B}(\ ))$$

and we also have to have prior distributions for $f(\beta), f(\theta)$

With normalizing constant, we can obtain the posterior distribution. Typically, we can't get analytical solution right away. Hence, we can use Markov Chain Monte Carlo (MCMC) to obtain samples from the posterior distribution.

Bayesian Methods:

- account for the uncertainty in parameters estimates and accommodate the propagation of that uncertainty through the model
- can adjust prior information (i.e., priori) in parameters
- Can extend beyond Gaussian distributions
- but hard to implement algorithms and might have problem converging

8.3 Inference

8.3.1 Parameters β

8.3.1.1 Wald test

We have

$$\hat{}\,(\ ) = \{\mathbf{X}'\mathbf{V}^{-\mathbf{1}}(\ )\mathbf{X}\}^{-\mathbf{1}}\mathbf{X}'\mathbf{V}^{-\mathbf{1}}(\ )\mathbf{Y}var(\hat{\beta}(\theta)) = \{\mathbf{X}'\mathbf{V}^{-\mathbf{1}}(\ )\mathbf{X}\}^{-\mathbf{1}}$$

We can use $\hat{\theta}$ in place of θ to approximate Wald test

$$H_0 : \mathbf{A}\ = \mathbf{d}$$

With

$$W = (\mathbf{A}\hat{}\ - \mathbf{d})'[\mathbf{A}(\mathbf{X}'\hat{\mathbf{V}}^{-\mathbf{1}}\mathbf{X})^{-\mathbf{1}}\mathbf{A}']^{-\mathbf{1}}(\mathbf{A}\hat{}\ - \mathbf{d})$$

where $W \sim \chi^2_{rank(A)}$ under H_0 is true. However, it does not take into account variability from using $\hat{\theta}$ in place of θ, hence the standard errors are underestimated

8.3.1.2 F-test

Alternatively, we can use the modified F-test, suppose we have $var(\mathbf{Y}) = \sigma^2 \mathbf{V}(\theta)$, then

$$F^* = \frac{(\mathbf{A}\hat{} - \mathbf{d})'[\mathbf{A}(\mathbf{X}'\hat{\mathbf{V}}^{-\mathbf{1}}\mathbf{X})^{-\mathbf{1}}\mathbf{A}']^{-\mathbf{1}}(\mathbf{A}\hat{} - \mathbf{d})}{\hat{\sigma}^2 \text{rank}(A)}$$

where $F^* \sim f_{rank(A),den(df)}$ under the null hypothesis. And den(df) needs to be approximated from the data by either:

- Satterthwaite method
- Kenward-Roger approximation

Under balanced cases, the Wald and F tests are similar. But for small sample sizes, they can differ in p-values. And both can be reduced to t-test for a single β

8.3.1.3 Likelihood Ratio Test

$$H_0 : \beta \in \Theta_{\beta,0}$$

where $\Theta_{\beta,0}$ is a subspace of the parameter space, Θ_β of the fixed effects β . Then

$$-2\log\lambda_N = -2\log\{\frac{\hat{L}_{ML,0}}{\hat{L}_{ML}}\}$$

where

- $\hat{L}_{ML,0}$, $\hat{L}_{ML}$ are the maximized likelihood obtained from maximizing over $\Theta_{\beta,0}$ and Θ_β
- $-2\log\lambda_N \dot{\sim} \chi^2_{df}$ where df is the difference in the dimension (i.e., number of parameters) of $\Theta_{\beta,0}$ and Θ_β

This method is not applicable for REML. But REML can still be used to test for covariance parameters between nested models.

8.3.2 Variance Components

- For ML and REML estimator, $\hat{\theta} \sim N(\theta, I(\theta))$ for large samples
- Wald test in variance components is analogous to the fixed effects case (see 8.3.1.1)

  - However, the normal approximation depends largely on the true value of θ. It will fail if the true value of θ is close to the boundary of the parameter space Θ_θ (i.e., $\sigma^2 \approx 0$)
  - Typically works better for covariance parameter, than vairance prarmetesr.
- The likelihood ratio tests can also be used with ML or REML estimates. However, the same problem of parameters

8.4 Information Criteria

- account for the likelihood and the number of parameters to assess model comparison.

8.4.1 Akaike's Information Criteria (AIC)

Derived as an estimator of the expected Kullback discrepancy between the true model and a fitted candidate model

$$AIC = -2l(\hat{\theta}, \hat{\beta}) + 2q$$

where

- $l(\hat{\theta}, \hat{\beta})$ is the log-likelihood
- q = the effective number of parameters; total of fixed and those associated with random effects (variance/covariance; those not estimated to be on a boundary constraint)

Note:

- In comparing models that differ in their random effects, this method is not advised to due the inability to get the correct number of effective parameters).
- We prefer smaller AIC values.
- If your program uses $l - q$ then we prefer larger AIC values (but rarely).
- can be used for mixed model section, (e.g., selection of the covariance structure), but the sample size must be very large to have adequate comparison based on the criterion
- Can have a large negative bias (e.g., when sample size is small but the number of parameters is large) due to the penalty term can't approximate the bias adjustment adequately

8.4.2 Corrected AIC (AICC)

- developed by (HURVICH and TSAI, 1989)
- correct small-sample adjustment

- depends on the candidate model class
- Only if you have fixed covariance structure, then AICC is justified, but not general covariance structure

8.4.3 Bayesian Information Criteria (BIC)

$$BIC = -2l(\hat{\theta}, \hat{\beta}) + q \log n$$

where n = number of observations.

- we prefer smaller BIC value
- BIC and AIC are used for both REML and MLE if we have the same mean structure. Otherwise, in general, we should prefer MLE

With our example presented at the beginning of Linear Mixed Models,

$$Y_{ik} = \begin{cases} \beta_0 + b_{1i} + (\beta_1 + \ b_{2i})t_{ij} + \epsilon_{ij} & L \\ \beta_0 + b_{1i} + (\beta_2 + \ b_{2i})t_{ij} + \epsilon_{ij} & H \\ \beta_0 + b_{1i} + (\beta_3 + \ b_{2i})t_{ij} + \epsilon_{ij} & C \end{cases}$$

where

- $i = 1, .., N$
- $j = 1, .., n_i$ (measures at time t_{ij})

Note:

- we have subject-specific intercepts,

$$\mathbf{Y}_i | b_i \sim N(\mathbf{X}_i \beta + 1 b_i, \sigma^2 \mathbf{I}) b_i \sim N(0, d_{11})$$

here, we want to estimate β, σ^2, d_{11} and predict b_i

8.5 Split-Plot Designs

- Typically used in the case that you have two factors where one needs much larger units than the other.

Example:

A: 3 levels (large units)

B: 2 levels (small units)

- A and B levels are randomized into 4 blocks.
- But it differs from Randomized Block Designs. In each block, both have one of the 6 (3x2) treatment combinations. But Randomized Block Designs assign in each block randomly, while split-plot does not randomize this step.

- Moreover, because A needs to be applied in large units, factor A is applied only once in each block while B can be applied multiple times.

Hence, we have our model

If A is our factor of interest

$$Y_{ij} = \mu + \rho_i + \alpha_j + e_{ij}$$

where

- i = replication (block or subject)
- j = level of Factor A
- μ = overall mean
- ρ_i = variation due to the i-th block
- $e_{ij} \sim N(0, \sigma^2_e)$ = whole plot error

If B is our factor of interest

$$Y_{ijk} = \mu + \phi_{ij} + \beta_k + \epsilon_{ijk}$$

where

- ϕ_{ij} = variation due to the ij-th main plot
- β_k = Factor B effect
- $\epsilon_{ijk} \sim N(0, \sigma^2_\epsilon)$ = subplot error
- $\phi_{ij} = \rho_i + \alpha_j + e_{ij}$

Together, the split-plot model

$$Y_{ijk} = \mu + \rho_i + \alpha_j + e_{ij} + \beta_k + (\alpha\beta)_{jk} + \epsilon_{ijk}$$

where

- i = replicate (blocks or subjects)
- j = level of factor A
- k = level of factor B
- μ = overall mean
- ρ_i = effect of the block
- α_j = main effect of factor A (fixed)
- $e_{ij} = (\rho\alpha)_{ij}$ = block by factor A interaction (the whole plot error, random)
- β_k = main effect of factor B (fixed)
- $(\alpha\beta)_{jk}$ = interaction between factors A and B (fixed)
- ϵ_{ijk} = subplot error (random)

We can approach sub-plot analysis based on

- the ANOVA perspective

    - Whole plot comparisons
        - Compare factor A to the whole plot error (i.e., α_j to e_{ij})
        - Compare the block to the whole plot error (i.e., ρ_i to e_{ij})
    - Sub-plot comparisons:
        - Compare factor B to the subplot error (β to ϵ_{ijk})
        - Compare the AB interaction to the subplot error ($(\alpha\beta)_{jk}$ to ϵ_{ijk})
- the mixed model perspective

$$\mathbf{Y} = \mathbf{X} \ + \mathbf{Zb} +$$

8.5.1 Application

8.5.1.1 Example 1

$$y_{ijk} = \mu + i_i + v_j + (iv)_{ij} + f_k + \epsilon_{ijk}$$

where

- y_{ijk} = observed yield
- μ = overall average yield
- i_i = irrigation effect
- v_j = variety effect
- $(iv)_{ij}$ = irrigation by variety interaction
- f_k = random field (block) effect
- ϵ_{ijk} = residual
- because variety-field combination is only observed once, we can't have the random interaction effects between variety and field

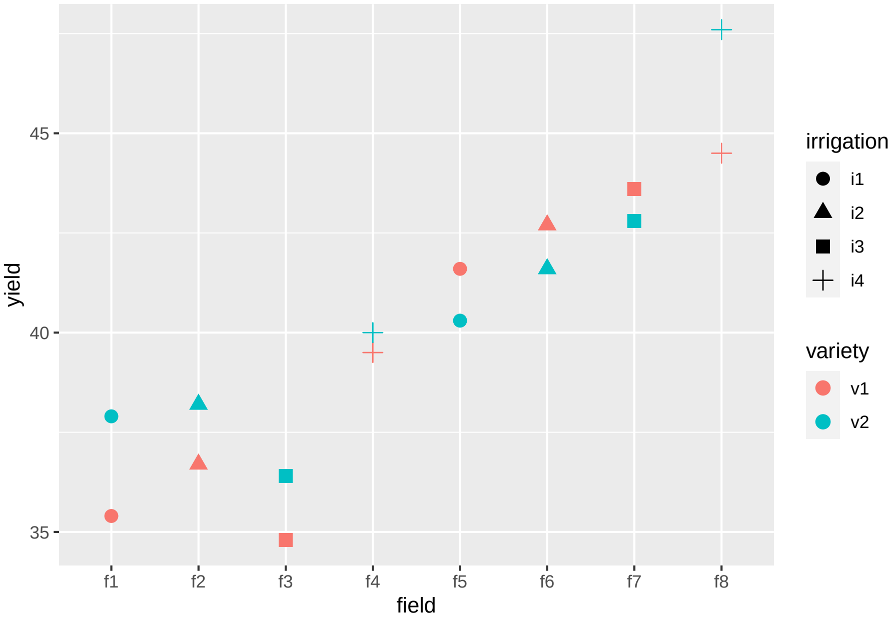
```
library(ggplot2)
data(irrigation, package = "faraway")
summary(irrigation)
```

```
##      field   irrigation variety     yield      
##  f1     :2   i1:4       v1:8    Min.   :34.80  
##  f2     :2   i2:4       v2:8    1st Qu.:37.60  
##  f3     :2   i3:4               Median :40.15  
##  f4     :2   i4:4               Mean   :40.23  
##  f5     :2                      3rd Qu.:42.73  
##  f6     :2                      Max.   :47.60  
##  (Other):4
```

```
head(irrigation, 4)
```

```
##   field irrigation variety yield
## 1    f1         i1      v1  35.4
## 2    f1          i1      v2  37.9
## 3    f2          i2      v1  36.7
## 4    f2          i2      v2  38.2
```

```
ggplot(irrigation,
       aes(
         x = field,
         y = yield,
         shape = irrigation,
         color = variety
       )) +
  geom_point(size = 3)
```



```
sp_model <- lmerTest::lmer(yield ~ irrigation * variety + (1 | field), irrigation)
summary(sp_model)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: yield ~ irrigation * variety + (1 | field)
##    Data: irrigation
##
## REML criterion at convergence: 45.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.7448 -0.5509  0.0000  0.5509  0.7448
```

```
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  field    (Intercept) 16.200   4.025
##  Residual              2.107   1.452
## Number of obs: 16, groups:  field, 8
##
## Fixed effects:
##                          Estimate Std. Error     df t value Pr(>|t|)
## (Intercept)                38.500      3.026  4.487  12.725 0.000109 ***
## irrigationi2                1.200      4.279  4.487   0.280 0.791591
## irrigationi3                0.700      4.279  4.487   0.164 0.877156
## irrigationi4                3.500      4.279  4.487   0.818 0.454584
## varietyv2                   0.600      1.452  4.000   0.413 0.700582
## irrigationi2:varietyv2     -0.400      2.053  4.000  -0.195 0.855020
## irrigationi3:varietyv2     -0.200      2.053  4.000  -0.097 0.927082
## irrigationi4:varietyv2      1.200      2.053  4.000   0.584 0.590265
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) irrgt2 irrgt3 irrgt4 vrtyv2 irr2:2 irr3:2
## irrigation2 -0.707
## irrigation3 -0.707  0.500
## irrigation4 -0.707  0.500  0.500
## varietyv2   -0.240  0.170  0.170  0.170
## irrgtn2:vr2  0.170 -0.240 -0.120 -0.120 -0.707
## irrgtn3:vr2  0.170 -0.120 -0.240 -0.120 -0.707  0.500
## irrgtn4:vr2  0.170 -0.120 -0.120 -0.240 -0.707  0.500  0.500
```

```
anova(sp_model,ddf = c("Kenward-Roger"))
```

```
## Type III Analysis of Variance Table with Kenward-Roger's method
##                     Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## irrigation          2.4545 0.81818     3     4  0.3882 0.7685
## variety             2.2500 2.25000     1     4  1.0676 0.3599
## irrigation:variety 1.5500 0.51667     3     4  0.2452 0.8612
```

Since p-value of the interaction term is insignificant, we consider fitting without it.

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
sp_model_additive <- lmer(yield ~ irrigation + variety + (1 | field), irrigation)
anova(sp_model_additive,sp_model,ddf = "Kenward-Roger")
```

```
## refitting model(s) with ML (instead of REML)

## Data: irrigation
## Models:
## sp_model_additive: yield ~ irrigation + variety + (1 | field)
## sp_model: yield ~ irrigation * variety + (1 | field)
##                   npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## sp_model_additive    7 83.959 89.368 -34.980   69.959
## sp_model            10 88.609 96.335 -34.305   68.609 1.3503  3     0.7172
```

Since p-value of Chi-square test is insignificant, we can't reject the additive model is already sufficient. Looking at AIC and BIC, we can also see that we would prefer the additive model

Random Effect Examination

`exactRLRT` test

- H_0: Var(random effect) (i.e., σ^2)= 0
- H_a: Var(random effect) (i.e., σ^2) > 0

```
sp_model <- lme4::lmer(yield ~ irrigation * variety + (1 | field), irrigation)
library(RLRsim)
exactRLRT(sp_model)
```

```
##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
## data:
## RLRT = 6.1118, p-value = 0.0088
```

Since the p-value is significant, we reject H_0

8.6 Repeated Measures in Mixed Models

$$Y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \delta_{i(k)} + \epsilon_{ijk}$$

where

- i-th group (fixed)
- j-th (repeated measure) time effect (fixed)
- k-th subject
- $\delta_{i(k)} \sim N(0, \sigma^2_\delta)$ (k-th subject in the i-th group) and $\epsilon_{ijk} \sim N(0, \sigma^2)$ (independent error) are random effects $(i = 1, .., n_A, j = 1, .., n_B, k = 1, ..., n_i)$

hence, the variance-covariance matrix of the repeated observations on the k-th subject of the i-th group, $\mathbf{Y}_{ik} = (Y_{i1k}, .., Y_{in_Bk})'$, will be

$$
\begin{aligned}
{}_{subject} &= \begin{pmatrix} \sigma_\delta^2 + \sigma^2 & \sigma_\delta^2 & ... & \sigma_\delta^2 \\ \sigma_\delta^2 & \sigma_\delta^2 + \sigma^2 & ... & \sigma_\delta^2 \\ . & . & . & . \\ \sigma_\delta^2 & \sigma_\delta^2 & ... & \sigma_\delta^2 + \sigma^2 \end{pmatrix} \\
&= (\sigma_\delta^2 + \sigma^2) \begin{pmatrix} 1 & \rho & ... & \rho \\ \rho & 1 & ... & \rho \\ . & . & . & . \\ \rho & \rho & ... & 1 \end{pmatrix} && \text{product of a scalar and a correlation matrix}
\end{aligned}
$$

where $\rho = \frac{\sigma_\delta^2}{\sigma_\delta^2 + \sigma^2}$, which is the compound symmetry structure that we discussed in Random-Intercepts Model

But if you only have repeated measurements on the subject over time, AR(1) structure might be more appropriate

Mixed model for a repeated measure

$$
Y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}
$$

where

- ϵ_{ijk} combines random error of both the whole and subplots.

In general,

$$
\mathbf{Y} = \mathbf{X} \ +
$$

where

- $\epsilon \sim N(0, \sigma^2 \ )$ where is block diagonal if the random error covariance is the same for each subject

The variance covariance matrix with AR(1) structure is

$$
{}_{subject} = \sigma^2 \begin{pmatrix} 1 & \rho & \rho^2 & ... & \rho^{n_B - 1} \\ \rho & 1 & \rho & ... & \rho^{n_B - 2} \\ . & . & . & . & . \\ \rho^{n_B - 1} & \rho^{n_B - 2} & \rho^{n_B - 3} & ... & 1 \end{pmatrix}
$$

Hence, the mixed model for a repeated measure can be written as

$$
Y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}
$$

where

- ϵ_{ijk} = random error of whole and subplots

Generally,

$$\mathbf{Y} = \mathbf{X} \; +$$

where $\epsilon \sim N(0, \;^{\mathbf{2}}\;)$ and Σ = block diagonal if the random error covariance is the same for each subject.

8.7 Unbalanced or Unequally Spaced Data

Consider the model

$$Y_{ikt} = \beta_0 + \beta_{0i} + \beta_1 t + \beta_{1i} t + \beta_2 t^2 + \beta_{2i} t^2 + \epsilon_{ikt}$$

where

- i = 1,2 (groups)
- $k = 1, ..., n_i$ (individuals)
- $t = (t_1, t_2, t_3, t_4)$ (times)
- β_{2i} = common quadratic term
- β_{1i} = common linear time trends
- β_{0i} = common intercepts

Then, we assume the variance-covariance matrix of the repeated measurements collected on a particular subject over time has the form

$$_{ik} = \sigma^2 \left(\begin{array}{cccc} 1 & \rho^{t_2-t_1} & \rho^{t_3-t_1} & \rho^{t_4-t_1} \\ \rho^{t_2-t_1} & 1 & \rho^{t_3-t_2} & \rho^{t_4-t_2} \\ \rho^{t_3-t_1} & \rho^{t_3-t_2} & 1 & \rho^{t_4-t_3} \\ \rho^{t_4-t_1} & \rho^{t_4-t_2} & \rho^{t_4-t_3} & 1 \end{array} \right)$$

which is called “power” covariance model

We can consider $\beta_{2i}, \beta_{1i}, \beta_{0i}$ accordingly to see whether these terms are needed in the final model

8.8 Application

R Packages for mixed models

- `nlme`
    - has nested structure
    - flexible for complex design

    - not user-friendly

- `lme4`
    - computationally efficient
    - user-friendly
    - can handle nonnormal response
    - for more detailed application, check Fitting Linear Mixed-Effects Models Using lme4
- Others
    - Bayesian setting: `MCMCglmm`, `brms`
    - For genetics: `ASReml`

8.8.1 Example 1 (Pulps)

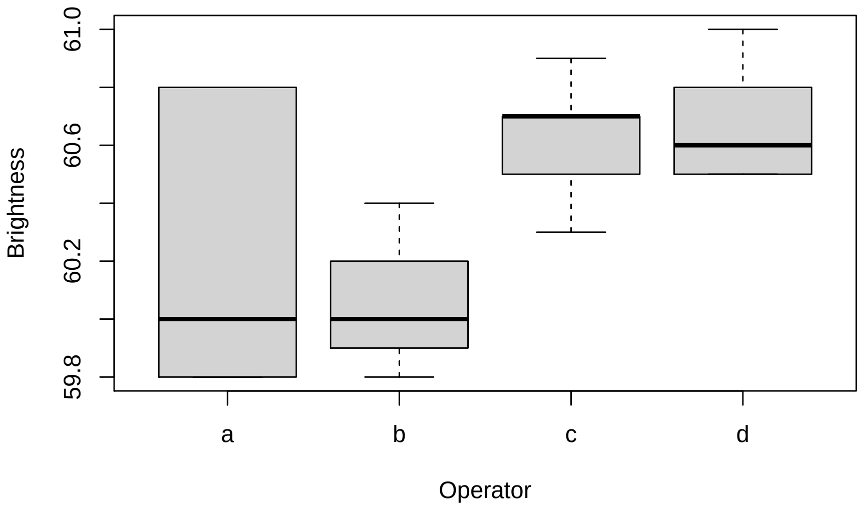
Model:

$$y_{ij} = \mu + \alpha_i + \epsilon_{ij}$$

where

- $i = 1,..,a$ groups for random effect α_i
- $j = 1,...,n$ individuals in each group
- $\alpha_i \sim N(0, \sigma^2_\alpha)$ is random effects
- $\epsilon_{ij} \sim N(0, \sigma^2_\epsilon)$ is random effects
- Imply compound symmetry model where the intraclass correlation coefficient is: $\rho = \frac{\sigma^2_\alpha}{\sigma^2_\alpha + \sigma^2_\epsilon}$
- If factor a does not explain much variation, low correlation within the levels: $\sigma^2_\alpha \to 0$ then $\rho \to 0$
- If factor a explain much variation, high correlation within the levels $\sigma^2_\alpha \to \infty$ hence, $\rho \to 1$

```
data(pulp, package = "faraway")
plot(
  y = pulp$bright,
  x = pulp$operator,
  xlab = "Operator",
  ylab = "Brightness"
)
```



```
pulp %>% dplyr::group_by(operator) %>% dplyr::summarise(average = mean(bright))
```

```
## # A tibble: 4 x 2
##   operator average
##   <fct>      <dbl>
## 1 a           60.2
## 2 b           60.1
## 3 c           60.6
## 4 d           60.7
```

`lmer` application

```
library(lme4)
mixed_model <- lmer(formula = bright ~ 1 + (1 | operator), # pipe (i..e, | ) denotes r
                    data = pulp)
summary(mixed_model)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: bright ~ 1 + (1 | operator)
##    Data: pulp
##
## REML criterion at convergence: 18.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.4666 -0.7595 -0.1244  0.6281  1.6012
```

```
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  operator (Intercept) 0.06808  0.2609
##  Residual             0.10625  0.3260
## Number of obs: 20, groups:  operator, 4
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  60.4000     0.1494   404.2
```

```
coef(mixed_model)
```

```
## $operator
##   (Intercept)
## a    60.27806
## b    60.14088
## c    60.56767
## d    60.61340
##
## attr(,"class")
## [1] "coef.mer"
```

```
fixef(mixed_model) # fixed effects
```

```
## (Intercept)
##        60.4
```

```
confint(mixed_model) # confidence interval
```

```
## Computing profile confidence intervals ...

##                 2.5 %     97.5 %
## .sig01       0.000000  0.6178987
## .sigma       0.238912  0.4821845
## (Intercept) 60.071299 60.7287012
```

```
ranef(mixed_model) # random effects
```

```
## $operator
##   (Intercept)
## a  -0.1219403
## b  -0.2591231
## c   0.1676679
## d   0.2133955
##
## with conditional variances for "operator"
```

```
VarCorr(mixed_model) # random effects standard deviation
```

```
##  Groups   Name        Std.Dev.
##  operator (Intercept) 0.26093
##  Residual             0.32596
```

```
re_dat = as.data.frame(VarCorr(mixed_model))
rho = re_dat[1,'vcov']/(re_dat[1,'vcov'] + re_dat[2,'vcov']) # rho based on the above
rho
```

```
## [1] 0.3905354
```

To Satterthwaite approximation for the denominator df, we use `lmerTest`

```
library(lmerTest)
```

```
##
## Attaching package: 'lmerTest'
```

```
## The following object is masked from 'package:lme4':
##
##     lmer
```

```
## The following object is masked from 'package:stats':
##
##     step
```

```
summary(lmerTest::lmer(bright ~ 1 + (1 | operator), pulp))$coefficients
```

```
##             Estimate Std. Error df  t value     Pr(>|t|)
## (Intercept)     60.4  0.1494434  3 404.1664 3.340265e-08
```

```
confint(mixed_model)[3,]
```

```
## Computing profile confidence intervals ...
```

```
##   2.5 %  97.5 %
## 60.0713 60.7287
```

In this example, we can see that the confidence interval computed by `confint` in `lmer` package is very close is `confint` in `lmerTest` model.

`MCMglmm` application

under the Bayesian framework

```
library(MCMCglmm)
```

```
## Loading required package: coda
```

```
## Loading required package: ape
```

```
mixed_model_bayes <- MCMCglmm(bright~1,random=~operator, data=pulp, verbose=FALSE)
summary(mixed_model_bayes)$solutions
```

```
##             post.mean l-95% CI u-95% CI eff.samp pMCMC
## (Intercept)  60.39357 60.07742 60.66047 1184.638 0.001
```

this method offers the confidence interval slightly more positive than `lmer` and `lmerTest`

8.8.1.1 Prediction

```
# random effects prediction (BLUPs)
ranef(mixed_model)$operator
```

```
##   (Intercept)
## a  -0.1219403
## b  -0.2591231
## c   0.1676679
## d   0.2133955
```

```
fixef(mixed_model) + ranef(mixed_model)$operator #prediction for each categories
```

```
##   (Intercept)
## a    60.27806
## b    60.14088
## c    60.56767
## d    60.61340
```

```
predict(mixed_model, newdata=data.frame(operator=c('a','b','c','d'))) # equivalent to the above m
```

```
##        1        2        3        4
## 60.27806 60.14088 60.56767 60.61340
```

use `bootMer()` to get bootstrap-based confidence intervals for predictions.

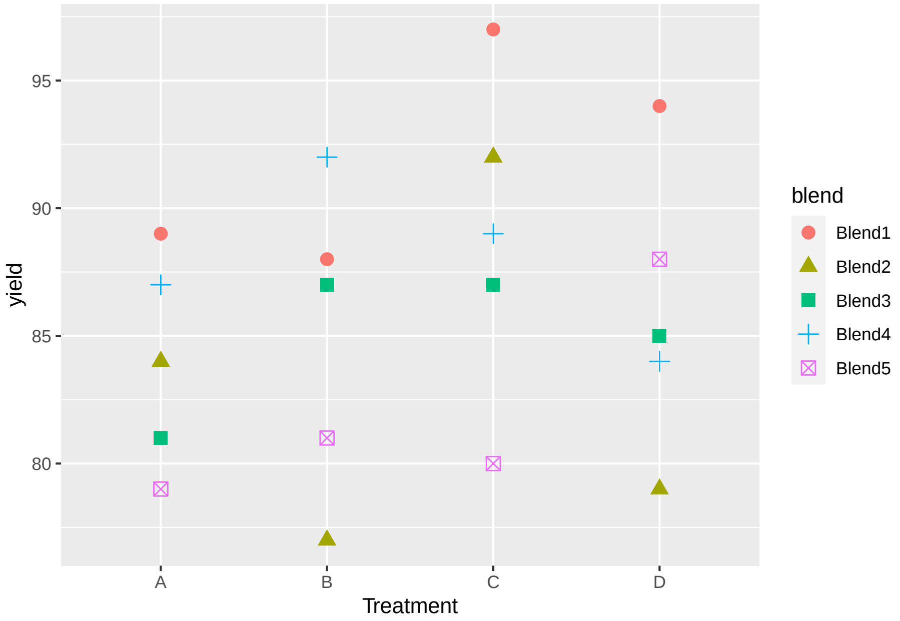
Another example using GLMM in the context of blocking

Penicillin data

```
data(penicillin, package = "faraway")
summary(penicillin)
```

```
##  treat    blend       yield   
##  A:5   Blend1:4   Min.   :77  
##  B:5   Blend2:4   1st Qu.:81  
##  C:5   Blend3:4   Median :87  
##  D:5   Blend4:4   Mean   :86  
##        Blend5:4   3rd Qu.:89  
##                   Max.   :97  
```

```r
library(ggplot2)
ggplot(penicillin, aes(
  y = yield,
  x = treat,
  shape = blend,
  color = blend
)) + # treatment = fixed effect, blend = random effects
  geom_point(size = 3) +
  xlab("Treatment")
```



```r
library(lmerTest) # for p-values
mixed_model <- lmerTest::lmer(yield ~ treat + (1 | blend),
                              data = penicillin)
summary(mixed_model)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: yield ~ treat + (1 | blend)
##    Data: penicillin
##
## REML criterion at convergence: 103.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.4152 -0.5017 -0.1644  0.6830  1.2836
```

```
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  blend    (Intercept) 11.79    3.434
##  Residual             18.83    4.340
## Number of obs: 20, groups:  blend, 5
##
## Fixed effects:
##             Estimate Std. Error     df t value Pr(>|t|)
## (Intercept)   84.000      2.475 11.075  33.941 1.51e-12 ***
## treatB         1.000      2.745 12.000   0.364   0.7219
## treatC         5.000      2.745 12.000   1.822   0.0935 .
## treatD         2.000      2.745 12.000   0.729   0.4802
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##        (Intr) treatB treatC
## treatB -0.555
## treatC -0.555  0.500
## treatD -0.555  0.500  0.500
```

```
#The BLUPs for the each blend
ranef(mixed_model)$blend
```

```
##        (Intercept)
## Blend1   4.2878788
## Blend2  -2.1439394
## Blend3  -0.7146465
## Blend4   1.4292929
## Blend5  -2.8585859
```

Examine treatment effect

```
anova(mixed_model) # p-value based on lmerTest
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##       Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## treat     70  23.333     3    12  1.2389 0.3387
```

Since the p-value is greater than 0.05, we can't reject the null hypothesis that there is no treatment effect.

```
library(pbkrtest)
full_model <- lmer(yield ~ treat + (1 | blend), penicillin, REML=FALSE) #REML is not appropriate
null_model <- lmer(yield ~ 1 + (1 | blend), penicillin, REML=FALSE)
KRmodcomp(full_model, null_model) # use  Kenward-Roger approximation for df
```

```
## large : yield ~ treat + (1 | blend)
## small : yield ~ 1 + (1 | blend)
##          stat     ndf     ddf F.scaling p.value
## Ftest  1.2389  3.0000 12.0000         1  0.3387
```

Since the p-value is greater than 0.05, and consistent with our previous observation, we conclude that we can't reject the null hypothesis that there is no treatment effect.

8.8.2 Example 2 (Rats)

```r
rats <- read.csv(
    "images/rats.dat",
    header = F,
    sep = ' ',
    col.names = c('Treatment', 'rat', 'age', 'y')
)
rats$t <- log(1 + (rats$age - 45)/10) #log transformed age
```

We are interested in whether treatment effect induces changes over time.

```r
rat_model <- lmerTest::lmer(y~t:Treatment+(1|rat),data=rats) #treatment = fixed effect
summary(rat_model)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: y ~ t:Treatment + (1 | rat)
##    Data: rats
##
## REML criterion at convergence: 932.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.25574 -0.65898 -0.01163  0.58356  2.88309
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  rat      (Intercept) 3.565    1.888
##  Residual             1.445    1.202
## Number of obs: 252, groups:  rat, 50
##
## Fixed effects:
##                Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)     68.6074     0.3312  89.0275  207.13   <2e-16 ***
## t:Treatmentcon   7.3138     0.2808 247.2762   26.05   <2e-16 ***
## t:Treatmenthig   6.8711     0.2276 247.7097   30.19   <2e-16 ***
## t:Treatmentlow   7.5069     0.2252 247.5196   33.34   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) t:Trtmntc t:Trtmnth
## t:Tretmntcn -0.327
## t:Tretmnthg -0.340  0.111
## t:Tretmntlw -0.351  0.115     0.119
```

```
anova(rat_model)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##             Sum Sq Mean Sq NumDF  DenDF F value    Pr(>F)
## t:Treatment 3181.9  1060.6     3 223.21  734.11 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Since the p-value is significant, we can be confident concluding that there is a treatment effect

8.8.3 Example 3 (Agridat)

```
library(agridat)
library(latticeExtra)
```

```
## Loading required package: lattice
```

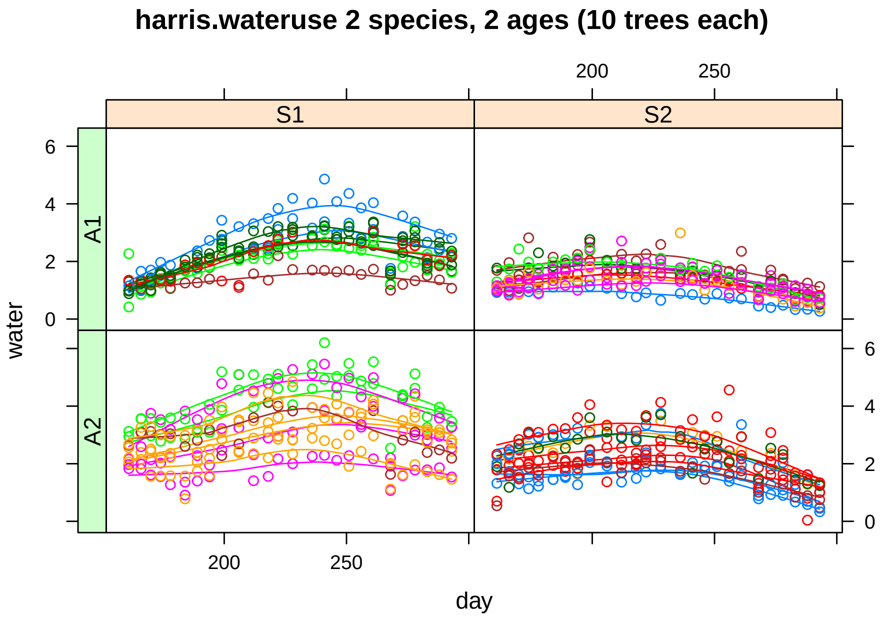
```
##
## Attaching package: 'lattice'
```

```
## The following object is masked from 'package:faraway':
##
##     melanoma
```

```
##
## Attaching package: 'latticeExtra'
```

```
## The following object is masked from 'package:ggplot2':
##
##     layer
```

```
dat <- harris.wateruse
# Compare to Schabenberger & Pierce, fig 7.23
useOuterStrips(
  xyplot(
    water ~ day | species * age,
    dat,
    as.table = TRUE,
    group = tree,
    type = c('p', 'smooth'),
    main = "harris.wateruse 2 species, 2 ages (10 trees each)"
  )
)
```

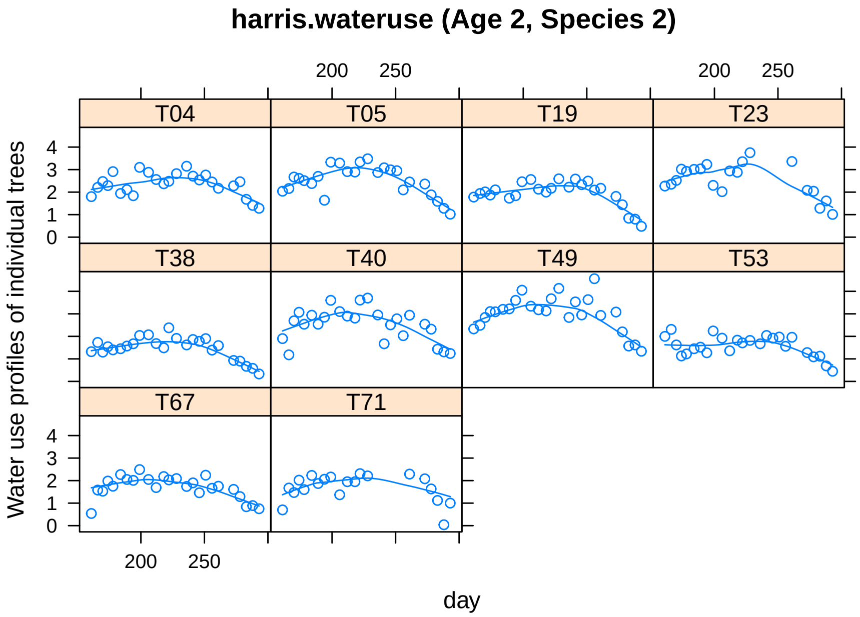


Remove outliers

```
dat <- subset(dat, day!=268)
```

Plot between age and species

```
xplot(
  water ~ day | tree,
  dat,
  subset = age == "A2" & species == "S2",
  as.table = TRUE,
  type = c('p', 'smooth'),
  ylab = "Water use profiles of individual trees",
  main = "harris.wateruse (Age 2, Species 2)"
)
```



```
# Rescale day for nicer output, and convergence issues, add quadratic term
dat <- transform(dat, ti = day / 100)
dat <- transform(dat, ti2 = ti * ti)
# Start with a subgroup: age 2, species 2
d22 <- droplevels(subset(dat, age == "A2" & species == "S2"))
```

`lme` function from `nlme` package

```
library(nlme)
```

```
##
## Attaching package: 'nlme'

## The following object is masked from 'package:lme4':
##
##     lmList

## The following object is masked from 'package:dplyr':
##
##     collapse
```

```
## We use pdDiag() to get uncorrelated random effects
m1n <- lme(
    water ~ 1 + ti + ti2, #intercept, time and time-squared = fixed effects
    data = d22,
    na.action = na.omit,
    random = list(tree = pdDiag( ~ 1 + ti + ti2)) # random intercept, time and time squared per t
)
ranef(m1n)
```

```
##     (Intercept)            ti           ti2
## T04   0.1985796  1.609864e-09  4.990101e-10
## T05   0.3492827  2.487690e-10 -4.845287e-11
## T19  -0.1978989 -7.681202e-10 -1.961453e-10
## T23   0.4519003 -3.270426e-10 -2.413583e-10
## T38  -0.6457494 -1.608770e-09 -3.298010e-10
## T40   0.3739432  3.264705e-10 -2.543109e-11
## T49   0.8620648  9.021831e-10 -5.402247e-12
## T53  -0.5655049 -8.279040e-10 -4.579291e-11
## T67  -0.4394623 -3.485113e-10  2.147434e-11
## T71  -0.3871552  7.930610e-10  3.718993e-10
```

```
fixef(m1n)
```

```
## (Intercept)          ti         ti2
##  -10.798799   12.346704   -2.838503
```

```
summary(m1n)
```

```
## Linear mixed-effects model fit by REML
##   Data: d22
##        AIC     BIC    logLik
##   276.5142 300.761 -131.2571
##
## Random effects:
##  Formula: ~1 + ti + ti2 | tree
##  Structure: Diagonal
##         (Intercept)           ti          ti2  Residual
## StdDev:   0.5187869 1.438333e-05 3.864019e-06 0.3836614
##
## Fixed effects:  water ~ 1 + ti + ti2
##                  Value Std.Error  DF   t-value p-value
## (Intercept) -10.798799 0.8814666 227 -12.25094       0
## ti           12.346704 0.7827112 227  15.77428       0
## ti2          -2.838503 0.1720614 227 -16.49704       0
##  Correlation:
##     (Intr) ti
## ti  -0.979
## ti2  0.970 -0.997
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -3.07588246 -0.58531056  0.01210209  0.65402695  3.88777402
##
```

```
## Number of Observations: 239
## Number of Groups: 10
```

`lmer` function from `lme4` package

```
m1lmer <- lmer(water~1+ti+ti2+(ti+ti2||tree),data = d22,na.action = na.omit)
```

```
## boundary (singular) fit: see ?isSingular
```

```
ranef(m1lmer)
```

```
## $tree
##     (Intercept) ti ti2
## T04   0.1985796  0   0
## T05   0.3492827  0   0
## T19  -0.1978989  0   0
## T23   0.4519003  0   0
## T38  -0.6457494  0   0
## T40   0.3739432  0   0
## T49   0.8620648  0   0
## T53  -0.5655049  0   0
## T67  -0.4394623  0   0
## T71  -0.3871552  0   0
##
## with conditional variances for "tree"
```

Notes:

- `||` double pipes= uncorrelated random effects
- To remove the intercept term:
  - `(0+ti|tree)`
  - `(ti-1|tree)`

```
fixef(m1lmer)
```

```
## (Intercept)          ti         ti2
##  -10.798799   12.346704   -2.838503
```

```
m1l <- lmer(water ~ 1 + ti + ti2 + (1 | tree) + (0 + ti | tree) + (0 + ti2 | tree), data = d22)
```

```
## boundary (singular) fit: see ?isSingular
```

```
ranef(m1l)
```

```
## $tree
##     (Intercept) ti ti2
## T04   0.1985796  0   0
## T05   0.3492827  0   0
## T19  -0.1978989  0   0
## T23   0.4519003  0   0
## T38  -0.6457494  0   0
## T40   0.3739432  0   0
## T49   0.8620648  0   0
## T53  -0.5655049  0   0
## T67  -0.4394623  0   0
## T71  -0.3871552  0   0
##
## with conditional variances for "tree"
```

```
fixef(m1l)
```

```
## (Intercept)          ti         ti2
##  -10.798799   12.346704   -2.838503
```

To include structured covariance terms, we can use the following way

```
m2n <- lme(
    water ~ 1 + ti + ti2,
    data = d22,
    random = ~ 1 | tree,
    cor = corExp(form =  ~ day | tree),
    na.action = na.omit
)
ranef(m2n)
```

```
##     (Intercept)
## T04   0.1929971
## T05   0.3424631
## T19  -0.1988495
## T23   0.4538660
## T38  -0.6413664
## T40   0.3769378
## T49   0.8410043
## T53  -0.5528236
## T67  -0.4452930
## T71  -0.3689358
```

```
fixef(m2n)
```

```
## (Intercept)          ti         ti2
##  -11.223310   12.712094   -2.913682
```

```
summary(m2n)
```

```
## Linear mixed-effects model fit by REML
##   Data: d22
##        AIC      BIC   logLik
##   263.3081 284.0911 -125.654
##
```

```
## Random effects:
##  Formula: ~1 | tree
##         (Intercept)  Residual
## StdDev:   0.5154042 0.3925777
##
## Correlation Structure: Exponential spatial correlation
##  Formula: ~day | tree
##  Parameter estimate(s):
##    range
## 3.794624
## Fixed effects:  water ~ 1 + ti + ti2
##                  Value Std.Error  DF   t-value p-value
## (Intercept) -11.223310 1.0988725 227 -10.21348       0
## ti           12.712094 0.9794235 227  12.97916       0
## ti2          -2.913682 0.2148551 227 -13.56115       0
##  Correlation:
##     (Intr) ti
## ti  -0.985
## ti2  0.976 -0.997
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -3.04861039 -0.55703950  0.00278101  0.62558762  3.80676991
##
## Number of Observations: 239
## Number of Groups: 10
```

Chapter 9

Nonlinear and Generalized Linear Mixed Models

- NLMMs extend the nonlinear model to include both fixed effects and random effects
- GLMMs extend the generalized linear model to include both fixed effects and random effects.

A nonlinear mixed model has the form of

$$Y_{ij} = f(\mathbf{x_{ij}}, \ , \ _{\mathbf{i}}) + \epsilon_{ij}$$

for the j-th response from cluster (or sujbect) i ($i = 1, ..., n$), where

- $j = 1, ..., n_i$
- are the fixed effects
- $_i$ are the random effects for cluster i
- $\mathbf{x}_{ij}$ are the regressors or design variables
- $f(.)$ is nonlinear mean response function

A GLMM can be written as:

we assume

$$y_i|\alpha_i \sim \text{indep } f(y_i|\alpha)$$

and $f(y_i| \ )$ is an exponential family distribution,

$$f(y_i|\alpha) = \exp[\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} - c(y_i, \phi)]$$

The conditional mean of y_i is related to θ_i

$$\mu_i = \frac{\partial b(\theta_i)}{\partial \theta_i}$$

The transformation of this mean will give us the desired linear model to model both the fixed and random effects.

$$E(y_i|\alpha) = \mu_i g(\mu_i) = \mathbf{x_i'} \quad + \mathbf{z_i'}$$

where $g()$ is a known link function and μ_i is the conditional mean. We can see similarity to GLM

We also have to specify the random effects distribution

$$\alpha \sim f(\alpha)$$

which is similar to the specification for mixed models.

Moreover, law of large number applies to fixed effects so that you know it is a normal distribution. But here, you can specify α subjectively.

Hence, we can show NLMM is a special case of the GLMM

$$\mathbf{Y}_i = \mathbf{f}(\mathbf{x}_i, \; , \; _i) + \; _i\mathbf{Y}_i = \mathbf{g}^{-1}(\mathbf{x}_i'\beta + \mathbf{z}_i' \; _i) + \; _i$$

where the inverse link function corresponds to a nonlinear transformation of the fixed and random effects.

Note:

- we can't derive the analytical formulation of the marginal distribution because nonlinear combination of normal variables is not normally distributed, even in the case of additive error (e_i) and random effects (α_i) are both normal.

Consequences of having random effects

The marginal mean of y_i is

$$E(y_i) = E_\alpha(E(y_i|\alpha)) = E_\alpha(\mu_i) = E(g^{-1}(\mathbf{x_i'} \quad + \mathbf{z_i'} \quad))$$

Because $g^{-1}()$ is nonlinear, this is the most simplified version we can go for.

In special cases such as log link ($g(\mu) = \log \mu$ or $g^{-1}() = \exp()$) then

$$E(y_i) = E(\exp(\mathbf{x_i'} \quad + \mathbf{z_i'} \quad)) = \exp(\mathbf{x_i'} \quad)E(\exp(\mathbf{z}_i'\alpha))$$

which is the moment generating function of α evaluated at $\mathbf{z}_i$

Marginal variance of y_i

$$\begin{aligned} var(y_i) &= var_\alpha(E(y_i|\alpha)) + E_\alpha(var(y_i|\alpha)) \\ &= var(\mu_i) + E(a(\phi)V(\mu_i)) \\ &= var(g^{-1}(\mathbf{x_i'}\ \ + \mathbf{z_i'}\ \ )) + E(a(\phi)V(g^{-1}(\mathbf{x_i'}\ \ + \mathbf{z_i'}\ \ ))) \end{aligned}$$

Without specific assumption about $g()$ and/or the conditional distribution of $\mathbf{y}$, this is the most simplified version.

Marginal covariance of y

In a linear mixed model, random effects introduce a dependence among observations which share any random effect in common

$$\begin{aligned} cov(y_i, y_j) &= cov_\alpha(E(y_i|\ ), E(y_j|\ )) + E_\alpha(cov(y_i, y_j|\ )) \\ &= cov(\mu_i, \mu_j) + E(0) \\ &= cov(g^{-1}(\mathbf{x}_i'\beta + \mathbf{z}_i'\ ), g^{-1}(\mathbf{x}_j'\beta + \mathbf{z}_j'\ )) \end{aligned}$$

- Important: conditioning to induce the covariability

Example:

Repeated measurements on the subjects. Let y_{ij} be the j-th count taken on the i-th subject.

then, the model is $y_{ij}|\ \sim$ indep $Pois(\mu_{ij})$. Here

$$\log(\mu_{ij}) = \mathbf{x}_{ij}'\beta + \alpha_i$$

where $\alpha_i \sim iidN(0, \sigma_\alpha^2)$

which is a log-link with a random patient effect.

9.1 Estimation

In linear mixed models, the marginal likelihood for $\mathbf{y}$ is the integration of the random effects from the hierarchical formulation

$$f(\mathbf{y}) = \int f(\mathbf{y}|\alpha)f(\alpha)d\alpha$$

For linear mixed models, we assumed that the 2 component distributions were Gaussian with linear relationships, which implied the marginal distribution was also linear and Gaussian and allows us to solve this integral analytically.

On the other hand, GLMMs, the distribution for $f(\mathbf{y}|\alpha)$ is not Gaussian in general, and for NLMMs, the functional form between the mean response and the random (and fixed) effects is nonlinear. In both cases, we can't perform the integral analytically, which means we have to solve it

- numerically and/or
- linearize the inverse link function.

9.1.1 Estimation by Numerical Integration

The marginal likelihood is

$$L(\beta; \mathbf{y}) = \int f(\mathbf{y}|\alpha) f(\alpha) d\alpha$$

Estimation fo the fixed effects requires $\frac{\partial l}{\partial \beta}$, where l is the log-likelihood

One way to obtain the marginal inference is to numerically integrate out the random effects through

- numerical quadrature
- Laplace approximation
- Monte Carlo methods

When the dimension of is relatively low, this is easy. But when the dimension of α is high, additional approximation is required.

9.1.2 Estimation by Linearization

Idea: Linearized version of the response (known as working response, or pseudo-response) called $\tilde{y}_i$ and then the conditional mean is

$$E(\tilde{y}_i|\alpha) = \mathbf{x}_i'\beta + \mathbf{z}_i'\alpha$$

and also estimate $var(\tilde{y}_i|\alpha)$. then, apply Linear Mixed Models estimation as usual.

The difference is only in how the linearization is done (i.e., how to expand $f(\mathbf{x}, \ , \ )$ or the inverse link function

9.1.2.1 Penalized quasi-likelihood

(PQL)

This is the more popular method

$$\tilde{y}_i^{(k)} = \hat{\eta}_i^{(k-1)} + (y_i - \hat{\mu}_i^{(k-1)})\frac{d\eta}{d\mu}|\hat{\eta}_i^{(k-1)}$$

where

- $\eta_i = g(\mu_i)$ is the linear predictor
- k = iteration of the optimization algorithm

The algorithm updates $\tilde{y}_i$ after each linear mixed model fit using $E(\tilde{y}_i|\alpha)$ and $var(\tilde{y}_i|\alpha)$

Comments:

- Easy to implement
- Inference is only asymptotically correct due to the linearizaton
- Biased estimates are likely for binomial response with small groups and worst for Bernoulli response. Similarly for Poisson models with small counts. (Faraway, 2016)
- Hypothesis testing and confidence intervals also have problems.

9.1.2.2 Generalized Estimating Equations

(GEE)

Let a marginal generalized linear model for the mean of y as a function of the predictors, which means we linearize the mean response function and assume a dependent error structure

Example
Binary data:

$$logit(E(\mathbf{y})) = \mathbf{X}\beta$$

If we assume a "working covariance matrix", $\mathbf{V}$ the the elements of $\mathbf{y}$, then the maximum likelihood equations for estimating β is

$$\mathbf{X'V^{-1}y} = \mathbf{X'V^{-1}}E(\mathbf{y})$$

If $\mathbf{V}$ is correct, then unbiased estimating equations

We typically define $\mathbf{V} = \mathbf{I}$. Solutions to unbiased estimating equation give consistent estimators.

In practice, we assume a covariance structure, and then do a logistic regression, and calculate its large sample variance

Let $y_{ij}, j = 1, .., n_i, i = 1, .., K$ be the j-th measurement on the i-th subject.

$$\mathbf{y}_i = \begin{pmatrix} y_{i1} \\ . \\ y_{in_i} \end{pmatrix}$$

with mean

$${}_i = \begin{pmatrix} \mu_{i1} \\ . \\ \mu_{in_i} \end{pmatrix}$$

and

$$\mathbf{x}_{ij} = \begin{pmatrix} X_{ij1} \\ . \\ X_{ijp} \end{pmatrix}$$

Let $\mathbf{V}_i = cov(\mathbf{y}_i)$, then based on(LIANG and ZEGER, 1986) GEE estimates for β can be obtained from solving the equation:

$$S(\beta) = \sum_{i=1}^{K} \frac{\partial \,{}'_i}{\partial \beta} \mathbf{V}^{-1}(\mathbf{y}_i - \;{}_i) = 0$$

Let $\mathbf{R}_i(\mathbf{c})$ be an $n_i \times n_i$ "working" correlation matrix specified up to some parameters $\mathbf{c}$. Then, $\mathbf{V}_i = a(\phi)\mathbf{B}_i^{1/2}\mathbf{R}(\mathbf{c})\mathbf{B}_i^{1/2}$, where $\mathbf{B}_i$ is an $n_i \times n_i$ diagonal matrix with $V(\mu_{ij})$ on the j-th diagonal

If $\mathbf{R}(\mathbf{c})$ is the true correlation matrix of $\mathbf{y}_i$, then $\mathbf{V}_i$ is the true covariance matrix

The working correlation matrix must be estimated iteratively by a fitting algorithm:

1. Compute the initial estimate of β (using GLM under the independence assumption)
2. Compute the working correlation matrix $\mathbf{R}$ based upon studentized residuals
3. Compute the estimate covariance $\hat{\mathbf{V}}_i$
4. Update β according to

$$\beta_{r+1} = \beta_r + (\sum_{i=1}^{K} \frac{\partial \,{}'_i}{\partial \beta} \hat{\mathbf{V}}_i^{-1} \frac{\partial \,{}_i}{\partial \beta})$$

5. Iterate until the algorithm converges

Note: Inference based on likelihoods is not appropriate because this is not a likelihood estimator

9.1.3 Estimation by Bayesian Hierarchical Models

Bayesian Estimation

$$f(\; , \; |\mathbf{y}) \propto f(\mathbf{y}| \; , \;)f(\;)f(\;)$$

Numerical techniques (e.g., MCMC) can be used to find posterior distribution. This method is best in terms of not having to make simplifying approximation and fully accounting for uncertainty in estimation and prediction, but it could be complex, time-consuming, and computationally intensive.

Implementation Issues:

- No valid joint distribution can be constructed from the given conditional model and random parameters
- The mean/ variance relationship and the random effects lead to constraints on the marginal covariance model
- Difficult to fit computationally

2 types of estimation approaches:

1. Approximate the objective function (marginal likelihood) through integral approximation
    1. Laplace methods
    2. Quadrature methods
    3. Monte Carlo integration
2. Approximate the model (based on Taylor series linearizations)

Packages in R

- GLMM: `MASS:glmmPQL` `lme4::glmer` `glmmTMB`
- NLMM: `nlme::nlme`; `lme4::nlmer` `brms::brm`
- Bayesian: `MCMCglmm` ; `brms:brm`

Example: Non-Gaussian Repeated measurements

- When the data are Gaussian, then Linear Mixed Models
- When the data are non-Gaussian, then Nonlinear and Generalized Linear Mixed Models

9.2 Application

9.2.1 Binomial (CBPP Data)

```
data(cbpp,package = "lme4")
head(cbpp)
```

```
##   herd incidence size period
## 1    1         2   14      1
## 2    1         3   12      2
## 3    1         4    9      3
## 4    1         0    5      4
## 5    2         3   22      1
## 6    2         1   18      2
```

PQL

Pro:

- Linearizes the response to have a pseudo-response as the mean response (like LMM)
- computationally efficient

Cons:

- biased for binary, Poisson data with small counts
- random effects have to be interpreted on the link scale
- can't interpret AIC/BIC value

```
library(MASS)
pql_cbpp <-
    glmmPQL(
        cbind(incidence, size - incidence) ~ period,
        random = ~ 1 | herd,
        data = cbpp,
        family = binomial(link = "logit"),
        verbose = F
    )
summary(pql_cbpp)
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: cbpp
##   AIC BIC logLik
##    NA  NA     NA
##
## Random effects:
##  Formula: ~1 | herd
##         (Intercept) Residual
## StdDev:   0.5563535 1.184527
##
## Variance function:
##  Structure: fixed weights
##  Formula: ~invwt
## Fixed effects:  cbind(incidence, size - incidence) ~ period
##                  Value Std.Error DF   t-value p-value
## (Intercept) -1.327364 0.2390194 38 -5.553372  0.0000
## period2     -1.016126 0.3684079 38 -2.758156  0.0089
## period3     -1.149984 0.3937029 38 -2.920944  0.0058
## period4     -1.605217 0.5178388 38 -3.099839  0.0036
##  Correlation:
##         (Intr) perid2 perid3
## period2 -0.399
## period3 -0.373  0.260
## period4 -0.282  0.196  0.182
##
## Standardized Within-Group Residuals:
##        Min         Q1        Med         Q3        Max
## -2.0591168 -0.6493095 -0.2747620  0.5170492  2.6187632
##
## Number of Observations: 56
## Number of Groups: 15
```

```
exp(0.556)
```

```
## [1] 1.743684
```

is how the herd specific outcome odds varies.

We can interpret the fixed effect coefficients just like in GLM. Because we use logit link function here, we can say that the log odds of the probability of having a case in period 2 is -1.016 less than period 1 (baseline).

```
summary(pql_cbpp)$tTable
```

```
##                  Value Std.Error DF   t-value      p-value
## (Intercept) -1.327364 0.2390194 38 -5.553372 2.333216e-06
## period2     -1.016126 0.3684079 38 -2.758156 8.888179e-03
## period3     -1.149984 0.3937029 38 -2.920944 5.843007e-03
## period4     -1.605217 0.5178388 38 -3.099839 3.637000e-03
```

Numerical Integration

Pro:

- more accurate

Con:

- computationally expensive
- won't work for complex models.

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
numint_cbpp <-
    glmer(
        cbind(incidence, size - incidence) ~ period + (1 | herd),
        data = cbpp,
        family = binomial(link = "logit")
    )
summary(numint_cbpp)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: cbind(incidence, size - incidence) ~ period + (1 | herd)
##    Data: cbpp
##
##      AIC      BIC   logLik deviance df.resid
##    194.1    204.2    -92.0    184.1       51
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.3816 -0.7889 -0.2026  0.5142  2.8791
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  herd   (Intercept) 0.4123   0.6421
## Number of obs: 56, groups:  herd, 15
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.3983     0.2312  -6.048 1.47e-09 ***
## period2      -0.9919     0.3032  -3.272 0.001068 **
## period3      -1.1282     0.3228  -3.495 0.000474 ***
## period4      -1.5797     0.4220  -3.743 0.000182 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##         (Intr) perid2 perid3
## period2 -0.363
## period3 -0.340  0.280
## period4 -0.260  0.213  0.198
```

For small data set, the difference between two approaches are minimal

```
library(rbenchmark)
benchmark(
    "MASS" = {
        pql_cbpp <-
            glmmPQL(
                cbind(incidence, size - incidence) ~ period,
                random = ~ 1 | herd,
                data = cbpp,
                family = binomial(link = "logit"),
                verbose = F
            )
    },
    "lme4" = {
        glmer(
            cbind(incidence, size - incidence) ~ period + (1 | herd),
            data = cbpp,
            family = binomial(link = "logit")
        )
    },
    replications = 50,
    columns = c("test", "replications", "elapsed", "relative"),
    order = "relative"
)
```

```
##   test replications elapsed relative
## 1 MASS           50    2.38    1.000
## 2 lme4           50    4.94    2.076
```

In numerical integration, we can set `nAGQ > 1` to switch the method of likelihood evaluation, which might increase accuracy

```
library(lme4)
numint_cbpp_GH <-
    glmer(
        cbind(incidence, size - incidence) ~ period + (1 | herd),
        data = cbpp,
        family = binomial(link = "logit"),
        nAGQ = 20
    )
summary(numint_cbpp_GH)$coefficients[, 1] - summary(numint_cbpp)$coefficients[, 1]
```

```
##   (Intercept)       period2       period3       period4
## -0.0008808634  0.0005160912  0.0004066218  0.0002644629
```

Bayesian approach to GLMMs

- assume the fixed effects parameters have distribution

- can handle models with intractable result under traditional methods
- computationally expensive

```
library(MCMCglmm)
```

```
## Loading required package: coda

## Loading required package: ape
```

```
Bayes_cbpp <-
    MCMCglmm(
        cbind(incidence, size - incidence) ~ period,
        random = ~ herd,
        data = cbpp,
        family = "multinomial2",
        verbose = FALSE
    )
summary(Bayes_cbpp)
```

```
##
##  Iterations = 3001:12991
##  Thinning interval  = 10
##  Sample size  = 1000
##
##  DIC: 539.1983
##
##  G-structure:  ~herd
##
##      post.mean  l-95% CI u-95% CI eff.samp
## herd   0.03313 1.037e-16   0.2131    60.82
##
##  R-structure:  ~units
##
##       post.mean l-95% CI u-95% CI eff.samp
## units     1.108   0.1491    2.184    197.5
##
##  Location effects: cbind(incidence, size - incidence) ~ period
##
##             post.mean l-95% CI u-95% CI eff.samp  pMCMC
## (Intercept)   -1.5272  -2.1949  -0.8480    884.6 <0.001 ***
## period2       -1.2328  -2.2987  -0.2133    860.3  0.026 *
## period3       -1.3746  -2.5082  -0.3640    708.7  0.010 **
## period4       -1.9741  -3.2607  -0.7588    388.6 <0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- `MCMCglmm` fits a residual variance component (useful with dispersion issues)

```
apply(Bayes_cbpp$VCV,2,sd) #explains less variability
```

```
##      herd     units
## 0.1246411 0.5608232
```

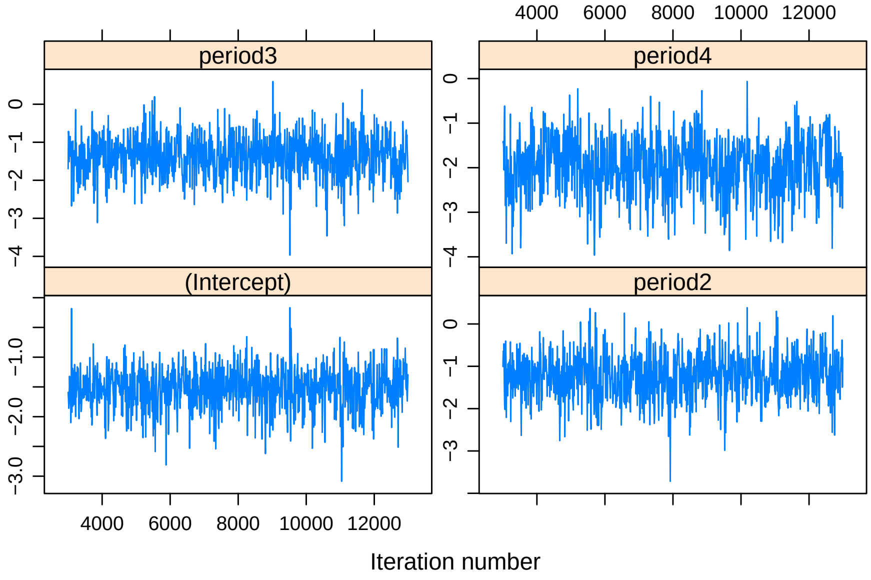
```
summary(Bayes_cbpp)$solutions
```

```
##             post.mean  l-95% CI   u-95% CI eff.samp pMCMC
## (Intercept) -1.527187 -2.194941 -0.8480449 884.5814 0.001
## period2     -1.232812 -2.298668 -0.2132777 860.2938 0.026
## period3     -1.374650 -2.508226 -0.3639825 708.6559 0.010
## period4     -1.974134 -3.260665 -0.7588418 388.6115 0.001
```

interpret Bayesian “credible intervals” similarly to confidence intervals

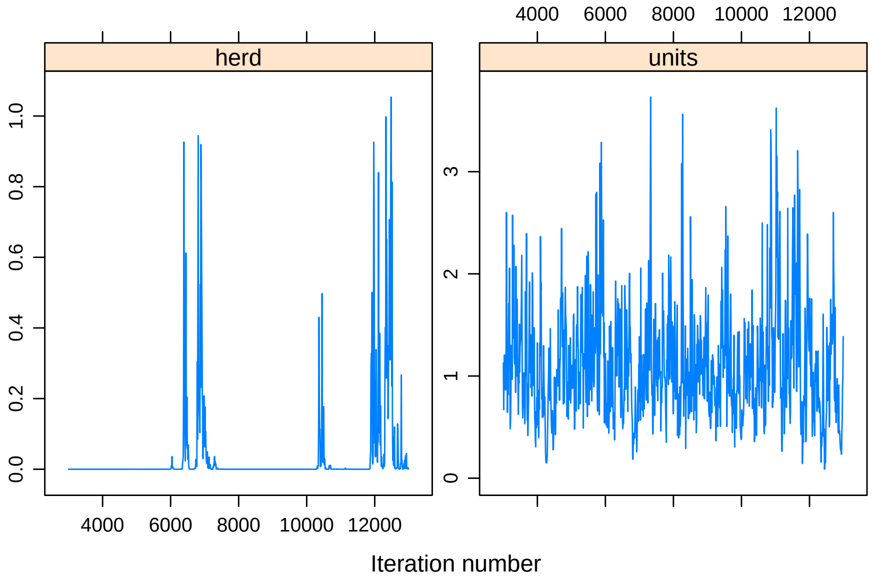
Make sure you make post-hoc diagnoses

```
library(lattice)
xyplot(as.mcmc(Bayes_cbpp$Sol), layout = c(2, 2))
```



There is no trend, well-mixed

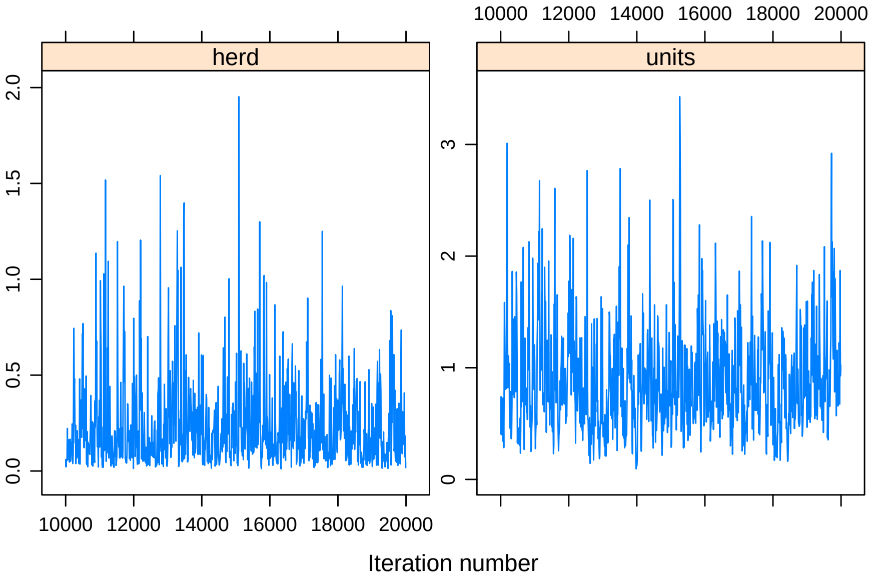
```
xyplot(as.mcmc(Bayes_cbpp$VCV),layout=c(2,1))
```



For the herd variable, a lot of them are 0, which suggests problem. To fix the instability in the herd effect sampling, we can either

- modify the prior distribution on the herd variation
- increases the number of iteration

```
library(MCMCglmm)
Bayes_cbpp2 <-
    MCMCglmm(
        cbind(incidence, size - incidence) ~ period,
        random = ~ herd,
        data = cbpp,
        family = "multinomial2",
        nitt = 20000,
        burnin = 10000,
        prior = list(G = list(list(
            V = 1, nu = .1
        ))),
        verbose = FALSE
    )
xyplot(as.mcmc(Bayes_cbpp2$VCV), layout = c(2, 1))
```



To change the shape of priors, in `MCMCglmm` use:

- `V` controls for the location of the distribution (default = 1)
- `nu` controls for the concentration around V (default = 0)

9.2.2 Count (Owl Data)

```
library(glmmTMB)
```

```
## Warning: package 'glmmTMB' was built under R version 4.0.5
```

```
## Warning in checkMatrixPackageVersion(): Package version inconsistency detected.
## TMB was built with Matrix version 1.2.18
## Current Matrix version is 1.3.2
## Please re-install 'TMB' from source using install.packages('TMB', type = 'source') or ask CRAN
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following objects are masked from 'package:stats':
##
```

```
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
data(Owls, package = "glmmTMB")
Owls <- Owls %>% rename(Ncalls = SiblingNegotiation)
```

In a typical Poisson model, λ (Poisson mean), is model as $\log(\lambda) = \mathbf{x}'$ But if the response is the rate (e.g., counts per BroodSize), we could model it as $\log(\lambda/b) = \mathbf{x}'$, equivalently $\log(\lambda) = \log(b) + \mathbf{x}'$ where b is BroodSize. Hence, we "offset" the mean by the log of this variable.

```
owls_glmer <-
    glmer(
        Ncalls ~ offset(log(BroodSize)) + FoodTreatment * SexParent +
            (1 | Nest),
        family = poisson,
        data = Owls
    )
summary(owls_glmer)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: poisson  ( log )
## Formula: Ncalls ~ offset(log(BroodSize)) + FoodTreatment * SexParent +
##     (1 | Nest)
##    Data: Owls
##
##      AIC      BIC   logLik deviance df.resid
##   5212.8   5234.8  -2601.4   5202.8      594
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.5529 -1.7971 -0.6842  1.2689 11.4312
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Nest   (Intercept) 0.2063   0.4542
## Number of obs: 599, groups:  Nest, 27
##
## Fixed effects:
##                                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)                            0.65585    0.09567   6.855 7.12e-12 ***
## FoodTreatmentSatiated                 -0.65612    0.05606 -11.705  < 2e-16 ***
## SexParentMale                         -0.03705    0.04501  -0.823   0.4104
## FoodTreatmentSatiated:SexParentMale  0.13135    0.07036   1.867   0.0619 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) FdTrtS SxPrnM
## FdTrtmntStt -0.225
## SexParentMl -0.292  0.491
## FdTrtmS:SPM  0.170 -0.768 -0.605
```

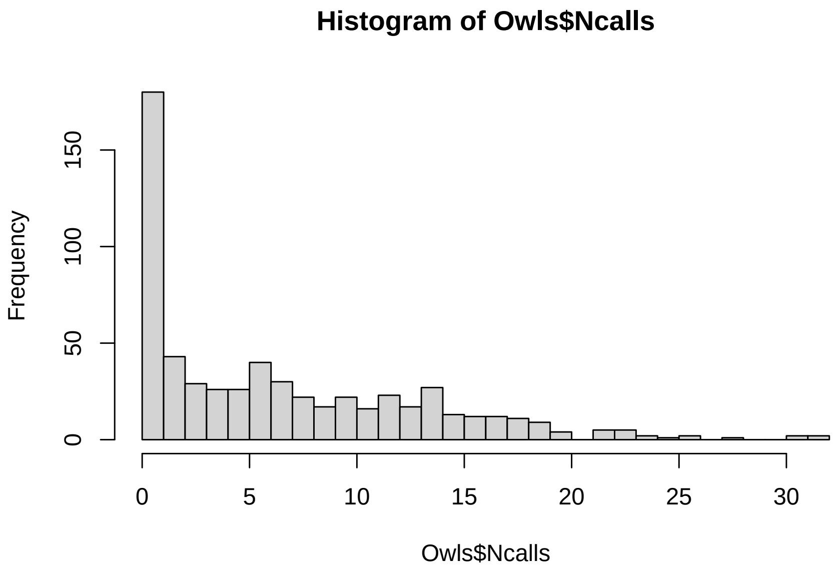
- nest explains a relatively large proportion of the variability (its standard deviation is larger than some coefficients)
- the model fit isn't great (deviance of 5202 on 594 df)

```
# Negative binomial model
owls_glmerNB <-
    glmer.nb(Ncalls ~ offset(log(BroodSize)) + FoodTreatment * SexParent
             + (1 | Nest), data = Owls)
c(Deviance = round(summary(owls_glmerNB)$AICtab["deviance"], 3),
  df = summary(owls_glmerNB)$AICtab["df.resid"])
```

```
## Deviance.deviance       df.df.resid
##          3483.616           593.000
```

There is an improvement using negative binomial considering overdispersion

```
hist(Owls$Ncalls,breaks=30)
```



To account for too many 0s in these data, we can use zero-inflated Poisson (ZIP) model.

- `glmmTMB` can handle ZIP GLMMs since it adds automatic differentiation to existing estimation strategies.

```
library(glmmTMB)
owls_glmm <-
    glmmTMB(
        Ncalls ~ FoodTreatment * SexParent + offset(log(BroodSize)) +
            (1 | Nest),
        ziformula =  ~ 0,
        family = nbinom2(link = "log"),
        data = Owls
    )
```

```
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you

## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you

## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you
```

```
owls_glmm_zi <-
    glmmTMB(
        Ncalls ~ FoodTreatment * SexParent + offset(log(BroodSize)) +
            (1 | Nest),
        ziformula =  ~ 1,
        family = nbinom2(link
                         = "log"),
        data = Owls
    )
```

```
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you

## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you

## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you
```

```
# Scale Arrival time to use as a covariate for zero-inflation parameter
Owls$ArrivalTime <- scale(Owls$ArrivalTime)
owls_glmm_zi_cov <- glmmTMB(
    Ncalls ~ FoodTreatment * SexParent +
        offset(log(BroodSize)) +
        (1 | Nest),
    ziformula =  ~ ArrivalTime,
    family = nbinom2(link
                     = "log"),
    data = Owls
)
```

```
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you

## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you

## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you
```

```
as.matrix(anova(owls_glmm, owls_glmm_zi))
```

```
##              Df      AIC      BIC    logLik deviance    Chisq Chi Df
## owls_glmm     6 3495.610 3521.981 -1741.805 3483.610       NA     NA
## owls_glmm_zi  7 3431.646 3462.413 -1708.823 3417.646 65.96373      1
##                Pr(>Chisq)
## owls_glmm              NA
## owls_glmm_zi 4.592983e-16
```

```
as.matrix(anova(owls_glmm_zi,owls_glmm_zi_cov))
```

```
##                  Df      AIC      BIC    logLik deviance    Chisq Chi Df
## owls_glmm_zi      7 3431.646 3462.413 -1708.823 3417.646       NA     NA
## owls_glmm_zi_cov  8 3422.532 3457.694 -1703.266 3406.532 11.11411      1
##                    Pr(>Chisq)
## owls_glmm_zi               NA
## owls_glmm_zi_cov 0.0008567362
```

```
summary(owls_glmm_zi_cov)
```

```
##  Family: nbinom2  ( log )
## Formula:
## Ncalls ~ FoodTreatment * SexParent + offset(log(BroodSize)) +      (1 | Nest)
## Zero inflation:          ~ArrivalTime
## Data: Owls
##
##      AIC      BIC   logLik deviance df.resid
##   3422.5   3457.7  -1703.3   3406.5      591
##
## Random effects:
##
```

```
## Conditional model:
##  Groups Name        Variance Std.Dev.
##  Nest   (Intercept) 0.07487  0.2736  
## Number of obs: 599, groups:  Nest, 27
## 
## Overdispersion parameter for nbinom2 family (): 2.22 
## 
## Conditional model:
##                                     Estimate Std. Error z value Pr(>|z|)    
## (Intercept)                          0.84778    0.09961   8.511  < 2e-16 ***
## FoodTreatmentSatiated               -0.39529    0.13742  -2.877  0.00402 ** 
## SexParentMale                       -0.07025    0.10435  -0.673  0.50079    
## FoodTreatmentSatiated:SexParentMale  0.12388    0.16449   0.753  0.45138    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Zero-inflation model:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  -1.3018     0.1261  -10.32  < 2e-16 ***
## ArrivalTime   0.3545     0.1074    3.30 0.000966 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can see ZIP GLMM with an arrival time covariate on the zero is best.

- arrival time has a positive effect on observing a nonzero number of calls
- interactions are non significant, the food treatment is significant (fewer calls after eating)
- nest variability is large in magnitude (without this, the parameter estimates change)

9.2.3 Binomial

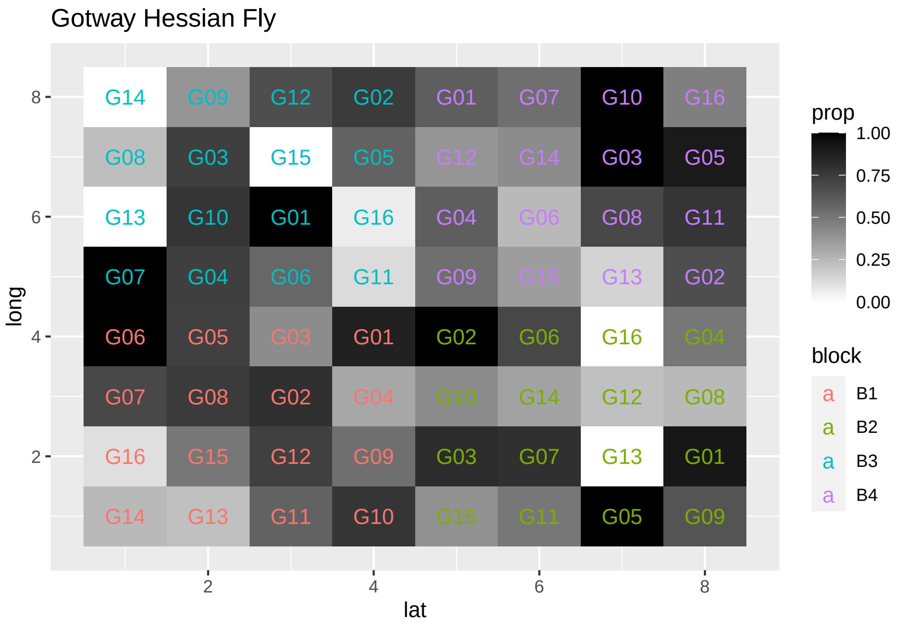
```
library(agridat)
library(ggplot2)
library(lme4)
library(spaMM)
```

```
## Warning: package 'spaMM' was built under R version 4.0.5
```

```
## Registered S3 methods overwritten by 'registry':
##   method               from 
##   print.registry_field proxy
##   print.registry_entry proxy
```

```
## spaMM (Rousset & Ferdy, 2014, version 3.7.34) is loaded.
## Type 'help(spaMM)' for a short introduction,
## 'news(package='spaMM')' for news,
## and 'citation('spaMM')' for proper citation.
```

```
data(gotway.hessianfly)
dat <- gotway.hessianfly
dat$prop <- dat$y / dat$n
ggplot(dat, aes(x = lat, y = long, fill = prop)) +
    geom_tile() +
    scale_fill_gradient(low = 'white', high = 'black') +
    geom_text(aes(label = gen, color = block)) +
    ggtitle('Gotway Hessian Fly')
```



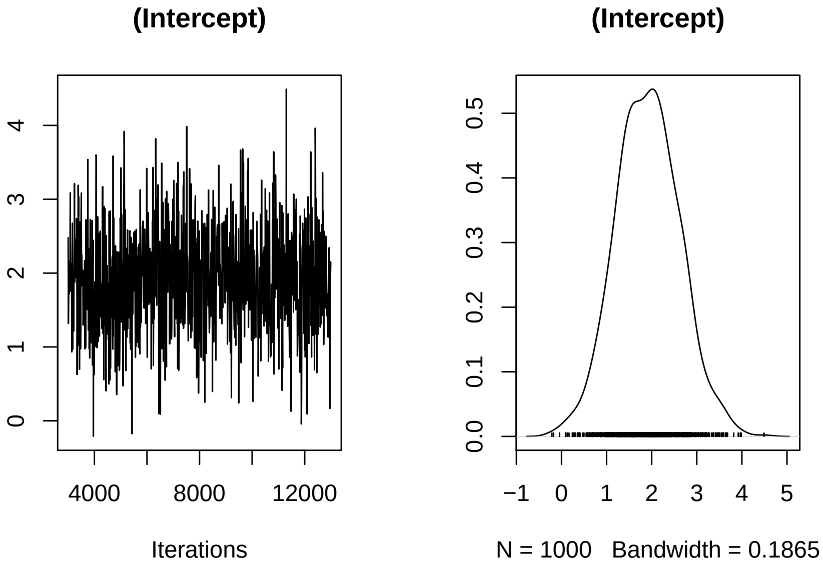
- Fixed effects (β) = genotype
- Random effects (α) = block

```
flymodel <-
    glmer(
        cbind(y, n - y) ~ gen + (1 | block),
        data = dat,
        family = binomial,
        nAGQ = 5
    )
summary(flymodel)
```

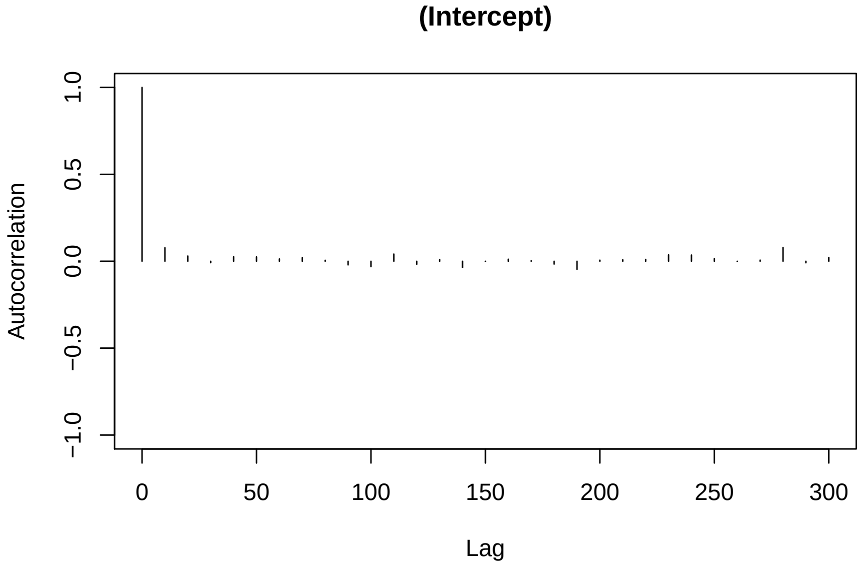
```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 5) [glmerMod]
##  Family: binomial  ( logit )
## Formula: cbind(y, n - y) ~ gen + (1 | block)
##    Data: dat
##
##      AIC      BIC   logLik deviance df.resid
##    162.2    198.9    -64.1    128.2       47
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.38644 -1.01188  0.09631  1.03468  2.75479
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  block  (Intercept) 0.001022 0.03196
## Number of obs: 64, groups:  block, 4
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.5035     0.3914   3.841 0.000122 ***
## genG02       -0.1939     0.5302  -0.366 0.714644
## genG03       -0.5408     0.5103  -1.060 0.289260
## genG04       -1.4342     0.4714  -3.043 0.002346 **
## genG05       -0.2037     0.5429  -0.375 0.707486
## genG06       -0.9783     0.5046  -1.939 0.052533 .
## genG07       -0.6041     0.5111  -1.182 0.237235
## genG08       -1.6774     0.4907  -3.418 0.000630 ***
## genG09       -1.3984     0.4725  -2.960 0.003078 **
## genG10       -0.6817     0.5333  -1.278 0.201181
## genG11       -1.4630     0.4843  -3.021 0.002522 **
## genG12       -1.4591     0.4918  -2.967 0.003010 **
## genG13       -3.5528     0.6600  -5.383 7.31e-08 ***
## genG14       -2.5073     0.5264  -4.763 1.90e-06 ***
## genG15       -2.0872     0.4851  -4.302 1.69e-05 ***
## genG16       -2.9697     0.5383  -5.517 3.46e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 16 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

Equivalently, we can use `MCMCglmm` , for a Bayesian approach

```
library(coda)
Bayes_flymodel <- MCMCglmm(
    cbind(y, n - y) ~ gen ,
    random = ~ block,
    data = dat,
    family = "multinomial2",
    verbose = FALSE
)
plot(Bayes_flymodel$Sol[, 1], main = dimnames(Bayes_flymodel$Sol)[[2]][1])
```



```
autocorr.plot(Bayes_flymodel$Sol[,1],main=dimnames(Bayes_flymodel$Sol)[[2]][1])
```

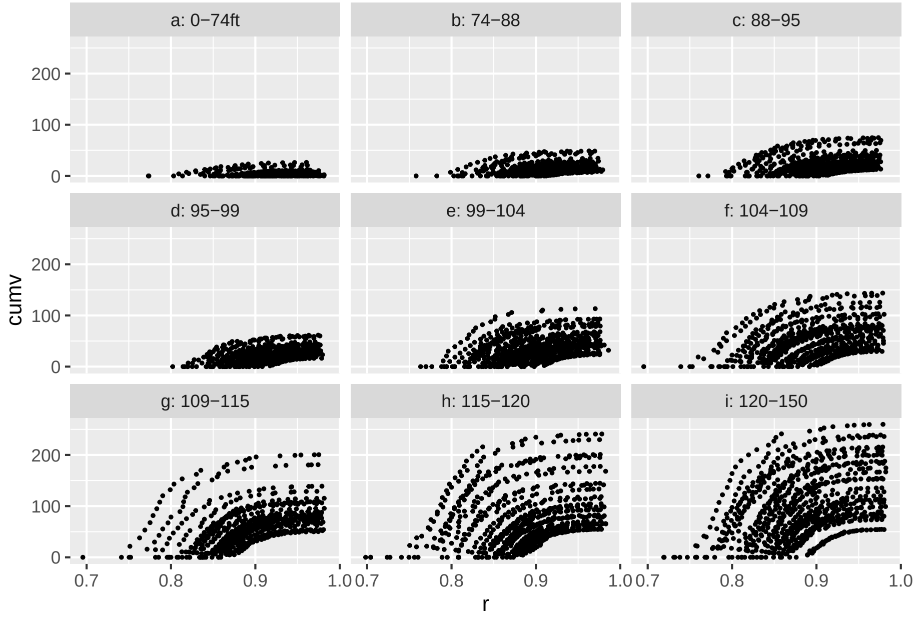


9.2.4 Example from (Schabenberger and Pierce, 2001) section 8.4.1

```
dat2 <- read.table("images/YellowPoplarData_r.txt")
names(dat2) <- c('tn', 'k', 'dbh', 'totht', 'dob', 'ht', 'maxd', 'cumv')
dat2$t <- dat2$dob / dat2$dbh
dat2$r <- 1 - dat2$dob / dat2$totht
```

The cumulative volume relates to the complementary diameter (subplots were created based on total tree height)

```
library(ggplot2)
library(dplyr)
dat2 <- dat2 %>% group_by(tn) %>% mutate(
    z = case_when(
        totht < 74 & totht >= 0 ~ 'a: 0-74ft',
        totht < 88 & totht >= 74 ~ 'b: 74-88',
        totht < 95 & totht >= 88 ~ 'c: 88-95',
        totht < 99 & totht >= 95 ~ 'd: 95-99',
        totht < 104 & totht >= 99 ~ 'e: 99-104',
        totht < 109 & totht >= 104 ~ 'f: 104-109',
        totht < 115 & totht >= 109 ~ 'g: 109-115',
        totht < 120 & totht >= 115 ~ 'h: 115-120',
        totht < 140 & totht >= 120 ~ 'i: 120-150',
    )
)
ggplot(dat2, aes(x = r, y = cumv)) + geom_point(size = 0.5) + facet_wrap(vars(z))
```



The proposed non-linear model:

$$V_{id_j} = (\beta_0 + (\beta_1 + b_{1i})\frac{D_i^2 H_i}{1000})(\exp[-(\beta_2 + b_{2i})t_{ij}\exp(\beta_3 t_{ij})]) + e_{ij}$$

where

- b_{1i}, b_{2i} are random effects
- e_{ij} are random errors

```
library(nlme)
```

```
##
## Attaching package: 'nlme'

## The following object is masked from 'package:dplyr':
##
##     collapse

## The following object is masked from 'package:lme4':
##
##     lmList
```

```
tmp <-
    nlme(
        cumv ~ (b0 + (b1 + u1) * (dbh * dbh * totht / 1000)) * (exp(-(b2 + u2) *
                                                                        (t / 1000) * e
        data = dat2,
        fixed = b0 + b1 + b2 + b3 ~ 1,
        # 1 on the right hand side of the formula indicates a single fixed effects for
        random = list(pdDiag(u1 + u2 ~ 1)),
        #uncorrelated random effects
        groups = ~ tn,
        #group on trees so each tree w/ have u1 and u2
        start = list(fixed = c(
            b0 = 0.25,
            b1 = 2.3,
            b2 = 2.87,
            b3 = 6.7
        ))
    )
summary(tmp)
```

```
## Nonlinear mixed-effects model fit by maximum likelihood
##   Model: cumv ~ (b0 + (b1 + u1) * (dbh * dbh * totht/1000)) * (exp(-(b2 +      u2)
##   Data: dat2
##        AIC      BIC    logLik
##   31103.73 31151.33 -15544.86
##
## Random effects:
##  Formula: list(u1 ~ 1, u2 ~ 1)
##  Level: tn
##  Structure: Diagonal
##                 u1       u2 Residual
## StdDev: 0.1508094 0.447829 2.226361
##
## Fixed effects:  b0 + b1 + b2 + b3 ~ 1
##       Value  Std.Error   DF  t-value p-value
## b0 0.249386 0.12894686 6297   1.9340  0.0532
## b1 2.288832 0.01266805 6297 180.6776  0.0000
## b2 2.500497 0.05606685 6297  44.5985  0.0000
## b3 6.848871 0.02140677 6297 319.9395  0.0000
##  Correlation:
##    b0     b1     b2
## b1 -0.639
## b2  0.054  0.056
## b3 -0.011 -0.066 -0.850
##
```

```
## Standardized Within-Group Residuals:
##           Min            Q1           Med            Q3           Max
## -6.694575e+00 -3.081861e-01 -8.910696e-05  3.469469e-01  7.855665e+00
##
## Number of Observations: 6636
## Number of Groups: 336
```

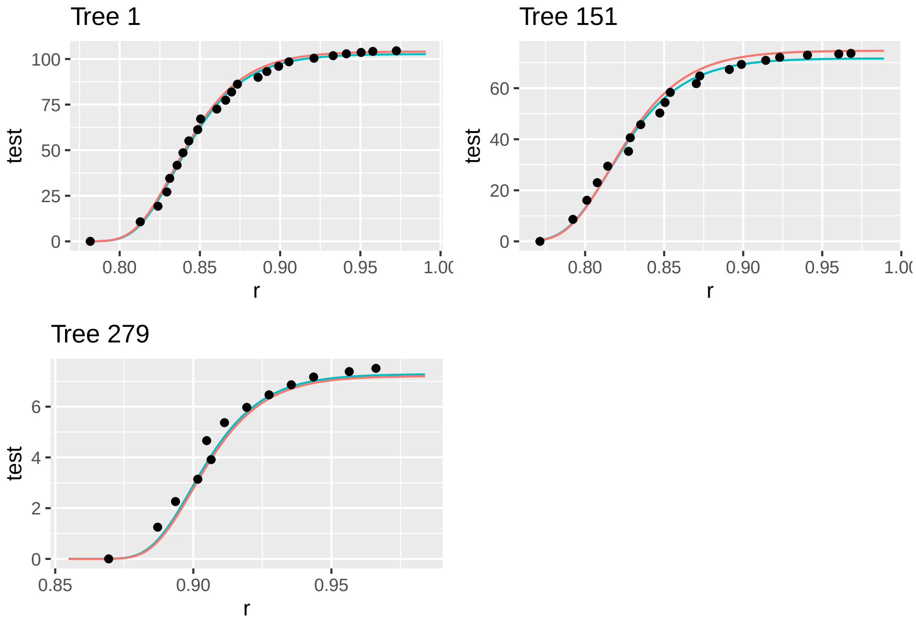
```
nlme::intervals(tmp)
```

```
## Approximate 95% confidence intervals
##
##  Fixed effects:
##           lower      est.     upper
## b0 -0.003318095 0.2493855 0.5020891
## b1  2.264006138 2.2888323 2.3136585
## b2  2.390619987 2.5004970 2.6103740
## b3  6.806919317 6.8488712 6.8908232
## attr(,"label")
## [1] "Fixed effects:"
##
##  Random Effects:
##   Level: tn
##            lower      est.     upper
## sd(u1) 0.1376080 0.1508094 0.1652772
## sd(u2) 0.4056135 0.4478290 0.4944382
##
##  Within-group standard error:
##    lower     est.    upper
## 2.187260 2.226361 2.266161
```

- Little different from the book because of different implementation of nonlinear mixed models.

```
library(cowplot)
nlmmfn <- function(fixed,rand,dbh,totht,t){
  b0 <- fixed[1]
  b1 <- fixed[2]
  b2 <- fixed[3]
  b3 <- fixed[4]
  u1 <- rand[1]
  u2 <- rand[2]
  #just made so we can predict w/o random effects
  return((b0+(b1+u1)*(dbh*dbh*totht/1000))*(exp(-(b2+u2)*(t/1000)*exp(b3*t))))
}


#Tree 1
```

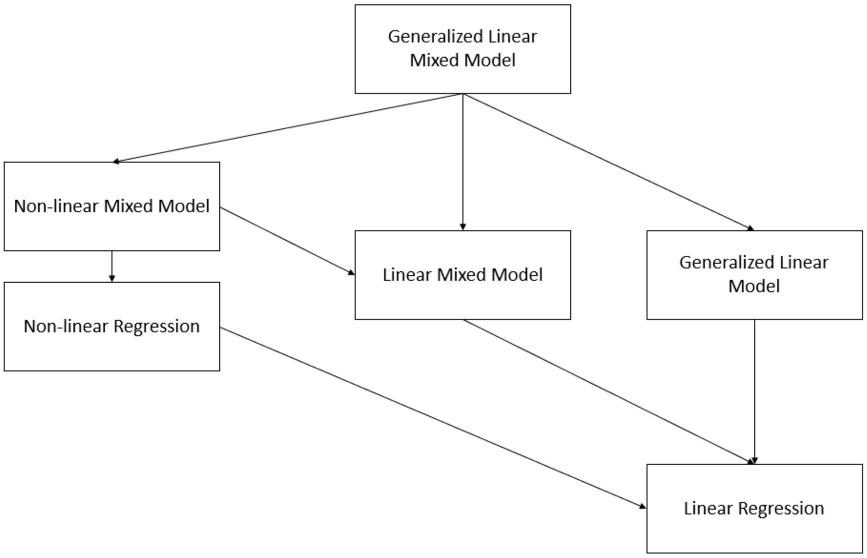
```
pred1 <- data.frame(seq(1,24,length.out=100))
names(pred1) <- 'dob'
pred1$tn <- 1
pred1$dbh <- unique(dat2[dat2$tn==1,]$dbh)
pred1$t <- pred1$dob/pred1$dbh
pred1$totht <- unique(dat2[dat2$tn==1,]$totht)
pred1$r <- 1-pred1$dob/pred1$totht


pred1$test <- predict(tmp,pred1)
pred1$testno <- nlmmfn(fixed=tmp$coefficients$fixed, rand = c(0,0),pred1$dbh,pred1$totl

p1 <- ggplot(pred1)+geom_line(aes(x=r,y=test,color='with random'))+geom_line(aes(x=r,y=


#Tree 151
pred151 <- data.frame(seq(1,21,length.out=100))
names(pred151) <- 'dob'
pred151$tn <- 151
pred151$dbh <- unique(dat2[dat2$tn==151,]$dbh)
pred151$t <- pred151$dob/pred151$dbh
pred151$totht <- unique(dat2[dat2$tn==151,]$totht)
pred151$r <- 1-pred151$dob/pred151$totht


pred151$test <- predict(tmp,pred151)
pred151$testno <- nlmmfn(fixed=tmp$coefficients$fixed, rand = c(0,0),pred151$dbh,pred15

p2 <- ggplot(pred151)+geom_line(aes(x=r,y=test,color='with random'))+geom_line(aes(x=r


#Tree 279
pred279 <- data.frame(seq(1,9,length.out=100))
names(pred279) <- 'dob'
pred279$tn <- 279
pred279$dbh <- unique(dat2[dat2$tn==279,]$dbh)
pred279$t <- pred279$dob/pred279$dbh
pred279$totht <- unique(dat2[dat2$tn==279,]$totht)
pred279$r <- 1-pred279$dob/pred279$totht


pred279$test <- predict(tmp,pred279)
pred279$testno <- nlmmfn(fixed=tmp$coefficients$fixed, rand = c(0,0),pred279$dbh,pred27

p3 <- ggplot(pred279)+geom_line(aes(x=r,y=test,color='with random'))+geom_line(aes(x=r
plot_grid(p1,p2,p3)
```



red line = predicted observations based on the common fixed effects

teal line = tree-specific predictions with random effects

9.3 Summary


Generalized Linear Mixed Model
Non-linear Mixed Model
Linear Mixed Model
Generalized Linear Model
Non-linear Regression
Linear Regression

Chapter 10

Generalized Method of Moments

Chapter 11

Minimum Distance

Chapter 12

Spline Regression

This chapter is based on CMU stat

Definition: a k-th order spline is a piecewise polynomial function of degree k, that is continuous and has continuous derivatives of orders 1,..., k -1, at its knot points

Equivalently, a function f is a k-th order spline with knot points at $t_1 < ... < t_m$ if

- f is a polynomial of degree k on each of the intervals $(-\infty, t_1], [t_1, t_2], ..., [t_m, \infty)$
- $f^{(j)}$, the j-th derivative of f, is continuous at $t_1, ..., t_m$ for each j = 0,1,...,k-1

A common case is when k = 3, called cubic splines. (piecewise cubic functions are continuous, and also continuous at its first and second derivatives)

To parameterize the set of splines, we could use **truncated power basis**, defined as

$$g_1(x) = 1 g_2(x) = x ... g_{k+1}(x) = x^k g_{k+1+j}(x) = (x - t_j)_+^k$$

where j = 1,...,m and x_+ = max{x,0}

However, now software typically use B-spline basis.

12.1 Regression Splines

To estimate the regression function $r(X) = E(Y|X = x)$, we can fit a k-th order spline with knots at some prespecified locations $t_1, ..., t_m$

Regression splines are functions of

$$\sum_{j=1}^{m+k+1} \beta_j g_j$$

where

$\beta_1, ..\beta_{m+k+1}$ are coefficients $g_1, ..., g_{m+k+1}$ are the truncated power basis functions for k-th order splines over the knots $t_1, ..., t_m$

To estimate the coefficients

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{m} \beta_j g_j(x_i))^2$$

then regression spline is

$$\hat{r}(x) = \sum_{j=1}^{m+k+1} \hat{\beta}_j g_j(x)$$

If we define the basis matrix $G \in R^{n\times(m+k+1)}$ by

$$G_{ij} = g_j(x_i)$$

where $i = 1, .., n$, $j = 1, .., m + k + 1$

Then,

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{m} \beta_j g_j(x_i))^2 = ||y - G\beta||_2^2$$

and the regression spline estimate at x is

$$\hat{r}(x) = g(x)^T \hat{\beta} = g(x)^T (G^T G)^{-1} G^T y$$

12.2 Natural splines

A natural spline of order k, with knots at $t_1 < ... < t_m$, is a piecewise polynomial function f such that

- f is polynomial of degree k on each of $[t_1, t_2], ..., [t_{m-1}, t_m]$
- f is a polynomial of degree $(k - 1)/2$ on $(-\infty, t_1]$ and $[t_m, \infty)$
- f is continuous and has continuous derivatives of orders 1,.„, k -1 at its knots $t_1, .., t_m$

Note

natural splines are only defined for odd orders k.

12.3 Smoothing splines

These estimators use a regularized regression over the natural spline basis: placing knots at all points $x_1, ...x_n$

For the case of cubic splines, the coefficients are the minimization of

$$||y - G\beta||_2^2 + \lambda\beta^T\Omega\beta$$

where $\Omega \in R^{n \times n}$ is the penalty matrix

$$\Omega_{ij} = \int g_i''(t)g_j''(t)dt,$$

and i,j = 1,..,n

and $\lambda\beta^T\Omega\beta$ is the **regularization term** used to shrink the components of $\hat{\beta}$ towards 0. $\lambda > 0$ is the tuning parameter (or smoothing parameter). Higher value of λ, faster shrinkage (shrinking away basis functions)

Note
smoothing splines have similar fits as kernel regression.

	Smoothing splines	kernel regression
tuning parameter	smoothing parameter λ	bandwidth h

12.4 Application

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(caret)
```

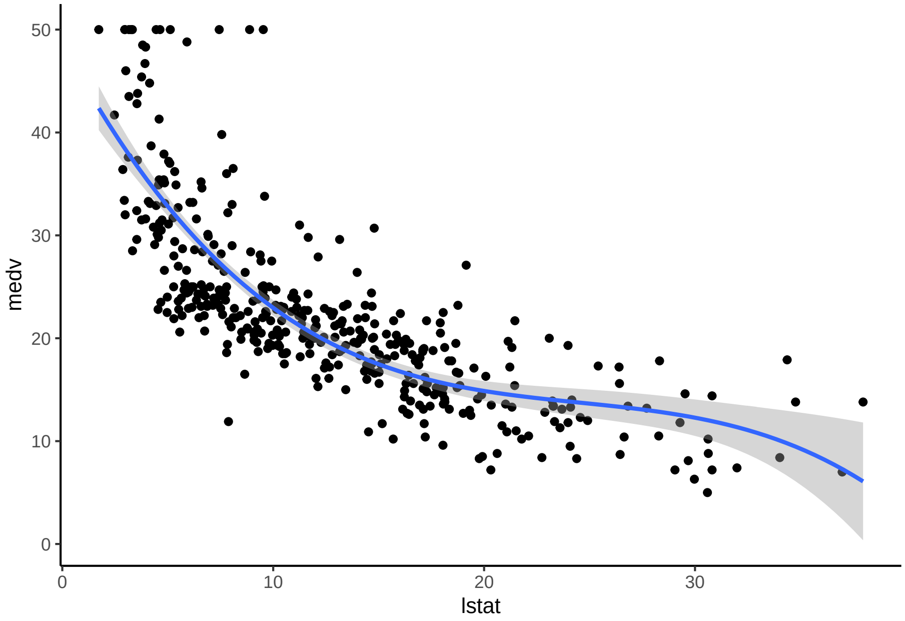
```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
## The following object is masked from 'package:purrr':
##
##     lift
```

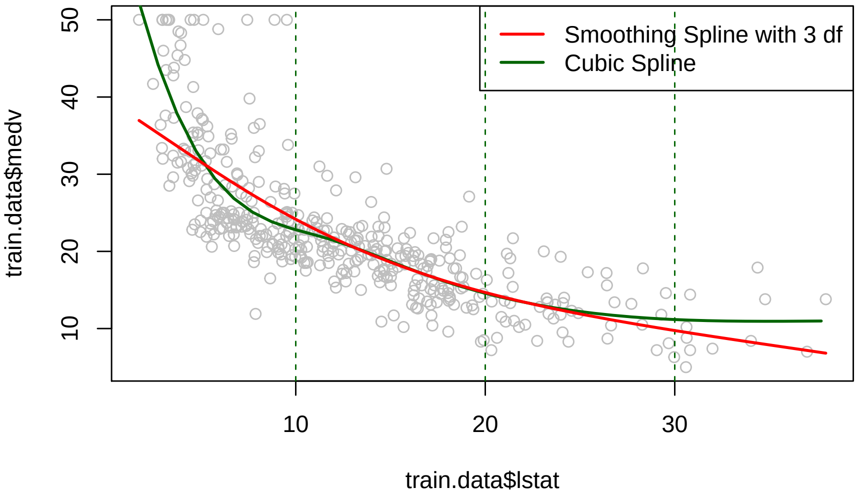
```
theme_set(theme_classic())

# Load the data
data("Boston", package = "MASS")
# Split the data into training and test set
set.seed(123)
training.samples <- Boston$medv %>%
  createDataPartition(p = 0.8, list = FALSE)
train.data  <- Boston[training.samples, ]
test.data <- Boston[-training.samples, ]

knots <- quantile(train.data$lstat, p = c(0.25, 0.5, 0.75)) # we use 3 knots at 25,50,

library(splines)
# Build the model
knots <- quantile(train.data$lstat, p = c(0.25, 0.5, 0.75))
model <- lm (medv ~ bs(lstat, knots = knots), data = train.data)
# Make predictions
predictions <- model %>% predict(test.data)
# Model performance
data.frame(
  RMSE = RMSE(predictions, test.data$medv),
  R2 = R2(predictions, test.data$medv)
)
```

```
##       RMSE        R2
## 1 5.317372 0.6786367
```

```
ggplot(train.data, aes(lstat, medv) ) +
  geom_point() +
  stat_smooth(method = lm, formula = y ~ splines::bs(x, df = 3))
```



```
attach(train.data)
#fitting smoothing splines using smooth.spline(X,Y,df=...)

fit1<-smooth.spline(train.data$lstat,train.data$medv,df=3 ) # 3 degrees of freedom
#Plotting both cubic and Smoothing Splines
plot(train.data$lstat,train.data$medv,col="grey")
lstat.grid = seq(from = range(lstat)[1], to = range(lstat)[2])
points(lstat.grid,predict(model,newdata = list(lstat=lstat.grid)),col="darkgreen",lwd=2,type="l")
#adding cutpoints
abline(v=c(10,20,30),lty=2,col="darkgreen")
lines(fit1,col="red",lwd=2)

legend("topright",c("Smoothing Spline with 3 df","Cubic Spline"),col=c("red","darkgreen"),lwd=2)
```


Smoothing Spline with 3 df
Cubic Spline
train.data$medv
train.data$lstat
10
20
30
40
50
10
20
30

Chapter 13

Generalized Additive Models

To overcome Spline Regression's requirements for specifying the knots, we can use Generalized Additive Models or GAM.

```
library(mgcv)
```

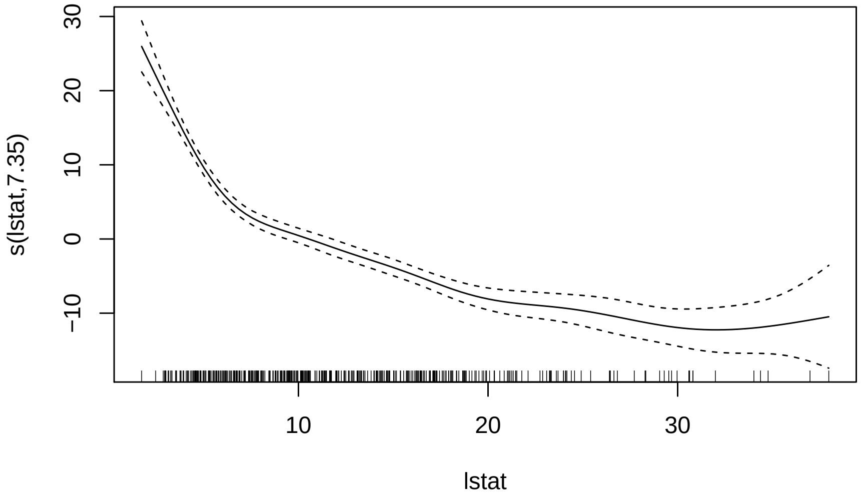
```
## Loading required package: nlme
```

```
##
## Attaching package: 'nlme'
```

```
## The following object is masked from 'package:dplyr':
##
##     collapse
```

```
## This is mgcv 1.8-34. For overview type 'help("mgcv-package")'.
```

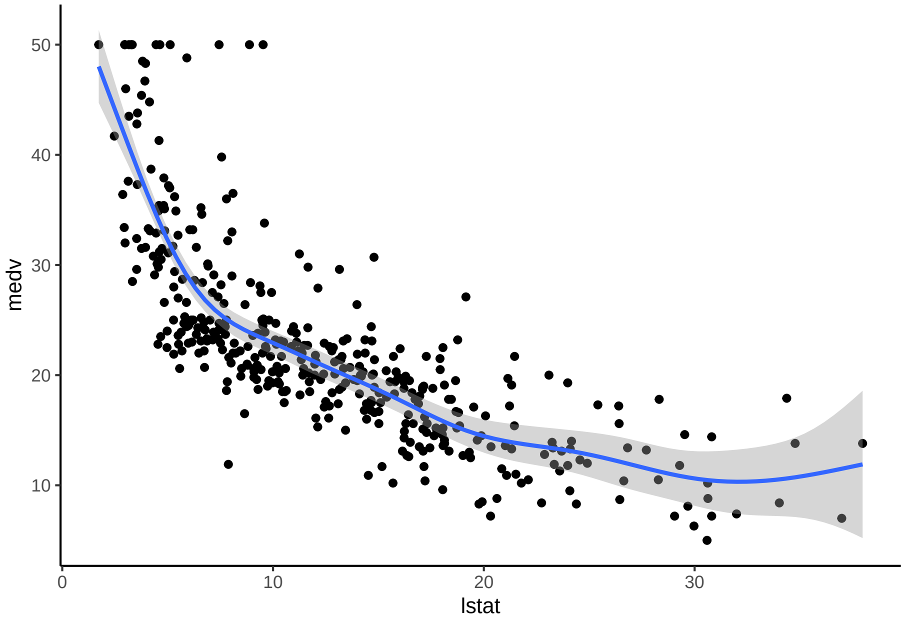
```
# Build the model
model <- gam(medv ~ s(lstat), data = train.data)
plot(model)
```



```
# Make predictions
# predictions <- model %>% predict(test.data)
# Model performance
data.frame(
  RMSE = RMSE(predictions, test.data$medv),
  R2 = R2(predictions, test.data$medv)
)
```

```
##       RMSE        R2
## 1 5.317372 0.6786367
```

```
ggplot(train.data, aes(lstat, medv) ) +
  geom_point() +
  stat_smooth(method = gam, formula = y ~ s(x))
```



```
detach(train.data)
```

Chapter 14

Quantile Regression

For academic review on quantile regression, check (Yu et al., 2003)

Linear Regression is based on the conditional mean function $E(y|x)$

In Quantile regression, we can view each points in the conditional distribution of y. Quantile regression estimates the conditional median or any other quantile of Y.

In the case that we're interested in the 50th percentile, quantile regression is median regression, also known as least-absolute-deviations (LAD) regression, minimizes $\sum_i |e_i|$

Properties of estimators β

- Asymptotically normally distributed

Advantages

- More robust to outliers compared to OLS
- In the case the dependent variable has a bimodal or multimodal (multiple humps with multiple modes) distribution, quantile regression can be extremely useful.
- Avoids parametric distribution assumption of the error process. In another word, no assumptions regarding the distribution of the error term.
- Better characterization of the data (not just its conditional mean)
- is invariant to monotonic transformations (such as log) while OLS is not. In another word, $E(g(y)) = g(E(y))$

Disadvantages

- The dependent variable needs to be continuous with no zeroes or too many repeated values.

$$y_i = x_i'\beta_q + e_i$$

Let $e(x) = y - \hat{y}(x)$, then $L(e(x)) = L(y - \hat{y}(x))$ is the loss function of the error term.

If $L(e) = |e|$ (called absolute-error loss function) then $\hat{\beta}$ can be estimated by minimizing $\sum_i |y_i - x_i'\beta|$

More specifically, the objective function is

$$Q(\beta_q) = \sum_{i:y_i \geq x_i'\beta}^{N} q|y_i - x_i'\beta_q| + \sum_{i:y_i < x_i'\beta}^{N} (1-q)|y_i - x_i'\beta_q$$

where $0 < q < 1$

The sum penalizes $q|e_i|$ for under-prediction and $(1-q)|e_i|$ for over-prediction

We use simplex method to minimize this function (cannot use analytical solution since it's non-differentiable). Standard errors can be estimated by bootstrap.

The absolute-error loss function is symmetric.

Interpretation For the jth regressor (x_j), the marginal effect is the coefficient for the qth quantile

$$\frac{\partial Q_q(y|x)}{\partial x_j} = \beta_{qj}$$

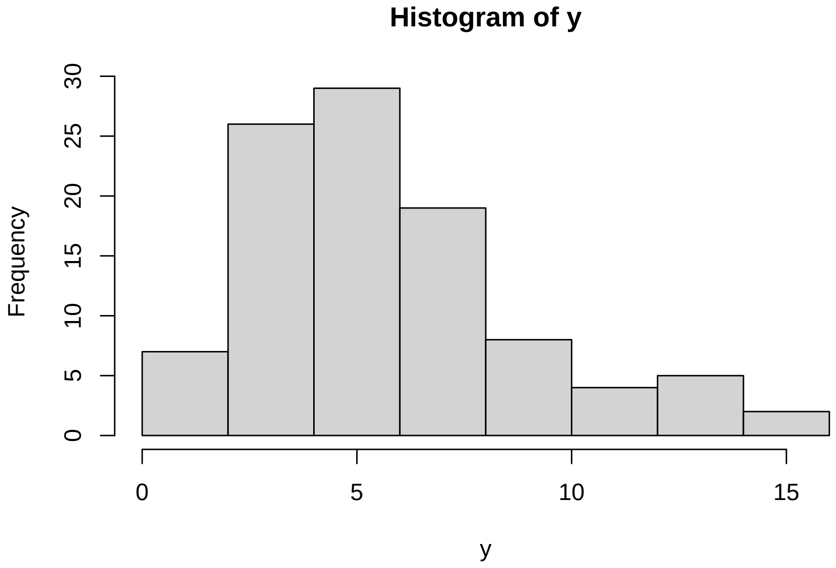
At the quantile q of the dependent variable y, β_q represents a one unit change in the independent variable x_j on the dependent variable y.

In other words, at the qth percentile, a one unit change in x results in β_q unit change in y.

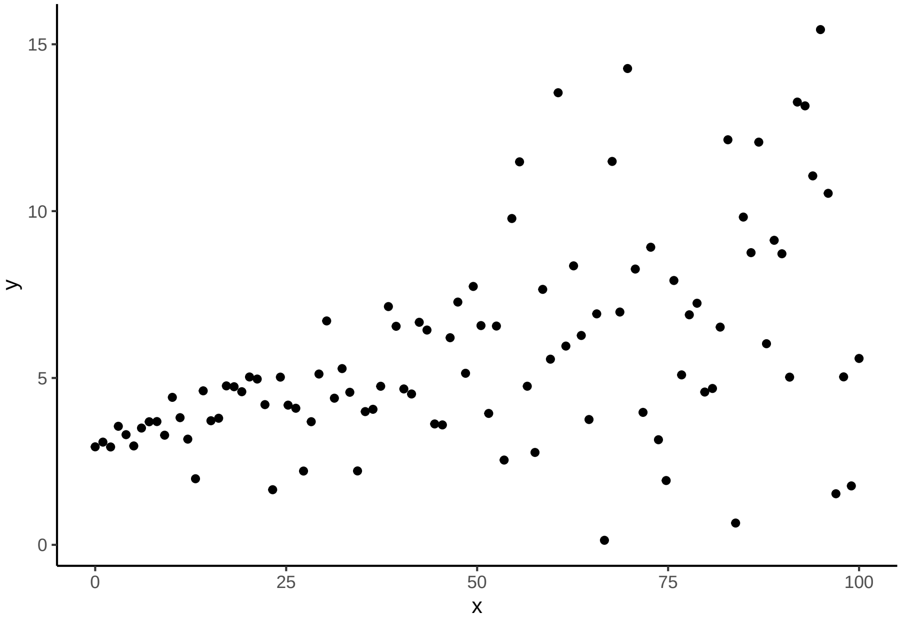
14.1 Application

```
# generate data with non-constant variance

x <- seq(0,100,length.out = 100)        # independent variable
sig <- 0.1 + 0.05*x                     # non-constant variance
b_0 <- 3                                # true intercept
b_1 <- 0.05                             # true slope
set.seed(1)                             # reproducibility
e <- rnorm(100,mean = 0, sd = sig)      # normal random error with non-constant varian
y <- b_0 + b_1*x + e                    # dependent variable
```

```
dat <- data.frame(x,y)
hist(y)
```

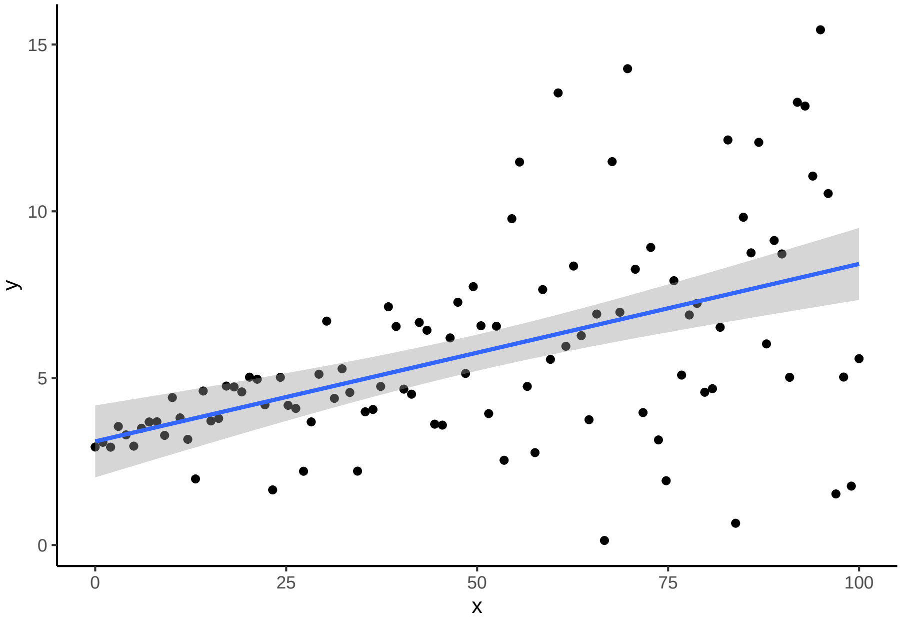


```
library(ggplot2)
ggplot(dat, aes(x,y)) + geom_point()
```



```
ggplot(dat, aes(x,y)) + geom_point() + geom_smooth(method="lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



We follow (Koenker, 1996) to estimate quantile regression

```
library(quantreg)
```

```
## Loading required package: SparseM
```

```
##
## Attaching package: 'SparseM'
```

```
## The following object is masked from 'package:base':
##
##     backsolve
```

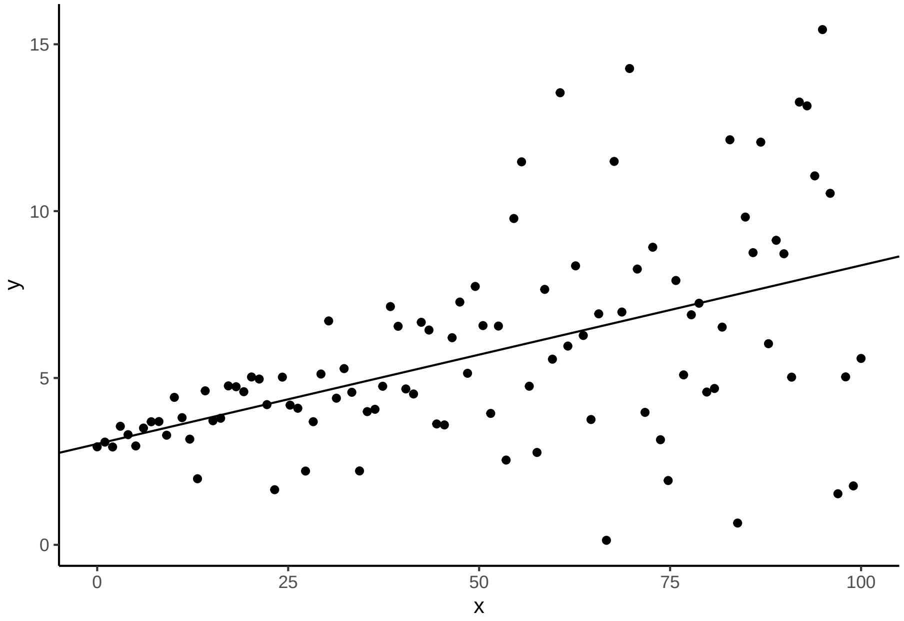
```
qr <- rq(y ~ x, data=dat, tau = 0.5) # tau: quantile of interest. Here we have it at 50th percent
summary(qr)
```

```
## Warning in rq.fit.br(x, y, tau = tau, ci = TRUE, ...): Solution may be nonunique
```

```
##
## Call: rq(formula = y ~ x, tau = 0.5, data = dat)
##
## tau: [1] 0.5
##
## Coefficients:
##             coefficients lower bd upper bd
## (Intercept) 3.02410      2.80975  3.29408
## x           0.05351      0.03838  0.06690
```

adding the regression line

```
ggplot(dat, aes(x,y)) + geom_point() +
  geom_abline(intercept=coef(qr)[1], slope=coef(qr)[2])
```



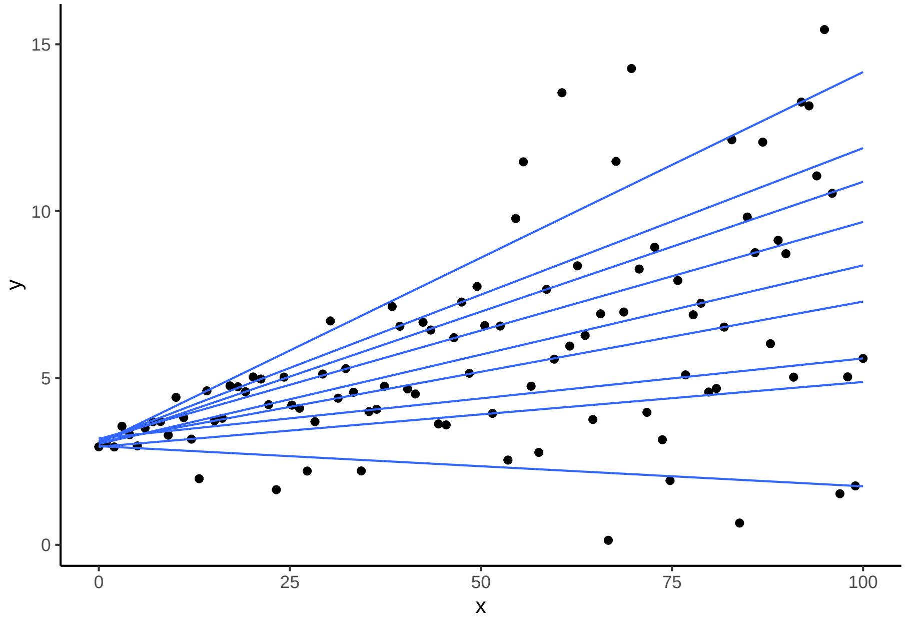
To have R estimate multiple quantile at once

```
qs <- 1:9/10
qr1 <- rq(y ~ x, data=dat, tau = qs)
#check for its coefficients
coef(qr1)
```

```
##                 tau= 0.1   tau= 0.2   tau= 0.3   tau= 0.4   tau= 0.5   tau= 0.6
## (Intercept)  2.95735740 2.93735462 3.19112214 3.08146314 3.02409828 3.16840820
## x           -0.01203696 0.01942669 0.02394535 0.04208019 0.05350556 0.06507385
##               tau= 0.7   tau= 0.8 tau= 0.9
## (Intercept) 3.09507770 3.10539343 3.041681
## x           0.07783556 0.08782548 0.111254
```

```
# plot
ggplot(dat, aes(x,y)) + geom_point() + geom_quantile(quantiles = qs)
```

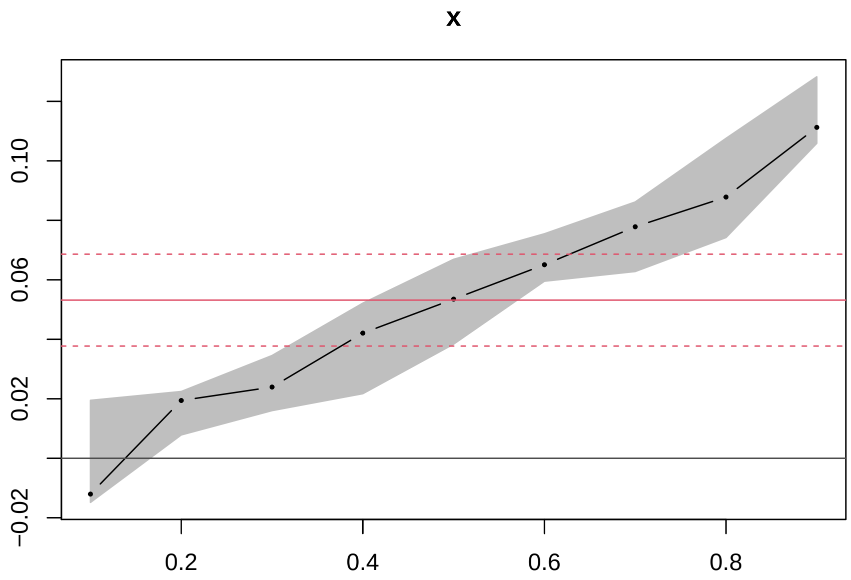
```
## Smoothing formula not specified. Using: y ~ x
```



To examine if the quantile regression is appropriate, we can see its plot compared to least squares regression

```
plot(summary(qr1), parm="x")
```

```
## Warning in rq.fit.br(x, y, tau = tau, ci = TRUE, ...): Solution may be nonunique

## Warning in rq.fit.br(x, y, tau = tau, ci = TRUE, ...): Solution may be nonunique

## Warning in rq.fit.br(x, y, tau = tau, ci = TRUE, ...): Solution may be nonunique

## Warning in rq.fit.br(x, y, tau = tau, ci = TRUE, ...): Solution may be nonunique

## Warning in rq.fit.br(x, y, tau = tau, ci = TRUE, ...): Solution may be nonunique

## Warning in rq.fit.br(x, y, tau = tau, ci = TRUE, ...): Solution may be nonunique

## Warning in rq.fit.br(x, y, tau = tau, ci = TRUE, ...): Solution may be nonunique

## Warning in rq.fit.br(x, y, tau = tau, ci = TRUE, ...): Solution may be nonunique

## Warning in rq.fit.br(x, y, tau = tau, ci = TRUE, ...): Solution may be nonunique
```

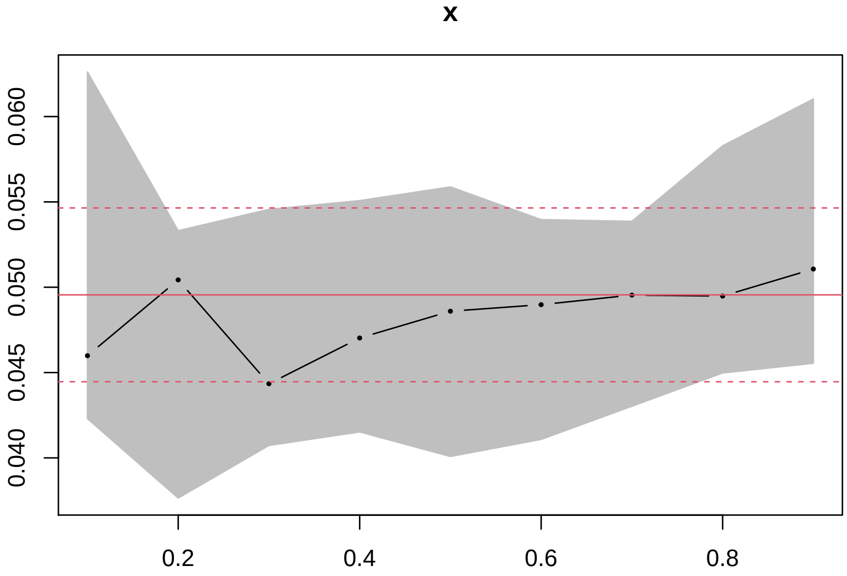


where red line is the least squares estimates, and its confidence interval. x-axis is the quantile y-axis is the value of the quantile regression coefficients at different quantile

If the error term is normally distributed, the quantile regression line will fall inside the coefficient interval of least squares regression.

```r
# generate data with constant variance

x <- seq(0, 100, length.out = 100)    # independent variable
b_0 <- 3                              # true intercept
b_1 <- 0.05                           # true slope
set.seed(1)                           # reproducibility
e <- rnorm(100, mean = 0, sd = 1)     # normal random error with constant variance
y <- b_0 + b_1 * x + e                # dependent variable
dat2 <- data.frame(x, y)
qr2 = rq(y ~ x, data = dat2, tau = qs)
plot(summary(qr2), parm = "x")
```

```
## Warning in rq.fit.br(x, y, tau = tau, ci = TRUE, ...): Solution may be nonunique

## Warning in rq.fit.br(x, y, tau = tau, ci = TRUE, ...): Solution may be nonunique

## Warning in rq.fit.br(x, y, tau = tau, ci = TRUE, ...): Solution may be nonunique

## Warning in rq.fit.br(x, y, tau = tau, ci = TRUE, ...): Solution may be nonunique
## Warning in rq.fit.br(x, y, tau = tau, ci = TRUE, ...): Solution may be nonunique

## Warning in rq.fit.br(x, y, tau = tau, ci = TRUE, ...): Solution may be nonunique

## Warning in rq.fit.br(x, y, tau = tau, ci = TRUE, ...): Solution may be nonunique

## Warning in rq.fit.br(x, y, tau = tau, ci = TRUE, ...): Solution may be nonunique

## Warning in rq.fit.br(x, y, tau = tau, ci = TRUE, ...): Solution may be nonunique
```



Part III

RAMIFICATIONS

Chapter 15

Model Specification

Test whether underlying assumptions hold true

- Nested Model (A1/A3)
- Non-Nested Model (A1/A3)
- Heteroskedasticity (A4)

15.1 Nested Model

$$
\begin{aligned}
y = \beta_0 + x_1\beta_1 + x_2\beta - 2 + x_3\beta_3 + \epsilon & \quad \text{unrestricted model} \\
y = \beta_0 + x_1\beta_1 + \epsilon & \quad \text{restricted model}
\end{aligned}
$$

Unrestricted model is always longer than the restricted model
The restricted model is "nested" within the unrestricted model
To determine which variables should be included or exclude, we could use the same Wald Test

Adjusted R^2

- R^2 will always increase with more variables included
- Adjusted R^2 tries to correct by penalizing inclusion of unnecessary variables.

$$
R^2 = 1 - \frac{SSR/n}{SST/n} R^2_{adj} = 1 - \frac{SSR/(n-k)}{SST/(n-1)} = 1 - \frac{(n-1)(1-R^2)}{(n-k)}
$$

- R^2_{adj} increases if and only if the t-statistic on the additional variable is greater than 1 in absolute value.
- R^2_{adj} is valid in models where there is no heteroskedasticity
- there fore it **should not** be used in determining which variables should be included in the model (the t or F-tests are more appropriate)

15.1.1 Chow test

Should we run two different regressions for two groups?

15.2 Non-Nested Model

compare models with different non-nested specifications

15.2.1 Davidson-Mackinnon test

15.2.1.1 Independent Variable

should the independent variables be logged? decide between non-nested alternatives

$$y = \beta_0 + x_1\beta_1 + x_2\beta_2 + \epsilon \quad \text{(level eq)}$$
$$y = \beta_0 + ln(x_1)\beta_1 + x_2\beta_2 + \epsilon \quad \text{(log eq)}$$

1. Obtain predict outcome when estimating the model in log equation $\check{y}$ and then estimate the following auxiliary equation,

$$y = \beta_0 + x_1\beta_1 + x_2\beta_2 + \check{y}\gamma + error$$

and evaluate the t-statistic for the null hypothesis $H_0 : \gamma = 0$

2. Obtain predict outcome when estimating the model in the level equation $\hat{y}$, then estimate the following auxiliary equation,

$$y = \beta_0 + ln(x_1)\beta_1 + x_2\beta_2 + \check{y}\gamma + error$$

and evaluate the t-statistic for the null hypothesis $H_0 : \gamma = 0$

- If you reject the null in the (1) step but fail to reject the null in the second step, then the log equation is preferred.
- If fail to reject the null in the (1) step but reject the null in the (2) step then, level equation is preferred.
- If reject in both steps, then you have statistical evidence that neither model should be used and should re-evaluate the functional form of your model.
- If fail to reject in both steps, you do not have sufficient evidence to prefer one model over the other. You can compare the R^2_{adj} to choose between the two models.

$$y = \beta_0 + ln(x)\beta_1 + \epsilon y = \beta_0 + x(\beta_1) + x^2\beta_2 + \epsilon$$

* Compare which better fits the data * Compare standard R^2 is unfair because the second model is less parsimonious (more parameters to estimate) * The

R^2_{adj} will penalize the second model for being less parsimonious + Only valid when there is no heteroskedasticity (A4 holds) * Should only compare after a Davidson-Mackinnon test

15.2.1.2 Dependent Variable

$$\begin{aligned} y &= \beta_0 + x_1\beta_1 + \epsilon \quad \text{level eq} \\ ln(y) &= \beta_0 + x_1\beta_1 + \epsilon \quad \text{log eq} \end{aligned}$$

- In the level model, regardless of how big y is, x has a constant effect (i.e., one unit change in x_1 results in a β_1 unit change in y)
- In the log model, the larger in y is, the effect of x is stronger (i.e., one unit change in x_1 could increase y from 1 to $1 + \beta_1$ or from 100 to 100+100xβ_1)
- Cannot compare R^2 or R^2_{adj} because the outcomes are complement different, the scaling is different (SST is different)

We need to "un-transform" the $ln(y)$ back to the same scale as y and then compare,

1. Estimate the model in the log equation to obtain the predicted outcome $\widehat{ln(y)}$
2. "Un-transform" the predicted outcome

$$\hat{m} = exp(\widehat{ln(y)})$$

3. Estimate the following model (without an intercept)

$$y = \alpha\hat{m} + error$$

and obtain predicted outcome $\hat{y}$

4. Then take the square of the correlation between $\hat{y}$ and y as a scaled version of the R^2 from the log model that can now compare with the usual R^2 in the level model.

15.3 Heteroskedasticity

- Using roust standard errors are always valid
- If there is significant evidence of heteroskedasticity implying A4 does not hold
  - Gauss-Markov Theorem no longer holds, OLS is not BLUE.
  - Should consider using a better linear unbiased estimator (Weighted Least Squares or Generalized Least Squares)

15.3.1 Breusch-Pagan test

A4 implies

$$E(\epsilon_i^2|\mathbf{x_i}) = \sigma^2$$

$$\epsilon_i^2 = \gamma_0 + x_{i1}\gamma_1 + ... + x_{ik-1}\gamma_{k-1} + error$$

and determining whether or not $\mathbf{x}_i$ has any predictive value

- if $\mathbf{x}_i$ has predictive value, then the variance changes over the levels of $\mathbf{x}_i$ which is evidence of heteroskedasticity
- if $\mathbf{x}_i$ does not have predictive value, the variance is constant for all levels of $\mathbf{x}_i$

The Breusch-Pagan test for heteroskedasticity would compute the F-test of total significance for the following model

$$e_i^2 = \gamma_0 + x_{i1}\gamma_1 + ... + x_{ik-1}\gamma_{k-1} + error$$

A low p-value means we reject the null of homoskedasticity

However, Breusch-Pagan test cannot detect heteroskedasticity in non-linear form

15.3.2 White test

test heteroskedasticity would allow for a non-linear relationship by computing the F-test of total significance for the following model (assume there are three independent random variables)

$$e_i^2 = \gamma_0 + x_i\gamma_1 + x_{i2}\gamma_2 + x_{i3}\gamma_3 + x_{i1}^2\gamma_4 + x_{i2}^2\gamma_5 + x_{i3}^2\gamma_6 + (x_{i1} \times x_{i2})\gamma_7 + (x_{i1} \times x_{i3})\gamma_8 + (x_{i2} \times x_{i3})\gamma_9 + error$$

A low p-value means we reject the null of homoskedasticity

Equivalently, we can compute LM as $LM = nR_{e^2}^2$ where the $R_{e^2}^2$ come from the regression with the squared residual as the outcome

- The LM statistic has a χ_k^2 distribution

Chapter 16

Endogeneity

Types of endogeneity

1. Endogenous Treatment

- Omitted Variables Bias

    - Motivation/choice

    - Ability/talent

    - Self-selection

- Feedback Effect (Simultaneity): also known as bidirectionality
- Measurement Error

2. Endogenous Sample Selection

16.1 Endogenous Treatment

Using the OLS estimates as a reference point

```
library(AER)
```

```
## Loading required package: car

## Loading required package: carData

## Loading required package: lmtest

## Loading required package: zoo

##
```

```
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

## Loading required package: survival
```

```
library(REndo)
```

```
## Registered S3 methods overwritten by 'lme4':
##   method                          from
##   cooks.distance.influence.merMod car
##   influence.merMod                car
##   dfbeta.influence.merMod         car
##   dfbetas.influence.merMod        car
```

```
set.seed(421)
data("CASchools")
school <- CASchools
school$stratio <- with(CASchools, students / teachers)
m1.ols <-
  lm(read ~ stratio + english + lunch + grades + income + calworks + county,
     data = school)
summary(m1.ols)$coefficients[1:7, ]
```

```
##                 Estimate Std. Error     t value      Pr(>|t|)
## (Intercept) 683.45305948 9.56214469  71.4748711 3.011667e-218
## stratio      -0.30035544 0.25797023  -1.1643027  2.450536e-01
## english      -0.20550107 0.03765408  -5.4576041  8.871666e-08
## lunch        -0.38684059 0.03700982 -10.4523759  1.427370e-22
## gradesKK-08  -1.91291321 1.35865394  -1.4079474  1.599886e-01
## income        0.71615378 0.09832843   7.2832829  1.986712e-12
## calworks     -0.05273312 0.06154758  -0.8567863  3.921191e-01
```

16.1.1 Instrumental Variable

A3a requires ϵ_i to be uncorrelated with $\mathbf{x}_i$

Assume A1 , A2, A5

$$
plim(\hat{\beta}_{OLS}) = \beta + [E(\mathbf{x_i'x_i})]^{-1}E(\mathbf{x_i'}\epsilon_i)
$$

A3a is the weakest assumption needed for OLS to be **consistent**

[A3] fails when x_{ik} is correlated with ϵ_i

- [Omitted Variables Bias] ϵ_i includes any other factors that may influence the dependent variable (linearly)
- [Feedback Effect (Simultaneity)] Demand and prices are simultaneously determined.
- [Endogenous sample design (sample selection)] we did not have iid sample
- [Measurement Error]

Note

- Omitted Variable: an omitted variable is a variable, omitted from the model (but is in the ϵ_i) and unobserved has predictive power towards the outcome.

- Omitted Variable Bias: is the bias (and inconsistency when looking at large sample properties) of the OLS estimator when the omitted variable.

The **structural equation** is used to emphasize that we are interested understanding a **causal relationship**

$$y_{i1} = \beta_0 + \mathbf{z}_i 1\beta_1 + y_{i2}\beta_2 + \epsilon_i$$

where

- y_{it} is the outcome variable (inherently correlated with ϵ_i)
- y_{i2} is the endogenous covariate (presumed to be correlated with ϵ_i)
- β_1 represents the causal effect of y_{i2} on y_{i1}
- $\mathbf{z}_{i1}$ is exogenous controls (uncorrelated with ϵ_i) ($E(z'_{1i}\epsilon_i) = 0$)

OLS is an inconsistent estimator of the causal effect β_2

If there was no endogeneity

- $E(y'_{i2}\epsilon_i) = 0$
- the exogenous variation in y_{i2} is what identifies the causal effect

If there is endogeneity

- Any wiggle in y_{i2} will shift simultaneously with ϵ_i

$$plim(\hat{\beta}_{OLS}) = \beta + [E(\mathbf{x_i'x_i})]^{-1}E(\mathbf{x_i'}\epsilon_i)$$

where

- β is the causal effect
- $[E(\mathbf{x_i'x_i})]^{-1}E(\mathbf{x_i'}\epsilon_i)$ is the endogenous effect

Hence $\hat{\beta}_{OLS}$ can be either more positive and negative than the true causal effect.

Motivation for **Two Stage Least Squares (2SLS)**

$$y_{i1} = \beta_0 + \mathbf{z}_{i1}\beta_1 + y_{i2}\beta_2 + \epsilon_i$$

We want to understand how movement in y_{i2} effects movement in y_{i1}, but whenever we move y_{i2}, ϵ_i also moves.

Solution
We need a way to move y_{i2} independently of ϵ_i, then we can analyze the response in y_{i1} as a causal effect

- Find an **instrumental variable(s)** z_{i2}
  - Instrument Relevance**: when** z_{i2} moves then y_{i2} also moves
  - Instrument Exogeneity**: when z_{i2} moves then ϵ_i does not move.
- z_{i2} is the **exogenous variation that identifies** the causal effect β_2

Finding an Instrumental variable:

- Random Assignment: + Effect of class size on educational outcomes: instrument is initial random
- Relation's Choice + Effect of Education on Fertility: instrument is parent's educational level
- Eligibility + Trade-off between IRA and 401K retirement savings: instrument is 401k eligibility

Example

Return to College

- education is correlated with ability - endogenous
- **Near 4year** as an instrument
  - Instrument Relevance: when **near** moves then education also moves
  - Instrument Exogeneity: when **near** moves then ϵ_i does not move.
- Other potential instruments; near a 2-year college. Parent's Education. Owning Library Card

$$y_{i1} = \beta_0 + \mathbf{z}_{i1}\beta_1 + y_{i2}\beta_2 + \epsilon_i$$

First Stage (Reduced Form) Equation:

$$y_{i2} = \pi_0 + \mathbf{z_{i1}}\ {}_{\mathbf{1}} + \mathbf{z_{i2}}\ {}_{\mathbf{2}} + v_i$$

where

- $\pi_0 + \mathbf{z_{i1}}\ {}_{\mathbf{1}} + \mathbf{z_{i2}}\ {}_{\mathbf{2}}$ is exogenous variation v_i is endogenous variation

This is called a **reduced form equation**
* Not interested in the causal interpretation of π_1 or π_2 * A linear projection of z_{i1} and z_{i2} on y_{i2} (simple correlations) * The projections π_1 and π_2 guarantee that $E(z_{i1}'v_i) = 0$ and $E(z_{i2}'v_i) = 0$

Instrumental variable z_{i2}

- **Instrument Relevance**: $\pi_2 \neq 0$
- **Instrument Exogeneity**: $E(\mathbf{z_{i2}}\ {}_{\mathbf{i}}) = 0$

Moving only the exogenous part of y_i2 is moving

$$\tilde{y}_{i2} = \pi_0 + \mathbf{z_{i1}}\ {}_{\mathbf{1}} + \mathbf{z_{i2}}\ {}_{\mathbf{2}}$$

two Stage Least Squares (2SLS)

$$y_{i1} = \beta_0 + \mathbf{z_{i1}}\ {}_{\mathbf{1}} + y_{i2}\beta_2 + \epsilon_i$$

$$y_{i2} = \pi_0 + \mathbf{z_{i2}}\ {}_{\mathbf{2}} + \mathbf{v_i}$$

Equivalently,

$$y_{i1} = \beta_0 + \mathbf{z_{i1}}\beta_1 + \tilde{y}_{i2}\beta_2 + u_i \tag{16.1}$$

where

- $\tilde{y}_{i2} = \pi_0 + \mathbf{z_{i2}}\ {}_{\mathbf{2}}$
- $u_i = v_i\beta_2 + \epsilon_i$

The (16.1) holds for A1, A5

- A2 holds if the instrument is relevant $\pi_2 \neq 0 + y_{i1} = \beta_0 + \mathbf{z_{i1}}\ {}_{\mathbf{1}} + (\ {}_{\mathbf{0}} + \mathbf{z_{i1}}\ {}_{\mathbf{1}} + \mathbf{z_{i2}}\ {}_{\mathbf{2}})\beta_2 + u_i$
- A3a holds if the instrument is exogenous $E(\mathbf{z}_{i2}\epsilon_i) = 0$

$$\begin{aligned}
E(\tilde{y}_{i2}'u_i) &= E((\pi_0 + \mathbf{z_{i1}}\ {}_{\mathbf{1}} + \mathbf{z_{i2}})(v_i\beta_2 + \epsilon_i)) \\
&= E((\pi_0 + \mathbf{z_{i1}}\ {}_{\mathbf{1}} + \mathbf{z_{i2}})(\epsilon_i)) \\
&= E(\epsilon_i)\pi_0 + E(\epsilon_i z_{i1})\pi_1 + E(\epsilon_i z_{i2}) \\
&= 0
\end{aligned}$$

Hence, (16.1) is consistent

The 2SLS Estimator
1. Estimate the first stage using OLS

$$y_{i2} = \pi_0 + {\mathbf{z_{i2}}}_{\mathbf{2}} + \mathbf{v_i}$$

and obtained estimated value $\hat{y}_{i2}$

2. Estimate the altered equation using OLS

$$y_{i1} = \beta_0 + {\mathbf{z_{i1}}}_{\mathbf{1}} + \hat{y}_{i2}\beta_2 + \epsilon_i$$

Properties of the 2SLS Estimator

- Under A1, A2, A3a (for z_{i1}), A5 and if the instrument satisfies the following two conditions, + **Instrument Relevance**: $\pi_2 \neq 0$ + **Instrument Exogeneity**: $E(\mathbf{z}'_{i2}\epsilon_i) = 0$ then the 2SLS estimator is consistent
- Can handle more than one endogenous variable and more than one instrumental variable

$$y_{i1} = \beta_0 + z_{i1}\beta_1 + y_{i2}\beta_2 + y_{i3}\beta_3 + \epsilon_i y_{i2} = \pi_0 + z_{i1}\pi_1 + z_{i2}\pi_2 + z_{i3}\pi_3 + z_{i4}\pi_4 + v_{i2} y_{i3} = \gamma_0 + z_{i1}\gamma_1 + z_{i2}\gamma_2 + z_{i3}\gamma_3 + z$$

```
    + **IV estimator**: one endogenous variable with a single instrument
    + **2SLS estimator**: one endogenous variable with multiple instruments
    + **GMM estimator**: multiple endogenous variables with multiple instruments
```

- Standard errors produced in the second step are not correct

    - Because we do not know $\tilde{y}$ perfectly and need to estimate it in the firs step, we are introducing additional variation

    - We did not have this problem with FGLS because "the first stage was orthogonal to the second stage." This is generally not true for most multi-step procedure.

    - If A4 does not hold, need to report robust standard errors.

- 2SLS is less efficient than OLS and will always have larger standard errors.

    - First, $Var(u_i) = Var(v_i\beta_2 + \epsilon_i) > Var(\epsilon_i)$

  - Second, $\hat{y}_{i2}$ is generally highly collinear with $\mathbf{z}_{i1}$

- The number of instruments need to be at least as many or more the number of endogenous variables.

Note

- 2SLS can be combined with FGLS to make the estimator more efficient: You have the same first-stage, and in the second-stage, instead of using OLS, you can use FLGS with the weight matrix $\hat{w}$

- Generalized Method of Moments can be more efficient than 2SLS.

- In the second-stage of 2SLS, you can also use MLE, but then you are making assumption on the distribution of the outcome variable, the endogenous variable, and their relationship (joint distribution).

16.1.1.1 Testing Assumption

1. Test of Endogeneity: Is y_{i2} truly endogenous (i.e., can we just use OLS instead of 2SLS)?
2. Testing Instrument's assumptions

   - Exogeneity: Cannot always test (and when you can it might not be informative)

   -

16.1.1.1.1 Test of Endogeneity

- 2SLS is generally so inefficient that we may prefer OLS if there is not much endogeneity
- Biased but inefficient vs efficient but biased
- Want a sense of "how endogenous" y_{i2} is

  - if "very" endgeneous - should use 2SLS

  - if not "very" endogenous - perhaps prefer OLS

Invalid Test of Endogeneity * y_{i2} is endogenous if it is correlated with ϵ_i,

$$\epsilon_i = \gamma_0 + y_{i2}\gamma_1 + error_i$$

where $\gamma_1 \neq 0$ implies that there is endogeneity

- ϵ_i is not observed, but using the residuals

$$e_i = \gamma_0 + y_{i2}\gamma_1 + error_i$$

is **NOT** a valid test of endogeneity + The OLS residual, e is mechanically uncorrelated with y_{i2} (by FOC for OLS) + In every situation, γ_1 will be essentially 0 and you will never be able to reject the null of no endogeneity

Valid test of endogeneity

- If y_{i2} is not endogenous then ϵ_i and v are uncorrelated

$$y_{i1} = \beta_0 + \mathbf{z}_{i1}\beta_1 + y_{i2}\beta_2 + \epsilon_i y_{i2} = \pi_0 + \mathbf{z}_{i1}\pi_1 + z_{i2}\pi_2 + v_i$$

variable Addition test: include the first stage residuals as an additional variable,

$$y_{i1} = \beta_0 + \mathbf{z}_{i1}\beta_1 + y_{i2}\beta_2 + \hat{v}_i\theta + error_i$$

Then the usual t-test of significance is a valid test to evaluate the following hypothesis. **note** this test requires your instrument to be valid instrument.

$$\begin{aligned} H_0 &: \theta = 0 \quad \text{(not endogenous)} \\ H_1 &: \theta \neq 0 \quad \text{(endogenous)} \end{aligned}$$

16.1.1.1.2 Testing Instrument's assumptions The instrumental variable must satisfy

1. Exogeneity
2. Relevancy

16.1.1.1.2.1 Exogeneity Why exogeneity matter?

$$E(\mathbf{z}'_{i2}\epsilon_i) = 0$$

- If A3a fails - 2SLS is also inconsistent
- If instrument is not exogenous, then we need to find a new one.
- Similar to Test of Endogeneity, when there is a single instrument

$$e_i = \gamma_0 + \mathbf{z}_{i2}\gamma_1 + error_i H_0 : \gamma_1 = 0$$

is **NOT** a valid test of endogeneity
* the OLS residual, e is mechanically uncorrelated with z_{i2}: $\hat{\gamma}_1$ will be essentially 0 and you will never be able to determine if the instrument is endogenous.

Solution

Testing Instrumental Exegeneity in an Over-identified Model * When there is more than one exogenous instrument (per endogenous variable), we can test for instrument exogeneity.
+ When we have multiple instruments, the model is said to be over-identiifed.
+ Could estimate the same model several ways (i.e., can identify/ estimate β_1 more than one way)
* Idea behind the test: if the controls and instruments are truly exogenous then OLS estimation of the following regression,

$$\epsilon_i = \gamma_0 + \mathbf{z}_{i1}\gamma_1 + \mathbf{z}_{i2}\gamma_2 + error_i$$

should have a very low R^2
* if the model is **just identified** (one instrument per endogenous variable) then the $R^2 = 0$

Steps:

(1) Estimate the structural equation by 2SLS (using all available instruments) and obtain the residuals e

(2) Regress e on all controls and instruments and obtain the R^2

(3) Under the null hypothesis (all IV's are uncorrelated), $nR^2 \sim \chi^2(q)$, where q is the number of instrumental variables minus the number of endogenous variables

    - if the model is just identified (one instrument per endogenous variable) then q = 0, and the distribution under the null collapses.

low p-value means you reject the null of exogenous instruments. Hence you would like to have high p-value in this test.

Pitfalls for the Overid test

- the overid test is essentially compiling the following information.

    - Conditional on first instrument being exogenous is the other instrument exogenous?

    - Conditional on the other instrument being exogenous, is the first instrument exogenous?

- If all instruments are endogenous than neither test will be valid
- really only useful if one instrument is thought to be truly exogenous (randomly assigned). even f you do reject the null, the test does not tell you which instrument is exogenous and which is endogenous.

Result	Implication
reject the null	you can be pretty sure there is an endogenous instrument, but don't know which one.
fail to reject	could be either (1) they are both exogenous, (2) they are both endogenous.

16.1.1.1.2.2 Relevancy

Why Relevance matter?

$$\pi_2 \neq 0$$

* used to show A2 holds + If $\pi_2 = 0$ (instrument is not relevant) then A2 fails - perfect multicollinearity
+ If π_2 is close to 0 (**weak instrument**) then there is near perfect multicollinearity - 2SLS is highly inefficient (Large standard errors).
* A weak instrument will exacerbate any inconsistency due to an instrument being (even slightly) endogenous.
+ In the simple case with no controls and a single endogenous variable and single instrumental variable,

$$plim(\hat{\beta}_{2_{2SLS}}) = \beta_2 + \frac{E(z_{i2}\epsilon_i)}{E(z_{i2}y_{i2})}$$

Testing Weak Instruments

- can use t-test (or F-test for over-identified models) in the first stage to determine if there is a weak instrument problem.
- (Stock and Yogo, 2005): a statistical rejection of the null hypothesis in the first stage at the 5% (or even 1%) level is not enough to insure the instrument is not weak

    - Rule of Thumb: need a F-stat of at least 10 (or a t-stat of at least 3.2) to reject the null hypothesis that the instrument is weak.

Summary of the 2SLS Estimator

$$y_{i1} = \beta_0 + \mathbf{z}_{i1}\beta_1 + y_{i2}\beta_2 + \epsilon_i y_{i2} = \pi_0 + \mathbf{z_{i1\ 1}} + \mathbf{z_{i2\ 2}} + v_i$$

- when A3a does not hold

$$E(y'_{i2}\epsilon_i) \neq 0$$

- Then the OLS estimator is no longer unbiased or consistent.
  * If we have valid instruments $\mathbf{z}_{i2}$

-

- Relevancy: $\pi_2 \neq 0$ Then the 2SLS estimator is consistent under A1, A2, A5a, and the above two conditions. + If A4 also holds, then the usual standard errors are valid. + If A4 does not hold then use the robust standard errors.

$$y_{i1} = \beta_0 + \mathbf{z}_{i1}\beta_1 + y_{i2}\beta_2 + \epsilon_i y_{i2} = \pi_0 + \mathbf{z_{i1}}_{\mathbf{1}} + \mathbf{z_{i2}}_{\mathbf{2}} + v_i$$

* When A3a does hold

$$E(y_{i2}'\epsilon_i) = 0$$

and we have valid instruments, then both the OLS and 2SLS estimators are consistent.
+ The OLS estimator is always more efficient + can use the variable addition test to determine if 2SLS is need (A3a does hold) or if OLS is valid (A3a does not hold)

Sometimes we can test the assumption for instrument to be valid:
+ Exogeneity: Only table when there are more instruments than endogenous variables. + Relevancy: Always testable, need the F-stat to be greater than 10 to rule out a weak instrument

Application

Expenditure as observed instrument

```
m2.2sls <-
        ivreg(
                read ~ stratio + english + lunch + grades + income + calworks +
                        county | expenditure + english + lunch + grades + income + calworks +
                        county ,
                data = school
        )
summary(m2.2sls)$coefficients[1:7, ]
```

```
##                 Estimate  Std. Error     t value      Pr(>|t|)
## (Intercept) 700.47891593 13.58064436  51.5792106 8.950497e-171
## stratio      -1.13674002  0.53533638  -2.1234126  3.438427e-02
## english      -0.21396934  0.03847833  -5.5607753  5.162571e-08
## lunch        -0.39384225  0.03773637 -10.4366757  1.621794e-22
## gradesKK-08  -1.89227865  1.37791820  -1.3732881  1.704966e-01
## income        0.62487986  0.11199008   5.5797785  4.668490e-08
## calworks     -0.04950501  0.06244410  -0.7927892  4.284101e-01
```

16.1.2 Internal instrumental variable

(also **instrument free methods**). This section is based on Raluca Gui's guide

alternative to external instrumental variable approaches

All approaches here assume a **continuous dependent variable**

Application

16.1.2.1 Non-hierarchical Data (Cross-classified)

$$Y_t = \beta_0 + \beta_1 P_t + \beta_2 X_t + \epsilon_t$$

where

- $t = 1, .., T$ (indexes either time or cross-sectional units)
- Y_t is a k x 1 response variable
- X_t is a k x n exogenous regressor
- P_t is a k x 1 continuous endogenous regressor
- ϵ_t is a structural error term with $\mu_\epsilon = 0$ and $E(\epsilon^2) = \sigma^2$
- β are model parameters

The endogeneity problem arises from the correlation of P_t and ϵ_t:

$$P_t = \gamma Z_t + v_t$$

where

- Z_t is a l x 1 vector of internal instrumental variables
- ν_t is a random error with $\mu_{v_t}, E(v^2) = \sigma_v^2, E(\epsilon v) = \sigma_{\epsilon v}$
- Z_t is assumed to be stochastic with distribution G
- ν_t is assumed to have density h($\cdot$)

16.1.2.1.1 Latent Instrumental Variable (Ebbes et al., 2005)

assume Z_t (unobserved) to be uncorrelated with ϵ_t, which is similar to Instrumental Variable. Hence, Z_t and ν_t can't be identified without distributional assumptions

The distributions of Z_t and ν_t need to be specified such that:

(1) endogeneity of P_t is corrected

(2) the distribution of P_t is empirically close to the integral that expresses the amount of overlap of Z as it is shifted over (= the convolution between Z_t and ν_t).

When the density h($\cdot$) = Normal, then G cannot be normal because the parameters would not be identified (Ebbes et al., 2005) .

Hence,

- in the LIV model the distribution of Z_t is discrete

- in the Higher Moments Method and Joint Estimation Using Copula methods, the distribution of Z_t is taken to be skewed.

Z_t are assumed **unobserved, discrete and exogenous**, with

- an unknown number of groups m

- γ is a vector of group means.

Identification of the parameters relies on the distributional assumptions of

- P_t: a non-Gaussian distribution

- Z_t discrete with $m \geq 2$

Note:

- If Z_t is continuous, the model is unidentified

- If $P_t \sim N$, you have inefficient estimates.

```
m3.liv <- latentIV(read ~ stratio, data=school)
```

```
## No start parameters were given. The linear model read ~ stratio is fitted to derive them.

## The start parameters c((Intercept)=706.449, stratio=-2.621, pi1=19.64, pi2=21.532, theta5=0.5,
```

```
summary(m3.liv)$coefficients[1:7,]
```

```
##                  Estimate    Std. Error       z-score     Pr(>|z|)
## (Intercept)   6.996014e+02  2.686186e+02  2.604441e+00 9.529597e-03
## stratio      -2.272673e+00  1.367757e+01 -1.661605e-01 8.681108e-01
## pi1          -4.896363e+01  5.526907e-08 -8.859139e+08 0.000000e+00
## pi2           1.963920e+01  9.225351e-02  2.128830e+02 0.000000e+00
## theta5       6.939432e-152 3.354672e-160  2.068587e+08 0.000000e+00
## theta6        3.787512e+02  4.249457e+01  8.912932e+00 1.541524e-17
## theta7       -1.227543e+00  4.885276e+01 -2.512741e-02 9.799653e-01
```

it will return a coefficient very different from the other methods since there is only one endogenous variable.

16.1.2.1.2 Joint Estimation Using Copula assume Z_t (unobserved) to be uncorrelated with ϵ_t, which is similar to Instrumental Variable. Hence, Z_t and ν_t can't be identified without distributional assumptions

(Park and Gupta, 2012) allows joint estimation of the continuous P_t and ϵ_t using Gaussian copulas, where a copula is a function that maps several conditional distribution functions (CDF) into their joint CDF).

The underlying idea is that using information contained in the observed data, one selects marginal distributions for P_t and ϵ_t. Then, the copula model constructs a flexible multivariate joint distribution that allows a wide range of correlations between the two marginals.

The method allows both continuous and discrete P_t.

In the special case of **one continuous** P_t, estimation is based on MLE Otherwise, based on Gaussian copulas, augmented OLS estimation is used.

Assumptions:

- skewed P_t
- the recovery of the correct parameter estimates
- $\epsilon_t \sim$ normal marginal distribution. The marginal distribution of P_t is obtained using the **Epanechnikov kernel density estimator**

  $$\hat{h}_p = \frac{1}{T.b} \sum_{t=1}^{T} K(\frac{p - P_t}{b})$$

  where
- P_t = endogenous variables
- $K(x) = 0.75(1 - x^2) I(||x|| \leq 1)$
- $b = 0.9T^{-1/5} \times min(s, IQR/1.34)$ suggested by (Silverman, 1969)
  - IQR = interquartile range
  - s = sample standard deviation
  - T = n of time periods observed in the data

In augmented OLS and MLE, the inference procedure occurs in two stages:

(1): the empirical distribution of P_t is computed
(2) used in it constructing the likelihood function)
Hence, the standard errors would not be correct.

So we use the sampling distributions (from bootstrapping) to get standard errors and the variance-covariance matrix. Since the distribution of the bootstraped parameters is highly skewed, we report the percentile confidence intervals is preferable.

```
set.seed(110)
m4.cc <-
        copulaCorrection(
                read ~ stratio + english + lunch + calworks +
                        grades + income + county | continuous(stratio),
                data = school,
                optimx.args = list(method = c("Nelder-Mead"), itnmax = 60000),
                num.boots = 2,
                verbose = FALSE
        )
```

```
## Warning: It is recommended to run 1000 or more bootstraps.
```

```
summary(m4.cc)$coefficients[1:7, ]
```

```
##             Point Estimate   Boots SE Lower Boots CI (95%) Upper Boots CI (95%)
## (Intercept)   683.06900891 2.80554212                   NA                   NA
## stratio        -0.32434608 0.02075999                   NA                   NA
## english        -0.21576110 0.01450666                   NA                   NA
## lunch          -0.37087664 0.01902052                   NA                   NA
## calworks       -0.05569058 0.02076781                   NA                   NA
## gradesKK-08    -1.92286128 0.25684614                   NA                   NA
## income          0.73595353 0.04725700                   NA                   NA
```

we run this model with only one endogenous continuous regressor (`stratio`). Sometimes, the code will not converge, in which case you can use different

- optimization algorithm
- starting values
- maximum number of iterations

16.1.2.1.3 Higher Moments Method suggested by (Lewbel, 1997) to identify ϵ_t caused by **measurement error**.

Identification is achieved by using third moments of the data, with no restrictions on the distribution of ϵ_t
The following instruments can be used with 2SLS estimation to obtain consistent estimates:

$$
\begin{aligned}
q_{1t} &= (G_t - \bar{G}) \\
q_{2t} &= (G_t - \bar{G})(P_t - \bar{P}) \\
q_{3t} &= (G_t - \bar{G})(Y_t - \bar{Y}) \\
q_{4t} &= (Y_t - \bar{Y})(P_t - \bar{P}) \\
q_{5t} &= (P_t - \bar{P})^2 \\
q_{6t} &= (Y_t - \bar{Y})^2
\end{aligned}
$$

where

- $G_t = G(X_t)$ for any given function G that has finite third own and cross moments

- X = exogenous variable

q_{5t}, q_{6t} can be used only when the measurement and ϵ_t are symmetrically distributed. The rest of the instruments does not require any distributional assumptions for ϵ_t.

Since the regressors $G(X) = X$ are included as instruments, $G(X)$ can't be a linear function of X in q_{1t}

Since this method has very strong assumptions, Higher Moments Method should only be used in case of overidentification

```r
set.seed(111)
m5.hetEr <-
        hetErrorsIV(
                read ~ stratio + english + lunch + calworks + income +
                        grades + county | stratio | IIV(income, english),
                data = school
        )
```

```
## Residuals were derived by fitting stratio ~ english + lunch + calworks + income + gr
```

```
## Warning: A studentized Breusch-Pagan test (stratio ~ english) indicates at a 95%
## confidence level that the assumption of heteroscedasticity for the variable is
## not satisfied (p-value: 0.2428). The instrument built from it therefore is weak.
```

```
## The following internal instruments were built: IIV(income), IIV(english).
```

```
## Fitting an instrumental variable regression with model read ~ stratio + english + lu
```

```r
summary(m5.hetEr)$coefficients[1:7, ]
```

```
##                 Estimate  Std. Error    t value     Pr(>|t|)
## (Intercept) 662.78791557 27.90173069 23.7543657 2.380436e-76
## stratio       0.71480686  1.31077325  0.5453322 5.858545e-01
## english      -0.19522271  0.04057527 -4.8113717 2.188618e-06
## lunch        -0.37834232  0.03927793 -9.6324402 9.760809e-20
## calworks     -0.05665126  0.06302095 -0.8989273 3.692776e-01
## income        0.82693755  0.17236557  4.7975797 2.335271e-06
## gradesKK-08  -1.93795843  1.38723186 -1.3969968 1.632541e-01
```

recommend using this approach to create additional instruments to use with external ones for better efficiency.

16.1.2.1.4 Heteroskedastic Error Approach

- using means of variables that are uncorrelated with the product of heteroskedastic errors to identify structural parameters.

- This method can be use either when you don't have external instruments or you want to use additional instruments to improve the efficiency of the IV estimator (Lewbel, 2012)

- The instruments are constructed as simple functions of data

- Model's assumptions:

$$E(X\epsilon) = 0E(Xv) = 0cov(Z, \epsilon v) = 0cov(Z, v^2) \neq 0 \text{ (for identification)}$$

Structural parameters are identified by 2SLS regression of Y on X and P, using X and [Z − E(Z)] as instruments.

$$\text{instrument's strength} \propto cov((Z - \bar{Z})v, v)$$

where $cov((Z - \bar{Z})v, v)$ is the degree of heteroskedasticity of with respect to Z (Lewbel, 2012), which can be empirically tested.

If it is zero or close to zero (i.e.,the instrument is weak), you might have imprecise estimates, with large standard errors.

- Under homoskedasticity, the parameters of the model are unidentified.

- Under heteroskedasticity related to at least some elements of X, the parameters of the model are identified.

16.1.2.2 Hierarchical Data

Multiple independent assumptions involving various random components at different levels mean that any moderate correlation between some predictors and a random component or error term can result in a significant bias of the coefficients and of the variance components. (Kim and Frees, 2007) proposed a generalized method of moments which uses both, the between and within variations of the exogenous variables, but only assumes the within variation of the variables to be endogenous.

Assumptions

- the errors at each level $\sim iidN$

- the slope variables are exogenous

- the level-1 $\epsilon \perp X, P$. If this is not the case, additional, external instruments are necessary

Hierarchical Model

$$Y_{cst} = Z^1_{cst}\beta^1_{cs} + X^1_{cst}\beta_1 + \epsilon^1_{cst}\beta^1_{cs} = Z^2_{cs}\beta^2_c + X^2_{cst}\beta_2 + \epsilon^2_{cst}\beta^2_c = X^3_c\beta_3 + \epsilon^3_c$$

Bias could stem from:

- errors at the higher two levels ($\epsilon^3_c, \epsilon^2_{cst}$) are correlated with some of the regressors

- only third level errors (ϵ^3_c) are correlated with some of the regressors

(Kim and Frees, 2007) proposed

- When all variables are assumed exogenous, the proposed estimator equals the random effects estimator

- When all variables are assumed endogenous, it equals the fixed effects estimator

- also use omitted variable test (based on the Hausman-test (Hausman, 1978) for panel data), which allows the comparison of a robust estimator and an estimator that is efficient under the null hypothesis of no omitted variables or the comparison of two robust estimators at different levels.

```
set.seed(113)
school$gr08 <- school$grades == "KK-06"
m7.multilevel <-
        multilevelIV(read ~ stratio + english + lunch + income + gr08 +
                            calworks + (1 | county) | endo(stratio),
                     data = school)
```

```
## Fitting linear mixed-effects model read ~ stratio + english + lunch + income + gr08

## Detected multilevel model with 2 levels.

## For county (Level 2), 45 groups were found.
```

```
summary(m7.multilevel)$coefficients[1:7, ]
```

```
##                Estimate Std. Error     z-score     Pr(>|z|)
## (Intercept) 675.8228656 5.58008680 121.1133248 0.000000e+00
## stratio      -0.4956054 0.23922638  -2.0717005 3.829339e-02
## english      -0.2599777 0.03413530  -7.6160948 2.614656e-14
## lunch        -0.3692954 0.03560210 -10.3728537 3.295342e-25
## income        0.6723141 0.08862012   7.5864728 3.287314e-14
## gr08TRUE      2.1590333 1.28167222   1.6845440 9.207658e-02
## calworks     -0.0570633 0.05711701  -0.9990596 3.177658e-01
```

Another example using simulated data

- level-1 regressors: $X_{11}, X_{12}, X_{13}, X_{14}, X_{15}$, where X_{15} is correlated with the level-2 error (i.e., endogenous).

- level-2 regressors: $X_{21}, X_{22}, X_{23}, X_{24}$

- level-3 regressors: X_{31}, X_{32}, X_{33}

We estimate a three-level model with X15 assumed endogenous. Having a three-level hierarchy, `multilevelIV()` returns five estimators, from the most robust to omitted variables (FE_L2), to the most efficient (REF) (i.e. lowest mean squared error).

- The random effects estimator (REF) is efficient assuming no omitted variables
- The fixed effects estimator (FE) is unbiased and asymptotically normal even in the presence of omitted variables.

- Because of the efficiency, the random effects estimator is preferable if you think there is no omitted. variables

- The robust estimator would be preferable if you think there is omitted variables.

```
data(dataMultilevelIV)
set.seed(114)
formula1 <-
        y ~ X11 + X12 + X13 + X14 + X15 + X21 + X22 + X23 + X24 +
        X31 + X32 + X33 + (1 | CID) + (1 | SID) | endo(X15)
m8.multilevel <-
        multilevelIV(formula = formula1, data = dataMultilevelIV)
```

```
## Fitting linear mixed-effects model y ~ X11 + X12 + X13 + X14 + X15 + X21 + X22 + X23 + X24 + X
```

```
## Detected multilevel model with 3 levels.
```

```
## For CID (Level 2), 1368 groups were found.
## For SID (Level 3), 40 groups were found.
```

```r
coef(m8.multilevel)
```

```
##                     REF      FE_L2      FE_L3     GMM_L2     GMM_L3
## (Intercept) 64.3168856  0.0000000  0.0000000 64.3485944 64.3168868
## X11          3.0213405  3.0459605  3.0214255  3.0146686  3.0213403
## X12          8.9522160  8.9839088  8.9524723  8.9747533  8.9522169
## X13         -2.0194178 -2.0145054 -2.0193321 -2.0021426 -2.0194171
## X14          1.9651420  1.9791437  1.9648317  1.9658681  1.9651421
## X15         -0.5647915 -0.9777361 -0.5647621 -0.9750309 -0.5648070
## X21         -2.3316225  0.0000000 -2.2845297 -2.3052516 -2.3316215
## X22         -3.9564944  0.0000000 -3.9553644 -4.0130975 -3.9564966
## X23         -2.9779887  0.0000000 -2.9756848 -2.9488487 -2.9779876
## X24          4.9078293  0.0000000  4.9084694  4.7933756  4.9078250
## X31          2.1142348  0.0000000  0.0000000  2.1164477  2.1142349
## X32          0.3934770  0.0000000  0.0000000  0.3799626  0.3934764
## X33          0.1082086  0.0000000  0.0000000  0.1108386  0.1082087
```

```r
summary(m8.multilevel, "REF")
```

```
##
## Call:
## multilevelIV(formula = formula1, data = dataMultilevelIV)
##
## Number of levels: 3
## Number of observations: 2824
## Number of groups: L2(CID): 1368  L3(SID): 40
##
## Coefficients for model REF:
##             Estimate Std. Error z-score Pr(>|z|)
## (Intercept) 64.31689    7.87332   8.169 3.11e-16 ***
## X11          3.02134    0.02576 117.306  < 2e-16 ***
## X12          8.95222    0.02572 348.131  < 2e-16 ***
## X13         -2.01942    0.02409 -83.835  < 2e-16 ***
## X14          1.96514    0.02521  77.937  < 2e-16 ***
## X15         -0.56479    0.01950 -28.962  < 2e-16 ***
## X21         -2.33162    0.16228 -14.368  < 2e-16 ***
## X22         -3.95649    0.13119 -30.160  < 2e-16 ***
## X23         -2.97799    0.06611 -45.044  < 2e-16 ***
## X24          4.90783    0.19796  24.792  < 2e-16 ***
## X31          2.11423    0.10433  20.264  < 2e-16 ***
## X32          0.39348    0.30426   1.293   0.1959
## X33          0.10821    0.05236   2.067   0.0388 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Omitted variable tests for model REF:
##               df     Chisq  p-value
## GMM_L2_vs_REF  7     18.74 0.009040 **
## GMM_L3_vs_REF 13 -12872.98 1.000000
## FE_L2_vs_REF  13     39.99 0.000139 ***
## FE_L3_vs_REF  13     39.99 0.000138 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

True $\beta_{X_{15}} = -1$. We can see that some estimators are bias because X_{15} is correlated with the level-two error, to which only FE_L2 and GMM_L2 are robust

To select the appropriate estimator, we use the omitted variable test.

In a three-level setting, we can have different estimator comparisons:

- Fixed effects vs. random effects estimators: Test for omitted level-two and level-three omitted effects, simultaneously, one compares FE_L2 to REF. But we will not know at which omitted variables exist.

- Fixed effects vs. GMM estimators: Once the existence of omitted effects is established but not sure at which level, we test for level-2 omitted effects by comparing FE_L2 vs GMM_L3. If you reject the null, the omitted variables are at level-2 The same is accomplished by testing FE_L2 vs. GMM_L2, since the latter is consistent only if there are no omitted effects at level-2.

- Fixed effects vs. fixed effects estimators: We can test for omitted level-2 effects, while allowing for omitted level-3 effects by comparing FE_L2 vs. FE_L3 since FE_L2 is robust against both level-2 and level-3 omitted effects while FE_L3 is only robust to level-3 omitted variables.

Summary, use the omitted variable test comparing `REF vs. FE_L2` first.

- If the null hypothesis is rejected, then there are omitted variables either at level-2 or level-3
- Next, test whether there are level-2 omitted effects, since testing for omitted level three effects relies on the assumption there are no level-two omitted effects. You can use any of these pair of comparisons:

  - `FE_L2 vs. FE_L3`

  - `FE_L2 vs. GMM_L2`

- If no omitted variables at level-2 are found, test for omitted level-3 effects by comparing either

  - FE_L3 vs. GMM_L3
  - GMM_L2 vs. GMM_L3

```
summary(m8.multilevel, "REF")
```

```
##
## Call:
## multilevelIV(formula = formula1, data = dataMultilevelIV)
##
## Number of levels: 3
## Number of observations: 2824
## Number of groups: L2(CID): 1368  L3(SID): 40
##
## Coefficients for model REF:
##              Estimate Std. Error z-score Pr(>|z|)
## (Intercept) 64.31689    7.87332   8.169 3.11e-16 ***
## X11          3.02134    0.02576 117.306  < 2e-16 ***
## X12          8.95222    0.02572 348.131  < 2e-16 ***
## X13         -2.01942    0.02409 -83.835  < 2e-16 ***
## X14          1.96514    0.02521  77.937  < 2e-16 ***
## X15         -0.56479    0.01950 -28.962  < 2e-16 ***
## X21         -2.33162    0.16228 -14.368  < 2e-16 ***
## X22         -3.95649    0.13119 -30.160  < 2e-16 ***
## X23         -2.97799    0.06611 -45.044  < 2e-16 ***
## X24          4.90783    0.19796  24.792  < 2e-16 ***
## X31          2.11423    0.10433  20.264  < 2e-16 ***
## X32          0.39348    0.30426   1.293   0.1959
## X33          0.10821    0.05236   2.067   0.0388 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Omitted variable tests for model REF:
##                df     Chisq  p-value
## GMM_L2_vs_REF  7      18.74 0.009040 **
## GMM_L3_vs_REF 13 -12872.98 1.000000
## FE_L2_vs_REF  13      39.99 0.000139 ***
## FE_L3_vs_REF  13      39.99 0.000138 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# compare REF with all the other estimators. Testing REF (the most efficient estimator
```

Since the null hypothesis is rejected (p = 0.000139), there is bias in the random effects estimator.

To test for level-2 omitted effects (regardless of level-3 omitted effects), we compare FE_L2 versus FE_L3

```
summary(m8.multilevel,"FE_L2")
```

```
##
## Call:
## multilevelIV(formula = formula1, data = dataMultilevelIV)
##
## Number of levels: 3
## Number of observations: 2824
## Number of groups: L2(CID): 1368  L3(SID): 40
##
## Coefficients for model FE_L2:
##               Estimate Std. Error z-score Pr(>|z|)
## (Intercept)  0.000e+00  4.275e-19    0.00        1
## X11          3.046e+00  2.978e-02  102.30   <2e-16 ***
## X12          8.984e+00  3.360e-02  267.41   <2e-16 ***
## X13         -2.015e+00  3.107e-02  -64.83   <2e-16 ***
## X14          1.979e+00  3.203e-02   61.80   <2e-16 ***
## X15         -9.777e-01  3.364e-02  -29.06   <2e-16 ***
## X21          0.000e+00  1.824e-18    0.00        1
## X22          0.000e+00  1.303e-18    0.00        1
## X23          0.000e+00  4.389e-18    0.00        1
## X24          0.000e+00  1.724e-18    0.00        1
## X31          0.000e+00  1.468e-17    0.00        1
## X32          0.000e+00  8.265e-18    0.00        1
## X33          0.000e+00  2.793e-17    0.00        1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Omitted variable tests for model FE_L2:
##                 df Chisq  p-value
## FE_L2_vs_REF    13 39.99 0.000139 ***
## FE_L2_vs_FE_L3   9 36.02 3.92e-05 ***
## FE_L2_vs_GMM_L2 12 39.99 7.21e-05 ***
## FE_L2_vs_GMM_L3 13 39.99 0.000139 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The null hypothesis of no omitted level-2 effects is rejected ($p = 3.92e - 05$). Hence, there are omitted effects at level-two. We should use FE_L2 which is consistent with the underlying data that we generated (level-2 error correlated with $X_1$5, which leads to biased FE_L3 coefficients.

The omitted variable test between FE_L2 and GMM_L2 should reject the null hypothesis of no omitted level-2 effects (p-value is 0).

If we assume an endogenous variable as exogenous, the RE and GMM estimators will be biased because of the wrong set of internal instrumental variables. To increase our confidence, we should compare the omitted variable tests when the variable is considered endogenous vs. exogenous to get a sense whether the variable is truly endogenous.

16.1.3 Proxy Variables

Can be in place of the omitted variable,
* will not be able to estimate the effect of the omitted variable * will be able to reduce some endogeneity caused bye the omitted variable

Criteria for a proxy variable:

1. The proxy is correlated with the omitted variable.
2. Having the omitted variable in the regression will solve the problem of endogeneity 3.The variation of the omitted variable unexplained by the proxy is uncorrelated with all independent variables, including the proxy.

IQ test can be a proxy for ability in the regression between wage explained education.

For the third requirement

$$ability = \gamma_0 + \gamma_1 IQ + \epsilon$$

where ϵ is uncorrelated with education and IQ test.

16.2 Endogenous Sample Selection

sample selection or self-selection problem

the omitted variable is how people were selected into the sample

Some disciplines consider nonresponse bias and selection bias as sample selection.

- When unobservable factors that affect who is in the sample are independent of unobservable factors that affect the outcome, the sample selection is not endogenous. Hence, the sample selection is ignorable and estimator that ignores sample selection is still consistent.

- when the unobservable factors that affect who is included in the sample are correlated with the unobservable factors that affect the outcome, the sample selection is endogenous and not ignorable, because estimators that ignore endogenous sample selection are not consistent (we don't know which part of the observable outcome is related to the causal relationship and which part is due to different people were selected for the treatment and control groups).

To combat Sample selection, we can

- Randomization: participants are randomly selected into treatment and control.

- Instruments that determine the treatment status (i.e., treatment vs. control) but not the outcome (Y)

- Functional form of the selection and outcome processes: originated from (Heckman, 1976), later on generalize by (Amemiya, 1984)

We have our main model

$$\mathbf{y}^* = \mathbf{xb} +$$

However, the pattern of missingness (i.e., censored) is related to the unobserved (latent) process:

$$\mathbf{z}^* = \mathbf{w} \quad + \mathbf{u}$$

and

$$z_i = \begin{cases} 1 & \text{if } z_i^* > 0 \\ 0 & \text{if } z_i^* \le 0 \end{cases}$$

Equivalently, $z_i = 1$ (y_i is observed) when

$$u_i \ge -w_i \gamma$$

Hence, the probability of observed y_i is

$$\begin{aligned} P(u_i \ge -w_i\gamma) &= 1 - \Phi(-w_i\gamma) \\ &= \Phi(w_i\gamma) && \text{symmetry of the standard normal distribution} \end{aligned}$$

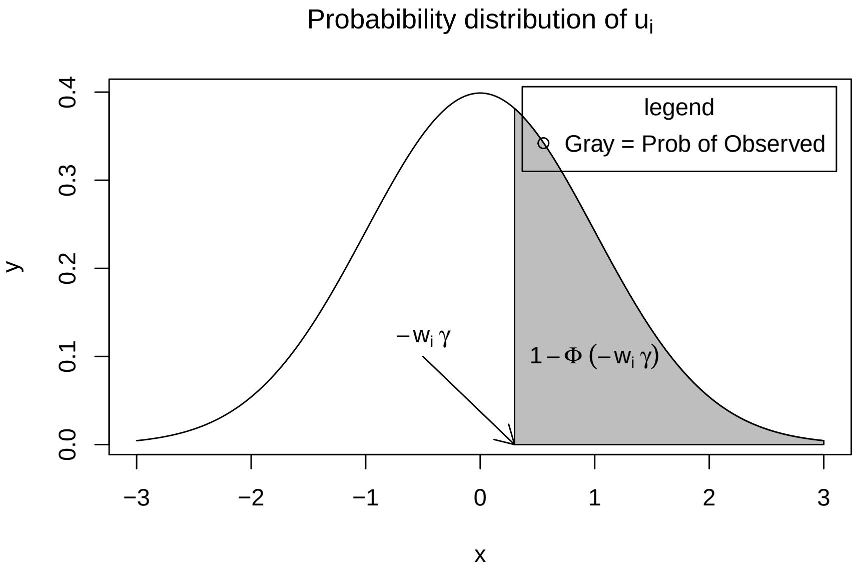
We will **assume**

- the error term of the selection $\mathbf{u} \sim \mathbf{N(0, I)}$

- $Var(u_i) = 1$ for identification purposes

Visually, $P(u_i \geq -w_i\gamma)$ is the shaded area.

```
x = seq(-3, 3, length = 200)
y = dnorm(x, mean = 0, sd = 1)
plot(x,
     y,
     type = "l",
     main = bquote("Probabibility distribution of" ~ u[i]))
x = seq(0.3, 3, length = 100)
y = dnorm(x, mean = 0, sd = 1)
polygon(c(0.3, x, 3), c(0, y, 0), col = "gray")
text(1, 0.1, bquote(1 - Phi ~ (-w[i] ~ gamma)))
arrows(-0.5, 0.1, 0.3, 0, length = .15)
text(-0.5, 0.12, bquote(-w[i] ~ gamma))
legend(
  "topright",
  "Gray = Prob of Observed",
  pch = 1,
  title = "legend",
  inset = .02
)
```



Hence in our observed model, we see

$$y_i = x_i\beta + \epsilon_i \text{when } z_i = 1 \tag{16.2}$$

and the joint distribution of the selection model (u_i), and the observed equation (ϵ_i) as

$$\left[\begin{array}{c} u \\ \epsilon \end{array}\right] \sim^{iid} N\left(\left[\begin{array}{c} 0 \\ 0 \end{array}\right], \left[\begin{array}{cc} 1 & \rho \\ \rho & \sigma^2_\epsilon \end{array}\right]\right)$$

The relation between the observed and selection models:

$$\begin{aligned}
E(y_i|y_i \text{ observed}) &= E(y_i|z^* > 0) \\
&= E(y_i| - w_i\gamma) \\
&= \mathbf{x}_i\beta + E(\epsilon_i|u_i > -w_i\gamma) \\
&= \mathbf{x}_i\beta + \rho\sigma_\epsilon \frac{\phi(w_i\gamma)}{\Phi(w_i\gamma)}
\end{aligned}$$

where $\frac{\phi(w_i\gamma)}{\Phi(w_i\gamma)}$ is the Inverse Mills Ratio. and $\rho\sigma_\epsilon \frac{\phi(w_i\gamma)}{\Phi(w_i\gamma)} \geq 0$

Great visualization of special cases of correlation patterns amongst data and errors by professor Rob Hick

Note:

(Bareinboim et al., 2014) is an excellent summary of cases that we can still do causal inference in case of selection bias. I'll try to summarize their idea here:

Let X be an action, Y be an outcome, and S be a binary indicator of entry into the data pool where (S = 1 = in the sample, S = 0 =out of sample) and Q be the conditional distribution $Q = P(y|x)$.

Usually we want to understand , but because of S, we only have $P(y, x|S = 1)$. Hence, we'd like to recover $P(y|x)$ from $P(y, x|S = 1)$

- If both X and Y affect S, we can't unbiasedly estimate $P(y|x)$

In the case of Omitted variable bias (U) and sample selection bias (S), you have unblocked extraneous "flow" of information between X and Y, which causes spurious correlation for X and Y. Traditionally, we would recover Q by parametric assumption of

(1) the data generating process (e.g., Heckman 2-step)

(2) type of data-generating model (e..g, treatment-dependent or outcome-dependent)

(3) selection's probability $P(S = 1|Pa_s)$ with non-parametrically based causal graphical models, the authors proposed more robust way to model mis-specification regardless of the type of data-generating model, and do not require selection's probability. Hence, you can recover Q

- without external data

- with external data
- causal effects with the Selection-backdoor criterion

16.2.1 Tobit-2

also known as Heckman's standard sample selection model
Assumption: joint normality of the errors

Data here is taken from (Mroz, 1987)'s paper.

We want to estimate the log(wage) for married women, with education, experience, experience squared, and a dummy variable for living in a big city. But we can only observe the wage for women who are working, which means a lot of married women in 1975 who were out of the labor force are unaccounted for. Hence, an OLS estimate of the wage equation would be bias due to sample selection. Since we have data on non-participants (i.e., those who are not working for pay), we can correct for the selection process.

The Tobit-2 estimates are consistent

16.2.1.1 Example 1

```
library(sampleSelection)
```

```
## Loading required package: maxLik

## Loading required package: miscTools

##
## Please cite the 'maxLik' package as:
## Henningsen, Arne and Toomet, Ott (2011). maxLik: A package for maximum likelihood es
##
## If you have questions, suggestions, or comments regarding the 'maxLik' package, plea
## https://r-forge.r-project.org/projects/maxlik/
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':
##
##     recode

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
```

```
##
##     intersect, setdiff, setequal, union
```

```
data("Mroz87") #1975 data on married women's pay and labor-force participation from the Panel Stu
head(Mroz87)
```

```
##   lfp hours kids5 kids618 age educ   wage repwage hushrs husage huseduc huswage
## 1   1  1610     1       0  32   12 3.3540    2.65   2708     34      12  4.0288
## 2   1  1656     0       2  30   12 1.3889    2.65   2310     30       9  8.4416
## 3   1  1980     1       3  35   12 4.5455    4.04   3072     40      12  3.5807
## 4   1   456     0       3  34   12 1.0965    3.25   1920     53      10  3.5417
## 5   1  1568     1       2  31   14 4.5918    3.60   2000     32      12 10.0000
## 6   1  2032     0       0  54   12 4.7421    4.70   1040     57      11  6.7106
##   faminc    mtr motheduc fatheduc unem city exper  nwifeinc wifecoll huscoll
## 1  16310 0.7215       12        7  5.0    0    14 10.910060    FALSE   FALSE
## 2  21800 0.6615        7        7 11.0    1     5 19.499981    FALSE   FALSE
## 3  21040 0.6915       12        7  5.0    0    15 12.039910    FALSE   FALSE
## 4   7300 0.7815        7        7  5.0    0     6  6.799996    FALSE   FALSE
## 5  27300 0.6215       12       14  9.5    1     7 20.100058     TRUE   FALSE
## 6  19495 0.6915       14        7  7.5    1    33  9.859054    FALSE   FALSE
```

```
Mroz87 = Mroz87 %>%
        mutate(kids = kids5+kids618)

library(nnet)
library(ggplot2)
library(reshape2)
```

2-stage Heckman's model:

(1) probit equation estimates the selection process (who is in the labor force?)
(2) the results from 1st stage are used to construct a variable that captures the selection effect in the wage equation. This correction variable is called the **inverse Mills ratio**.

```
# OLS: log wage regression on LF participants only
ols1 = lm(log(wage) ~ educ + exper + I( exper^2 ) + city, data=subset(Mroz87, lfp==1))
# Heckman's Two-step estimation with LFP selection equation
heck1 = heckit( lfp ~ age + I( age^2 ) + kids + huswage + educ, # the selection process, lfp = 1
                 log(wage) ~ educ + exper + I( exper^2 ) + city, data=Mroz87 )
```

Use only variables that affect the selection process in the selection equation. Technically, the selection equation and the equation of interest could have the same set of regressors. But it is not recommended because we should only use variables (or at least one) in the selection equation that affect the selection process, but not the wage process (i.e., instruments). Here, variable `kids` fulfill that role: women with kids may be more likely to stay home, but working moms with kids would not have their wages change.

Alternatively,

```
# ML estimation of selection model
ml1 = selection( lfp ~ age + I( age^2 ) + kids + huswage + educ,
                    log(wage) ~ educ + exper + I( exper^2 ) + city, data=Mroz87 )
```

```
library("stargazer")
```

```
##
## Please cite as:
##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics
##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```
library("Mediana")
library("plm")
```

```
##
## Attaching package: 'plm'
## The following objects are masked from 'package:dplyr':
##
##     between, lag, lead
```

```
# function to calculate corrected SEs for regression
cse = function(reg) {
  rob = sqrt(diag(vcovHC(reg, type = "HC1")))
  return(rob)
}

# stargazer table
stargazer(ols1, heck1, ml1,
          se=list(cse(ols1),NULL,NULL),
          title="Married women's wage regressions", type="text",
          df=FALSE, digits=4, selection.equation = T)
```

```
##
## Married women's wage regressions
## ==============================================================
##                                 Dependent variable:
##                     ------------------------------------------
##                     log(wage)                lfp
##                        OLS        Heckman        selection
##                                  selection
##                        (1)          (2)             (3)
## --------------------------------------------------------------
## age                              0.1861***       0.1842***
##                                  (0.0652)        (0.0658)
##
```

```
## I(age2)                          -0.0024***      -0.0024***
##                                   (0.0008)        (0.0008)
##
## kids                             -0.1496***      -0.1488***
##                                   (0.0383)        (0.0385)
##
## huswage                          -0.0430***      -0.0434***
##                                   (0.0122)        (0.0123)
##
## educ                0.1057***     0.1250***       0.1256***
##                      (0.0130)     (0.0228)        (0.0229)
##
## exper               0.0411***
##                      (0.0154)
##
## I(exper2)            -0.0008*
##                      (0.0004)
##
## city                  0.0542
##                      (0.0653)
##
## Constant            -0.5308***   -4.1815***      -4.1484***
##                      (0.2032)     (1.4024)        (1.4109)
##
## --------------------------------------------------------------
## Observations           428           753             753
## R2                    0.1581       0.1582
## Adjusted R2           0.1501       0.1482
## Log Likelihood                                    -914.0777
## rho                                0.0830      0.0505 (0.2317)
## Inverse Mills Ratio            0.0551 (0.2099)
## Residual Std. Error   0.6667
## F Statistic         19.8561***
## ==============================================================
## Note:                              *p<0.1; **p<0.05; ***p<0.01
```

Rho is an estimate of the correlation of the errors between the selection and wage equations. In the lower panel, the estimated coefficient on the inverse Mills ratio is given for the Heckman model. The fact that it is not statistically different from zero is consistent with the idea that selection bias was not a serious problem in this case.

If the estimated coefficient of the inverse Mills ratio in the Heckman model is not statistically different from zero, then selection bias was not a serious problem.

16.2.1.2 Example 2

This code is from R package sampleSelection

```
set.seed(0)
library("sampleSelection")
library("mvtnorm")
eps <- rmvnorm(500, c(0,0), matrix(c(1,-0.7,-0.7,1), 2, 2)) # bivariate normal disturb
xs <- runif(500)# uniformly distributed explanatory variable (vectors of explanatory v
ys <- xs + eps[,1] > 0 # probit data generating process
xo <- runif(500) # vectors of explanatory variables for outcome equation
yoX <- xo + eps[,2] # latent outcome
yo <- yoX*(ys > 0) # observable outcome
# true intercepts = 0 and our true slopes = 1
# xs and xo are independent. Hence, exclusion restriction is fulfilled
summary( selection(ys~xs, yo ~xo))
```

```
## --------------------------------------------
## Tobit 2 model (sample selection model)
## Maximum Likelihood estimation
## Newton-Raphson maximisation, 5 iterations
## Return code 1: gradient close to zero (gradtol)
## Log-Likelihood: -712.3163
## 500 observations (172 censored and 328 observed)
## 6 free parameters (df = 494)
## Probit selection equation:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.2228     0.1081  -2.061   0.0399 *
## xs            1.3377     0.2014   6.642 8.18e-11 ***
## Outcome equation:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0002265  0.1294178  -0.002    0.999
## xo           0.7299070  0.1635925   4.462 1.01e-05 ***
##    Error terms:
##       Estimate Std. Error t value Pr(>|t|)
## sigma   0.9190     0.0574  16.009  < 2e-16 ***
## rho    -0.5392     0.1521  -3.544 0.000431 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## --------------------------------------------
```

without the exclusion restriction, we generate yo using xs instead of xo.

```
yoX <- xs + eps[,2]
yo <- yoX*(ys > 0)
summary(selection(ys ~ xs, yo ~ xs))
```

```
## --------------------------------------------
## Tobit 2 model (sample selection model)
## Maximum Likelihood estimation
## Newton-Raphson maximisation, 14 iterations
## Return code 8: successive function values within relative tolerance limit (reltol)
## Log-Likelihood: -712.8298
## 500 observations (172 censored and 328 observed)
## 6 free parameters (df = 494)
## Probit selection equation:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.1984     0.1114  -1.781   0.0756 .
## xs            1.2907     0.2085   6.191 1.25e-09 ***
## Outcome equation:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.5499     0.5644  -0.974  0.33038
## xs            1.3987     0.4482   3.120  0.00191 **
##    Error terms:
##       Estimate Std. Error t value Pr(>|t|)
## sigma  0.85091    0.05352  15.899   <2e-16 ***
## rho   -0.13226    0.72684  -0.182    0.856
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## --------------------------------------------
```

We can see that our estimates are still unbiased but standard errors are substantially larger. The exclusion restriction (i.e., independent information about the selection process) has a certain identifying power that we desire. Hence, it's better to have different set of variable for the selection process from the interested equation. Without the exclusion restriction, we solely rely on the functional form identification.

16.2.2 Tobit-5

Also known as the switching regression model
Condition: There is at least one variable in X in the selection process not included in the observed process. Used when there are separate models for participants, and non-participants.

```r
set.seed(0)
vc <- diag(3)
vc[lower.tri(vc)] <- c(0.9, 0.5, 0.1)
vc[upper.tri(vc)] <- vc[lower.tri(vc)]
eps <- rmvnorm(500, c(0,0,0), vc) # 3 disturbance vectors by a 3-dimensional normal distribution
xs <- runif(500) # uniformly distributed on [0, 1]
ys <- xs + eps[,1] > 0
xo1 <- runif(500) # uniformly distributed on [0, 1]
yo1 <- xo1 + eps[,2]
xo2 <- runif(500) # uniformly distributed on [0, 1]
```

```
yo2 <- xo2 + eps[,3]
```

exclusion restriction is fulfilled when x's are independent.

```
summary(selection(ys~xs, list(yo1 ~ xo1, yo2 ~ xo2))) # one selection equation and a l
```

```
## --------------------------------------------
## Tobit 5 model (switching regression model)
## Maximum Likelihood estimation
## Newton-Raphson maximisation, 11 iterations
## Return code 1: gradient close to zero (gradtol)
## Log-Likelihood: -895.8201
## 500 observations: 172 selection 1 (FALSE) and 328 selection 2 (TRUE)
## 10 free parameters (df = 490)
## Probit selection equation:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.1550     0.1051  -1.474    0.141
## xs            1.1408     0.1785   6.390 3.86e-10 ***
## Outcome equation 1:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.02708    0.16395   0.165    0.869
## xo1          0.83959    0.14968   5.609  3.4e-08 ***
## Outcome equation 2:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.1583     0.1885   0.840    0.401
## xo2           0.8375     0.1707   4.908 1.26e-06 ***
##    Error terms:
##        Estimate Std. Error t value Pr(>|t|)
## sigma1  0.93191    0.09211  10.118   <2e-16 ***
## sigma2  0.90697    0.04434  20.455   <2e-16 ***
## rho1    0.88988    0.05353  16.623   <2e-16 ***
## rho2    0.17695    0.33139   0.534    0.594
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## --------------------------------------------
```

All the estimates are close to the true values.

Example of functional form misspecification

```
set.seed(5)
eps <- rmvnorm(1000, rep(0, 3), vc)
eps <- eps^2 - 1 # subtract 1 in order to get the mean zero disturbances
xs <- runif(1000, -1, 0) # interval [-1, 0] to get an asymmetric distribution over obs
ys <- xs + eps[,1] > 0
xo1 <- runif(1000)
yo1 <- xo1 + eps[,2]
```

```r
xo2 <- runif(1000)
yo2 <- xo2 + eps[,3]
summary(selection(ys~xs, list(yo1 ~ xo1, yo2 ~ xo2), iterlim=20))
```

```
## Warning in sqrt(diag(vc)): NaNs produced

## Warning in sqrt(diag(vc)): NaNs produced

## Warning in sqrt(diag(vcov(object, part = "full"))): NaNs produced

## --------------------------------------------
## Tobit 5 model (switching regression model)
## Maximum Likelihood estimation
## Newton-Raphson maximisation, 4 iterations
## Return code 3: Last step could not find a value above the current.
## Boundary of parameter space?
## Consider switching to a more robust optimisation method temporarily.
## Log-Likelihood: -1665.936
## 1000 observations: 760 selection 1 (FALSE) and 240 selection 2 (TRUE)
## 10 free parameters (df = 990)
## Probit selection equation:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.53698    0.05808  -9.245  < 2e-16 ***
## xs           0.31268    0.09395   3.328 0.000906 ***
## Outcome equation 1:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.70679    0.03573  -19.78   <2e-16 ***
## xo1          0.91603    0.05626   16.28   <2e-16 ***
## Outcome equation 2:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.1446         NA      NA       NA
## xo2           1.1196     0.5014   2.233   0.0258 *
##    Error terms:
##        Estimate Std. Error t value Pr(>|t|)
## sigma1  0.67770    0.01760   38.50   <2e-16 ***
## sigma2  2.31432    0.07615   30.39   <2e-16 ***
## rho1   -0.97137         NA      NA       NA
## rho2    0.17039         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## --------------------------------------------
```

Although we still have an exclusion restriction (xo1 and xo2 are independent), we now have problems with the intercepts (i.e., they are statistically significantly different from the true values zero), and convergence problems.

If we don't have the exclusion restriction, we will have a larger variance of xs

```r
set.seed(6)
xs <- runif(1000, -1, 1)
ys <- xs + eps[,1] > 0
yo1 <- xs + eps[,2]
yo2 <- xs + eps[,3]
summary(tmp <- selection(ys~xs, list(yo1 ~ xs, yo2 ~ xs), iterlim=20))
```

```
## --------------------------------------------
## Tobit 5 model (switching regression model)
## Maximum Likelihood estimation
## Newton-Raphson maximisation, 16 iterations
## Return code 8: successive function values within relative tolerance limit (reltol)
## Log-Likelihood: -1936.431
## 1000 observations: 626 selection 1 (FALSE) and 374 selection 2 (TRUE)
## 10 free parameters (df = 990)
## Probit selection equation:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.3528     0.0424  -8.321 2.86e-16 ***
## xs            0.8354     0.0756  11.050  < 2e-16 ***
## Outcome equation 1:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.55448    0.06339  -8.748   <2e-16 ***
## xs           0.81764    0.06048  13.519   <2e-16 ***
## Outcome equation 2:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.6457     0.4994   1.293    0.196
## xs            0.3520     0.3197   1.101    0.271
##    Error terms:
##        Estimate Std. Error t value Pr(>|t|)
## sigma1  0.59187    0.01853  31.935   <2e-16 ***
## sigma2  1.97257    0.07228  27.289   <2e-16 ***
## rho1    0.15568    0.15914   0.978    0.328
## rho2   -0.01541    0.23370  -0.066    0.947
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## --------------------------------------------
```

Usually it will not converge. Even if it does, the results may be seriously biased.

Note

The log-likelihood function of the models might not be globally concave. Hence, it might not converge, or converge to a local maximum. To combat this, we can use

- Different starting value
- Different maximization methods.

- refer to Non-linear Least Squares for suggestions.

16.2.2.0.1 Pattern-Mixture Models

- compared to the Heckman's model where it assumes the value of the missing data is predetermined, pattern-mixture models assume missingness affect the distribution of variable of interest (e.g., Y)
- To read more, you can check NCSU, stefvanbuuren.

Chapter 17

Imputation (Missing Data)

Imputation is usually seen as the illegitimate child of statistical analysis. Several reasons that contribute to this negative views could be:

(1) Peopled hardly do imputation correctly
(2) Imputation can only be applied to a small range of problems correctly

If you have missing data on y (dependent variable), you probability would not be able to do any imputation appropriately. However, if you have certain type of missing data (e.g., non-random missing data) in the xs variable (independent variables), then you can still salvage your collected data points with imputation.

We also need to talk why you would want to do imputation in the first place. If your purpose is inference/ explanation (valid statistical inference not optimal point prediction), then imputation would not offer much help (Rubin, 1996). However, if your purpose is prediction, you would want your standard error to be reduced by including information (non-missing data) on other variables of a data point. Then imputation could be the tool that you're looking for.

For most software packages, it will use listwise deletion or casewise deletion to have complete case analysis (analysis with only observations with all information). Not until recently that statistician can propose some methods that are a bit better than listwise deletion which are maximum likelihood and multiple imputation.

"Judging the quality of missing data procedures by their ability to recreate the individual missing values (according to hit rate, mean square error, etc) does not lead to choosing procedures that result in valid inference", (Rubin, 1996)

17.1 Assumptions

17.1.1 Missing Completely at Random (MCAR)

Missing Completely at Random, MCAR, means there is no relationship between the missingness of the data and any values, observed or missing. Those missing data points are a random subset of the data. There is nothing systematic going on that makes some data more likely to be missing than others.

The probability of missing data on a variable is unrelated to the value of it or to the values of any other variables in the data set.

Note: the "missingness" on Y can be correlated with the "missingness" on X We can compare the value of other variables for the observations with missing data, and observations without missing data. If we reject the t-test for mean difference, we can say there is evidence that the data are not MCAR. But we cannot say that our data are MCAR if we fail to reject the t-test.

- the propensity for a data point to be missing is completely random.
- There's no relationship between whether a data point is missing and any values in the data set, missing or observed.
- The missing data are just a random subset of the data.

17.1.2 Missing at Random (MAR)

Missing at Random, MAR, means there is a systematic relationship between the propensity of missing values and the observed data, but not the missing data. Whether an observation is missing has nothing to do with the missing values, but it does have to do with the values of an individual's observed variables. So, for example, if men are more likely to tell you their weight than women, weight is MAR.

MAR is weaker than MCAR

$$P(Y_{missing}|Y,X) = P(Y_{missing}|X)$$

The probability of Y missing given Y and X equal to the probability of of Y missing given X. However, it is impossible to provide evidence to the MAR condition.

- the propensity for a data point to be missing is not related to the missing data, but it is related to some of the observed data. In another word, there is a systematic relationship between the propensity of missing values and the observed data, but not the missing data.
    - For example, if men are more likely to tell you their weight than women, weight is MAR
- MAR requires that the cause of the missing data is unrelated to the missing values but may be related to the observed values of other variables.

- MAR means that the missing values are related to observed values on other variables. As an example of CD missing data, missing income data may be unrelated to the actual income values but are related to education. Perhaps people with more education are less likely to reveal their income than those with less education

17.1.3 Ignorable

The missing data mechanism is ignorable when

(1) The data are MAR
(2) the parameters in the function of the missing data process are unrelated to the parameters (of interest) that need to be estimated.

In this case, you actually don't need to model the missing data mechanisms unless you would like to improve on your accuracy, in which case you still need to be very rigorous about your approach to improve efficiency in your parameters.

17.1.4 Nonignorable

Missing Not at Random, MNAR, means there is a relationship between the propensity of a value to be missing and its values.

Example: people with the lowest education are missing on education or the sickest people are most likely to drop out of the study.

MNAR is called Nonignorable because the missing data mechanism itself has to be modeled as you deal with the missing data. You have to include some model for why the data are missing and what the likely values are.

Hence, in the case of nonignorable, the data are not MAR. Then, your parameters of interest will be biased if you do not model the missing data mechanism. One of the most widely used approach for nonignorable missing data is (Heckman, 1976)

- Another name: Missing Not at Random (MNAR): there is a relationship between the propensity of a value to be missing and its values
  - For example, people with low education will be less likely to report it.

- We need to model why the data are missing and what the likely values are.

- the missing data mechanism is related to the missing values

- It commonly occurs when people do not want to reveal something very personal or unpopular about themselves

- Complete case analysis can give highly biased results for NI missing data. If proportionally more low and moderate income individuals are left in the sample because high income people are missing, an estimate of the mean income will be lower than the actual population mean.

17.2 Solutions to Missing data

17.2.1 Listwise Deletion

Advantages:

- Can be applied to any statistical test (SEM, multi-level regression, etc.)

- In the case of MCAR, both the parameters estimates and its standard errors are unbiased.

- In the case of MAR among independent variables (not depend on the values of dependent variables), then listwise deletion parameter estimates can still be unbiased. (Little, 1992) For example, you have a model $y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$ if the probability of missing data on X1 is independent of Y, but dependent on the value of X1 and X2, then the model estimates are still unbiased.

  - The missing data mechanism the depends on the values of the independent variables are the same as stratified sampling. And stratified sampling does not bias your estimates
  - In the case of logistic regression, if the probability of missing data on any variable depends on the value of the dependent variable, but independent of the value of the independent variables, then the listwise deletion will yield biased intercept estimate, but consistent estimates of the slope and their standard errors (Vach, 1994). However, logistic regression will still fail if the probability of missing data is dependent on both the value of the dependent and independent variables.
  - Under regression analysis, listwise deletion is more robust than maximum likelihood and multiple imputation when MAR assumption is violated.

Disadvantages:

- It will yield a larger standard errors than other more sophisticated methods discussed later.
- If the data are not MCAR, but MAR, then your listwise deletion can yield biased estimates.
- In other cases than regression analysis, other sophisticated methods can yield better estimates compared to listwise deletion.

17.2.2 Pairwise Deletion

This method could only be used in the case of linear models such as linear regression, factor analysis, or SEM. The premise of this method based on that the coefficient estimates are calculated based on the means, standard deviations, and correlation matrix. Compared to listwise deletion, we still utilized as many correlation between variables as possible to compute the correlation matrix.

Advantages:

- If the true missing data mechanism is MCAR, pair wise deletion will yield consistent estimates, and unbiased in large samples
- Compared to listwise deletion: (Glasser, 1964)
  - If the correlation among variables are low, pairwise deletion is more efficient estimates than listwise
  - If the correlations among variables are high, listwise deletion is more efficient than pairwise.

Disadvantages:

- If the data mechanism is MAR, pairwise deletion will yield biased estimates.
- In small sample, sometimes covariance matrix might not be positive definite, which means coefficients estimates cannot be calculated.

Note: You need to read carefully on how your software specify the sample size because it will alter the standard errors.

17.2.3 Dummy Variable Adjustment

Also known as Missing Indicator Method or Proxy Variable

Add another variable in the database to indicate whether a value is missing.

Create 2 variables

$$D = \begin{cases} 1 & \text{data on X are missing} \\ 0 & \text{otherwise} \end{cases} \tag{17.1}$$

$$X^* = \begin{cases} X & \text{data are available} \\ c & \text{data are missing} \end{cases} \tag{17.2}$$

Note: A typical choice for c is usually the mean of X

Interpretation:

- Coefficient of D is the the difference in the expected value of Y between the group with data and the group without data on X.

- Coefficient of X* is the effect of the group with data on Y

Disadvantages:

- This method yields bias estimates of the coefficient even in the case of MCAR (Jones, 1996)

17.2.4 Imputation

17.2.4.1 Mean, Mode, Median Imputation

- Bad:
    - Mean imputation does not preserve the relationships among variables
    - Mean imputation leads to An Underestimate of Standard Errors → you're making Type I errors without realizing it.
    - Biased estimates of variances and covariances (Haitovsky, 1968)

17.2.4.2 Maximum Likelihood

When missing data are MAR and monotonic (such as in the case of panel studies), ML can be adequately in estimating coefficients.

Monotonic means that if you are missing data on X1, then that observation also has missing data on all other variables that come after it.

ML can generally handle linear models, log-linear model, but beyond that, ML still lacks both theory and software to implement.

17.2.4.2.1 Expectation-Maximization Algorithm (EM Algorithm) An iterative process:

(1) Other variables are used to impute a value (Expectation).
(2) Check whether the value is most likely (Maximization).
(3) If not, it re-imputes a more likely value.

You start your regression with your estimates based on either listwise deletion or pairwise deletion. After regressing missing variables on available variables, you obtain a regression model. Plug the missing data back into the original model, with modified variances and covariances For example, if you have missing data on X_{ij} you would regress it on available data of $X_{i(j)}$, then plug the expected value of X_{ij} back with its X_{ij}^2 turn into $X_{ij}^2 + s_{j(j)}^2$ where $s_{j(j)}^2$ stands for the residual variance from regressing X_{ij} on $X_{i(j)}$ With the new estimated model, you rerun the process until the estimates converge.

Advantages:

(1) easy to use
(2) preserves the relationship with other variables (important if you use Factor Analysis or Linear Regression later on), but best in the case of Factor Analysis, which doesn't require standard error of individuals item.

Disadvantages:

(1) Standard errors of the coefficients are incorrect (biased usually downward - underestimate)
(2) Models with overidentification, the estimates will not be efficient

17.2.4.2.2 Direct ML (raw maximum likelihood)

Advantages

(1) efficient estimates and correct standard errors.

Disadvantages:

(1) Hard to implements

17.2.4.3 Multiple Imputation

MI is designed to use "the Bayesian model-based approach to *create* procedures, and the frequentist (randomization-based approach) to *evaluate* procedures". (Rubin, 1996)

MI estimates have the same properties as ML when the data is MAR

- Consistent
- Asymptotically efficient
- Asymptotically normal

MI can be applied to any type of model, unlike Maximum Likelihood that is only limited to a small set of models.

A drawback of MI is that it will produce slightly different estimates every time you run it. To avoid such problem, you can set seed when doing your analysis to ensure its reproducibility.

17.2.4.3.1 Single Random Imputation

Random draws form the residual distribution of each imputed variable and add those random numbers to the imputed values.

For example, if we have missing data on X, and it's MCAR, then

(1) regress X on Y (Listwise Deletion method) to get its residual distribution.

(2) For every missing value on X, we substitute with $\tilde{x_i} = \hat{x_i} + \rho u_i$ where

   - u_i is a random draw from a standard normal distribution
   - x_i is the predicted value from the regression of X and Y
   - ρ is the standard deviation of the residual distribution of X regressed on Y.

However, the model you run with the imputed data still thinks that your data are collected, not imputed, which leads your standard error estimates to be too low and test statistics too high.

To address this problem, we need to repeat the imputation process which leads us to repeated imputation or multiple random imputation.

17.2.4.3.2 Repeated Imputation "Repeated imputations are draws from the posterior predictive distribution of the missing values under a specific model , a particular Bayesian model for both the data and the missing mechanism".(Rubin, 1996)

Repeated imputation, also known as, multiple random imputation, allows us to have multiple "completed" data sets. The variability across imputations will adjust the standard errors upward.

The estimate of the standard error of $\bar{r}$ (mean correlation estimates between X and Y) is

$$SE(\bar{r}) = \sqrt{\frac{1}{M}\sum_k s_k^2 + (1+\frac{1}{M})(\frac{1}{M-1})\sum_k (r_k - \bar{r})^2}$$

where M is the number of replications, r_k is the the correlation in replication k, s_k is the estimated standard error in replication k.

However, this method still considers the parameter in predicting $\tilde{x}$ is still fixed, which means we assume that we are using the true parameters to predict $\tilde{x}$. To overcome this challenge, we need to introduce variability into our model for $\tilde{x}$ by treating the parameters as a random variables and use Bayesian posterior distribution of the parameters to predict the parameters.

However, if your sample is large and the proportion of missing data is small, the extra Bayesian step might not be necessary. If your sample is small or the proportion of missing data is large, the extra Bayesian step is necessary.

Two algorithms to get random draws of the regression parameters from its posterior distribution:

- Data Augmentation
- Sampling importance/resampling (SIR)

Authors have argued for SIR superiority due to its computer time (King et al., 2001)

17.2.4.3.2.1 Data Augmentation Steps for data augmentation:

(1) Choose starting values for the parameters (e.g., for multivariate normal, choose means and covariance matrix). These values can come from previous values, expert knowledge, or from listwise deletion or pairwise deletion or EM estimation.
(2) Based on the current values of means and covariances calculate the coefficients estimates for the equation that variable with missing data is

regressed on all other variables (or variables that you think will help predict the missing values, could also be variables that are not in the final estimation model)

(3) Use the estimates in step (2) to predict values for missing values. For each predicted value, add a random error from the residual normal distribution for that variable.

(4) From the "complete" data set, recalculate the means and covariance matrix. And take a random draw from the posterior distribution of the means and covariances with Jeffreys' prior.

(5) Using the random draw from step (4), repeat step (2) to (4) until the means and covariances stabilize (converged).

The iterative process allows us to get random draws from the joint posterior distribution of both data nd parameters, given the observed data.

Rules of thumb regarding convergence:

- The higher the proportion of missing, the more iterations
- the rate of convergence for EM algorithm should be the minimum threshold for DA.
- You can also check if your distribution has been converged by diagnostic statistics Can check Bayesian Diagnostics for some introduction.

Types of chains

1. **Parallel**: Run a separate chain of iterations for each of data set. Different starting values are encouraged. For example, one could use bootstrap to generate different data set with replacement, and for each data set, calculate the starting values by EM estimates.

    - Pro: Run faster, and less likely to have dependence in the resulting data sets.
    - Con: Sometimes it will not converge

2. **Sequential** one long chain of data augmentation cycles. After burn-in and thinning, you will have to data sets

    - Pro: Converged to the true posterior distribution is more likely.
    - Con: The resulting data sets are likely to be dependent. Remedies can be thinning and burn-in.

Note on Non-normal or categorical data The normal-based methods still work well, but you will need to do some transformation. For example,

- If the data is skewed, then log-transform, then impute, the exponentiate to have the missing data back to its original metric.
- If the data is proportion, logit-transform, impute, then de-transform the missing data.

If you want to impute non-linear relationship, such as interaction between 2 variables and 1 variable is categorical. You can do separate imputation for different levels of that variable separately, then combined for the final analysis.

- If all variables that have missing data are categorical, then **unrestricted multinomial model** or **log-linear model** is recommended.
- If a single categorical variable, **logistic (logit) regression** would be sufficient.

17.2.4.4 Nonparametric/ Semiparametric Methods

17.2.4.4.1 Hot Deck Imputation

- Used by U.S. Census Bureau for public datasets
- approximate Bayesian bootstrap
  A randomly chosen value from an individual in the sample who has similar values on other variables. In other words, find all the sample subjects who are similar on other variables, then randomly choose one of their values on the missing variable.

When we have n_1 cases with complete data on Y and n_0 cases with missing data on Y

- Step 1: From n_1, take a random sample (with replacement) of n_1 cases
- Step 2: From the retrieved sample take a random sample (with replacement) of n_0 cases
- Step 3: Assign the n_0 cases in step 2 to n_0 missing data cases.
- Step 4: Repeat the process for every variable.
- Step 5: For multiple imputation, repeat the four steps multiple times.

Note:

- If we skip step 1, it reduce variability for estimating standard errors.
- Good:
  - Constrained to only possible values.
  - Since the value is picked at random, it adds some variability, which might come in handy when calculating standard errors.

17.2.4.4.2 Cold Deck Imputation Contrary to Hot Deck, Cold Deck choose value systematically from an observation that has similar values on other variables, which remove the random variation that we want.

17.2.4.4.3 Predictive Mean Matching Steps:

1. Regress Y on X (matrix of covariates) for the n_1 (i.e., non-missing cases) to get coefficients b (a $k \times 1$ vector) and residual variance estimates s^2
2. Draw randomly from the posterior predictive distribution of the residual variance (assuming a noninformative prior) by calculating $\frac{(n_1-k)s^2}{\chi^2}$, where χ^2 is a random draw from a $\chi^2_{n_1-k}$ and let $s^2_{[1]}$ be an i-th random draw

3. Randomly draw from the posterior distribution of the coefficients b, by drawing from $MVN(b, s^2_{[1]}(X'X)^{-1})$, where X is an $n_1 \times k$ matrix of X values. Then we have b_1
4. Using step 1, we can calculate standardized residuals for n_1 cases: $e_i = \frac{y_i - bx_i}{\sqrt{s^2(1-k/n_1)}}$
5. Randomly draw a sample (with replacement) of n_0 from the n_1 residuals in step 4
6. With n_0 cases, we can calculate imputed values of Y: $y_i = b_{[1]}x_i + s_{[1]}e_i$ where e_i are taken from step 5, and $b_{[1]}$ taken from step 3, and $s_{[1]}$ taken from step 2.
7. Repeat steps 2 through 6 except for step 4.

Notes:

- can be used for multiple variables where each variable is imputed using all other variables as predictor.
- can also be used for heteroskedasticity in imputed values.

Example from Statistics Globle

```
set.seed(918273)                                    # Seed
N <- 3000                                           # Sample size
y <- round(runif(N, -10, 10))                       # Target variable Y
x1 <- y + round(runif(N, 0, 50))                    # Auxiliary variable 1
x2 <- round(y + 0.25 * x1 + rnorm(N, - 3, 15))      # Auxiliary variable 2
x3 <- round(0.1 * x1 + rpois(N, 2))                 # Auxiliary variable 3
x4 <- as.factor(round(0.02 * y + runif(N)))         # Auxiliary variable 4 (categorical variable)
y[rbinom(N, 1, 0.2) == 1] <- NA                     # Insert 20% missing data in Y
data <- data.frame(y, x1, x2, x3, x4)               # Store data in dataset
head(data)                                          # First 6 rows of our data
```

```
##    y x1  x2 x3 x4
## 1  8 38  -3  6  1
## 2  1 50  -9  5  0
## 3  5 43  20  5  1
## 4 NA  9  13  3  0
## 5 -4 40 -10  6  0
## 6 NA 29  -6  5  1
```

```
library("mice")                                     # Load mice package
```

```
##
## Attaching package: 'mice'

## The following object is masked from 'package:stats':
##
##     filter

## The following objects are masked from 'package:base':
```

```
##
##     cbind, rbind
```

```
##### Impute data via predictive mean matching (single imputation)#####

imp_single <- mice(data, m = 1, method = "pmm") # Impute missing values
```

```
##
##  iter imp variable
##   1   1  y
##   2   1  y
##   3   1  y
##   4   1  y
##   5   1  y
```

```
data_imp_single <- complete(imp_single)         # Store imputed data
# head(data_imp_single)

# SInce single imputation underestiamtes stnadard errors, we use multiple imputaiton

##### Predictive mean matching (multiple imputation)#####

imp_multi <- mice(data, m = 5, method = "pmm")  # Impute missing values multiple times
```

```
##
##  iter imp variable
##   1   1  y
##   1   2  y
##   1   3  y
##   1   4  y
##   1   5  y
##   2   1  y
##   2   2  y
##   2   3  y
##   2   4  y
##   2   5  y
##   3   1  y
##   3   2  y
##   3   3  y
##   3   4  y
##   3   5  y
##   4   1  y
##   4   2  y
##   4   3  y
##   4   4  y
##   4   5  y
##   5   1  y
##   5   2  y
##   5   3  y
##   5   4  y
##   5   5  y
```

```
data_imp_multi_all <- complete(imp_multi,       # Store multiply imputed data
                           "repeated",
                           include = TRUE)
```

```
## New names:
## * y -> y...1
## * x1 -> x1...2
## * x2 -> x2...3
## * x3 -> x3...4
## * x4 -> x4...5
## * ...
```

```
data_imp_multi <- data.frame(                   # Combine imputed Y and X1-X4 (for convenience)
  data_imp_multi_all[ , 1:6], data[, 2:5])
head(data_imp_multi)                            # First 6 rows of our multiply imputed data
```

```
##   y.0 y.1 y.2 y.3 y.4 y.5 x1  x2 x3 x4
## 1   8   8   8   8   8   8 38  -3  6  1
## 2   1   1   1   1   1   1 50  -9  5  0
## 3   5   5   5   5   5   5 43  20  5  1
## 4  NA   1  -2  -4   9  -8  9  13  3  0
## 5  -4  -4  -4  -4  -4  -4 40 -10  6  0
## 6  NA   4   7   7   6   0 29  -6  5  1
```

Example from UCLA Statistical Consulting (Bruin, 2011)

```
library(mice)
library(VIM)
```

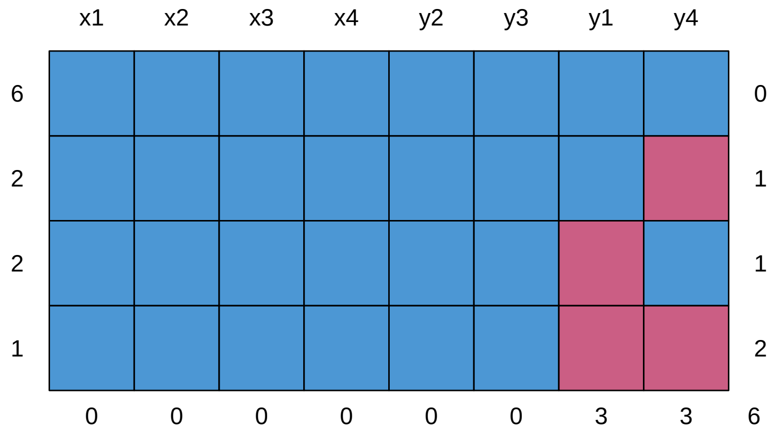
```
## Loading required package: colorspace

## Loading required package: grid

## VIM is ready to use.

## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues

##
## Attaching package: 'VIM'

## The following object is masked from 'package:datasets':
##
##     sleep
```

```
library(lattice)
library(ggplot2)
```

```
## set observations to NA
anscombe <- within(anscombe, {
    y1[1:3] <- NA
    y4[3:5] <- NA
})
## view
head(anscombe)
```

```
##   x1 x2 x3 x4   y1   y2    y3   y4
## 1 10 10 10  8   NA 9.14  7.46 6.58
## 2  8  8  8  8   NA 8.14  6.77 5.76
## 3 13 13 13  8   NA 8.74 12.74   NA
## 4  9  9  9  8 8.81 8.77  7.11   NA
## 5 11 11 11  8 8.33 9.26  7.81   NA
## 6 14 14 14  8 9.96 8.10  8.84 7.04
```

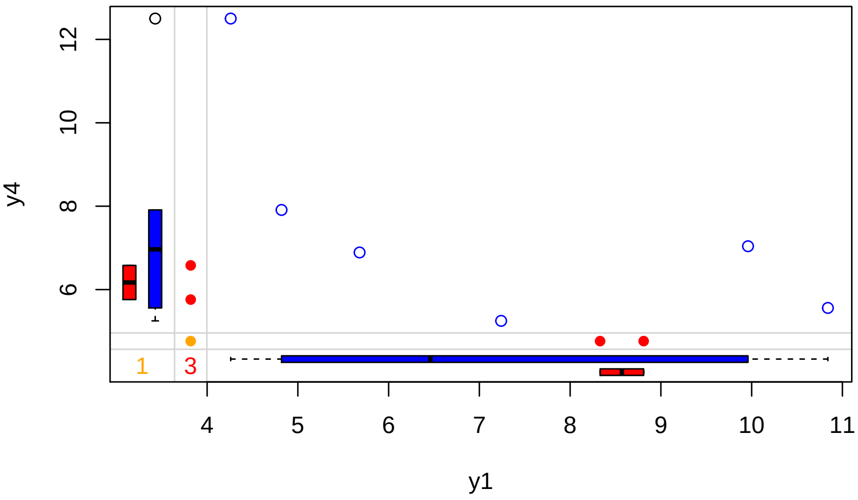
```
## check missing data patterns
md.pattern(anscombe)
```



```
##   x1 x2 x3 x4 y2 y3 y1 y4  
## 6  1  1  1  1  1  1  1  1 0
## 2  1  1  1  1  1  1  1  0 1
## 2  1  1  1  1  1  1  0  1 1
## 1  1  1  1  1  1  1  0  0 2
##    0  0  0  0  0  0  3  3 6
## Number of observations per patterns for all pairs of variables
p <- md.pairs(anscombe)
p # rr = number of observations where both pairs of values are observed
```

```
## $rr
##    x1 x2 x3 x4 y1 y2 y3 y4
## x1 11 11 11 11  8 11 11  8
## x2 11 11 11 11  8 11 11  8
## x3 11 11 11 11  8 11 11  8
## x4 11 11 11 11  8 11 11  8
## y1  8  8  8  8  8  8  8  6
## y2 11 11 11 11  8 11 11  8
## y3 11 11 11 11  8 11 11  8
## y4  8  8  8  8  6  8  8  8
##
## $rm
##    x1 x2 x3 x4 y1 y2 y3 y4
## x1  0  0  0  0  3  0  0  3
## x2  0  0  0  0  3  0  0  3
## x3  0  0  0  0  3  0  0  3
## x4  0  0  0  0  3  0  0  3
## y1  0  0  0  0  0  0  0  2
## y2  0  0  0  0  3  0  0  3
## y3  0  0  0  0  3  0  0  3
## y4  0  0  0  0  2  0  0  0
##
## $mr
##    x1 x2 x3 x4 y1 y2 y3 y4
## x1  0  0  0  0  0  0  0  0
## x2  0  0  0  0  0  0  0  0
## x3  0  0  0  0  0  0  0  0
## x4  0  0  0  0  0  0  0  0
## y1  3  3  3  3  0  3  3  2
## y2  0  0  0  0  0  0  0  0
## y3  0  0  0  0  0  0  0  0
## y4  3  3  3  3  2  3  3  0
##
## $mm
##    x1 x2 x3 x4 y1 y2 y3 y4
## x1  0  0  0  0  0  0  0  0
## x2  0  0  0  0  0  0  0  0
## x3  0  0  0  0  0  0  0  0
## x4  0  0  0  0  0  0  0  0
## y1  0  0  0  0  3  0  0  1
## y2  0  0  0  0  0  0  0  0
## y3  0  0  0  0  0  0  0  0
## y4  0  0  0  0  1  0  0  3
```

```
# rm = the number of observations where both variables are missing values
# mr = the number of observations where the first variable's value (e.g. the row varia
# mm = the number of observations where the second variable's value (e.g. the col vari

## Margin plot of y1 and y4
marginplot(anscombe[c(5, 8)], col = c("blue", "red", "orange"))
```



```
## 5 imputations for all missing values
imp1 <- mice(anscombe, m = 5)
```

```
##
##  iter imp variable
##   1   1  y1  y4
##   1   2  y1  y4
##   1   3  y1  y4
##   1   4  y1  y4
##   1   5  y1  y4
##   2   1  y1  y4
##   2   2  y1  y4
##   2   3  y1  y4
##   2   4  y1  y4
##   2   5  y1  y4
##   3   1  y1  y4
##   3   2  y1  y4
##   3   3  y1  y4
##   3   4  y1  y4
##   3   5  y1  y4
##   4   1  y1  y4
##   4   2  y1  y4
##   4   3  y1  y4
##   4   4  y1  y4
##   4   5  y1  y4
##   5   1  y1  y4
##   5   2  y1  y4
##   5   3  y1  y4
##   5   4  y1  y4
##   5   5  y1  y4
```

```
## Warning: Number of logged events: 52
```

```
## linear regression for each imputed data set - 5 regression are run
fitm <- with(imp1, lm(y1 ~ y4 + x1))
summary(fitm)
```

```
## # A tibble: 15 x 6
##    term        estimate std.error statistic p.value  nobs
##    <chr>          <dbl>     <dbl>     <dbl>   <dbl> <int>
##  1 (Intercept)    8.60      2.67      3.23  0.0121     11
##  2 y4            -0.533     0.251    -2.12  0.0667     11
##  3 x1             0.334     0.155     2.16  0.0628     11
##  4 (Intercept)    4.19      2.93      1.43  0.190      11
##  5 y4            -0.213     0.273    -0.782 0.457      11
##  6 x1             0.510     0.167     3.05  0.0159     11
##  7 (Intercept)    6.51      2.35      2.77  0.0244     11
##  8 y4            -0.347     0.215    -1.62  0.145      11
##  9 x1             0.395     0.132     3.00  0.0169     11
## 10 (Intercept)    5.48      3.02      1.81  0.107      11
## 11 y4            -0.316     0.282    -1.12  0.295      11
## 12 x1             0.486     0.173     2.81  0.0230     11
## 13 (Intercept)    7.12      1.81      3.92  0.00439    11
## 14 y4            -0.436     0.173    -2.53  0.0355     11
## 15 x1             0.425     0.102     4.18  0.00308    11
```

```
## pool coefficients and standard errors across all 5 regression models
pool(fitm)
```

```
## Class: mipo    m = 5
##          term m   estimate       ubar           b           t dfcom       df
## 1 (Intercept) 5  6.3808015 6.72703243 2.785088109 10.06913816     8 3.902859
## 2          y4 5 -0.3690455 0.05860053 0.014674911  0.07621042     8 4.716160
## 3          x1 5  0.4301588 0.02191260 0.004980516  0.02788922     8 4.856052
##         riv    lambda       fmi
## 1 0.4968172 0.3319158 0.5254832
## 2 0.3005074 0.2310693 0.4303733
## 3 0.2727480 0.2142985 0.4143230
```

```
## output parameter estimates
summary(pool(fitm))
```

```
##          term   estimate std.error statistic       df    p.value
## 1 (Intercept)  6.3808015 3.1731905  2.010847 3.902859 0.11643863
## 2          y4 -0.3690455 0.2760624 -1.336819 4.716160 0.24213491
## 3          x1  0.4301588 0.1670007  2.575791 4.856052 0.05107581
```

17.2.4.4.4 Stochastic Imputation

`Regression imputation + random residual = Stochastic Imputation`

Most multiple imputation is based off of some form of stochastic regression imputation.

Good:

- Has all the advantage of Regression Imputation
- and also has the random components

Bad:

- might lead to implausible values (e.g. negative values)
- can't handle heteroskadastic data

Note
Multiple Imputation usually based on some form of stochastic regression imputation.

```
# Income data

set.seed(91919)                              # Set seed
N <- 1000                                    # Sample size

income <- round(rnorm(N, 0, 500))            # Create some synthetic income data
income[income < 0] <- income[income < 0] * (- 1)

x1 <- income + rnorm(N, 1000, 1500)          # Auxiliary variables
x2 <- income + rnorm(N, - 5000, 2000)

income[rbinom(N, 1, 0.1) == 1] <- NA         # Create 10% missingness in income

data_inc_miss <- data.frame(income, x1, x2)
```

Single stochastic regression imputation

```
imp_inc_sri <- mice(data_inc_miss, method = "norm.nob", m = 1)
```

```
##
##  iter imp variable
##   1   1  income
##   2   1  income
##   3   1  income
##   4   1  income
##   5   1  income
```

```
data_inc_sri <- complete(imp_inc_sri)
```

Single predictive mean matching

```
imp_inc_pmm <- mice(data_inc_miss, method = "pmm", m = 1)
```

```
##
##  iter imp variable
##   1   1  income
##   2   1  income
##   3   1  income
##   4   1  income
##   5   1  income
```

```
data_inc_pmm <- complete(imp_inc_pmm)
```

Stochastic regression imputation contains negative values

```
data_inc_sri$income[data_inc_sri$income < 0]
```

```
## [1]  -66.055957  -96.980053  -28.921432   -4.175686  -54.480798  -27.207102
## [7] -143.603500  -80.960488
```

```
data_inc_pmm$income[data_inc_pmm$income < 0] # No values below 0
```

```
## numeric(0)
```

Proof for heteroskadastic data

```
# Heteroscedastic data

set.seed(654654)                              # Set seed
N <- 1:5000                                   # Sample size

a <- 0
b <- 1
sigma2 <- N^2
eps <- rnorm(N, mean = 0, sd = sqrt(sigma2))

y <- a + b * N + eps                          # Heteroscedastic variable
```

```
x <- 30 * N + rnorm(N[length(N)], 1000, 200) # Correlated variable

y[rbinom(N[length(N)], 1, 0.3) == 1] <- NA   # 30% missings

data_het_miss <- data.frame(y, x)
```

Single stochastic regression imputation

```
imp_het_sri <- mice(data_het_miss, method = "norm.nob", m = 1)
```

```
##
##  iter imp variable
##   1   1  y
##   2   1  y
##   3   1  y
##   4   1  y
##   5   1  y
```

```
data_het_sri <- complete(imp_het_sri)
```

Single predictive mean matching

```
imp_het_pmm <- mice(data_het_miss, method = "pmm", m = 1)
```

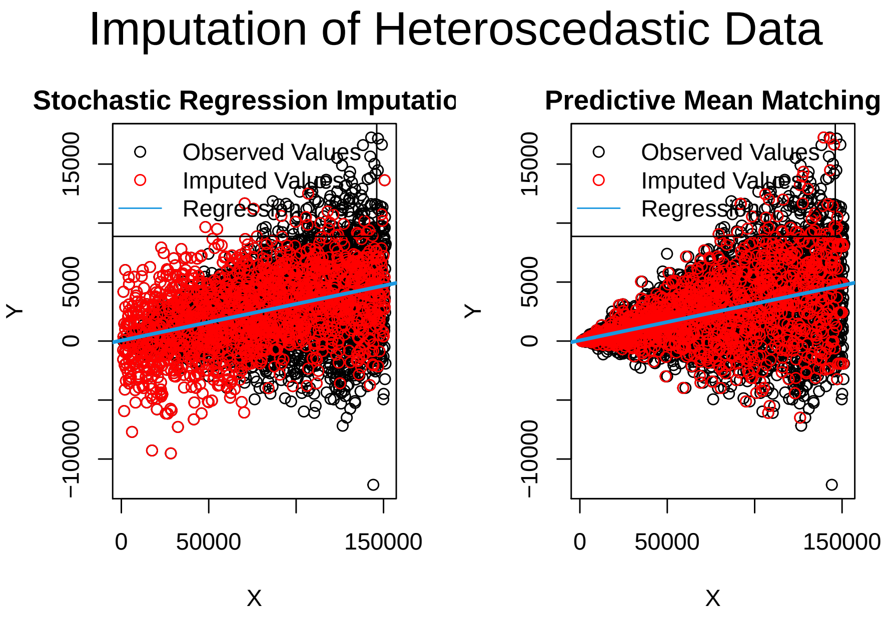
```
##
##  iter imp variable
##   1   1  y
##   2   1  y
##   3   1  y
##   4   1  y
##   5   1  y
```

```
data_het_pmm <- complete(imp_het_pmm)
```

Comparison between predictive mean matching and stochastic regression imputation

```
par(mfrow = c(1, 2))                              # Both plots in one graphic

plot(x[!is.na(data_het_sri$y)],                   # Plot of observed values
     data_het_sri$y[!is.na(data_het_sri$y)],
     main = "",
     xlab = "X", ylab = "Y")
points(x[is.na(y)], data_het_sri$y[is.na(y)],     # Plot of missing values
       col = "red")
title("Stochastic Regression Imputation",         # Title of plot
      line = 0.5)
abline(lm(y ~ x, data_het_sri),                   # Regression line
       col = "#1b98e0", lwd = 2.5)
```

```
legend("topleft",                                # Legend
       c("Observed Values", "Imputed Values", "Regression Y ~ X"),
       pch = c(1, 1, NA),
       lty = c(NA, NA, 1),
       col = c("black", "red", "#1b98e0"))

plot(x[!is.na(data_het_pmm$y)],                   # Plot of observed values
     data_het_pmm$y[!is.na(data_het_pmm$y)],
     main = "",
     xlab = "X", ylab = "Y")
points(x[is.na(y)], data_het_pmm$y[is.na(y)],     # Plot of missing values
       col = "red")
title("Predictive Mean Matching",                 # Title of plot
      line = 0.5)
abline(lm(y ~ x, data_het_pmm),
       col = "#1b98e0", lwd = 2.5)
legend("topleft",                                # Legend
       c("Observed Values", "Imputed Values", "Regression Y ~ X"),
       pch = c(1, 1, NA),
       lty = c(NA, NA, 1),
       col = c("black", "red", "#1b98e0"))

mtext("Imputation of Heteroscedastic Data",       # Main title of plot
      side = 3, line = - 1.5, outer = TRUE, cex = 2)
```



17.2.4.5 Regression Imputation

Also known as conditional mean imputation Missing value is based (regress) on other variables.

- Good:
  - Maintain the relationship with other variables
  - If the data are MCAR, least-squares coefficients estimates will be consistent, and approximately unbiased in large samples (Gourieroux and Monfort, 1981)
    - * Can have improvement on efficiency by using weighted least squares (Beale and Little, 1975) or generalized least squares (Gourieroux and Monfort, 1981).
- Bad:
  - No variability left. treated data as if they were collected.
  - Underestimate the standard errors and overestimate test statistics

17.2.4.6 Interpolation and Extrapolation

An estimated value from other observations from the same individual. It usually only works in longitudinal data.

17.2.4.7 K-nearest neighbor (KNN) imputation

The above methods are model-based imputation (regression).
This is an example of neighbor-based imputation (K-nearest neighbor).

For every observation that needs to be imputed, the algorithm identifies ‘k’ closest observations based on some types distance (e.g., Euclidean) and computes the weighted average (weighted based on distance) of these ‘k’ obs.

For a discrete variable, it uses the most frequent value among the k nearest neighbors.

- Distance metrics: Hamming distance.

For a continuous variable, it uses the mean or mode.

- Distance metrics:
  - Euclidean
  - Mahalanobis
  - Manhattan

17.2.4.8 Bayesian Ridge regression implementation

17.2.5 Other methods

- For panel data, or clustered data, use `pan` package by Schafer (1997)

17.3 Criteria for Choosing an Effective Approach

Criteria for an ideal technique in treating missing data:

1. Unbiased parameter estimates
2. Adequate power
3. Accurate standard errors (p-values, confidence intervals)

The Multiple Imputation and Full Information Maximum Likelihood are the the most ideal candidate. Single imputation will generally lead to underestimation of standard errors.

17.4 Another Perspective

Model bias can arisen from various factors including:

- Imputation method
- Missing data mechanism (MCAR vs. MAR)
- Proportion of the missing data
- Information available in the data set

Since the imputed observations are themselves estimates, their values have corresponding random error. But when you put in that estimate as a data point, your software doesn't know that. So it overlooks the extra source of error, resulting in too-small standard errors and too-small p-values. So multiple imputation comes up with multiple estimates.

Because multiple imputation have a random component, the multiple estimates are slightly different. This re-introduces some variation that your software can incorporate in order to give your model accurate estimates of standard error. Multiple imputation was a huge breakthrough in statistics about 20 years ago. It solves a lot of problems with missing data (though, unfortunately not all) and if done well, leads to unbiased parameter estimates and accurate standard errors. If your rate of missing data is very, very small (2-3%) it doesn't matter what technique you use.

Remember that there are three goals of multiple imputation, or any missing data technique:

- Unbiased parameter estimates in the final analysis (regression coefficients, group means, odds ratios, etc.)

- accurate standard errors of those parameter estimates, and therefore, accurate p-values in the analysis
- adequate power to find meaningful parameter values significant.

Hence,

1. Don't round off imputations for dummy variables. Many common imputation techniques, like MCMC, require normally distributed variables. Suggestions for imputing categorical variables were to dummy code them, impute them, then round off imputed values to 0 or 1. Recent research, however, has found that rounding off imputed values actually leads to biased parameter estimates in the analysis model. You actually get better results by leaving the imputed values at impossible values, even though it's counter-intuitive.

2. Don't transform skewed variables. Likewise, when you transform a variable to meet normality assumptions before imputing, you not only are changing the distribution of that variable but the relationship between that variable and the others you use to impute. Doing so can lead to imputing outliers, creating more bias than just imputing the skewed variable.

3. Use more imputations. The advice for years has been that 5-10 imputations are adequate. And while this is true for unbiasedness, you can get inconsistent results if you run the multiple imputation more than once. (Bodner, 2008) recommends having as many imputations as the percentage of missing data. Since running more imputations isn't any more work for the data analyst, there's no reason not to.

4. Create multiplicative terms before imputing. When the analysis model contains a multiplicative term, like an interaction term or a quadratic, create the multiplicative terms first, then impute. Imputing first, and then creating the multiplicative terms actually biases the regression parameters of the multiplicative term (von Hippel, 2009)

17.5 Diagnosing the Mechanism

17.5.1 MAR vs. MNAR

The only true way to distinguish between MNAR and MAR is to measure some of that missing data.

It's a common practice among professional surveyors to, for example, follow-up on a paper survey with phone calls to a group of the non-respondents and ask a few key survey items. This allows you to compare respondents to non-respondents.

If their responses on those key items differ by very much, that's good evidence that the data are MNAR.

However in most missing data situations, we can't get a hold of the missing data. So while we can't test it directly, we can examine patterns in the data get an idea of what's the most likely mechanism.

The first thing in diagnosing randomness of the missing data is to use your substantive scientific knowledge of the data and your field. The more sensitive the issue, the less likely people are to tell you. They're not going to tell you as much about their cocaine usage as they are about their phone usage.

Likewise, many fields have common research situations in which non-ignorable data is common. Educate yourself in your field's literature.

17.5.2 MCAR vs. MAR

There is a very useful test for MCAR, Little's test.

A second technique is to create dummy variables for whether a variable is missing.

1 = missing 0 = observed

You can then run t-tests and chi-square tests between this variable and other variables in the data set to see if the missingness on this variable is related to the values of other variables.

For example, if women really are less likely to tell you their weight than men, a chi-square test will tell you that the percentage of missing data on the weight variable is higher for women than men.

17.6 Application

How many imputation:

Usually 5. (unless you have extremely high portion of missing, in which case you probably need to check your data again)

According to Rubin, the relative efficiency of an estimate based on m imputations to infinity imputation is approximately

$$(1 + \frac{\lambda}{m})^{-1}$$

where λ is the rate of missing data

Example 50% of missing data means an estimate based on 5 imputation has standard deviation that is only 5% wider compared to an estimate based on infinity imputation
$(\sqrt{1 + 0.5/5} = 1.049)$

```
library(missForest)
```

```
## Loading required package: randomForest

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

## Loading required package: foreach

## Loading required package: itertools

## Loading required package: iterators

##
## Attaching package: 'missForest'

## The following object is masked from 'package:VIM':
##
##     nrmse
```

```
#load data
data <- iris

#Generate 10% missing values at Random
set.seed(1)
iris.mis <- prodNA(iris, noNA = 0.1)

#remove categorical variables
iris.mis.cat <- iris.mis
iris.mis <- subset(iris.mis, select = -c(Species))
```

17.6.1 Imputation with mean / median / mode

```
# whole data set
e1071::impute(iris.mis, what = "mean") # replace with mean
e1071::impute(iris.mis, what = "median") # replace with median

# by variables
Hmisc::impute(iris.mis$Sepal.Length, mean)  # mean
Hmisc::impute(iris.mis$Sepal.Length, median)  # median
Hmisc::impute(iris.mis$Sepal.Length, 0)  # replace specific number
```

check accurary

```
library(DMwR)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

##
## Attaching package: 'DMwR'

## The following object is masked from 'package:VIM':
##
##     kNN
```

```
actuals <- iris$Sepal.Width[is.na(iris.mis$Sepal.Width)]
predicteds <- rep(mean(iris$Sepal.Width, na.rm=T), length(actuals))
regr.eval(actuals, predicteds)
```

```
##       mae       mse      rmse      mape
## 0.2870303 0.1301598 0.3607767 0.1021485
```

17.6.2 KNN

```
library(DMwR)
# iris.mis[,!names(iris.mis) %in% c("Sepal.Length")]
# data should be this line. But since knn cant work with 3 or less variables, we need to use at l

# knn is not appropriate for categorical variables
knnOutput <-
    knnImputation(data = iris.mis.cat,
                  #k = 10,
                  meth = "median" # could use "median" or "weighAvg"
                  )  # should exclude the dependent variable: Sepal.Length
anyNA(knnOutput)
```

```
## [1] FALSE
```

```
library(DMwR)
actuals <- iris$Sepal.Width[is.na(iris.mis$Sepal.Width)]
predicteds <- knnOutput[is.na(iris.mis$Sepal.Width), "Sepal.Width"]
regr.eval(actuals, predicteds)
```

```
##       mae       mse      rmse      mape
## 0.2318182 0.1038636 0.3222788 0.0823571
```

Compared to mape (mean absolute percentage error) of mean imputation, we see almost always see improvements.

17.6.3 rpart

For categorical (factor) variables, rpart can handle

```
library(rpart)
class_mod <- rpart(Species ~ . - Sepal.Length, data=iris.mis.cat[!is.na(iris.mis.cat$Sp

anova_mod <- rpart(Sepal.Width ~ . - Sepal.Length, data=iris.mis[!is.na(iris.mis$Sepal
species_pred <- predict(class_mod, iris.mis.cat[is.na(iris.mis.cat$Species), ])
width_pred <- predict(anova_mod, iris.mis[is.na(iris.mis$Sepal.Width), ])
```

17.6.4 MICE (Multivariate Imputation via Chained Equations)

Assumption: data are MAR

It imputes data per variable by specifying an imputation model for each variable

Example

We have $X_1, X_2, .., X_k$. If X_1 has missing data, then it is regressed on the rest of the variables. Same procedure applies if X_2 has missing data. Then, predicted values are used in place of missing values.

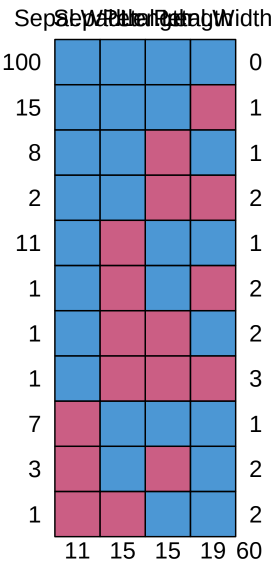
By default,

- **Continuous variables** use linear regression.
- **Categorical Variables** use logistic regression.

Methods in MICE:

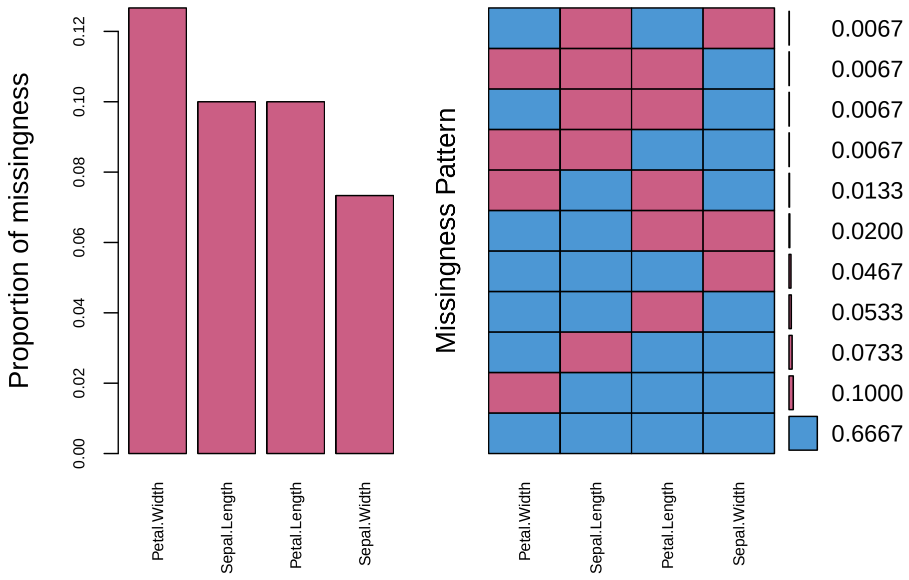
- PMM (Predictive Mean Matching) – For numeric variables
- logreg(Logistic Regression) – For Binary Variables(with 2 levels)
- polyreg(Bayesian polytomous regression) – For Factor Variables (>= 2 levels)
- Proportional odds model (ordered, >= 2 levels)

```
# load package
library(mice)
library(VIM)

# check missing values
md.pattern(iris.mis)
```



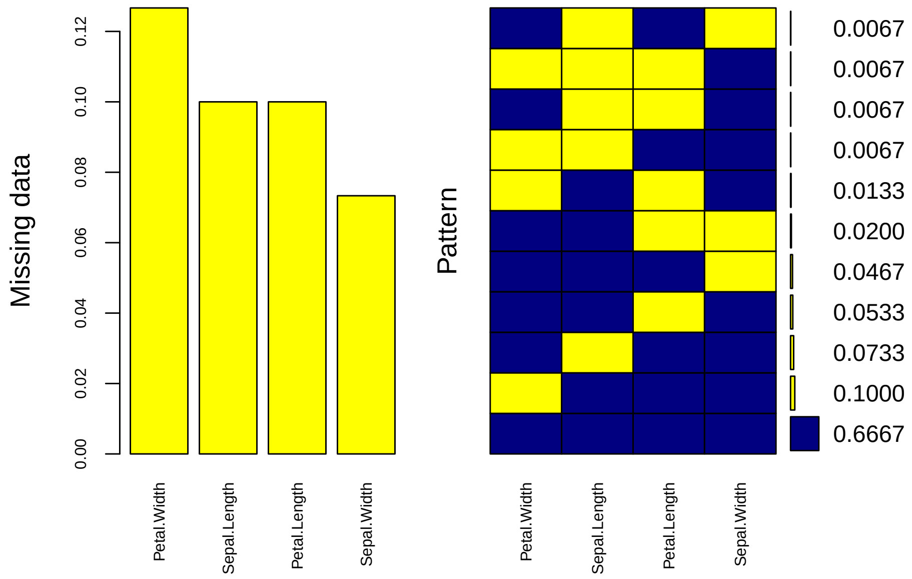
```
##     Sepal.Width Sepal.Length Petal.Length Petal.Width   
## 100           1            1            1           1  0
## 15            1            1            1           0  1
## 8             1            1            0           1  1
## 2             1            1            0           0  2
## 11            1            0            1           1  1
## 1             1            0            1           0  2
## 1             1            0            0           1  2
## 1             1            0            0           0  3
## 7             0            1            1           1  1
## 3             0            1            0           1  2
## 1             0            0            1           1  2
##              11           15           15          19 60
```

```
#plot the missing values
aggr(iris.mis, col=mdc(1:2), numbers=TRUE, sortVars=TRUE, labels=names(iris.mis), cex.axis=.7, ga
```



```
##
##  Variables sorted by number of missings:
##      Variable      Count
##   Petal.Width 0.12666667
##  Sepal.Length 0.10000000
##  Petal.Length 0.10000000
##   Sepal.Width 0.07333333
```

```
mice_plot <- aggr(iris.mis, col=c('navyblue','yellow'),
                    numbers=TRUE, sortVars=TRUE,
                    labels=names(iris.mis), cex.axis=.7,
                    gap=3, ylab=c("Missing data","Pattern"))
```



```
##
##  Variables sorted by number of missings:
##      Variable      Count
##   Petal.Width 0.12666667
##  Sepal.Length 0.10000000
##  Petal.Length 0.10000000
##   Sepal.Width 0.07333333
```

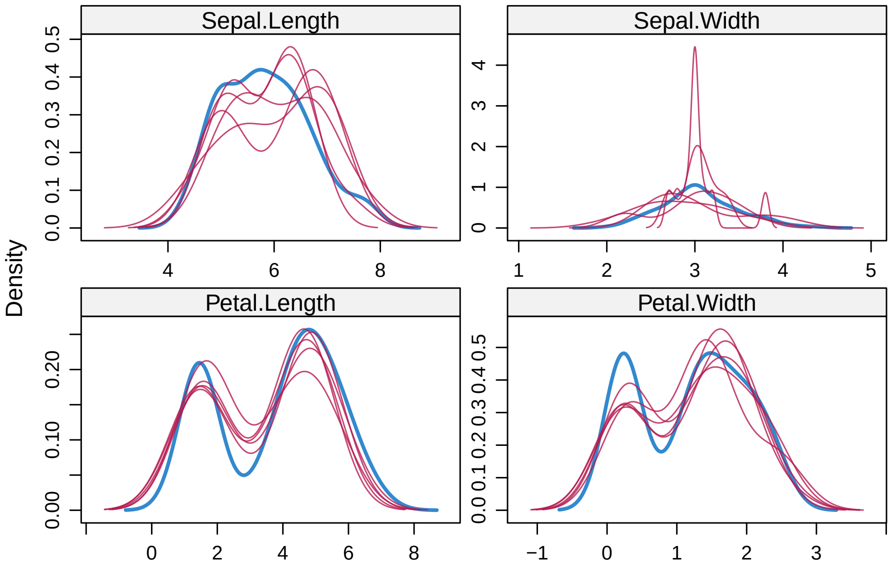
Impute Data

```
imputed_Data <-
    mice(
        iris.mis,
        m = 5, # number of imputed datasets
        maxit = 50, # number of iterations taken to impute missing values
        method = 'pmm', # method used in imputation. Here, we used predictive mean matching
        # other methods can be
        # "pmm": Predictive mean matching
        # "midastouch" : weighted predictive mean matching
        # "sample": Random sample from observed values
        # "cart": classification and regression trees
        # "rf": random forest imputations.
        # "2lonly.pmm": Level-2 class predictive mean matching
        # Other methods based on whether variables are (1) numeric, (2) binary, (3) ordered, (4),
        seed = 500
    )
```

```
summary(imputed_Data)
```

```
## Class: mids
## Number of multiple imputations:  5
## Imputation methods:
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##        "pmm"        "pmm"        "pmm"        "pmm"
## PredictorMatrix:
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length            0           1            1           1
## Sepal.Width             1           0            1           1
## Petal.Length            1           1            0           1
## Petal.Width             1           1            1           0
```

```
#make a density plot
densityplot(imputed_Data)
```



```
#the red (imputed values) should be similar to the blue (observed)
```

Check imputed dataset

```
# 1st dataset
completeData <- complete(imputed_Data,1)

# 2nd dataset
complete(imputed_Data,2)
```

Regression model using imputed datasets

```
# regression model
fit <- with(data = imputed_Data, exp = lm(Sepal.Width ~ Sepal.Length + Petal.Width))

#combine results of all 5 models
combine <- pool(fit)
summary(combine)
```

```
##           term   estimate  std.error statistic       df      p.value
## 1  (Intercept)  1.8963130 0.32453912  5.843095 131.0856 3.838556e-08
## 2 Sepal.Length  0.2974293 0.06679204  4.453066 130.2103 1.802241e-05
## 3  Petal.Width -0.4811603 0.07376809 -6.522608 108.8253 2.243032e-09
```

17.6.5 Amelia

- Use bootstrap based EMB algorithm (faster and robust to impute many variables including cross sectional, time series data etc)
- Use parallel imputation feature using multicore CPUs.

Assumptions

- All variables follow Multivariate Normal Distribution (MVN). Hence, this package works best when data is MVN, or transformation to normality.
- Missing data is Missing at Random (MAR)

Steps:

1. m bootstrap samples and applies EMB algorithm to each sample. Then we have m different estimates of mean and variances.
2. the first set of estimates are used to impute first set of missing values using regression, then second set of estimates are used for second set and so on.

However, Amelia is different from MICE

- MICE imputes data on variable by variable basis whereas MVN uses a joint modeling approach based on multivariate normal distribution.
- MICE can handle different types of variables while the variables in MVN need to be normally distributed or transformed to approximate normality.
- MICE can manage imputation of variables defined on a subset of data whereas MVN cannot.

```
library(Amelia)
```

```
## Loading required package: Rcpp
```

```
## ## 
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.6, built: 2019-11-24)
## ## Copyright (C) 2005-2021 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```
data("iris")
#seed 10% missing values
iris.mis <- prodNA(iris, noNA = 0.1)

# idvars - keep all ID variables and other variables which you don't want to impute
# noms - keep nominal variables here

#specify columns and run amelia
amelia_fit <- amelia(iris.mis, m=5, parallel = "multicore", noms = "Species")
```

```
## -- Imputation 1 --
##
##   1  2  3  4  5  6  7  8
##
## -- Imputation 2 --
##
##   1  2  3  4  5  6  7  8
##
## -- Imputation 3 --
##
##   1  2  3  4  5
##
## -- Imputation 4 --
##
##   1  2  3  4  5  6  7
##
## -- Imputation 5 --
##
##   1  2  3  4  5  6  7
```

```
# access imputed outputs
# amelia_fit$imputations[[1]]
```

17.6.6 missForest

- an implementation of random forest algorithm (a non parametric imputation method applicable to various variable types). Hence, no assumption about function form of f. Instead, it tries to estimate f such that it can be as close to the data points as possible.
- builds a random forest model for each variable. Then it uses the model to predict missing values in the variable with the help of observed values.
- It yields out of bag imputation error estimate. Moreover, it provides high level of control on imputation process.
- Since bagging works well on categorical variable too, we don't need to remove them here.

```
library(missForest)
#impute missing values, using all parameters as default values
iris.imp <- missForest(iris.mis)
```

```
##   missForest iteration 1 in progress...done!
##   missForest iteration 2 in progress...done!
##   missForest iteration 3 in progress...done!
##   missForest iteration 4 in progress...done!
```

```
# check imputed values
# iris.imp$ximp


# check imputation error
# NRMSE is normalized mean squared error. It is used to represent error derived from imputing con
# PFC (proportion of falsely classified) is used to represent error derived from imputing categor
iris.imp$OOBerror
```

```
##      NRMSE        PFC
## 0.13631893 0.04477612
```

```
#comparing actual data accuracy
iris.err <- mixError(iris.imp$ximp, iris.mis, iris)
iris.err
```

```
##     NRMSE       PFC
## 0.1501524 0.0625000
```

This means categorical variables are imputed with 5% error and continuous variables are imputed with 14% error.

This can be improved by tuning the values of `mtry` and `ntree` parameter.

- `mtry` refers to the number of variables being randomly sampled at each split.
- `ntree` refers to number of trees to grow in the forest.

17.6.7 Hmisc

- `impute()` function imputes missing value using user defined statistical method (mean, max, mean). It's default is median.

- `aregImpute()` allows mean imputation using additive regression, bootstrapping, and predictive mean matching.

1. In bootstrapping, different bootstrap resamples are used for each of multiple imputations. Then, a flexible additive model (non parametric regression method) is fitted on samples taken with replacements from original data and missing values (acts as dependent variable) are predicted

using non-missing values (independent variable).

2. it uses predictive mean matching (default) to impute missing values. Predictive mean matching works well for continuous and categorical (binary & multi-level) without the need for computing residuals and maximum likelihood fit.

Note

- For predicting categorical variables, Fisher's optimum scoring method is used.
- `Hmisc` automatically recognizes the variables types and uses bootstrap sample and predictive mean matching to impute missing values.
- `missForest` can outperform `Hmisc` if the observed variables have sufficient information.

Assumption

- linearity in the variables being predicted.

```
library(Hmisc)
```

```
## Loading required package: survival

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
# impute with mean value
iris.mis$imputed_age <- with(iris.mis, impute(Sepal.Length, mean))

# impute with random value
iris.mis$imputed_age2 <- with(iris.mis, impute(Sepal.Length, 'random'))

# could also use min, max, median to impute missing value

# using argImpute
impute_arg <- aregImpute(~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width +
Species, data = iris.mis, n.impute = 5) # argImpute() automatically identifies the var
```

```
## Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Iteration 6
Iteration 7
Iteration 8
```

```
impute_arg # R-squares are for predicted missing values.
```

```
##
## Multiple Imputation using Bootstrap and PMM
##
## aregImpute(formula = ~Sepal.Length + Sepal.Width + Petal.Length +
##     Petal.Width + Species, data = iris.mis, n.impute = 5)
##
## n: 150   p: 5    Imputations: 5      nk: 3
##
## Number of NAs:
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
##           11           11           13           24           16
##
##              type d.f.
## Sepal.Length    s    2
## Sepal.Width     s    2
## Petal.Length    s    2
## Petal.Width     s    2
## Species         c    2
##
## Transformation of Target Variables Forced to be Linear
##
## R-squares for Predicting Non-Missing Values for Each Variable
## Using Last Imputations of Predictors
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
##        0.907        0.660        0.978        0.963        0.993
```

```
# check imputed variable Sepal.Length
impute_arg$imputed$Sepal.Length
```

```
##     [,1] [,2] [,3] [,4] [,5]
## 19   5.2  5.2  5.2  5.8  5.7
## 21   5.1  5.0  5.1  5.7  5.4
## 31   4.8  5.0  5.2  5.0  4.8
## 35   4.6  4.9  4.9  4.9  4.8
## 49   5.0  5.1  5.1  5.1  5.1
## 62   6.2  5.7  6.0  6.4  5.6
## 65   5.5  5.5  5.2  5.8  5.5
## 67   6.5  5.8  5.8  6.3  6.5
## 82   5.2  5.1  5.7  5.8  5.5
## 113  6.4  6.5  7.4  7.2  6.3
## 122  6.2  5.8  5.5  5.8  6.7
```

17.6.8 mi

1. allows graphical diagnostics of imputation models and convergence of imputation process.
2. uses Bayesian version of regression models to handle issue of separation.
3. automatically detects irregularities in data (e.g., high collinearity among variables).
4. adds noise to imputation process to solve the problem of additive constraints.

```
library(mi)
# default values of parameters
# 1. rand.imp.method as "bootstrap"
# 2. n.imp (number of multiple imputations) as 3
# 3. n.iter ( number of iterations) as 30
mi_data <- mi(iris.mis, seed = 335)
summary(mi_data)
```

Chapter 18

Data

There are multiple ways to categorize data. For example,

- Qualitative vs. Quantitative:

Qualitative	Quantitative
in-depth interviews, documents, focus groups, case study, ethnography. open-ended questions. observations in words	experiments, observation in words, survey with closed-end questions, structured interviews
language, descriptive	quantities, numbers
Text-based	Numbers-based
Subjective	Objectivity

18.1 Cross-Sectional

18.2 Time Series

$$y_t = \beta_0 + x_{t1}\beta_1 + x_{t2}\beta_2 + ... + x_{t(k-1)}\beta_{k-1} + \epsilon_t$$

Examples

- Static Model
  - $y_t = \beta_0 + x_1\beta_1 + x_2\beta_2 - x_3\beta_3 - \epsilon_t$
- Finite Distributed Lag model
  - $y_t = \beta_0 + pe_t\delta_0 + pe_{t-1}\delta_1 + pe_{t-2}\delta_2 + \epsilon_t$
  - **Long Run Propensity (LRP)** is $LRP = \delta_0 + \delta_1 + \delta_2$

- Dynamic Model
  - $GDP_t = \beta_0 + \beta_1 GDP_{t-1} - \epsilon_t$

Finite Sample Properties for Time Series:

- A1-A3: OLS is unbiased
- A1-A4: usual standard errors are consistent and Gauss-Markov Theorem holds (OLS is BLUE)
- A1-A6, A6: Finite Sample Wald Test (t-test and F-test) are valid

A3 might not hold under time series setting

- Spurious Time Trend - solvable
- Strict vs Contemporaneous Exogeneity - not solvable

In time series data, there are many processes:

- Autoregressive model of order p: AR(p)
- Moving average model of order q: MA(q)
- Autoregressive model of order p and moving average model of order q: ARMA(p,q)
- Autoregressive conditional heteroskedasticity model of order p: ARCH(p)
- Generalized Autoregressive conditional heteroskedasticity of orders p and q; GARCH(p.q)

18.2.1 Deterministic Time trend

Both the dependent and independent variables are trending over time

Spurious Time Series Regression

$$y_t = \alpha_0 + t\alpha_1 + v_t$$

and x takes the form

$$x_t = \lambda_0 + t\lambda_1 + u_t$$

- $\alpha_1 \neq 0$ and $\lambda_1 \neq 0$
- v_t and u_t are independent
- there is no relationship between y_t and x_t

If we estimate the regression,

$$y_t = \beta_0 + x_t\beta_1 + \epsilon_t$$

so the true $\beta_1 = 0$

- Inconsistent: $plim(\hat{\beta}_1) = \frac{\alpha_1}{\lambda_1}$

- Invalid Inference: $|t| \to^d \infty$ for $H_0 : \beta_1 = 0$, will always reject the null as $n \to \infty$
- Uninformative R^2: $plim(R^2) = 1$ will be able to perfectly predict as $n \to \infty$

We can rewrite the equation as

$$y_t = \beta_0 + \beta_1 x_t + \epsilon_t \epsilon_t = \alpha_1 t + v_t$$

where $\beta_0 = \alpha_0$ and $\beta_1 = 0$. Since x_t is a deterministic function of time, ϵ_t is correlated with x_t and we have the usual omitted variable bias.
Even when y_t and x_t are related ($\beta_1 \neq 0$) but they are both trending over time, we still get spurious results with the simple regression on y_t on x_t

Solutions to Spurious Trend

1. Include time trend t as an additional control
   - consistent parameter estimates and valid inference
2. Detrend both dependent and independent variables and then regress the detrended outcome on detrended independent variables (i.e., regress residuals $\hat{u}_t$ on residuals $\hat{v}_t$)
   - Detrending is the same as partialling out in the Frisch-Waugh-Lovell Theorem
     - Could allow for non-linear time trends by including t t^2, and exp(t)
     - Allow for seasonality by including indicators for relevant "seasons" (quarters, months, weeks).

A3 does not hold under:

- Feedback Effect
  - ϵ_t influences next period's independent variables
- Dynamic Specification
  - include last time period outcome as an explanatory variable
- Dynamically Complete
  - For finite distrusted lag model, the number of lags needs to be absolutely correct.

18.2.2 Feedback Effect

$$y_t = \beta_0 + x_t \beta_1 + \epsilon_t$$

A3

$$E(\epsilon_t|\mathbf{X}) = E(\epsilon_t|x_1, x_2, ..., x_t, x_{t+1}, ..., x_T)$$

will not equal 0, because y_t will likely influence $x_{t+1}, .., x_T$

- A3 is violated because we require the error to be uncorrelated with all time observation of the independent regressors (**strict exogeneity**)

18.2.3 Dynamic Specification

$$y_t = \beta_0 + y_{t-1}\beta_1 + \epsilon_t$$

$$E(\epsilon_t|\mathbf{X}) = E(\epsilon_t|y_1, y_2, ..., y_t, y_{t+1}, ..., y_T)$$

will not equal 0, because y_t and ϵ_t are inherently correlated

- A3 is violated because we require the error to be uncorrelated with all time observation of the independent regressors (**strict exogeneity**)
- Dynamic Specification is not allowed under A3

18.2.4 Dynamically Complete

$$y_t = \beta_0 + x_t\delta_0 + x_{t-1}\delta_1 + \epsilon_t$$

$$E(\epsilon_t|\mathbf{X}) = E(\epsilon_t|x_1, x_2, ..., x_t, x_{t+1}, ..., x_T)$$

will not equal 0, because if we did not include enough lags, x_{t-2} and ϵ_t are correlated

- A3 is violated because we require the error to be uncorrelated with all time observation of the independent regressors (strict exogeneity)
- Can be corrected by including more lags (but when stop?)

Without A3

- OLS is biased
- Gauss-Markov Theorem
- Finite Sample Properties are invalid

then, we can

- Focus on Large Sample Properties
- Can use A3a instead of A3

A3a in time series become

$$A3a : E(\mathbf{x}_t'\epsilon_t) = 0$$

only the regressors in this time period need to be independent from the error in this time period (**Contemporaneous Exogeneity**)

- ϵ_t can be correlated with $..., x_{t-2}, x_{t-1}, x_{t+1}, x_{t+2}, ...$
- can have a dynamic specification $y_t = \beta_0 + y_{t-1}\beta_1 + \epsilon_t$

Deriving Large Sample Properties for Time Series

- Assumptions A1, A2, A3a
- Weak Law and Central Limit Theorem depend on A5
  - x_t and ϵ_t are dependent over t
  - without Weak Law or Central Limit Theorem depend on A5, we cannot have Large Sample Properties for OLS
  - Instead of A5, we consider A5a
- Derivation of the Asymptotic Variance depends on A4
  - time series setting introduces **Serial Correlation**: $Cov(\epsilon_t, \epsilon_s) \neq 0$

under A1, A2, A3a, and A5a, OLS estimator is **consistent**, and **asymptotically normal**

18.2.5 Highly Persistent Data

If $y_t, \mathbf{x}_t$ are not weakly dependent stationary process
* y_t and y_{t-h} are not almost independent for large h * A5a does not hold and OLS is not **consistent** and does not have a limiting distribution. * Example + Random Walk $y_t = y_{t-1} + u_t$ + Random Walk with a drift: $y_t = \alpha + y_{t-1} + u_t$

Solution First difference is a stationary process

$$y_t - y_{t-1} = u_t$$

- If u_t is a weakly dependent process (also called integrated of order 0) then y_t is said to be difference-stationary process (integrated of order 1)
- For regression, if $\{y_t, \mathbf{x}_t\}$ are random walks (integrated at order 1), can consistently estimate the first difference equation

$$\begin{aligned} y_t - y_{t-1} &= (\mathbf{x}_t - \mathbf{x}_{t-1}\beta + \epsilon_t - \epsilon_{t-1}) \\ \Delta y_t &= \Delta \mathbf{x}\beta + \Delta u_t \end{aligned}$$

Unit Root Test

$$y_t = \alpha + \alpha y_{t-1} + u_t$$

tests if $\rho = 1$ (integrated of order 1)

- Under the null $H_0 : \rho = 1$, OLS is not consistent or asymptotically normal.
- Under the alternative $H_a : \rho < 1$, OLS is consistent and asymptotically normal.
- usual t-test is not valid, will need to use the transformed equation to produce a valid test.

Dickey-Fuller Test

$$\Delta y_t = \alpha + \theta y_{t-1} + v_t$$

where $\theta = \rho - 1$

- $H_0 : \theta = 0$ and $H_a : \theta < 0$
- Under the null, Δy_t is weakly dependent but y_{t-1} is not.
- Dickey and Fuller derived the non-normal asymptotic distribution. If you reject the null then y_t is not a random walk.

Concerns with the standard Dickey Fuller Test
1. Only considers a fairly simplistic dynamic relationship

$$\Delta y_t = \alpha + \theta y_{t-1} + \gamma_1 \Delta_{t-1} + .. + \gamma_p \Delta_{t-p} + v_t$$

- with one additional lag, under the null Δ_{y_t} is an AR(1) process and under the alternative y_t is an AR(2) process.
- Solution: include lags of Δ_{y_t} as controls.

2. Does not allow for time trend

$$\Delta y_t = \alpha + \theta y_{t-1} + \delta t + v_t$$

- allows y_t to have a quadratic relationship with t
- Solution: include time trend (changes the critical values).

Adjusted Dickey-Fuller Test

$$\Delta y_t = \alpha + \theta y_{t-1} + \delta t + \gamma_1 \Delta y_{t-1} + ... + \gamma_p \Delta y_{t-p} + v_t$$

where $\theta = 1 - \rho$

- $H_0 : \theta_1 = 0$ and $H_a : \theta_1 < 0$
- Under the null, Δy_t is weakly dependent but y_{t-1} is not
- Critical values are different with the time trend, if you reject the null then y_t is not a random walk.

18.2.5.0.1 Newey West Standard Errors If A4 does not hold, we can use Newey West Standard Errors (HAC - Heteroskedasticity Autocorrelation Consistent)

$$\hat{B} = T^{-1}\sum_{t=1}^{T} e_t^2 \mathbf{x_t'x_t} + \sum_{h=1}^{g}(1-\frac{h}{g+1})T^{-1}\sum_{t=h+1}^{T} e_t e_{t-h}(\mathbf{x_t'x_{t-h}} + \mathbf{x_{t-h}'x_t})$$

- estimates the covariances up to a distance g part
- downweights to insure $\hat{B}$ is PSD
- How to choose g:
  - For yearly data: g = 1 or 2 is likely to account for most of the correlation
  - For quarterly or monthly data: g should be larger (g = 4 or 8 for quarterly and g = 12 or 14 for monthly)
  - can also take integer part of $4(T/100)^{2/9}$ or integer part of $T^{1/4}$

Testing for Serial Correlation

1. Run OLS regression of y_t on $\mathbf{x_t}$ and obtain residuals e_t
2. Run OLS regression of e_t on $\mathbf{x}_t, e_{t-1}$ and test whether coefficient on e_{t-1} is significant.
3. Reject the null of no serial correlation if the coefficient is significant at the 5% level.
   - Test using heteroskedastic robust standard errors
   - can include $e_{t-2}, e_{t-3}, ..$ in step 2 to test for higher order serial correlation (t-test would now be an F-test of joint significance)

18.3 Repeated Cross Sections

For each time point (day, month, year, etc.), a set of data is sampled. This set of data can be different among different time points.

For example, you can sample different groups of students each time you survey.

Allowing structural change in pooled cross section

$$y_i = \mathbf{x}_i\beta + \delta_1 y_1 + ... + \delta_T y_T + \epsilon_i$$

Dummy variables for all but one time period

- allows different intercept for each time period
- allows outcome to change on average for each time period

Allowing for structural change in pooled cross section

$$y_i = \mathbf{x}_i\beta + \mathbf{x}_i y_1\gamma_1 + ... + \mathbf{x}_i y_T\gamma_T + \delta_1 y_1 + ... + \delta_T y_T + \epsilon_i$$

Interact x_i with time period dummy variables

- allows different slopes for each time period
- allows effects to change based on time period (**structural break**)
- Interacting all time period dummies with x_i can produce many variables - use hypothesis testing to determine which structural breaks are needed.

18.3.1 Pooled Cross Section

$$y_i = \mathbf{x_i} \ + \mathbf{x_i} \times \mathbf{y1} \ _{\mathbf{1}} + ... + \mathbf{x_i} \times \mathbf{yT} \ _{\mathbf{T}} + \ _{\mathbf{1}}\mathbf{y_1} + ... + \ _{\mathbf{T}}\mathbf{y_T} + \ _{\mathbf{i}}$$

Interact x_i with time period dummy variables

- allows different slopes for each time period
- allows effect to change based on time period (structural break)
  - interacting all time period dummies with x_i can produce many variables - use hypothesis testing to determine which structural breaks are needed.

18.4 Panel Data

Detail notes in R can be found here

Follows an individual over T time periods.

Panel data structure is like having n samples of time series data

Characteristics

- Information both across individuals and over time (cross-sectional and time-series)
- N individuals and T time periods
- Data can be either
  - Balanced: all individuals are observed in all time periods
  - Unbalanced: all individuals are not observed in all time periods.
- Assume correlation (clustering) over time for a given individual, with independence over individuals.

Types

- Short panel: many individuals and few time periods.
- Long panel: many time periods and few individuals
- Both: many time periods and many individuals

Time Trends and Time Effects

- Nonlinear

- Seasonality
- Discontinuous shocks

Regressors

- Time-invariant regressors $x_{it} = x_i$ for all t (e.g., gender, race, education) have zero within variation
- Individual-invariant regressors $x_{it} = x_t$ for all i (e.g., time trend, economy trends) have zero between variation

Variation for the dependent variable and regressors

- Overall variation: variation over time and individuals.
- Between variation: variation between individuals
- Within variation: variation within individuals (over time).

Estimate	Formula
Individual mean	$\bar{x_i} = \frac{1}{T}\sum_t x_{it}$
Overall mean	$\bar{x} = \frac{1}{NT}\sum_i \sum_t x_{it}$
Overall Variance	$s^2_O = \frac{1}{NT-1}\sum_i \sum_t (x_{it} - \bar{x})^2$
Between variance	$s^2_B = \frac{1}{N-1}\sum_i (\bar{x_i} - \bar{x})^2$
Within variance	$s^2_W = \frac{1}{NT-1}\sum_i \sum_t (x_{it} - \bar{x_i})^2 = \frac{1}{NT-1}\sum_i \sum_t (x_{it} - \bar{x_i} + \bar{x})^2$

Note: $s^2_O \approx s^2_B + s^2_W$

Since we have n observation for each time period t, we can control for each time effect separately by including time dummies (time effects)

$$y_{it} = \mathbf{x_{it}}\ + d_1\delta_1 + ... + d_{T-1}\delta_{T-1} + \epsilon_{it}$$

Note: we cannot use these many time dummies in time series data because in time series data, our n is 1. Hence, there is no variation, and sometimes not enough data compared to variables to estimate coefficients.

Unobserved Effects Model Similar to group clustering, assume that there is a random effect that captures differences across individuals but is constant in time.

$$y_it = \mathbf{x_{it}}\ + d_1\delta_1 + ... + d_{T-1}\delta_{T-1} + c_i + u_{it}$$

where

- $c_i + u_{it} = \epsilon_{it}$
- c_i unobserved individual heterogeneity (effect)
- u_{it} idiosyncratic shock
- ϵ_{it} unobserved error term.

18.4.1 Pooled OLS Estimator

If c_i is uncorrelated with x_{it}

$$E(\mathbf{x'_{it}}(c_i + u_{it})) = 0$$

then A3a still holds. And we have Pooled OLS consistent.

If A4 does not hold, OLS is still consistent, but not efficient, and we need cluster robust SE.

Sufficient for A3a to hold, we need

- **Exogeneity** for u_{it} A3a (contemporaneous exogeneity): $E(\mathbf{x'_{it}}u_{it}) = 0$ time varying error
- **Random Effect Assumption** (time constant error): $E(\mathbf{x'_{it}}c_i) = 0$

Pooled OLS will give you consistent coefficient estimates under A1, A2, A3a (for both u_{it} and RE assumption), and A5 (randomly sampling across i).

18.4.2 Individual-specific effects model

- If we believe that there is unobserved heterogeneity across individual (e.g., unobserved ability of an individual affects y), If the individual-specific effects are correlated with the regressors, then we have the Fixed Effects Estimator. and if they are not correlated we have the Random Effects Estimator.

18.4.2.1 Random Effects Estimator

Random Effects estimator is the Feasible GLS estimator that assumes u_{it} is serially uncorrelated and homoskedastic

- Under A1, A2, A3a (for both u_{it} and RE assumption) and A5 (randomly sampling across i), RE estimator is consistent.
  - If A4 holds for u_{it}, RE is the most efficient estimator
  - If A4 fails to hold (may be heteroskedasticity across i, and serial correlation over t), then RE is not the most efficient, but still more efficient than pooled OLS.

18.4.2.2 Fixed Effects Estimator

also known as **Within Estimator** uses within variation (over time)

If the **RE assumption** is not hold ($E(\mathbf{x'_{it}} c_i) \neq 0$), then A3a does not hold ($E(\mathbf{x'_{it}} \epsilon_i) \neq 0$). Hence, the OLS and RE are inconsistent/biased (because of omitted variable bias)

To deal with violation in c_i, we have

$$y_{it} = \mathbf{x_{it}} \ + c_i + u_{it}$$

$$\bar{y_i} = \bar{\mathbf{x_i}} \beta + c_i + \bar{u_i}$$

where the second equation is the time averaged equation

using **within transformation**, we have

$$y_{it} - \bar{y_i} = (\mathbf{x_{it}} - \bar{\mathbf{x_i}} \ ) + u_{it} - \bar{u_i}$$

because c_i is time constant.

The Fixed Effects estimator uses POLS on the transformed equation

$$y_{it} - \bar{y_i} = (\mathbf{x_{it}} - \bar{\mathbf{x_i}} \ ) + d_1\delta_1 + ... + d_{T-2}\delta_{T-2} + u_{it} - \bar{u_i}$$

- we need A3 (strict exogeneity) ($E((\mathbf{x_{it}} - \bar{\mathbf{x_i}})'(u_{it} - \bar{u_i}) = 0$) to have FE consistent.
- Variables that are time constant will be absorbed into c_i. Hence we cannot make inference on time constant independent variables.
  - If you are interested in the effects of time-invariant variables, you could consider the OLS or **between estimator**
- It's recommended that you should still use cluster robust standard errors.

Equivalent to the within transformation, we can have the fixed effect estimator be the same with the dummy regression

$$y_{it} = x_{it}\beta + d_1\delta_1 + ... + d_{T-2}\delta_{T-2} + c_1\gamma_1 + ... + c_{n-1}\gamma_{n-1} + u_{it}$$

where

$$c_i = \begin{cases} 1 & \text{if observation is i} \\ 0 & \text{otherwise} \end{cases} \tag{18.1}$$

- The standard error is incorrectly calculated.

- the FE within transformation is controlling for any difference across individual which is allowed to correlated with observables.

18.4.3 Tests for Assumptions

We typically don't test heteroskedasticity because we will use robust covariance matrix estimation anyway.

Dataset

```
library("plm")
data("EmplUK", package="plm")
data("Produc", package="plm")
data("Grunfeld", package="plm")
data("Wages", package="plm")
```

18.4.3.1 Poolability

also known as an F test of stability (or Chow test) for the coefficients

H_0: All individuals have the same coefficients (i.e., equal coefficients for all individuals).

H_a Different individuals have different coefficients.

Notes:

- Under a within (i.e., fixed) model, different intercepts for each individual are assumed
- Under random model, same intercept is assumed

```
library(plm)
plm::pooltest(inv~value+capital, data=Grunfeld, model="within")
```

```
##
##  F statistic
##
## data:  inv ~ value + capital
## F = 5.7805, df1 = 18, df2 = 170, p-value = 1.219e-10
## alternative hypothesis: unstability
```

Hence, we reject the null hypothesis that coefficients are stable. Then, we should use the random model.

18.4.3.2 Individual and time effects

use the Lagrange multiplier test to test the presence of individual or time or both (i.e., individual and time).

Types:

- `honda`: (Honda, 1985) Default
- `bp`: (Breusch and Pagan, 1980) for unbalanced panels
- `kw`: (King and Wu, 1997) unbalanced panels, and two-way effects
- `ghm`: (Gourieroux et al., 1982): two-way effects

```
pFtest(inv~value+capital, data=Grunfeld, effect="twoways")
```

```
##
##  F test for twoways effects
##
## data:  inv ~ value + capital
## F = 17.403, df1 = 28, df2 = 169, p-value < 2.2e-16
## alternative hypothesis: significant effects
```

```
pFtest(inv~value+capital, data=Grunfeld, effect="individual")
```

```
##
##  F test for individual effects
##
## data:  inv ~ value + capital
## F = 49.177, df1 = 9, df2 = 188, p-value < 2.2e-16
## alternative hypothesis: significant effects
```

```
pFtest(inv~value+capital, data=Grunfeld, effect="time")
```

```
##
##  F test for time effects
##
## data:  inv ~ value + capital
## F = 0.23451, df1 = 19, df2 = 178, p-value = 0.9997
## alternative hypothesis: significant effects
```

18.4.3.3 Cross-sectional dependence/contemporaneous correlation

- Null hypothesis: residuals across entities are not correlated.

```

pcdtest(inv~value+capital, data=Grunfeld, model="within")
```

18.4.3.3.1 Global cross-sectional dependence

```
##
##  Pesaran CD test for cross-sectional dependence in panels
##
## data:  inv ~ value + capital
## z = 4.6612, p-value = 3.144e-06
## alternative hypothesis: cross-sectional dependence
```

18.4.3.3.2 Local cross-sectional dependence use the same command, but supply matrix `w` to the argument.

```
pcdtest(inv~value+capital, data=Grunfeld, model="within")
```

```
##
##  Pesaran CD test for cross-sectional dependence in panels
##
## data:  inv ~ value + capital
## z = 4.6612, p-value = 3.144e-06
## alternative hypothesis: cross-sectional dependence
```

18.4.3.4 Serial Correlation

- Null hypothesis: there is no serial correlation
- usually seen in macro panels with long time series (large N and T), not seen in micro panels (small T and large N)
- Serial correlation can arise from individual effects(i.e., time-invariant error component), or idiosyncratic error terms (e..g, in the case of AR(1) process). But typically, when we refer to serial correlation, we refer to the second one.
- Can be
  - **marginal** test: only 1 of the two above dependence (but can be biased towards rejection)
  - **joint** test: both dependencies (but don't know which one is causing the problem)
  - **conditional** test: assume you correctly specify one dependence structure, test whether the other departure is present.

18.4.3.4.1 Unobserved effect test

- semi-parametric test (the test statistic $W\dot{\sim}N$ regardless of the distribution of the errors) with $H_0 : \sigma^2_\mu = 0$ (i.e., no unobserved effects in the residuals), favors pooled OLS.
  - Under the null, covariance matrix of the residuals = its diagonal (off-diagonal = 0)
- It is robust against both **unobserved effects** that are constant within every group, and any kind of **serial correlation**.

```
pwtest(log(gsp)~log(pcap)+log(pc)+log(emp)+unemp, data=Produc)
```

```
##
##  Wooldridge's test for unobserved individual effects
##
```

```
## data:  formula
## z = 3.9383, p-value = 8.207e-05
## alternative hypothesis: unobserved effect
```

Here, we reject the null hypothesis that the no unobserved effects in the residuals. Hence, we will exclude using pooled OLS.

18.4.3.4.2 Locally robust tests for random effects and serial correlation

- A joint LM test for **random effects** and **serial correlation** assuming normality and homoskedasticity of the idiosyncratic errors (Baltagi and Li, 1991)(Baltagi and Li, 1995)

```r
pbsytest(log(gsp)~log(pcap)+log(pc)+log(emp)+unemp, data=Produc, test="j")
```

```
##
##  Baltagi and Li AR-RE joint test - balanced panel
##
## data:  formula
## chisq = 4187.6, df = 2, p-value < 2.2e-16
## alternative hypothesis: AR(1) errors or random effects
```

Here, we reject the null hypothesis that there is no presence of **serial correlation,** and **random effects**. But we still do not know whether it is because of serial correlation, of random effects or of both

To know the departure from the null assumption, we can use (Bera et al., 2001)'s test for first-order serial correlation or random effects (both under normality and homoskedasticity assumption of the error).

BSY for serial correlation

```r
pbsytest(log(gsp)~log(pcap)+log(pc)+log(emp)+unemp, data=Produc)
```

```
##
##  Bera, Sosa-Escudero and Yoon locally robust test - balanced panel
##
## data:  formula
## chisq = 52.636, df = 1, p-value = 4.015e-13
## alternative hypothesis: AR(1) errors sub random effects
```

BSY for random effects

```r
pbsytest(log(gsp)~log(pcap)+log(pc)+log(emp)+unemp, data=Produc, test="re")
```

```
##
##  Bera, Sosa-Escudero and Yoon locally robust test (one-sided) -
##  balanced panel
##
## data:  formula
## z = 57.914, p-value < 2.2e-16
## alternative hypothesis: random effects sub AR(1) errors
```

Since BSY is only locally robust, if you "know" there is no serial correlation, then this test is based on LM test is more superior:

```
plmtest(inv ~ value + capital, data = Grunfeld, type = "honda")
```

```
##
##  Lagrange Multiplier Test - (Honda) for balanced panels
##
## data:  inv ~ value + capital
## normal = 28.252, p-value < 2.2e-16
## alternative hypothesis: significant effects
```

On the other hand, if you know there is no random effects, to test for serial correlation, use (BREUSCH, 1978)-(Godfrey, 1978)'s test

```
lmtest::bgtest()
```

If you "know" there are random effects, use (Baltagi and Li, 1995)'s. to test for serial correlation in both AR(1) and MA(1) processes.

H_0: Uncorrelated errors.

Note:

- one-sided only has power against positive serial correlation.
- applicable to only balanced panels.

```
pbltest(log(gsp)~log(pcap)+log(pc)+log(emp)+unemp,
        data=Produc, alternative="onesided")
```

```
##
##  Baltagi and Li one-sided LM test
##
## data:  log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp
## z = 21.69, p-value < 2.2e-16
## alternative hypothesis: AR(1)/MA(1) errors in RE panel model
```

General serial correlation tests

- applicable to random effects model, OLS, and FE (with large T, also known as long panel).
- can also test higher-order serial correlation

```
plm::pbgtest(plm::plm(inv~value+capital, data = Grunfeld, model = "within"), order = 2
```

```
##
##  Breusch-Godfrey/Wooldridge test for serial correlation in panel models
##
## data:  inv ~ value + capital
## chisq = 42.587, df = 2, p-value = 5.655e-10
## alternative hypothesis: serial correlation in idiosyncratic errors
```

in the case of short panels (small T and large n), we can use

```
pwartest(log(emp) ~ log(wage) + log(capital), data=EmplUK)
```

```
##
##  Wooldridge's test for serial correlation in FE panels
##
## data:  plm.model
## F = 312.3, df1 = 1, df2 = 889, p-value < 2.2e-16
## alternative hypothesis: serial correlation
```

18.4.3.5 Unit roots/stationarity

- Dickey-Fuller test for stochastic trends.
- Null hypothesis: the series is non-stationary (unit root)
- You would want your test to be less than the critical value (p<.5) so that there is evidence there is not unit roots.

18.4.3.6 Heteroskedasticity

- Breusch-Pagan test
- Null hypothesis: the data is homoskedastic
- If there is evidence for heteroskedasticity, robust covariance matrix is advised.
- To control for heteroskedasticity: Robust covariance matrix estimation (Sandwich estimator)
  - "white1" - for general heteroskedasticity but no serial correlation (check serial correlation first). Recommended for random effects.
  - "white2" - is "white1" restricted to a common variance within groups. Recommended for random effects.
  - "arellano" - both heteroskedasticity and serial correlation. Recommended for fixed effects

18.4.4 Model Selection

18.4.4.1 POLS vs. RE

The continuum between RE (used FGLS which more assumption) and POLS check back on the section of FGLS

Breusch-Pagan LM test

- Test for the random effect model based on the OLS residual

- Null hypothesis: variances across entities is zero. In another word, no panel effect.
- If the test is significant, RE is preferable compared to POLS

18.4.4.2 FE vs. RE

- RE does not require strict exogeneity for consistency (feedback effect between residual and covariates)

Hypothesis	If true
$H_0 : Cov(c_i, \mathbf{x_{it}}) = 0$	$\hat{\beta}_{RE}$ is consistent and efficient, while $\hat{\beta}_{FE}$ is consistent
$H_0 : Cov(c_i, \mathbf{x_{it}}) \neq 0$	$\hat{\beta}_{RE}$ is inconsistent, while $\hat{\beta}_{FE}$ is consistent

Hausman Test

For the Hausman test to run, you need to assume that

- strict exogeneity hold
- A4 to hold for u_{it}

Then,

- Hausman test statistic: $H = (\hat{\beta}_{RE} - \hat{\beta}_{FE})'(V(\hat{\beta}_{RE}) - V(\hat{\beta}_{FE}))(\hat{\beta}_{RE} - \hat{\beta}_{FE}) \sim \chi^2_{n(X)}$ where $n(X)$ is the number of parameters for the time-varying regressors.
- A low p-value means that we would reject the null hypothesis and prefer FE
- A high p-value means that we would not reject the null hypothesis and consider RE estimator.

```
gw <- plm(inv~value+capital, data=Grunfeld, model="within")
gr <- plm(inv~value+capital, data=Grunfeld, model="random")
phtest(gw, gr)
```

```
##
##  Hausman Test
##
## data:  inv ~ value + capital
## chisq = 2.3304, df = 2, p-value = 0.3119
## alternative hypothesis: one model is inconsistent
```

Violation Estimator	Basic Estimator	Instrumental variable Estimator	Variable Coefficients estimator	Generalized Method of Moments estimator	General FGLS estimator	Means groups estimator	CCEMG	Estimator for limited dependent variables

18.4.5 Summary

- All three estimators (POLS, RE, FE) require A1, A2, A5 (for individuals) to be consistent. Additionally,
- POLS is consistent under A3a(for u_{it}): $E(\mathbf{x}'_{it}u_{it}) = 0$, and RE Assumption $E(\mathbf{x}'_{it}c_i) = 0$
  - If A4 does not hold, use cluster robust SE but POLS is not efficient
- RE is consistent under A3a(for u_{it}): $E(\mathbf{x}'_{it}u_{it}) = 0$, and RE Assumption $E(\mathbf{x}'_{it}c_i) = 0$
  - If A4 (for u_{it}) holds then usual SE are valid and RE is most efficient
  - If A4 (for u_{it}) does not hold, use cluster robust SE ,and RE is no longer most efficient (but still more efficient than POLS)
- FE is consistent under A3 $E((\mathbf{x}_{it} - \bar{\mathbf{x}}_{it})'(u_{it} - \bar{u}_{it})) = 0$
  - Cannot estimate effects of time constant variables
  - A4 generally does not hold for $u_{it} - \bar{u}_{it}$ so cluster robust SE are needed

Note: A5 for individual (not for time dimension) implies that you have A5a for the entire data set.

Estimator / True Model	POLS	RE	FE
POLS	Consistent	Consistent	Inconsistent
FE	Consistent	Consistent	Consistent
RE	Consistent	Consistent	Inconsistent

Based on table provided by Ani Katchova

18.4.6 Application

Recommended application of `plm` can be found here and here by Yves Croissant

```
#install.packages("plm")
library("plm")

library(foreign)
Panel <- read.dta("http://dss.princeton.edu/training/Panel101.dta")

attach(Panel)
Y <- cbind(y)
X <- cbind(x1, x2, x3)

# Set data as panel data
pdata <- pdata.frame(Panel, index=c("country","year"))
```

```
# Pooled OLS estimator
pooling <- plm(Y ~ X, data=pdata, model= "pooling")
summary(pooling)

# Between estimator
between <- plm(Y ~ X, data=pdata, model= "between")
summary(between)

# First differences estimator
firstdiff <- plm(Y ~ X, data=pdata, model= "fd")
summary(firstdiff)

# Fixed effects or within estimator
fixed <- plm(Y ~ X, data=pdata, model= "within")
summary(fixed)

# Random effects estimator
random <- plm(Y ~ X, data=pdata, model= "random")
summary(random)

# LM test for random effects versus OLS
# Accept Null, then OLS, Reject Null then RE
plmtest(pooling,effect = "individual", type = c("bp")) # other type: "honda", "kw"," "


# B-P/LM and Pesaran CD (cross-sectional dependence) test
pcdtest(fixed, test = c("lm")) # Breusch and Pagan's original LM statistic
pcdtest(fixed, test = c("cd")) # Pesaran's CD statistic

# Serial Correlation
pbgtest(fixed)

# stationary
library("tseries")
adf.test(pdata$y, k = 2)

# LM test for fixed effects versus OLS
pFtest(fixed, pooling)

# Hausman test for fixed versus random effects model
phtest(random, fixed)

# Breusch-Pagan heteroskedasticity
library(lmtest)
bptest(y ~ x1 + factor(country), data = pdata)
```

```
# If there is presence of heteroskedasticity
## For RE model
coeftest(random) #orginal coef
coeftest(random, vcovHC) # Heteroskedasticity consistent coefficients

t(sapply(c("HC0", "HC1", "HC2", "HC3", "HC4"), function(x) sqrt(diag(vcovHC(random, type = x)))))
# HC0 - heteroskedasticity consistent. The default.
# HC1,HC2, HC3 - Recommended for small samples. HC3 gives less weight to influential observations
# HC4 - small samples with influential observations
# HAC - heteroskedasticity and autocorrelation consistent

## For FE model
coeftest(fixed) # Original coefficients
coeftest(fixed, vcovHC) # Heteroskedasticity consistent coefficients
coeftest(fixed, vcovHC(fixed, method = "arellano")) # Heteroskedasticity consistent coefficients
t(sapply(c("HC0", "HC1", "HC2", "HC3", "HC4"), function(x) sqrt(diag(vcovHC(fixed, type = x)))))
```

Advanced

Other methods to estimate the random model:

- `"swar"`: *default* (Swamy and Arora, 1972)
- `"walhus"`: (Wallace and Hussain, 1969)
- `"amemiya"`: (Fuller and Battese, 1974)
- `"nerlove"`" (Nerlove, 1971)

Other effects:

- Individual effects: *default*
- Time effects: `"time"`
- Individual and time effects: `"twoways"`

Note: no random two-ways effect model for `random.method = "nerlove"`

```
amemiya <- plm(Y ~ X, data=pdata, model= "random",random.method = "amemiya",effect = "twoways")
```

To call the estimation of the variance of the error components

```
ercomp(Y~X, data=pdata, method = "amemiya", effect = "twoways")
```

Check for the unbalancedness. Closer to 1 indicates balanced data (Ahrens and Pincus, 1981)

```
punbalancedness(random)
```

Instrumental variable

- `"bvk"`: default (Balestra and Varadharajan-Krishnakumar, 1987)
- `"baltagi"`: (Baltagi, 1981)
- `"am"` (Amemiya and MaCurdy, 1986)

- "bms": (Breusch et al., 1989)

```
instr <- plm(Y ~ X | X_ins, data = pdata, random.method = "ht", model = "random", inst
```

18.4.7 Other Estimators

18.4.7.1 Variable Coefficients Model

```
fixed_pvcm <- pvcm(Y~X, data=pdata, model="within")
random_pvcm <- pvcm(Y~X, data=pdata, model="random")
```

More details can be found here

18.4.7.2 Generalized Method of Moments Estimator

Typically use in dynamic models. Example is from plm package

```
z2 <- pgmm(log(emp) ~ lag(log(emp), 1)+ lag(log(wage), 0:1) +
           lag(log(capital), 0:1) | lag(log(emp), 2:99) +
           lag(log(wage), 2:99) + lag(log(capital), 2:99),
           data = EmplUK, effect = "twoways", model = "onestep",
           transformation = "ld")
summary(z2, robust = TRUE)
```

18.4.7.3 General Feasible Generalized Least Squares Models

Assume there is no cross-sectional correlation Robust against intragroup heteroskedasticity and serial correlation. Suited when n is much larger than T (long panel) However, inefficient under groupwise heteorskedasticity.

```
# Random Effects
zz <- pggls(log(emp)~log(wage)+log(capital), data=EmplUK, model="pooling")

# Fixed
zz <- pggls(log(emp)~log(wage)+log(capital), data=EmplUK, model="within")
```

Chapter 19

Hypothesis Testing

Note:

- Always written in terms of the population parameter (β) not the estimator/estimate ($\hat{\beta}$)
- Sometimes, different disciplines prefer to use β (i.e., standardized coefficient), or $\mathbf{b}$ (i.e., unstandardized coefficient)
  - β and $\mathbf{b}$ are similar in interpretation; however, β is scale free. Hence, you can see the relative contribution of β to the dependent variable. On the other hand, $\mathbf{b}$ can be more easily used in policy decisions.
  - $$\beta_j = \mathbf{b}\frac{s_{x_j}}{s_y}$$
- Assuming the null hypothesis is true, what is the (asymptotic) distribution of the estimator
- Two-sided

$$H_0 : \beta_j = 0 H_1 : \beta_j \neq 0$$

then under the null, the OLS estimator has the following distribution

$$A1 - A3a, A5 : \sqrt{n}\hat{\beta}_j \sim N(0, Avar(\sqrt{n}\hat{\beta}_j))$$

- For the one-sided test, the null is a set of values, so now you choose the worst case single value that is hardest to prove and derive the distribution under the null
- One-sided

$$H_0 : \beta_j \geq 0 H_1 : \beta_j < 0$$

then the hardest null value to prove is $H_0 : \beta_j = 0$. Then under this specific null, the OLS estimator has the following asymptotic distribution

$$A1 - A3a, A5 : \sqrt{n}\hat{\beta}_j \sim N(0, Avar(\sqrt{n}\hat{\beta}_j))$$

19.1 Types of hypothesis testing

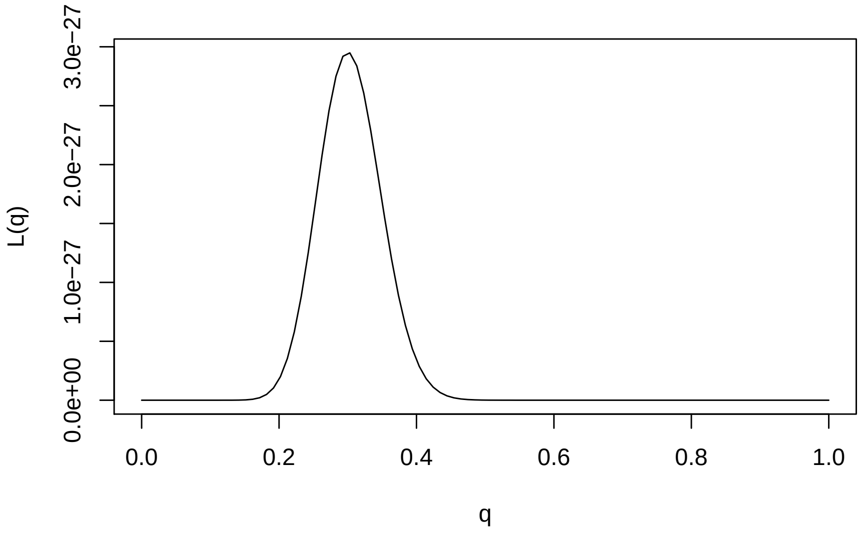
$H_0 : \theta = \theta_0$

$H_1 : \theta \neq \theta_0$

How far away / extreme θ can be if our null hypothesis is true

Assume that our likelihood function for q is $L(q) = q^{30}(1-q)^{70}$ **Likelihood function**

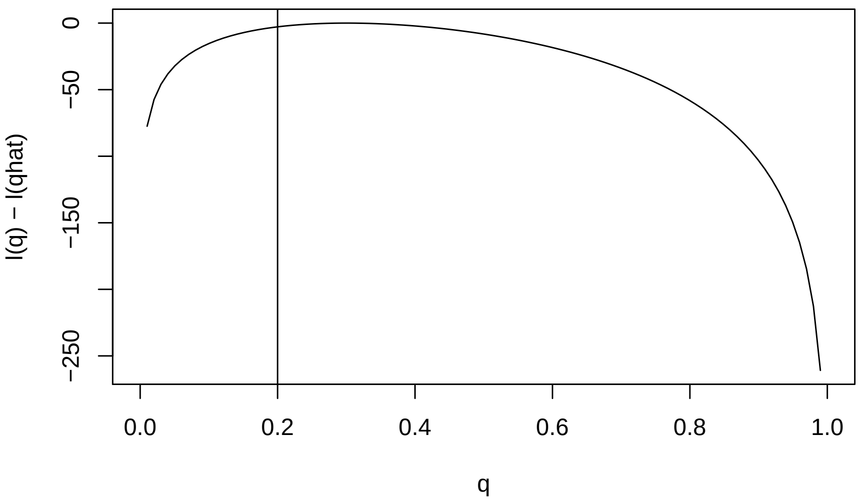
```
q = seq(0,1,length=100)
L= function(q){q^30 * (1-q)^70}
plot(q,L(q),ylab="L(q)",xlab="q",type="l")
```

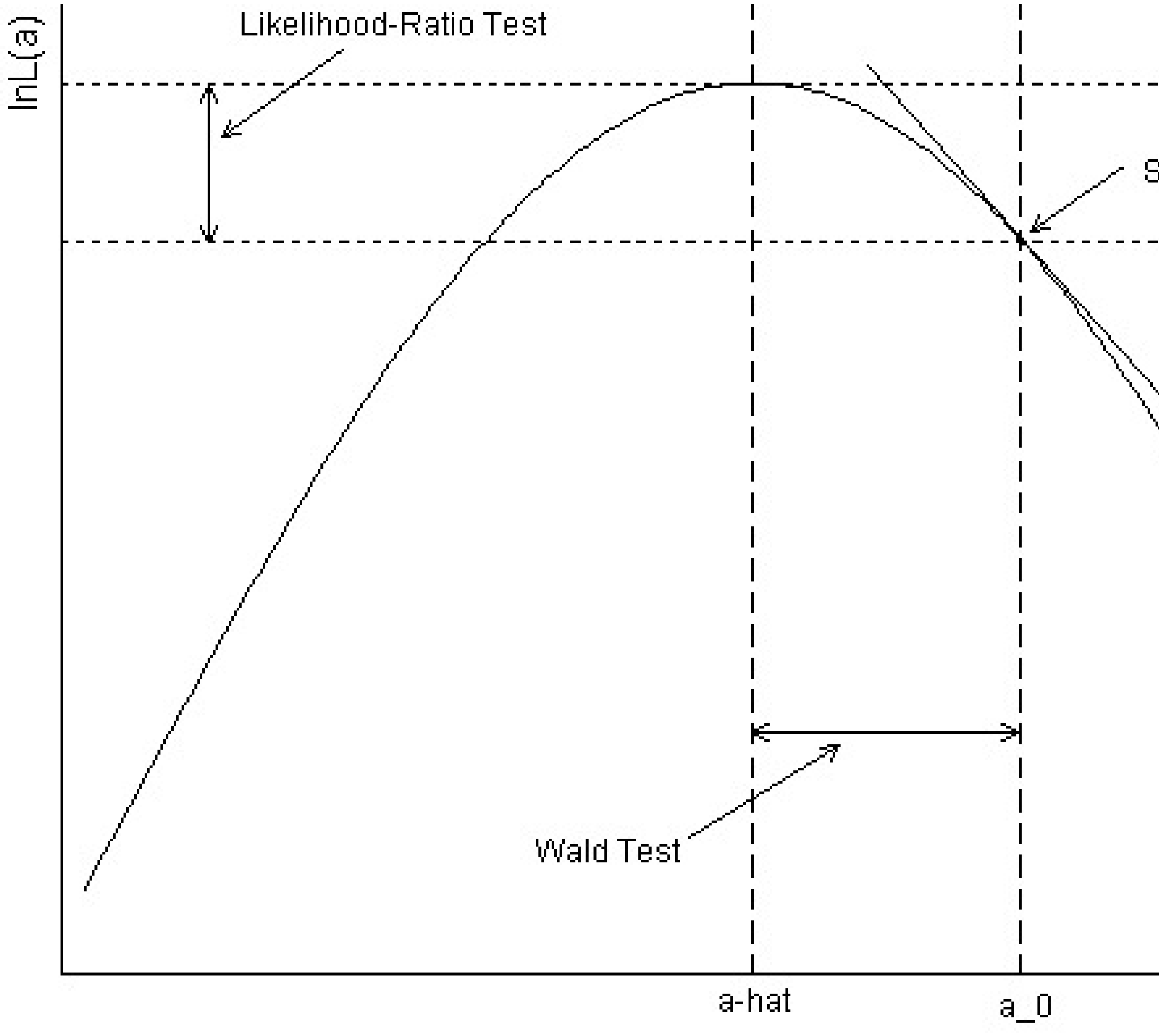


Log-Likelihood function

```
q = seq(0,1,length=100)
l= function(q){30*log(q) + 70 * log(1-q)}
```

```
plot(q,l(q)-l(0.3),ylab="l(q) - l(qhat)",xlab="q",type="l")
abline(v=0.2)
```





(Fox, 1991)

typically, The likelihood ratio test (and Lagrange Multiplier (Score)) performs better with small to moderate sample sizes, but the Wald test only requires one maximization (under the full model).

19.2 Wald test

$$W = (\hat{\theta} - \theta_0)'[cov(\hat{\theta})]^{-1}(\hat{\theta} - \theta_0)W \sim \chi^2_q$$

where $cov(\hat{\theta})$ is given by the inverse Fisher Information matrix evaluated at $\hat{\theta}$ and q is the rank of $cov(\hat{\theta})$, which is the number of non-redundant parameters in θ

Alternatively,

$$t_W = \frac{(\hat{\theta} - \theta_0)^2}{I(\theta_0)^{-1}} \sim \chi^2_{(v)}$$

where v is the degree of freedom.

Equivalently,

$$s_W = \frac{\hat{\theta} - \theta_0}{\sqrt{I(\hat{\theta})^{-1}}} \sim Z$$

How far away in the distribution your sample estimate is from the hypothesized population parameter.

For a null value, what is the probability you would obtained a realization "more extreme" or "worse" than the estimate you actually obtained.

Significance Level (α) and Confidence Level ($1 - \alpha$)

- The significance level is the benchmark in which the probability is so low that we would have to reject the null
- The confidence level is the probability that sets the bounds on how far away the realization of the estimator would have to be to reject the null.

Test Statistics

- Standardized (transform) the estimator and null value to a test statistic that always has the same distribution
- Test Statistic for the OLS estimator for a single hypothesis

$$T = \frac{\sqrt{n}(\hat{\beta}_j - \beta_{j0})}{\sqrt{n}SE(\hat{\beta}_j)} \sim^a N(0,1)$$

Equivalently,

$$T = \frac{(\hat{\beta}_j - \beta_{j0})}{SE(\hat{\beta}_j)} \sim^a N(0,1)$$

the test statistic is another random variable that is a function of the data and null hypothesis.

- T denotes the random variable test statistic

- t denotes the single realization of the test statistic

Evaluating Test Statistic: determine whether or not we reject or fail to reject the null hypothesis at a given significance / confidence level

Three equivalent ways

1. Critical Value

2. P-value

3. Confidence Interval

4. Critical Value

For a given significance level, will determine the critical value (c)
* One-sided: $H_0 : \beta_j \geq \beta_{j0}$

$$P(T < c|H_0) = \alpha$$

Reject the null if $t < c$

- One-sided: $H_0 : \beta_j \leq \beta_{j0}$

$$P(T > c|H_0) = \alpha$$

Reject the null if $t > c$

- TWo-sided: $H_0 : \beta_j \neq \beta_{j0}$

$$P(|T| > c|H_0) = \alpha$$

Reject the null if $|t| > c$

2. p-value

Calculate the probability that the test statistic was worse than the realization you have

- One-sided: $H_0 : \beta_j \geq \beta_{j0}$

$$\text{p-value} = P(T < t|H_0)$$

- One-sided: $H_0 : \beta_j \leq \beta_{j0}$

$$\text{p-value} = P(T > t|H_0)$$

- Two-sided: $H_0 : \beta_j \neq \beta_{j0}$

$$\text{p-value} = P(|T| < t|H_0)$$

reject the null if p-value $< \alpha$

3. Confidence Interval

Using the critical value associated with a null hypothesis and significance level, create an interval

$$CI(\hat{\beta}_j)_\alpha = [\hat{\beta}_j - (c \times SE(\hat{\beta}_j)), \hat{\beta}_j + (c \times SE(\hat{\beta}_j))]$$

If the null set lies outside the interval then we reject the null.

- We are not testing whether the true population value is close to the estimate, we are testing that given a field true population value of the parameter, how like it is that we observed this estimate.
- Can be interpreted as we believe with $(1-\alpha) \times 100\%$ probability that the confidence interval captures the true parameter value.

With stronger assumption (A1-A6), we could consider Finite Sample Properties

$$T = \frac{\hat{\beta}_j - \beta_{j0}}{SE(\hat{\beta}_j)} \sim T(n-k)$$

- This above distributional derivation is strongly dependent on A4 and A5
- T has a student t-distribution because the numerator is normal and the denominator is χ^2.
- Critical value and p-values will be calculated from the student t-distribution rather than the standard normal distribution.
- $n \to \infty$, $T(n-k)$ is asymptotically standard normal.

Rule of thumb

- if $n-k > 120$: the critical values and p-values from the t-distribution are (almost) the same as the critical values and p-values from the standard normal distribution.

- if $n-k < 120$
  - if (A1-A6) hold then the t-test is an exact finite distribution test
  - if (A1-A3a, A5) hold, because the t-distribution is asymptotically normal, computing the critical values from a t-distribution is still a valid asymptotic test (i.e., not quite the right critical values and p0values, the difference goes away as $n \to \infty$)

19.2.1 Multiple Hypothesis

- test multiple parameters as the same time
  - $H_0 : \beta_1 = 0$ & $\beta_2 = 0$
  - $H_0 : \beta_1 = 1$ & $\beta_2 = 0$
- perform a series of simply hypothesis does not answer the question (joint distribution vs. two marginal distributions).
- The test statistic is based on a restriction written in matrix form.

$$y = \beta_0 + x_1\beta_1 + x_2\beta_2 + x_3\beta_3 + \epsilon$$

Null hypothesis is $H_0 : \beta_1 = 0$ & $\beta_2 = 0$ can be rewritten as $H_0 : \mathbf{R}\beta - \mathbf{q} = 0$ where

- $\mathbf{R}$ is a m x k matrix where m is the number of restrictions and k is the number of parameters. $\mathbf{q}$ is a k x 1 vector
- $\mathbf{R}$ "picks up" the relevant parameters while $\mathbf{q}$ is a the null value of the parameter

$$\mathbf{R} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \mathbf{q} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Test Statistic for OLS estimator for a multiple hypothesis

$$F = \frac{(\mathbf{R}\hat{} - \mathbf{q})\hat{\Sigma}^{-1}(\mathbf{R}\hat{} - \mathbf{q})}{m} \sim^a F(m, n-k)$$

- $\hat{\Sigma}^{-1}$ is the estimator for the asymptotic variance-covariance matrix
  - if A4 holds, both the homoskedastic and heteroskedastic versions produce valid estimator
  - If A4 does not hold, only the heteroskedastic version produces valid estimators.
- When m = 1, there is only a single restriction, then the F-statistic is the t-statistic squared.
- F distribution is strictly positive, check F-Distribution for more details.

19.2.2 Linear Combination

Testing multiple parameters as the same time

$$H_0 : \beta_1 - \beta_2 = 0 H_0 : \beta_1 - \beta_2 > 0 H_0 : \beta_1 - 2 * \beta_2 = 0$$

Each is a single restriction on a function of the parameters.

Null hypothesis:

$$H_0 : \beta_1 - \beta_2 = 0$$

can be rewritten as

$$H_0 : \mathbf{R}\beta - \mathbf{q} = 0$$

where $\mathbf{R}$=(0 1 -1 0 0) and $\mathbf{q} = 0$

19.2.3 Application

```
library("car")
```

```
## Loading required package: carData
```

```
# Multiple hypothesis
mod.davis <- lm(weight ~ repwt, data=Davis)
linearHypothesis(mod.davis, c("(Intercept) = 0", "repwt = 1"),white.adjust = TRUE)
```

```
## Linear hypothesis test
##
## Hypothesis:
## (Intercept) = 0
## repwt = 1
##
## Model 1: restricted model
## Model 2: weight ~ repwt
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F  Pr(>F)
## 1    183
## 2    181  2 3.3896 0.03588 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Linear Combination
mod.duncan <- lm(prestige ~ income + education, data=Duncan)
linearHypothesis(mod.duncan, "1*income - 1*education = 0")
```

```
## Linear hypothesis test
##
## Hypothesis:
## income - education = 0
##
```

```
## Model 1: restricted model
## Model 2: prestige ~ income + education
##
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     43 7518.9
## 2     42 7506.7  1    12.195 0.0682 0.7952
```

19.2.4 Nonlinear

Suppose that we have q nonlinear functions of the parameters

$$\mathbf{h}(\theta) = \{h_1(\theta), ..., h_q(\theta)\}'$$

The,n, the Jacobian matrix ($\mathbf{H}(\theta)$), of rank q is

$$\mathbf{H}_{q\times p}(\theta) = \left(\begin{array}{ccc} \frac{\partial h_1(\theta)}{\partial \theta_1} & \dots & \frac{\partial h_1(\theta)}{\partial \theta_p} \\ . & . & . \\ \frac{\partial h_q(\theta)}{\partial \theta_1} & \dots & \frac{\partial h_q(\theta)}{\partial \theta_p} \end{array}\right)$$

where the null hypothesis $H_0 : \mathbf{h}(\theta) = 0$ can be tested agiasnt the 2-sided alternative with the Wald statistic

$$W = \frac{\mathbf{h}(\hat{}\ )'\{\mathbf{H}(\hat{}\ )[\mathbf{F}(\hat{}\ )'\mathbf{F}(\hat{}\ )]^{-\mathbf{1}}\mathbf{H}(\hat{}\ )'\}^{-\mathbf{1}}\mathbf{h}(\hat{}\ )}{s^2 q} \sim F_{q,n-p}$$

19.3 The likelihood ratio test

$$t_{LR} = 2[l(\hat{\theta}) - l(\theta_0)] \sim \chi^2_v$$

where v is the degree of freedom.

Compare the height of the log-likelihood of the sample estimate in relation to the height of log-likelihood of the hypothesized population parameter

Alternatively,

This test considers a ratio of two maximizations,

$L_r =$ maximized value of the likelihood under H_0 (the reduced model)$L_f =$ maximized value of the like

Then, the likelihood ratio is:

$$\Lambda = \frac{L_r}{L_f}$$

which can't exceed 1 (since L_f is always at least as large as $L-r$ because L_r is the result of a maximization under a restricted set of the parameter values).

The likelihood ratio statistic is:

$$-2ln(\Lambda) = -2ln(L_r/L_f) = -2(l_r - l_f) \lim_{n\to\infty}(-2ln(\Lambda)) \sim \chi^2_v$$

where v is the number of parameters in the full model minus the number of parameters in the reduced model.

If L_r is much smaller than L_f (the likelihood ratio exceeds $\chi^2_{\alpha,v}$), then we reject he reduced model and accept the full model at $\alpha \times 100\%$ significance level

19.4 Lagrange Multiplier (Score)

$$t_S = \frac{S(\theta_0)^2}{I(\theta_0)} \sim \chi^2_v$$

where v is the degree of freedom.

Compare the slope of the log-likelihood of the sample estimate in relation to the slope of the log-likelihood of the hypothesized population parameter

Part IV

EXPERIMENTAL DESIGN

Chapter 20

Analysis of Variance (ANOVA)

ANOVA is using the same underlying mechanism as linear regression. However, the angle that ANOVA chooses to look at is slightly different from the traditional linear regression. It can be more useful in the case with **qualitative variables** and **designed experiments**.

Experimental Design

- **Factor**: explanatory or predictor variable to be studied in an investigation
- **Treatment** (or Factor Level): "value" of a factor applied to the experimental unit
- **Experimental Unit**: person, animal, piece of material, etc. that is subjected to treatment(s) and provides a response
- **Single Factor Experiment**: one explanatory variable considered
- **Multifactor Experiment**: more than one explanatory variable
- **Classification Factor**: A factor that is not under the control of the experimenter (observational data)
- **Experimental Factor**: assigned by the experimenter

Basics of experimental design:

- Choices that a statistician has to make:
    - set of treatments
    - set of experimental units
    - treatment assignment (selection bias)
    - measurement (measurement bias, blind experiments)
- Advancements in experimental design:

1. **Factorial Experiments**:
   consider multiple factors at the same time (interaction)
2. **Replication**: repetition of experiment
   - assess mean squared error
   - control over precision of experiment (power)
3. **Randomization**
   - Before R.A. Fisher (1900s), treatments were assigned systematically or subjectively
   - randomization: assign treatments to experimental units at random, which averages out systematic effects that cannot be control by the investigator
4. **Local control**: Blocking or Stratification
   - Reduce experimental errors and increase power by placing restrictions on the randomization of treatments to experimental units.

Randomization may also eliminate correlations due to time and space.

20.1 Completely Randomized Design (CRD)

Treatment factor A with $a \geq 2$ treatments levels. Experimental units are randomly assinged to each treatment. The number of experiemntal units in each group can be

- equal (balanced): n
- unequal (unbalanced): n_i for the i-th group (i = 1,…,a).

The total sample size is $N = \sum_{i=1}^{a} n_i$

Possible assignments of units to treatments are $k = \frac{N!}{n_1!n_2!\dots n_a!}$

Each has probability 1/k of being selected. Each experimental unit is measured with a response Y_{ij}, in which j denotes unit and i denotes treatment.

Treatment

	1	2	…	a
	Y_{11}	Y_{21}	…	Y_{a1}
	Y_{12}	…	…	…
	…	…	…	…
Sample Mean	$\bar{Y_{1.}}$	$\bar{Y_{2.}}$	…	$\bar{Y_{a.}}$
Sample SD	s_1	s_2	…	s_a

where $\bar{Y}_{i.} = \frac{1}{n_i}\sum_{j=1}^{n_i} Y_{ij}$

$s_i^2 = \frac{1}{n_i - 1}\sum_{j=1}^{n_i}(Y_{ij} - \bar{Y}_i)^2$

And the grand mean is $\bar{Y}_{..} = \frac{1}{N}\sum_i \sum_j Y_{ij}$

20.1.1 Single Factor Fixed Effects Model

also known as Single Factor (One-Way) ANOVA or ANOVA Type I model.

Partitioning the Variance

The total variability of the Y_{ij} observation can be measured as the deviation of Y_{ij} around the overall mean $\bar{Y}_{..}$: $Y_{ij} - \bar{Y}_{..}$

This can be rewritten as:

$$
\begin{aligned}
Y_{ij} - \bar{Y}_{..} &= Y_{ij} - \bar{Y}_{..} + \bar{Y}_{i.} - \bar{Y}_{i.} \\
&= (\bar{Y}_{i.} - \bar{Y}_{..}) + (Y_{ij} - \bar{Y}_{i.})
\end{aligned}
$$

where

- the first term is the *between* treatment differences (i.e., the deviation of the treatment mean from the overall mean)
- the second term is *within* treatment differences (i.e., the deviation of the observation around its treatment mean)

$$
\begin{aligned}
\sum_i \sum_j (Y_{ij} - \bar{Y}_{..})^2 &= \sum_i n_i (\bar{Y}_{i.} - \bar{Y}_{..})^2 + \sum_i \sum_j (Y_{ij} - \bar{Y}_{i.})^2 \\
SSTO &= SSTR + SSE \\
total\ SS &= treatment\ SS + error\ SS \\
(N-1)\ d.f. &= (a-1)\ d.f. + (N-a)\ d.f.
\end{aligned}
$$

we lose a d.f. for the total corrected SSTO because of the estimation of the mean ($\sum_i \sum_j (Y_{ij} - \bar{Y}_{..}) = 0$)
And, for the SSTR $\sum_i n_i(\bar{Y}_{i.} - \bar{Y}_{..}) = 0$

Accordingly, $MSTR = \frac{SST}{a-1}$ and $MSR = \frac{SSE}{N-a}$

ANOVA Table

Source of Variation	SS	df	MS
Between Treatments	$\sum_i n_i(\bar{Y}_{i.} - \bar{Y}_{..})^2$	a-1	SSTR/(a-1)
Error (within treatments)	$\sum_i \sum_j (Y_{ij} - \bar{Y}_{i.})^2$	N-a	SSE/(N-a)
Total (corrected)	$\sum_i n_i(\bar{Y}_{i.} - \bar{Y}_{..})^2$	N-1	

Linear Model Explanation of ANOVA

20.1.1.1 Cell means model

$$Y_{ij} = \mu_i + \epsilon\_ij$$

where

- Y_{ij} response variable in j-th subject for the i-th treatment

- μ_i: parameters (fixed) representing the unknown population mean for the i-th treatment

- ϵ_{ij} independent $N(0, \sigma^2)$ errors
- $E(Y_{ij}) = \mu_i$ $var(Y_{ij}) = var(\epsilon_{ij}) = \sigma^2$

- All observations have the same variance

Example:

a = 3 (3 treatments) $n_1 = n_2 = n_3 = 2$

$$\begin{aligned}
\left(\begin{array}{c} Y_{11}\\ Y_{12}\\ Y_{21}\\ Y_{22}\\ Y_{31}\\ Y_{32} \end{array}\right) &= \left(\begin{array}{ccc} 1 & 0 & 0\\ 1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 0 & 1 \end{array}\right) \left(\begin{array}{c} \mu_1\\ \mu_2\\ \mu_3 \end{array}\right) + \left(\begin{array}{c} \epsilon_{11}\\ \epsilon_{12}\\ \epsilon_{21}\\ \epsilon_{22}\\ \epsilon_{31}\\ \epsilon_{32} \end{array}\right)\\
\mathbf{y} = \mathbf{X} &+
\end{aligned}$$

$X_{k,ij} = 1$ if the k-th treatment is used

$X_{k,ij} = 0$ Otherwise

Note: no intercept term.

$$\begin{aligned}
\mathbf{b} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix} &= (\mathbf{x}'\mathbf{x})^{-1}\mathbf{x}'\mathbf{y} \\
&= \begin{bmatrix} n_1 & 0 & 0 \\ 0 & n_2 & 0 \\ 0 & 0 & n_3 \end{bmatrix}^{-1} \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} \qquad (20.1) \\
&= \begin{bmatrix} \bar{Y_1} \\ \bar{Y_2} \\ \bar{Y_3} \end{bmatrix}
\end{aligned}$$

is the BLUE (best linear unbiased estimator) for $\beta = [\mu_1\mu_2\mu_3]'$

$$E(\mathbf{b}) = \beta$$

$$var(\mathbf{b}) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1} = \sigma^2 \begin{bmatrix} 1/n_1 & 0 & 0 \\ 0 & 1/n_2 & 0 \\ 0 & 0 & 1/n_3 \end{bmatrix}$$

$var(b_i) = var(\hat{\mu_i}) = \sigma^2/n_i$ where $\mathbf{b} \sim N(\beta, \sigma^2(\mathbf{X}'\mathbf{X})^{-1})$

$$\begin{aligned}
MSE &= \frac{1}{N-a}\sum_i\sum_j(Y_{ij} - \bar{Y_{i.}})^2 \\
&= \frac{1}{N-a}\sum_i[(n_i - 1)\frac{\sum_i(Y_{ij} - \bar{Y_{i.}})^2}{n_i - 1}] \\
&= \frac{1}{N-a}\sum_i(n_i - 1)s_1^2
\end{aligned}$$

We have $E(s_i^2) = \sigma^2$

$E(MSE) = \frac{1}{N-a}\sum_i(n_i - 1)\sigma^2 = \sigma^2$

Hence, MSE is an unbiased estimator of σ^2, regardless of whether the treatment means are equal or not.

$E(MSTR) = \sigma^2 + \frac{\sum_i n_i(\mu_i - \mu_.)^2}{a-1}$
where $\mu_. = \frac{\sum_{i=1}^a n_i\mu_i}{\sum_{i=1}^a n_i}$
If all treatment means are equals $(=\mu_.)$, $E(MSTR) = \sigma^2$.

Then we can use an F-test for teh equality of all treatment means:

$$H_0 : \mu_1 = \mu_2 = .. = \mu_a$$

$$H_a : not\ all\ \mu_i\ are\ equal$$

$F = \frac{MSTR}{MSE}$
where large values of F support H_a (since MSTR will tend to exceed MSE when H_a holds)
and F near 1 support H_0 (upper tail test)

Equivalently, when H_0 is true, $F \sim f_{(a-1,N-a)}$

- If $F \leq f_{(a-1,N-a;1-\alpha)}$, we cannot reject H_0
- If $F \geq f_{(a-1,N-a;1-\alpha)}$, we reject H_0

Note: If a = 2 (2 treatments), F-test = two sample t-test

20.1.1.2 Treatment Effects (Factor Effects)

Besides Cell means model, we have another way to formalize one-way ANOVA:

$$Y_{ij} = \mu + \tau_i + \epsilon_{ij}$$

where

- Y_{ij} is the j-th response for the i-th treatment
- τ_i i-th treatment effect
- μ constant component, common to all observations
- ϵ_{ij} independent random errors $\sim N(0, \sigma^2)$

For example, a = 3, $n_1 = n_2 = n_3 = 2$

$$\begin{pmatrix} Y_{11} \\ Y_{12} \\ Y_{21} \\ Y_{22} \\ Y_{31} \\ Y_{32} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mu \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{pmatrix} + \begin{pmatrix} \epsilon_{11} \\ \epsilon_{12} \\ \epsilon_{21} \\ \epsilon_{22} \\ \epsilon_{31} \\ \epsilon_{32} \end{pmatrix} \tag{20.2}$$

$$\mathbf{y} = \mathbf{X}\ +$$

However,

$$\mathbf{X}'\mathbf{X} = \begin{pmatrix} \sum_i n_i & n_1 & n_2 & n_3 \\ n_1 & n_1 & 0 & 0 \\ n_2 & 0 & n_2 & 0 \\ n_3 & 0 & 0 & n_3 \end{pmatrix}$$

is **singular** thus does not exist, **b** is insolvable (infinite solutions)

Hence, we have to impose restrictions on the parameters to a model matrix **X** of full rank.

Whatever restriction we use, we still have:

$E(Y_{ij}) = \mu + \tau_i = \mu_i = mean\ response\ for\ i-th\ treatment$

20.1.1.2.1 Restriction on sum of tau $\sum_{i=1}^{a} \tau_i = 0$

implies

$$\mu = \mu + \frac{1}{a}\sum_{i=1}^{a}(\mu + \tau_i)$$

is the average of the treatment mean (grand mean) (overall mean)

$$\begin{aligned} \tau_i &= (\mu + \tau_i) - \mu = \mu_i - \mu \\ &= \text{treatment mean} - \text{grand mean} \\ &= \text{treatment effect} \end{aligned}$$

$$\tau_a = -\tau_1 - \tau_2 - ... - \tau_{a-1}$$

Hence, the mean for the a-th treatment is

$$\mu_a = \mu + \tau_a = \mu - \tau_1 - \tau_2 - ... - \tau_{a-1}$$

Hence, the model need only "a" parameters:

$$\mu, \tau_1, \tau_2, .., \tau_{a-1}$$

Equation (20.2) becomes

$$\left(\begin{array}{c} Y_{11} \\ Y_{12} \\ Y_{21} \\ Y_{22} \\ Y_{31} \\ Y_{32} \end{array}\right) = \left(\begin{array}{ccc} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & -1 & -1 \\ 1 & -1 & -1 \end{array}\right) \left(\begin{array}{c} \mu \\ \tau_1 \\ \tau_2 \end{array}\right) + \left(\begin{array}{c} \epsilon_{11} \\ \epsilon_{12} \\ \epsilon_{21} \\ \epsilon_{22} \\ \epsilon_{31} \\ \epsilon_{32} \end{array}\right) \tag{20.3}$$

$$\mathbf{y} = \mathbf{X} \; +$$

where $\beta \equiv [\mu, \tau_1, \tau_2]'$

Equation (20.1) with $\sum_i \tau_i = 0$ becomes

$$\begin{aligned}
\mathbf{b} = \left[\begin{array}{c} \hat{\mu} \\ \hat{\tau_1} \\ \hat{\tau_2} \end{array}\right] &= (\mathbf{x}'\mathbf{x})^{-1}\mathbf{x}'\mathbf{y} \\
&= \left[\begin{array}{ccc} \sum_i n_i & n_1 - n_3 & n_2 - n_3 \\ n_1 - n_3 & n_1 + n_3 & n_3 \\ n_2 - n_3 & n_3 & n_2 - n_3 \end{array}\right]^{-1} \left[\begin{array}{c} Y_{..} \\ Y_{1.} - Y_{3.} \\ Y_{2.} - Y_{3.} \end{array}\right] \\
&= \left[\begin{array}{c} \frac{1}{3}\sum_{i=1}^{3}\bar{Y_{i.}} \\ \bar{Y_{1.}} - \frac{1}{3}\sum_{i=1}^{3}\bar{Y_{i.}} \\ \bar{Y_{2.}} - \frac{1}{3}\sum_{i=1}^{3}\bar{Y_{i.}} \end{array}\right] \\
&= \left[\begin{array}{c} \hat{\mu} \\ \hat{\tau_1} \\ \hat{\tau_2} \end{array}\right]
\end{aligned}$$

and $\hat{\tau_3} = -\hat{\tau_1} - \hat{\tau_2} = \bar{Y_3} - \frac{1}{3}\sum_i \bar{Y_{i.}}$

20.1.1.2.2 Restriction on first tau

In R, `lm()` uses the restriction $\tau_1 = 0$

For the previous example, for $n_1 = n_2 = n_3 = 2$, and $\tau_1 = 0$. Then the treatment means can be written as:

$$\mu_1 = \mu + \tau_1 = \mu + 0 = \mu\mu_2 = \mu + \tau_2\mu_3 = \mu + \tau_3$$

Hence, μ is the mean response for the first treatment

In the matrix form,

$$\begin{pmatrix} Y_{11} \\ Y_{12} \\ Y_{21} \\ Y_{22} \\ Y_{31} \\ Y_{32} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mu \\ \tau_2 \\ \tau_3 \end{pmatrix} + \begin{pmatrix} \epsilon_{11} \\ \epsilon_{12} \\ \epsilon_{21} \\ \epsilon_{22} \\ \epsilon_{31} \\ \epsilon_{32} \end{pmatrix}$$

$$\mathbf{y} = \mathbf{X} \; +$$

$\beta = [\mu, \tau_2, \tau_3]'$

$$\begin{aligned} \mathbf{b} = \begin{bmatrix} \hat{\mu} \\ \hat{\tau_2} \\ \hat{\tau_3} \end{bmatrix} &= (\mathbf{x}'\mathbf{x})^{-1}\mathbf{x}'\mathbf{y} \\ &= \begin{bmatrix} \sum_i n_i & n_2 & n_3 \\ n_2 & n_2 & 0 \\ n_3 & 0 & n_3 \end{bmatrix}^{-1} \begin{bmatrix} Y_{..} \\ Y_{2.} \\ Y_{3.} \end{bmatrix} \\ &= \begin{bmatrix} \bar{Y_{1.}} \\ \bar{Y_{2.}} - \bar{Y_{1.}} \\ \bar{Y_{3.}} - \bar{Y_{1.}} \end{bmatrix} \end{aligned}$$

$$E(\mathbf{b}) = \beta = \begin{bmatrix} \mu \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 - \mu_1 \\ \mu_3 - \mu_1 \end{bmatrix}$$

$var(\mathbf{b}) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1} var(\hat{\mu}) = var(\bar{Y_{1.}}) = \sigma^2/n_1 var(\hat{\tau_2}) = var(\bar{Y_{2.}} - \bar{Y_{1.}}) = \sigma^2/n_2 + \sigma^2/n_1 var(\hat{\tau_3}) = var(\bar{Y_{3.}} - \bar{Y_{1.}}) = \sigma^2/$

Note For all three parameterization, the ANOVA table is the same

- Model 1: $Y_{ij} = \mu_i + \epsilon_{ij}$
- Model 2: $Y_{ij} = \mu + \tau_i + \epsilon_{ij}$ where $\sum_i \tau_i = 0$
- Model 3: $Y_{ij} = \mu + \tau_i + \epsilon_{ij}$ where $\tau_1 = 0$

All models have the same calculation for $\hat{Y}$ as

$$\hat{\mathbf{Y}} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} = \mathbf{PY} = \mathbf{Xb}$$

ANOVA Table

Source of Variation	SS	df	MS	F
Between Treatments	$\sum_i n_i(\bar{Y_{i.}} - \bar{Y_{..}})^2 = \mathbf{Y}'(\mathbf{P} - \mathbf{P_1})\mathbf{Y}$	a-1	$\frac{SSTR}{a-1}$	$\frac{MSTR}{MSE}$
Error (within treatments)	$\sum_i \sum_j (Y_{ij} - \bar{Y_{i.}})^2 = \mathbf{e}'\mathbf{e}$	N-a	$\frac{SSE}{N-a}$	
Total (corrected)	$\sum_i n_i(\bar{Y_{i.}} - \bar{Y_{..}})^2 = \mathbf{Y}'\mathbf{Y} - \mathbf{Y}'\mathbf{P_1}\mathbf{Y}$	N-1		

where $\mathbf{P_1} = \frac{1}{n}\mathbf{J}$

The F-statistic here has (a-1,N-a) degrees of freedom, which gives the same value for all three parameterization, but the hypothesis test is written a bit different:

$$H_0 : \mu_1 = \mu_2 = ... = \mu_a H_0 : \mu + \tau_1 = \mu + tau_2 = ... = \mu + \tau_a H_0 : \tau_1 = \tau_2 = ... = \tau_a$$

The F-test here serves as a preliminary analysis, to see if there is any difference at different factors. For more in-depth analysis, we consider different testing of treatment effects.

20.1.1.3 Testing of Treatment Effects

- A Single Treatment Mean μ_i
- A Differences Between Treatment Means
- A Contrast Among Treatment Means
- A Linear Combination of Treatment Means

20.1.1.3.1 Single Treatment Mean We have $\hat{\mu_i} = \bar{Y_{i.}}$ where

- $E(\bar{Y_{i.}}) = \mu_i$
- $var(\bar{Y_i}) = \sigma^2/n_i$ estimated by $s^2(\bar{Y_{i.}}) = MSE/n_i$

Since $\frac{\bar{Y_{i.}} - \mu_i}{s(\bar{Y_{i.}})} \sim t_{N-a}$ and the confidence interval for μ_i is $\bar{Y_{i.}} \pm t_{1-\alpha/2;N-a}s(\bar{Y_{i.}})$, then we can do a t-test for the means difference with some constant c

$$H_0 : \mu_i = c H_1 : \mu_i \neq c$$

where

$$T = \frac{\bar{Y_{i.}} - c}{s(\bar{Y_{i.}})}$$

follows t_{N-a} when H_0 is true.
If $|T| > t_{1-\alpha/2;N-a}$, we can reject H_0

20.1.1.3.2 Differences Between Treatment Means Let $D = \mu_i - \mu_i'$, also known as **pairwise comparison**
D can be estimated by $\hat{D} = \bar{Y_i} - \bar{Y_i}'$ is unbiased ($E(\hat{D}) = \mu_i - \mu_i'$)

Since $\bar{Y_i}$ and $\bar{Y_i}'$ are independent, then

$$var(\hat{D}) = var(\bar{Y_i}) + var(\bar{Y_{i'}}) = \sigma^2(1/n_i + 1/n_i')$$

can be estimated with

$$s^2(\hat{D}) = MSE(1/n_i + 1/n_i')$$

With the single treatment inference,

$$\frac{\hat{D} - D}{s(\hat{D})} \sim t_{N-a}$$

hence,

$$\hat{D} \pm t_{(1-\alpha/2;N-a)} s(\hat{D})$$

Hypothesis tests:

$$H_0 : \mu_i = \mu_i' H_a : \mu_i \neq \mu_i'$$

can be tested by the following statistic

$$T = \frac{\hat{D}}{s(\hat{D})} \sim t_{1-\alpha/2;N-a}$$

reject H_0 if $|T| > t_{1-\alpha/2;N-a}$

20.1.1.3.3 Contrast Among Treatment Means generalize the comparison of two means, we have **contrasts**

A contrast is a linear combination of treatment means:

$$L = \sum_{i=1}^{a} c_i \mu_i$$

where each c_i is non-random constant and sum to 0:

$$\sum_{i=1}^{a} c_i = 0$$

An unbiased estimator of a contrast L is

$$\hat{L} = \sum_{i=1}^{a} c_i \bar{Y}_{i.}$$

and $E(\hat{L}) = L$. Since the $\bar{Y}_{i.}$, i = 1,…, a are independent.

$$var(\hat{L}) = var(\sum_{i=1}^{a} c_i \bar{Y}_{i.}) = \sum_{i=1}^{a} var(c_i \bar{Y}_i) = \sum_{i=1}^{a} c_i^2 var(\bar{Y}_i) = \sum_{i=1}^{a} c_i^2 \sigma^2 / n_i = \sigma^2 \sum_{i=1}^{a} c_i^2 / n_i$$

Estimation of the variance:

$$s^2(\hat{L}) = MSE \sum_{i=1}^{a} \frac{c_i^2}{n_i}$$

$\hat{L}$ is normally distributed (since it is a linear combination of independent normal random variables).

Then, since SSE/σ^2 is χ^2_{N-a}

$$\frac{\hat{L} - L}{s(\hat{L})} \sim t_{N-a}$$

A $1 - \alpha$ confidence limits are given by

$$\hat{L} \pm t_{1-\alpha/2;N-a} s(\hat{L})$$

Hypothesis testing

$$H_0 : L = 0 H_a : L \neq 0$$

with

$$T = \frac{\hat{L}}{s(\hat{L})}$$

reject H_0 if $|T| > t_{1-\alpha/2;N-a}$

20.1.1.3.4 Linear Combination of Treatment Means just like contrast $L = \sum_{i=1}^a c_i\mu_i$ but no restrictions on the c_i coefficients.

Tests og a single treatment mean, two treatment means, and contrasts can all be considered form the same perspective.

$$H_0 : \sum c_i\mu_i = cH_a : \sum c_i\mu_i \neq c$$

The test statistics (t-stat) can be considered equivalently as F-tests; $F = (T)^2$ where $F \sim F_{1,N-a}$. Since the numerator degrees of freedom is always 1 in these cases, we refer to them as single-degree-of-freedom tests.

Multiple Contrasts

To test simultaneously $k \geq 2$ contrasts, let $T_1, ..., T_k$ be the t-stat. The joint distribution of these random variables is a multivariate t-distribution (the tests are dependent since they re based on the same data).

Limitations for comparing multiple contrasts:

1. The confidence coefficient $1 - \alpha$ only applies to a particular estimate, not a series of estimates; similarly, the Type I error rate, α, applies to a particular test, not a series of tests. Example: 3 t-tests at $\alpha = 0.05$, if tests are independent (which they are not), $0.95^3 = 0.857$ (thus $\alpha - 0.143$ not 0.05)

2. The confidence coefficient $1 - \alpha$ and significance level α are appropriate only if the test was not suggest by the data.
   - often, the results of an experiment suggest important (ie..g, potential significant) relationships.

   - the process of studying effects suggests by the data is called **data snooping**

Multiple Comparison Procedures:

- Tukey
- Scheffe
- Bonferroni

20.1.1.3.4.1 Tukey All pairwise comparisons of factor level means. All pairs $D = \mu_i - \mu_i'$ or all tests of the form:

$$H_0 : \mu_i - \mu_i' = 0H_a : \mu_i - \mu_i' \neq 0$$

- When all sample sizes are equal ($n_1 = n_2 = ... = n_a$) then the Tukey method family confidence coefficient is exactly $1 - \alpha$ and the significance level is exactly α

- When the sample sizes are not equal, the family confidence coefficient is greater than $1-\alpha$ (i.e., the significance level is less than α) so the test i**conservative**

- Tukey considers the **studentized range distribution**. If we have $Y_1, .., Y_r$, observations from a normal distribution with mean α and variance σ^2. Define:
$$w = max(Y_i) - min(Y_i)$$
as the range of the observations. Let s^2 be an estimate of σ^2 with v degrees of freedom. Then,
$$q(r,v) = \frac{w}{s}$$
is called the studentized range. The distribution of q uses a special table.

Notes

- when we are not interested in testing all pairwise comparison,s the confidence coefficient for the family of comparisons under consideration will be greater than $1-\alpha$ (with the significance level less than α)

- Tukey can be used for "data snooping" as long as the effects to be studied on the basis of preliminary data analysis are pairwise comparisons.

20.1.1.3.4.2 Scheffe This method applies when the family of interest is the set of possible contrasts among the treatment means:

$$L = \sum_{i=1}^{a} c_i \mu_i$$

where $\sum_{i=1}^{a} c_i = 0$

That is, the family of all possible contrasts L or

$$H_0 : L = 0 H_a : L \neq 0$$

The family confidence level for the Scheffe procedure is exactly $1-\alpha$ (i.e., significance level $= \alpha$) whether the sample sizes are equal or not.

For simultaneous confidence intervals,

$$\hat{L} \pm Ss(\hat{L})$$

where $\hat{L} = \sum c_i \bar{Y}_{i.}, s^2(\hat{L}) = MSE \sum c_i^2/n_i$ and $S^2 = (a-1) f_{1-\alpha;a-1,N-a}$

The Scheffe procedure considers

$$F = \frac{\hat{L}^2}{(a-1)s^2(\hat{L})}$$

where we reject H_0 at the family significance level α if $F > f_{(1-\alpha;a-1,N-a)}$

Note

- Since applications of the Scheffe never involve all conceivable contrasts, the **finite family** confidence coefficient will be larger than $1-\alpha$, so $1-\alpha$ is a lower bound. Thus, people often consider a larger α (e.g., 90% confidence interval)

- Scheffe can be used for "data scooping" since the family of statements contains all possible contrasts.

- If only pairwise comparisons are to be considered, The Tukey procedure gives narrower confidence limits.

20.1.1.3.4.3 Bonferroni Applicable whether the sample sizes are equal or unequal.

For the confidence intervals,

$$\hat{L} \pm Bs(\hat{L})$$

where $B = t_{(1-\alpha/(2g);N-a)}$ and g is the number of comparisons in the family.

Hypothesis testing

$$H_0 : L = 0 H_a : L \neq 0$$

Let $T = \frac{\hat{L}}{s(\hat{L})}$ and reject H_0 if $|T| > t_{1-\alpha/(2g),N-a}$

Notes

- If all pairwise comparisons are of interest, the Tukey procedure is superior (narrower confidence intervals). If not, Bonferroni may be better.
- Bonferroni is better than Scheffe when the number of contrasts is about the same as the treatment levels (or less).
- Recommendation: compute all threes and pick the smallest.
- Bonferroni can't be used for **data snooping**

20.1.1.3.4.4 Fisher's LSD does not control for family error rate

use t-stat for testing

$$H_0 : \mu_i = \mu_j$$

t-stat

$$t = \frac{\bar{y}_i - \bar{y}_j}{\sqrt{MSE(\frac{1}{n_i} + \frac{1}{n_j})}}$$

20.1.1.3.4.5 Newman-Keuls Do not recommend using this test since it has less power than ANOVA.

20.1.1.3.5 Multiple comparisons with a control

20.1.1.3.5.1 Dunnett We have a groups where the last group is the control group, and the $a - 1$ treatment groups.

Then, we compare treatment groups to the control group. Hence, we have $a-1$ contrasts (i.e., $a - 1$ pairwise comparisons)

20.1.1.3.6 Summary When choosing a multiple contrast method:

- Pairwise
  - Equal groups sizes: Tukey
  - Unequal groups sizes: Tukey, Scheffe
- Not pairwise
  - with control: Dunnett
  - general: Bonferroni, Scheffe

20.1.2 Single Factor Random Effects Model

Also known as ANOVA Type II models.

Treatments are chosen at from from larger population. We extend inference to all treatments in the population and not restrict our inference to those treatments that happened to be selected for the study.

20.1.2.1 Random Cell Means

$$Y_{ij} = \mu_i + \epsilon_{ij}$$

where

- $\mu_i \sim N(\mu, \sigma^2_\mu)$ and independent
- $\epsilon_{ij} \sim N(0, \sigma^2)$ and independent

μ_i and ϵ_{ij} are mutually independent for $i = 1, ..., a; j = 1, ..., n$

With all treatment sample sizes are equal

$$E(Y_{ij}) = E(\mu_i) = \mu var(Y_{ij}) = var(\mu_i) + var(\epsilon_i) = \sigma^2_\mu + \sigma^2$$

Since Y_{ij} are not independent

$$\begin{aligned}
cov(Y_{ij}, Y_{ij'}) &= E(Y_{ij}Y_{ij'}) - E(Y_{ij})E(Y_{ij'}) \\
&= E(\mu_i^2 + \mu_i\epsilon_{ij'} + \mu_i\epsilon_{ij} + \epsilon_{ij}\epsilon_{ij'}) - \mu^2 \\
&= \sigma^2_\mu + \mu^2 - \mu^2 && \text{if} j \neq j' \\
&= \sigma^2_\mu && \text{if} j \neq j'
\end{aligned}$$

$$\begin{aligned}
cov(Y_{ij}, Y_{i'j'}) &= E(\mu_i\mu_{i'} + \mu_i\epsilon_{i'j'} + \mu_{i'}\epsilon_{ij} + \epsilon_{ij}\epsilon_{i'j'}) - \mu^2 \\
&= \mu^2 - \mu^2 && \text{if } i \neq i' \\
&= 0
\end{aligned}$$

Hence,

- all observations have the same variance
- any two observations from the same treatment have covariance σ^2_μ
- The correlation between any two responses from the same treatment:

$$\rho(Y_{ij}, Y_{ij'}) = \frac{\sigma^2_\mu}{\sigma^2_\mu + \sigma^2} \quad j \neq j'$$

Inference

Intraclass Correlation Coefficient

$$\frac{\sigma^2_\mu}{\sigma^2 + \sigma^2_\mu}$$

which measures the proportion of total variability of Y_{ij} accounted for by the variance of μ_i

$$H_0 : \sigma^2_\mu = 0 H_a : \sigma^2_\mu \neq 0$$

H_0 implies $\mu_i = \mu$ for all i, which can be tested by the F-test in ANOVA.

The understandings of the Single Factor Fixed Effects Model and the Single Factor Random Effects Model are different, the ANOVA is same for the one factor model. The difference is in the expected mean squares

Random Effects Model	**Fixed Effects** Model
$E(MSE) = \sigma^2$	$E(MSE) = \sigma^2$
$E(MSTR) = \sigma^2 - n\sigma^2_\mu$	$E(MSTR) = \sigma^2 + \frac{\sum_i n_i(\mu_i - \mu)^2}{a-1}$

If σ^2_μ, then MSE and MSTR have the same expectation (σ^2). Otherwise, $E(MSTR) > E(MSE)$. Large values of the statistic

$$F = \frac{MSTR}{MSE}$$

suggest we reject H_0.

Since $F \sim F_{(a-1,a(n-1))}$ when H_0 holds. If $F > f_{(1-\alpha;a-1,a(n-1))}$ we reject H_0. If sample sizes are not equal, F-test can still be used, but the df are $a-1$ and $N-a$.

20.1.2.1.1 Estimation of μ An unbiased estimator of $E(Y_{ij}) = \mu$ is the grand mean: $\hat{\mu} = \hat{Y}_{..}$

The variance of this estimator is

$$\begin{aligned}
var(\bar{Y}_{..}) &= var(\sum_i \bar{Y}_{i.}/a) \\
&= \frac{1}{a^2} \sum_i var(\bar{Y}_{i.}) \\
&= \frac{1}{a^2} \sum_i (\sigma^2_\mu + \sigma^2/n) \\
&= \frac{1}{a^2} (\sigma^2_\mu + \sigma^2/n) \\
&= \frac{n\sigma^2_\mu + \sigma^2}{an}
\end{aligned}$$

An unbiased estimator of this variance is $s^2(\bar{Y}) = \frac{MSTR}{an}$. Thus $\frac{\bar{Y}_{..} - \mu}{s(\bar{Y}_{..})} \sim t_{a-1}$

A $1-\alpha$ confidence interval is $\bar{Y}_{..} \pm t_{(1-\alpha/2;a-1)} s(\bar{Y}_{..})$

20.1.2.1.2 Estimation of $\sigma^2_\mu/(\sigma^2_\mu + \sigma^2)$ In the random and fixed effects model, MSTR and MSE are independent. When the sample sizes are equal ($n_i = n$ for all i),

$$\frac{\frac{MSTR}{n\sigma^2_\mu+\sigma^2}}{\frac{MSE}{\sigma^2}} \sim f_{(a-1,a(n-1))}$$

$$P(f_{(\alpha/2;a-1,a(n-1))} \leq \frac{\frac{MSTR}{n\sigma^2_\mu+\sigma^2}}{\frac{MSE}{\sigma^2}} \leq f_{(1-\alpha/2;a-1,a(n-1))}) = 1-\alpha$$

$$L = \frac{1}{n}(\frac{MSTR}{MSE}(\frac{1}{f_{(1-\alpha/2;a-1,a(n-1))}})-1)U = \frac{1}{n}(\frac{MSTR}{MSE}(\frac{1}{f_{(\alpha/2;a-1,a(n-1))}})-1)$$

The lower and upper (L^*, U^*) confidence limits for $\frac{\sigma^2_\mu}{\sigma^2_\mu+\sigma^2}$

$$L^* = \frac{L}{1+L}U^* = \frac{U}{1+U}$$

If the lower limit for $\frac{\sigma^2_\mu}{\sigma^2}$ is negative, it is customary to set L = 0.

20.1.2.1.3 Estimation of σ^2 $a(n-1)MSE/\sigma^2 \sim \chi^2_{a(n-1)}$, the $(1-\alpha)$ confidence interval for σ^2:

$$\frac{a(n-1)MSE}{\chi^2_{1-\alpha/2;a(n-1)}} \leq \sigma^2 \leq \frac{a(n-1)MSE}{\chi^2_{\alpha/2;a(n-1)}}$$

can also be used in case sample sizes are not equal - then df is N-a.

20.1.2.1.4 Estimation of σ^2_μ $E(MSE) = \sigma^2$ $E(MSTR) = \sigma^2 + n\sigma^2_\mu$. Hence,

$$\sigma^2_\mu = \frac{E(MSTR) - E(MSE)}{n}$$

An unbiased estimator of σ^2_μ is given by

$$s^2_\mu = \frac{MSTR - MSE}{n}$$

if $s^2_\mu < 0$, set $s^2_\mu = 0$

If sample sizes are not equal,

$$s_{\mu}^2 = \frac{MSTR - MSE}{n'}$$

where $n' = \frac{1}{a-1}(\sum_i n_i - \frac{\sum_i n_i^2}{\sum_i n_i})$

no exact confidence intervals for σ_{μ}^2, but we can approximate intervals.

Satterthewaite Procedure can be used to construct approximate confidence intervals for linear combination of expected mean squares
A linear combination:

$$\sigma_{\mu}^2 = \frac{1}{n}E(MSTR) + (-\frac{1}{n})E(MSE)$$

$$S = d_1 E(MS_1) + .. + d_h E(MS_h)$$

where d_i are coefficients.

An unbiased estimator of S is

$$\hat{S} = d_1 MS_1 + ... + d_h MS_h$$

Let df_i be the degrees of freedom associated with teh mean square MS_i. The **Satterthwaite** approximation:

$$\frac{(df)\hat{S}}{S} \sim \chi^2_{df}$$

where

$$df = \frac{(d_1 MS_1 + ... + d_h MS_h)^2}{(d_1 MS_1)^2/df_1 + ... + (d_h MS_h)^2/df_h}$$

An approximate $1-\alpha$ confidence interval for S:

$$\frac{(df)\hat{S}}{\chi^2_{1-\alpha/2;df}} \le S \le \frac{(df)\hat{S}}{\chi^2_{\alpha/2;df}}$$

For the single factor random effects model

$$\frac{(df)s_{\mu}^2}{\chi^2_{1-\alpha/2;df}} \le \sigma_{\mu}^2 \le \frac{(df)s_{\mu}^2}{\chi^2_{\alpha/2;df}}$$

where

$$df = \frac{(sn_\mu^2)^2}{\frac{(MSTR)^2}{a-1} + \frac{(MSE)^2}{a(n-1)}}$$

20.1.2.2 Random Treatment Effects Model

$$\tau_i = \mu_i - E(\mu_i) = \mu_i - \mu$$

we have $\mu_i = \mu + \tau_i$ and

$$Y_{ij} = \mu + \tau_i + \epsilon_{ij}$$

where

- μ = constant, common to all observations
- $\tau_i \sim N(0, \sigma_\tau^2)$ independent (random variables)
- $\epsilon_{ij} \sim N(0, \sigma^2)$ independent.
- τ_i, ϵ_{ij} are independent (i=1,…,a; j =1,..,n)
- our model is concerned with only balanced single factor ANOVA.

Diagnostics Measures

- Non-constant error variance (plots, Levene test, Hartley test).
- Non-independence of errors (plots, Durban-Watson test).
- Outliers (plots, regression methods).
- Non-normality of error terms (plots, Shapiro-Wilk, Anderson-Darling).
- Omitted Variable Bias (plots)

Remedial

- Weighted Least Squares
- Transformations
- Non-parametric Procedures.

Note

- Fixed effect ANOVA is relatively robust to
  - non-normality

  - unequal variances when sample sizes are approximately equal; at least the F-test and multiple comparisons. However, single comparisons of treatment means are sensitive to unequal variances.

- Lack of independence can seriously affect both fixed and random effect ANVOA.

20.1.3 Two Factor Fixed Effect ANOVA

The multi-factor experiment is

- more efficient
- provides more info

- gives more validity to the findings.

20.1.3.1 Balanced

Assumption:

- All treatment sample sizes are equal

- All treatment means are of equal importance

Assume:

- Factor A has `a` levels and Factor B has `b` levels. All a x b factor levels are considered.

- The number of treatments for each level is n. $N = abn$ observations in the study.

20.1.3.1.1 Cell Means Model

$$
Y_{ijk} = \mu_{ij} + \epsilon_{ijk}
$$

where

- μ_{ij} are fixed parameters (cell means)

- $i = 1,...,a$ = the levels of Factor A

- $j = 1,...,b$ = the levels of Factor B.

- $\epsilon_{ijk} \sim$ indep $N(0,\sigma^2)$ for $i = 1,...,a$, $j = 1,..,b$ and $k = 1,..,n$

And

$$E(Y_{ijk}) = \mu_{ij} var(Y_{ijk}) = var(\epsilon_{ijk}) = \sigma^2$$

Hence,

$$Y_{ijk} \sim \text{indep } N(\mu_{ij}, \sigma^2)$$

And the model is

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon$$

Thus,

$$E(\mathbf{Y}) = \mathbf{X}\beta var(\mathbf{Y}) = \sigma^2\mathbf{I}$$

Interaction

$$(\alpha\beta)_{ij} = \mu_{ij} - (\mu_{..} + \alpha_i + \beta_j)$$

where

- $\mu_{..} = \sum_i \sum_j \mu_{ij}/ab$ is the grand mean
- $\alpha_i = \mu_{i.} - \mu_{..}$ is the main effect for factor A at the i-th level
- $\beta_j = \mu_{.j} - \mu_{..}$ is the main effect for factor B at the j-th level
- $(\alpha\beta)_{ij}$ is the interaction effect when factor A is at the i-th level and factor B is at the j-th level.
- $(\alpha\beta)_{ij} = \mu_{ij} - \mu_{i.} - \mu_{.j} + \mu_{..}$

Examine interactions:

- Examine whether all μ_{ij} can be expressed as the sums $\mu_{..} + \alpha_i + \beta_j$
- Examine whether the difference between the mean responses for any two levels of factor B is the same for all levels of factor A.
- Examine whether the difference between the mean response for any two levels of factor A is the same for all levels of factor B
- Examine whether the treatment mean curves for the different factor levels in a treatment plot are parallel.

For $j = 1, ..., b$

$$\begin{aligned}
\sum_i (\alpha\beta)_{ij} &= \sum_i (\mu_{ij} - \mu_{..} - \alpha_i - \beta_j) \\
&= \sum_i \mu_{ij} - a\mu_{..} - \sum_i \alpha_i - a\beta_j \\
&= a\mu_{.j} - a\mu_{..} - \sum_i (\mu_{i.} - \mu_{..}) - a(\mu_{.j} - \mu_{..}) \\
&= a\mu_{.j} - a\mu_{..} - a\mu_{..} + a\mu_{..} - a(\mu_{.j} - \mu_{..}) \\
&= 0
\end{aligned}$$

Similarly, $\sum_j (\alpha\beta) = 0, i = 1, ..., a$ and $\sum_i \sum_j (\alpha\beta)_{ij} = 0$, $\sum_i \alpha_i = 0$, $\sum_j \beta_j = 0$

20.1.3.1.2 Factor Effects Model

$$\mu_{ij} = \mu_{..} + \alpha_i + \beta_j + (\alpha\beta)_{ij} Y_{ijk} = \mu_{..} + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}$$

where

- $\mu_{..}$ is a constant
- α_i are constants subject to the restriction $\sum_i \alpha_i = 0$
- β_j are constants subject to the restriction $\sum_j \beta_j = 0$
- $(\alpha\beta)_{ij}$ are constants subject to the restriction $\sum_i (\alpha\beta)_{ij} = 0$ for $j = 1, ..., b$ and $\sum_j (\alpha\beta)_{ij} = 0$ for $i = 1, ..., a$
- $\epsilon_{ijk} \sim$ indep $N(0, \sigma^2)$ for $k = 1, .., n$

We have

$$E(Y_{ijk}) = \mu_{..} + \alpha_i + \beta_j + (\alpha\beta)_{ij} var(Y_{ijk}) = \sigma^2 Y_{ijk} \sim N(\mu_{..} + \alpha_i + \beta_j + (\alpha\beta)_{ij}, \sigma^2)$$

We have $1 + a + b + ab$ parameters. But there are ab parameters in the Cell Means Model. In the Factor Effects Model, the restrictions limit the number of parameters that can be estimated:

$$1 \text{ for } \mu_{..} (a-1) \text{ for } \alpha_i (b-1) \text{ for } \beta_j (a-1)(b-1) \text{ for } (\alpha\beta)_{ij}$$

Hence, there are

$$1 + a - 1 + b - 1 + ab - a - b + 1 = ab$$

parameters in the model.

We can have several restrictions when considering the model in the form $\mathbf{Y} = \mathbf{X}\beta + \epsilon$

One way:

$$\alpha_a = \alpha_1 - \alpha_2 - ... - \alpha_{a-1}\beta_b = -\beta_1 - \beta_2 - ... - \beta_{b-1}(\alpha\beta)_{ib} = -(\alpha\beta)_{i1} - (\alpha\beta)_{i2} - ... - (\alpha\beta)_{i,b-1}; i = 1, .., a(\alpha\beta)_{aj} = -(\alpha\beta)_{1j} - ($$

We can fit the model by least squares or maximum likelihood

Cell Means Model
minimize

$$Q = \sum_i \sum_j \sum_k (Y_{ijk} - \mu_{ij})^2$$

estimators

$$\hat{\mu}_{ij} = \bar{Y}_{ij}\hat{Y}_{ijk} = \bar{Y}_{ij}e_{ijk} = Y_{ijk} - \hat{Y}_{ijk} = Y_{ijk} - \bar{Y}_{ij}$$

Factor Effects Model

$$Q = \sum_i \sum_j \sum_k (Y_{ijk} - \mu_{..} - \alpha_i = \beta_j - (\alpha\beta)_{ij})^2$$

subject to the restrictions

$$\sum_i \alpha_i = 0 \sum_j \beta_j = 0 \sum_i (\alpha\beta)_{ij} = 0 \sum_j (\alpha\beta)_{ij} = 0$$

estimators

$$\hat{\mu}_{..} = \bar{Y}_{...}\hat{\alpha}_i = \bar{Y}_{i..} - \bar{Y}_{...}\hat{\beta}_j = \bar{Y}_{.j.} - \bar{Y}_{...}(\widehat{\alpha\beta})_{ij} = \bar{Y}_{ij.} - \bar{Y}_{i..} - \bar{Y}_{.j.} + \bar{Y}_{...}$$

The fitted values

$$\hat{Y}_{ijk} = \bar{Y}_{...} + (\bar{Y}_{i..} - \bar{Y}_{...}) + (\bar{Y}_{.j.} - \bar{Y}_{...}) + (\bar{Y}_{ij.} - \bar{Y}_{i..} - \bar{Y}_{.j.} + \bar{Y}_{...}) = \bar{Y}_{ij.}$$

where

$$e_{ijk} = Y_{ijk} - \bar{Y}_{ij.}e_{ijk} \sim \text{ indep } (0, \sigma^2)$$

and

$$s^2_{\hat{\mu}_{..}} = \frac{MSE}{nab} s^2_{\hat{\alpha}_i} = MSE(\frac{1}{nb} - \frac{1}{nab}) s^2_{\hat{\beta}_j} = MSE(\frac{1}{na} - \frac{1}{nab}) s^2_{(\hat{\alpha\beta})_{ij}} = MSE(\frac{1}{n} - \frac{1}{na} - \frac{1}{nb} + \frac{1}{nab})$$

20.1.3.1.2.1 Partitioning the Total Sum of Squares

$$Y_{ijk} - \bar{Y}_{...} = \bar{Y}_{ij.} - \bar{Y}_{...} + Y_{ijk} - \bar{Y}_{ij.}$$

$Y_{ijk} - \bar{Y}_{...}$: Total deviation
$\bar{Y}_{ij.} - \bar{Y}_{...}$: Deviation of treatment mean from overall mean
$Y_{ijk} - \bar{Y}_{ij.}$: Deviation of observation around treatment mean (residual).

$$\sum_i \sum_j \sum_k (Y_{ijk} - \bar{Y}_{...})^2 = n \sum_i \sum_j (\bar{Y}_{ij.} - \bar{Y}_{...})^2 + \sum_i \sum_j sum_k (Y_{ijk} - \bar{ij}.)^2$$
$$SSTO = SSTR + SSE$$

(cross product terms are 0)

$$\bar{Y}_{ij.} - \bar{Y}_{...} = \bar{Y}_{i..} - \bar{Y}_{...} + \bar{Y}_{.j.} - \bar{Y}_{...} + \bar{Y}_{ij.} - \bar{Y}_{i..} - \bar{Y}_{.j.} + \bar{Y}_{...}$$

squaring and summing:

$$n \sum_i \sum_j (\bar{Y}_{ij.} - \bar{Y}_{...})^2 = nb \sum_i (\bar{Y}_{i..} - \bar{Y}_{...})^2 + na \sum_j (\bar{Y}_{.j.} - \bar{Y}_{...})^2 + n \sum_i \sum_j (\bar{Y}_{ij.} - \bar{Y}_{i..} - \bar{Y}_{.j.} + \bar{Y}_{...})^2$$
$$SSTR = SSA + SSB + SSAB$$

The interaction term from

$$SSAB = SSTO - SSE - SSA - SSBSSAB = SSTR - SSA - SSB$$

where

- SSA is the factor A sum of squares (measures the variability of the estimated factor A level means $\bar{Y}_{i..}$)- the more variable, the larger SSA
- SSB is the factor B sum of squares
- SSAB is the interaction sum of squares, measuring the variability of the estimated interactions.

20.1.3.1.2.2 Partitioning the df

$N = abn$ cases and ab treatments.

For one-way ANOVA and regression, the partition has df:

$$SS : SSTO = SSTR + SSEdf : N-1 = (ab-1) + (N-ab)$$

we must further partition the $ab-1$ df with SSTR

$$SSTR = SSA + SSB + SSABab-1 = (a-1)+(b-1)+(a-1)(b-1)$$

- $df_{SSA} = a-1$: a treatment deviations but 1 df is lost due to the restriction $\sum(\bar{Y}_{i..} - \bar{Y}_{...}) = 0$
- $df_{SSB} = b-1$: b treatment deviations but 1 df is lost due to the restriction $\sum(\bar{Y}_{.j.} - \bar{Y}_{...}) = 0$
- $df_{SSAB} = (a-1)(b-1) = (ab-1)-(a-1)-(b-1)$: ab interactions, there are (a+b-1) restrictions, so df = ab-a-(b-1)= (a-1)(b-1)

20.1.3.1.2.3 Mean Squares

$$MSA = \frac{SSA}{a-1} MSB = \frac{SSB}{b-1} MSAB = \frac{SSAB}{(a-1)(b-1)}$$

The expected mean squares are

$$E(MSE) = \sigma^2 E(MSA) = \sigma^2 + nb\frac{\sum \alpha_i^2}{a-1} = \sigma^2 + nb\frac{\sum(\sum_{i.} - \mu_{..})^2}{a-1} E(MSB) = \sigma^2 + na\frac{\sum \beta_i^2}{b-1} = \sigma^2 + na\frac{\sum(\sum_{.j} - \mu_{..})^2}{b-1} E($$

If there are no factor A main effects (all $\mu_{i.} = 0$ or $\alpha_i = 0$) the MSA and MSE have the same expectation; otherwise MSA > MSE. Same for factor B, and interaction effects. which case we can examine F-statistics.

Interaction

$$\begin{aligned} H_0 &: \mu_{ij} - \mu_{i.} - \mu_{.j} + \mu_{..} = 0 \quad \text{for all i,j} \\ H_a &: \mu_{ij} - \mu_{i.} - \mu_{.j} + \mu_{..} \neq 0 \quad \text{for some i,j} \end{aligned}$$

or

$$H_0 : \text{All}(\alpha\beta)_{ij} = 0 H_a : \text{Not all}(\alpha\beta) = 0$$

Let $F = \frac{MSAB}{MSE}$. When H_0 is true $F \sim f_{((a-1)(b-1),ab(n-1))}$. So reject H_0 when $F > f_{((a-1)(b-1),ab(n-1))}$

Factor A main effects:

$$H_0 : \mu_{1.} = \mu_{2.} = ... = \mu_{a.} H_a : \text{Not all } \mu_{i.} \text{ are equal}$$

or

$$H_0 : \alpha_1 = ... = \alpha_a = 0 H_a : \text{Not all } \alpha_i \text{ are equal to 0}$$

$F = \frac{MSA}{MSE}$ and reject H_0 if $F > f_{(1-\alpha;a-1,ab(n-1))}$

20.1.3.1.2.4 Two-way ANOVA

Source of Variation	SS	df	MS	F
Factor A	SSA	a-1	MSA = SSA/(a-1)	MSA/MSE
Factor B	SSB	b-1	MSB = SSB/(b-1)	MSB/MSE
AB interactions	SSAB	(a-1)(b-1)	MSAB = SSAB /MSE	
Error	SSE	ab(n-1)	MSE = SSE/ab(n-1)	
Total (corrected)	SSTO	abn - 1		

Doing 2-way ANOVA means you always check interaction first, because if there are significant interactions, checking the significance of the main effects becomes moot.

The main effects concern the mean responses for levels of one factor averaged over the levels of the other factor. WHen interaction is present, we can't conclude that a given factor has no effect, even if these averages are the same. It means that the effect of the factor depends on the level of the other factor.

On the other hand, if you can establish that there is no interaction, then you can consider inference on the factor main effects, which are then said to be **additive**.
And we can also compare factor means like the Single Factor Fixed Effects Model using Tukey, Scheffe, Bonferroni.

We can also consider contrasts in the 2-way model

$$L = \sum c_i \mu_i$$

where $\sum c_i = 0$
which is estimated by

$$\hat{L} = \sum c_i \bar{Y}_{i..}$$

with variance

$$\sigma^2(\hat{L}) = \frac{\sigma^2}{bn} \sum c_i^2$$

and variance estimate

$$\frac{MSE}{bn} \sum c_i^2$$

Orthogonal Contrasts

$$L_1 = \sum c_i \mu_i, \sum c_i = 0 L_2 = \sum d_i \mu_i, \sum d_i = 0$$

these contrasts are said to be **orthogonal** if

$$\sum \frac{c_i d_i}{n_i} = 0$$

in balanced case $\sum c_i d_i = 0$

$$\begin{aligned}
cov(\hat{L}_1, \hat{L}_2) &= cov(\sum_i c_i \bar{Y}_{i..}, \sum_l d_l \bar{Y}_{l..}) \\
&= \sum_i \sum_l c_i d_l cov(\bar{Y}_{i..}, \bar{Y}_{l..}) \\
&= \sum_i c_i d_i \frac{\sigma^2}{bn} = 0
\end{aligned}$$

Orthogonal contrasts can be used to further partition the model sum of squares. There are many sets of orthogonal contrasts and thus, many ways to partition the sum of squares.

A special set of orthogonal contrasts that are used when the levels of a factor can be assigned values on a metric scale are called **orthogonal polynomials**

Coefficients can be found for the special case of

- equal spaced levels (e.g., (0 15 30 45 60))
- equal sample sizes ($n_1 = n_2 = ... = n_{ab}$)

We can define the SS for a given contrast:

$$SS_L = \frac{\hat{L}^2}{\sum_{i=1}^a (c_i^2/bn_i)}$$

$$T = \frac{\hat{L}}{\sqrt{MSE \sum_{i=1}^a (c_i^2/bn_i)}} \sim t$$

Moreover,

$$t^2_{(1-\alpha/2;df)} = F_{(1-\alpha;1,df)}$$

So,

$$\frac{SS_L}{MSE} \sim F_{(1-\alpha;1,df_{MSE})}$$

all contrasts have d.f = 1

20.1.3.2 Unbalanced

We could have unequal numbers of replications for all treatment combinations:

- observational studies
- dropouts in designed studies
- larger sample sizes for inexpensive treatments
- Sample sizes to match population makeup.

Assume that each factor combination has at least 1 observation (no empty cells)

Consider the same model as:

$$Y_{ijk} = \mu_{..} + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}$$

where sample sizes are: n_{ij}:

$$n_{i.} = \sum_j n_{ij} n_{.j} = \sum_i n_{ij} n_T = \sum_i \sum_j n_{ij}$$

Problem here is that

$$SSTO \neq SSA + SSB + SSAB + SSE$$

(the design is **non-orthogonal**)

- For $i = 1, ..., a-1$,
  \begin{equation} u_i = \begin{cases} +1 & \text{if the obs is from the i-th level of Factor 1} \\ -1 & \text{if the obs is from the a-th level of Factor 1} \\ 0 & \text{otherwise} \\ \end{cases} \end{equation}
- For $j = 1, ..., b-1$
  \begin{equation} v_i = \begin{cases} +1 & \text{if the obs is from the j-th level of Factor 1} \\ -1 & \text{if the obs is from the b-th level of Factor 1} \\ 0 & \text{otherwise} \\ \end{cases} \end{equation}

We can use these indicator variables as predictor variables and $\mu_{..}, \alpha_i, \beta_j, (\alpha\beta)_{ij}$ as unknown parameters.

$$Y = \mu_{..} + \sum_{i=1}^{a-1} \alpha_i u_i + \sum_{j=1}^{b-1} \beta_j v_j + \sum_{i=1}^{a-1} \sum_{j=1}^{b-1} (\alpha\beta)_{ij} u_i v_j + \epsilon$$

To test hypotheses, we use the extra sum of squares idea.

For interaction effects

$$H_0 : all (\alpha\beta)_{ij} = 0 H_a : \text{not all } (\alpha\beta)_{ij} = 0$$

Or to test

$$H_0 : \beta_1 = \beta_2 = \beta_3 = 0 H_a : \text{not all } \beta_j = 0$$

Analysis of Factor Means

(e.g., contrasts) is analogous to the balanced case, with modifications in the formulas for means and standard errors to account for unequal sample sizes.

Or , we can fit the cell means model and consider it from a regression perspective

If you have empty cells (i.e., some factor combinations have no observation), then the equivalent regression approach can't be used. But you can still do partial analyses

20.1.4 Two-Way Random Effects ANOVA

$$Y_{ijk} = \mu_{..} + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ij}$$

where

- $\mu_{..}$: constant
- $\alpha_i \sim N(0, \sigma^2_\alpha), i = 1, .., a$ (independent)
- $\beta_j \sim N(0, \sigma^2_\beta), j = 1, .., b$ (independent)
- $(\alpha\beta)_{ij} \sim N(0, \sigma^2_{\alpha\beta}), i = 1, ..., a, j = 1, .., b$ (independent)
- $\epsilon_{ijk} \sim N(0, \sigma^2)$ (independent)

All $\alpha_i, \beta_j, (\alpha\beta)_{ij}$ are pairwise independent

Theoretical means, variances, and covariances are

$$E(Y_{ijk}) = \mu_{..} var(Y_{ijk}) = \sigma^2_Y = \sigma^2_\alpha + \sigma^2_\beta + \sigma^2_{\alpha\beta} + \sigma^2$$

So

$Y_{ijk} \sim N(\mu_{..}, \sigma^2_\alpha + \sigma^2_\beta + \sigma^2_{\alpha\beta} + \sigma^2)$

$cov(Y_{ijk}, Y_{ij'k'}) = \sigma^2_\alpha, j \neq j' cov(Y_{ijk}, Y_{i'jk'}) = \sigma^2_\beta, i \neq i' cov(Y_{ijk}, Y_{ijk'}) = \sigma^2_\alpha + \sigma^2_\beta + \sigma^2_{\alpha\beta}, k \neq k' cov(Y_{ijk}, Y_{i'}$

20.1.5 Two-Way Mixed Effects ANOVA

20.1.5.1 Balanced

One fixed factor, while other is random treatment levels, we have a **mixed effects model** or a **mixed model**

Restricted mixed model for 2-way ANOVA:

$$Y_{ijk} = \mu_{..} + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}$$

where

- $\mu_{..}$: constant
- α_i: fixed effects with constraints subject to restriction $\sum \alpha_i = 0$
- $\beta_j \sim indepN(0, \sigma^2_\beta)$
- $(\alpha\beta)_{ij} \sim N(0, \frac{a-1}{a}\sigma^2_{\alpha\beta})$ subject to restriction $\sum_i (\alpha\beta)_{ij} = 0$ for all j, the variance here is written as the proportion for convenience; it makes the expected mean squares simpler (other assumed $var((\alpha\beta)_{ij} = \sigma^2_{\alpha\beta})$
- $cov((\alpha\beta)_{ij}, (\alpha\beta)_{i'j'}) = -\frac{1}{a}\sigma^2_{\alpha\beta}, i \neq i'$

- $\epsilon_{ijk} \sim indepN(0, \sigma^2)$
- $\beta_j, (\alpha\beta)_{ij}, \epsilon_{ijk}$ are pairwise independent

Two-way mixed models are written in an "unrestricted" form, with no restrictions on the interaction effects $(\alpha\beta)_{ij}$, they are pairwise independent. Let $\beta^*, (\alpha\beta)^*_{ij}$ be the unrestricted random effects, and $(\bar{\alpha\beta})^*_{ij}$ the means averaged over the fixed factor for each level of random factor B.

$$\beta_j = \beta^*_j + (\bar{\alpha\beta})^*_{ij}(\alpha\beta)_{ij} = (\alpha\beta)^*_{ij} - (\bar{\alpha\beta})^*_{ij}$$

Some consider the restricted model to be more general. but here we consider the restricted form.

$$E(Y_{ijk}) = \mu_{..} + \alpha_i var(Y_{ijk}) = \sigma^2_\beta + \frac{a-1}{a}\sigma^2_{\alpha\beta} + \sigma^2$$

Responses from the same random factor (B) level are correlated

$$cov(Y_{ijk}, Y_{ijk'}) = E(Y_{ijk}Y_{ijk'}) - E(Y_{ijk})E(Y_{ijk'}) = \sigma^2_\beta + \frac{a-1}{a}\sigma^2_{\alpha\beta}, k \neq k'$$

Similarly,

$$cov(Y_{ijk}, Y_{i'jk'}) = \sigma^2_\beta - \frac{1}{a}\sigma^2_{\alpha\ \beta}, i \neq i' cov(Y_{ijk}, Y_{i'j'k'}) = 0, j \neq j'$$

Hence, you can see that the only way you don't have dependence in the Y is when they don't share the same random effect.

An advantage of the **restricted mixed model** is that 2 observations from the same random factor b level can be positively or negatively correlated. In the **unrestricted model**, they can only be positively correlated.

Mean Square	Fixed ANOVA (A, B Fixed)	Random ANOVA (A,B random)	Mixed ANVOA (A fixed, B random)
MSA	a - 1	$\sigma^2 + nb\frac{\sum \alpha_i^2}{a-1}$	$\sigma^2 + nb\sigma^2_\alpha + n\sigma^2_{\alpha\beta}$
MSB	b-1	$\sigma^2 + na\frac{\sum \beta_j^2}{b-1}$	$\sigma^2 + na\sigma^2_\beta + n\sigma^2_{\alpha\beta}$
MSAB	(a-1)(b-1)	$\sigma^2 + n\frac{\sum\sum(\alpha\beta)^2_{ij}}{(a-1)(b-1)}$	$\sigma^2 + n\sigma^2_{\alpha\beta}$
MSE	(n-1)ab	σ^2	σ^2

For fixed, random, and mixed models (balanced), the ANOVA table sums of squares calculations are identical. (also true for df and mean squares). The only difference is with the expected mean squares, thus the test statistics.

In Random ANOVA, we test

$$H_0 : \sigma^2 = 0 H_a : \sigma^2 > 0$$

by considering $F = \frac{MSA}{MSAB} \sim F_{a-1;(a-1)(b-1)}$

The same test statistic is used for mixed models, but in that case we are testing null hypothesis that all of the $\alpha_i = 0$

The test statistic different for the same null hypothesis under the fixed effects model.

Test for effects of	Fixed ANOVA (A&B fixed)	Random ANOVA (A&B random)	Mixed ANOVA (A fixed, B random)
Factor A	$\frac{MSA}{MSE}$	$\frac{MSA}{MSAB}$	$\frac{MSA}{MSAB}$
Factor B	$\frac{MSB}{MSE}$	$\frac{MSB}{MSAB}$	$\frac{MSB}{MSE}$
AB interactions	$\frac{MSAB}{MSE}$	$\frac{MSAB}{MSE}$	$\frac{MSAB}{MSE}$

Estimation Of Variance Components

In random and mixed effects models, we are interested in estimating the **variance components**
Variance component σ^2_β in the mixed ANOVA.

$$E(\sigma^2_\beta) = \frac{E(MSB) - E(MSE)}{na} = \frac{\sigma^2 + na\sigma^2_\beta - \sigma^2}{na} = \sigma^2_\beta$$

which can be estimated with

$$\hat{\sigma}^2_\beta = \frac{MSB - MSE}{na}$$

Confidence intervals for variance components can be constructed (approximately) by using the **Satterthwaite** procedure or the MLS procedure (like the 1-way random effects)

Estimation of Fixed Effects in Mixed Models

$$\hat{\alpha}_i = \bar{Y}_{i..} - \bar{Y}_{...} \hat{\mu}_{i.} = \bar{Y}_{...} + (\bar{Y}_{i..} - \bar{Y}_{...}) = \bar{Y}_{i..} \sigma^2(\hat{\alpha}_i) = \frac{\sigma^2 + n\sigma^2_{\alpha\beta}}{bn} = \frac{E(MSAB)}{bn} s^2(\hat{\alpha}_i) = \frac{MSAB}{bn}$$

Contrasts on the **Fixed Effects**

$$L = \sum c_i \alpha_i \sum c_i = 0 \hat{L} = \sum c_i \hat{\alpha}_i \sigma^2(\hat{L}) = \sum c_i^2 \sigma^2(\hat{\alpha}_i) s^2(\hat{L}) = \frac{MSAB}{bn} \sum c_i^2$$

Confidence intervals and tests can be constructed as usual

20.1.5.2 Unbalanced

For a mixed model with a = 2, b = 4

$$Y_{ijk} = \mu_{..} + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk} var(\beta_j) = \sigma^2_\beta var((\alpha\beta)_{ij}) = \frac{2-1}{2}\sigma^2_{\alpha\beta} = \frac{\sigma^2_{\alpha\beta}}{2} var(\epsilon_{ijk}) = \sigma^2 E(Y_{ijk}) = \mu_{..} + \alpha_i var(Y_{ijk}) =$$

assume

$$\mathbf{Y} \sim N(\mathbf{X}\beta, M)$$

where M is block diagonal

density function

$$f(\mathbf{Y}) = \frac{1}{(2\pi)^{N/2}|M|^{1/2}} exp(-\frac{1}{2}(\mathbf{Y} - \mathbf{X}\ )'\mathbf{M^{-1}}(\mathbf{Y} - \mathbf{X}\ ))$$

if we knew the variance components, we could use GLS:

$$\hat{\beta}_{GLS} = (\mathbf{X'M^{-1}X})^{-1}\mathbf{X'M^{-1}Y}$$

but we usually don't know the variance components $\sigma^2, \sigma^2_\beta, \sigma^2_{\alpha\beta}$ that make up M
Another way to get estimates is by **Maximum likelihood estimation**

we try to maximize its log

$$\ln L = -\frac{N}{2}\ln(2\pi) - \frac{1}{2}\ln|M| - \frac{1}{2}(\mathbf{Y} - \mathbf{X}\ )'\ ^{\mathbf{-1}}(\mathbf{Y} - \mathbf{X}\ )$$

20.2 Nonparametric ANOVA

20.2.1 Kruskal-Wallis

Generalization of independent samples Wilcoxon Rank sum test for 2 independent samples (like F-test of one-way ANOVA is a generalization to several independent maples of the two sample t-test)

Consider the one-way case:

We have

- $a \geq 2$ treatments
- n_i is the sample size for the ith treatment
- Y_{ij} is the j-th observation from the ith treatment.
- we make **no** assumption of normality
- We only assume that observations on the ith treatment are a random sample from the continuous CDF F_i, i = 1,..,n, and are mutually independent.

$$H_0 : F_1 = F_2 = ... = F_a H_a : F_i < F_j \text{ for some } i \neq j$$

or if distribution is from the location-scale family, $H_0 : \theta_1 = \theta_2 = ... = \theta_a$)

Procedure

- Rank all $N = \sum_{i=1}^a n_i$ observations in ascending order. Let $r_{ij} = rank(Y_{ij})$, note $\sum_i \sum_j r_{ij} = 1 + 2.. + N = \frac{N(N+1)}{2}$
- Calculate the rank sums and averages:

  $$r_{i.} = \sum_{j=1}^{n_i} r_{ij}$$

  and

  $$\bar{r}_{i.} = \frac{r_{i.}}{n_i}, i = 1, .., a$$
- Calculate the test statistic on the ranks:

  $$\chi^2_{KW} = \frac{SSTR}{\frac{SSTO}{N-1}}$$

  where $SSTR = \sum n_i(\bar{r}_{i.} - \bar{r}_{..})^2$ and $SSTO = \sum \sum (\bar{r}_{ij} - \bar{r}_{..})^2$
- FOr large n_i ($\geq$ 5 observations) the Kruskal-Wallis statistic is approximated by a χ^2_{a-1} distribution when all the treatment means are equal. Hence, reject H_0 if $\chi^2_{KW} > \chi^2_{(1-\alpha;a-1)}$.

- If sample sizes are small, one can exhaustively work out all possible distinct ways of assigning N ranks to the observations from a treatments and calculate the value of the KW statistic in each case ($\frac{N!}{n_1!..n_a!}$ possible combinations). Under H_0 all of these assignments are equally likely.

20.2.2 Friedman Test

When the responses $Y_{ij} = 1,..,n, j = 1,..,r$ in a randomized complete block design are not normally distributed (or do not have constant variance), a nonparametric test is more helpful.

A distribution-free rank-based test for comparing the treatments in this setting is the Friedman test. Let F_{ij} be the CDF of random Y_{ij}, corresponding to the observed value y_{ij}

Under the null hypothesis, F_{ij} are identical for all treatments j separately for each block i.

$$H_0 : F_{i1} = F_{i2} = ... = F_{ir} \text{ for all i} H_a : F_{ij} < F_{ij'} \text{ for some } j \neq j' \text{ for all } i$$

For location parameter distributions, treatment effects can be tested:

$$H_0 : \tau_1 = \tau_2 = ... = \tau_r H_a : \tau_j > \tau_{j'} \text{ for some } j \neq j'$$

Procedure

- Rank observations from the r treatments separately within each block (in ascending order; if ties, each tied observation is given the mean of ranks involved). Let the ranks be called r_{ij}

- Calculate the Friedman test statistic

$$\chi^2_F = \frac{SSTR}{\frac{SSTR+SSE}{n(r-1)}}$$

where

$$SSTR = n\sum(\bar{r}_{.j} - \bar{r}_{..})^2 SSE = \sum\sum(r_{ij} - \bar{r}_{.j})^2 \bar{r}_{.j} = \frac{\sum_i r_{ij}}{n} \bar{r}_{..} = \frac{r+1}{2}$$

If there is no ties, it can be rewritten as

$$\chi^2_F = [\frac{12}{nr(n+1)} \sum_j r^2_{.j}] - 3n(r+1)$$

with large number of blocks, χ^2_F is approximately χ^2_{r-1} under H_0. Hence, we reject H_0 if $\chi^2_F > \chi^2_{(1-\alpha;r-1)}$

The exact null distribution for χ_F^2 can be derived since there are r! possible ways of assigning ranks 1,2,…,r to the r observations within each block. There are n blocks and thus $(r!)^n$ possible assignments to the ranks, which are equally likely when H_0 is true.

20.3 Sample Size Planning for ANOVA

20.3.1 Balanced Designs

20.3.1.1 Single Factor Studies

20.3.1.1.1 Fixed cell means

$$P(F > f_{(1-\alpha;a-1,N-a)}|\phi) = 1 - \beta$$

where ϕ is the **noncentrality parameter** (measures how unequal the treatment means μ_i are)

$$\phi = \frac{1}{\sigma}\sqrt{\frac{n}{a}\sum_i (\mu_i - \mu_.)^2}, (n_i \equiv n)$$

and

$$\mu_. = \frac{\sum \mu_i}{a}$$

To decide on the power probabilities we use the noncetral F distribution.

We could use the power table directly when effects are fixed and design is balanced by using **minimum range** of factor level means for your desired differences

$$\Delta = \max(\mu_i) - \min(\mu_i)$$

Hence, we need

- α level
- Δ
- σ
- β

Notes:

- When Δ/σ is small greatly affects sample size, but if Δ/σ is large.
- Reducing α or β increases the required sample sizes.
- Error in estimating σ can make a large difference.

20.3.1.2 Multi-factor Studies

The same noncentral F tables can be used here

For two-factor fixed effect model

Test for interactions:

$$\phi = \frac{1}{\sigma}\sqrt{\frac{n\sum\sum(\alpha\beta_{ij})^2}{(a-1)(b-1)+1}} = \frac{1}{\sigma}\sqrt{\frac{n\sum\sum(\mu_{ij}-\mu_{i.}-\mu_{.j}+\mu_{..})^2}{(a-1)(b-1)+1}}\upsilon_1 = (a–1)(b–1)\upsilon_2 = ab(n–1)$$

Test for Factor A main effects:

$$\phi = \frac{1}{\sigma}\sqrt{\frac{nb\sum\alpha_i^2}{a}} = \frac{1}{\sigma}\sqrt{\frac{nb\sum(\mu_{i.}-\mu_{..})^2}{a}}\upsilon_1 = a-1\upsilon_2 = ab(n-1)$$

Test for Factor B main effects:

$$\phi = \frac{1}{\sigma}\sqrt{\frac{na\sum\beta_j^2}{b}} = \frac{1}{\sigma}\sqrt{\frac{na\sum(\mu_{.j}-\mu_{..})^2}{b}}\upsilon_1 = b-1\upsilon_2 = ab(n-1)$$

Procedure:

1. Specify the minimu range of Factor A means
2. Obtain sample sizes with r = a. The resulting sample size is bn, from which n can be obtained.
3. Repeat the first 2 steps for Factor B minimum range.
4. Choose the greater number of sample size between A and B.

20.3.2 Randomized Block Experiments

Analogous to completely randomized designs . The power of the F-test for treatment effects for randomized block design uses the same non-centrality parameter as completely randomized design:

$$\phi = \frac{1}{\sigma}\sqrt{\frac{n}{r}\sum(\mu_i-\mu_.)^2}$$

However, the power level is different from the randomized block design because

- error variance σ^2 is different
- df(MSE) is different.

20.4 Randomized Block Designs

To improve the precision of treatment comparisons, we can reduce variability among the experimental units. We can group experimental units into **blocks** so that each block contains relatively homogeneous units.

- Within each block, random assignment treatments to units (separate random assignment for each block)
- The number of units per block is a multiple of the number of factor combinations.
- Commonly, use each treatment once in each block.

Benefits of **Blocking**

- Reduction in variability of estimators for treatment means
  - Improved power for t-tests and F-tests
  - Narrower confidence intervals
  - Smaller MSE
- Compare treatments under different conditions (related to different blocks).

Loss from **Blocking** (little to lose)

- If you don't do blocking well, you waste df on negligible block effects that could have been used to estimate σ^2
- hence, the df for t-tests and denominator df for F-tests will be reduced without reducing MSE and small loss of power for both tests.

Consider

$$Y_{ij} = \mu_{..} + \rho_i + \tau_j + \epsilon_{ij} i = 1,2,...,n j = 1,2,..,r$$

where

- $\mu_{..}$: overall mean response, averaging across all blocks and treatments
- ρ_i: block effect, average difference in response for i-th block ($\sum \rho_i = 0$)
- τ_j treatment effect, average across blocks ($\sum \tau_j = 0$)
- $\epsilon_{ij} \sim iidN(0,\sigma^2)$: random experimental error.

Here, we assume that the block and treatment effects are additive. The difference in average response for any pair of treatments i the same **within** each block

$$(\mu_{..} + \rho_i + \tau_j) - (\mu_{..} + \rho_i + \tau_j') = \tau_j - \tau_j'$$

for all $i = 1,..,n$ blocks

$$\hat{\mu} = \bar{Y}_{..} \hat{\rho}_i = \bar{Y}_{i.} - \bar{Y}_{..} \hat{\tau}_j = \bar{Y}_{.j} - \bar{Y}_{..}$$

Hence,

$$\hat{Y}_{ij} = \bar{Y}_{..} + (\bar{Y}_{i.} - \bar{Y}_{..}) + (\bar{Y}_{.j} - \bar{Y}_{..}) = \bar{Y}_{i.} + \bar{Y}_{.j} - \bar{Y}_{..} e_{ij} = Y_{ij} - \hat{Y}_{ij} = Y_{ij} - \bar{Y}_{i.} - \bar{Y}_{.j} + \bar{Y}_{..}$$

ANOVA table

Source of Variation	SS	df	Fixed Treatments E(MS)	Random Treatments E(MS)
Blocks	$r\sum_i (\bar{Y}_{i.} - \bar{Y}_{..})^2$	n - 1	$\sigma^2 + r\frac{\sum \rho_i^2}{n-1}$	$\sigma^2 + r\frac{\sum \rho_i^2}{n-1}$
Treatments	$n\sum_j (\bar{Y}_{.j} - \bar{Y}_{..})^2$	r - 1	$\sigma^2 + n\frac{\sum \tau_j^2}{r-1}$	$\sigma^2 + n\sigma_\tau^2$
Error	$\sum_i \sum_j (Y_{ij} - \bar{Y}_{i.} - \bar{Y}_{.j} + \bar{Y}_{..})^2$	(n-1)(r-1)	σ^2	σ^2
Total	SSTO	nr-1		

F-tests

$$\begin{aligned} H_0 &: \tau_1 = \tau_2 = ... = \tau_r = 0 \quad \text{Fixed Treatment Effects} \\ H_a &: \text{not all } \tau_j = 0 \end{aligned}$$

$$\begin{aligned} H_0 &: \sigma_\tau^2 = 0 \quad \text{Random Treatment Effects} \\ H_a &: \sigma_\tau^2 \neq 0 \end{aligned}$$

In both cases $F = \frac{MSTR}{MSE}$, reject H_0 if $F > f_{(1-\alpha;r-1,(n-1)(r-1))}$

we don't use F-test to compare blocks, because

- We have a priori that blocs are different
- Randomization is done "within" block.

To estimate the efficiency that was gained by blocking (relative to completely randomized design).

$$
\begin{aligned}
\hat{\sigma}^2_{CR} &= \frac{(n-1)MSBL + n(r-1)MSE}{nr-1} \\
\hat{\sigma}^2_{RB} &= MSE \\
\frac{\hat{\sigma}^2_{CR}}{\hat{\sigma}^2_{RB}} &= \text{above 1}
\end{aligned}
$$

then a completely randomized experiment would

$$
(\frac{\hat{\sigma}^2_{CR}}{\hat{\sigma}^2_{RB}} - 1)\%
$$

more observations than the randomized block design to get the same MSE

If batches are randomly selected then they are random effects. That is , if the experiment was repeated, a new sample of i batches would be selected,d yielding new values for $\rho_1, \rho_2, ..., \rho_i$ then.

$$
\rho_1, \rho_2, ..., \rho_j \sim N(0, \sigma^2_\rho)
$$

Then,

$$
Y_{ij} = \mu_{..} + \rho_i + \tau_j + \epsilon_{ij}
$$

where

- $\mu_{..}$ fixed
- ρ_i: random iid $N(0, \sigma^2_p)$
- τ_j fixed (or random) $\sum \tau_j = 0$
- $\epsilon_{ij} \sim iid N(0, \sigma^2)$

**Fixed Treatment&&

$E(Y_{ij}) = \mu_{..} + \tau_j var(Y_{ij}) = \sigma^2_\rho + \sigma^2 cov(Y_{ij}, Y_{ij'}) = \sigma^2, j \neq j'$ treatments within same block are correlated

Correlation between 2 observations in teh same block

$$
\frac{\sigma^2_\rho}{\sigma^2 + \sigma^2_\rho}
$$

The expected MS for the additive fixed treatment effect, random block effect is

Source	SS	E(MS)
Blocks	SSBL	$\sigma^2 + r\sigma_\rho^2$
Treatment	SSTR	$\sigma^2 + n\frac{\sum \tau_j^2}{r-1}$
Error	SSE	σ^2

Interactions and Blocks
without replications within each block for each treatment, we can't consider interaction between block and treatment when the block effect is fixed. Hence, only in the random block effect, we have

$$Y_{ij} = \mu_{..} + \rho_i + \tau_j + (\rho\tau)_{ij} + \epsilon_{ij}$$

where

- $\mu_{..}$ constant
- $\rho_i \sim iddN(0, \sigma_\rho^2)$ random
- τ_j fixed ($\sum \tau_j = 0$)
- $(\rho\tau)_{ij} \sim N(0, \frac{r-1}{r}\sigma_{\rho\tau}^2)$ with $\sum_j (\rho\tau)_{ij} = 0$ for all i
- $cov((\rho\tau)_{ij}, (\rho\tau)_{ij'}) = -\frac{1}{r}\sigma_{\rho\tau}^2$ for $j \neq j'$
- $\epsilon_{ij} \sim iidN(0, \sigma^2)$ random

Note: a special case of mixed 2-factor model with 1 observation per "cell"

$$E(Y_{ij}) = \mu_{..} + \tau_j var(Y_{ij}) = \sigma_\rho^2 + \frac{r-1}{r}\sigma_{\rho\tau}^2 + \sigma^2 cov(Y_{ij}, Y_{ij'}) = \sigma_\rho^2 - \frac{1}{r}\sigma_{\rho\tau}^2, j \neq j' \text{ obs from the same block are correlatedc}$$

The sum of squares and degrees of freedom for interaction model are the same as those for the additive model. The difference exists only with the expected mean squares

Source	SS	df	E(MS)
Blocks	SSBL	n-1	$\sigma^2 + r\sigma_\rho^2$
Treatment	SSTR	r -1	$\sigma^2 + \sigma_{\rho\tau}^2 + n\frac{\sum \tau_j^2}{r-1}$
Error	SSE	(n-1)(r-1)	$\sigma^2 + \sigma_{\rho\tau}^2$

- No exact test is possible for block effects when interaction is present (Not important if blocks are used primarily to reduce experimental error variability)

- $E(MSE) = \sigma^2 + \sigma^2_{\rho\tau}$ the error term variance and interaction variance $\sigma^2_{\rho\tau}$. We can't estimate these components separately with this model. The two are **confounded**.
- If more than 1 observation per treatment block combination, one can consider interaction with fixed block effects, which is called **generalized randomized block designs** (multifactor analysis).

20.4.1 Tukey Test of Additivity

(Tukey's 1 df test for additivity)

formal test of interaction effects between blocks and treatments for a randomized block design. can also considered for testing additivity in 2-way analyses when there is only one observation per cell.

we consider a less restricted interaction term

$$(\rho\tau)_{ij} = D\rho_i\tau_j (\text{D: Constant})$$

So,

$$Y_{ij} = \mu_{..} + \rho_i + \tau_j + D\rho_i\tau_j + \epsilon_{ij}$$

the least square estimate or MLE for D

$$\hat{D} = \frac{\sum_i \sum_j \rho_i \tau_j Y_{ij}}{\sum_i \rho_i^2 \sum_j \tau_j^2}$$

replacing the parameters by their estimates

$$\hat{D} = \frac{\sum_i \sum_j (\bar{Y}_{i.} - \bar{Y}_{..})(\bar{Y}_{.j} - \bar{Y}_{..})Y_{ij}}{\sum_i (\bar{Y}_{i.} - \bar{Y}_{..})^2 \sum_j (\bar{Y}_{.j} - \bar{Y}_{..})^2}$$

Thus, the interaction sum of squares

$$SSint = \sum_i \sum_j \hat{D}^2 (\bar{Y}_{i.} - \bar{Y}_{..})^2 (\bar{Y}_{.j} - \bar{Y}_{..})^2$$

The ANOVA decomposition

$$SSTO = SSBL + SSTR + SSint + SSRem$$

where $SSRem$: remainder sum of squares

$$SSRem = SSTO - SSBL - SSTR - SSint$$

if D = 0 (i.e., no interactions of the type $D\rho_i\tau_j$). SSint and SSRem are independent $\chi^2_{1,rn-r-n}$.

If D = 0,

$$F = \frac{SSint/1}{SSRem/(rn-r-n)} \sim f_{(1-\alpha;rn-r-n)}$$

if

$H_0 : D = 0$ no interaction present$H_a : D \neq 0$ interaction of form $D\rho_i\tau_j$ present

we reject H_0 if $F > f_{(1-\alpha;1,nr-r-n)}$

20.5 Nested Designs

Let μ_{ij} be the mean response when factor A is at the i-th level and factor B is at the j-th level.
If the factors are crossed, the jth level of B is the same for all levels of A.
If factor B is nested within A, the j-th level of B when A is at level 1 has nothing in common with the j-th level of B when A is at level 2.

Factors that can't be manipulated are designated as **classification factors**, as opposed to **experimental factors** (i.e., you assign to the experimental units).

20.5.1 Two-Factor Nested Designs

- Consider B is nested within A.
- both factors are fixed
- All treatment means are equally important.

Mean responses

$$\mu_{i.} = \sum_j \mu_{ij}/b$$

Main effect factor A

$$\alpha_i = \mu_{i.} - \mu_{..}$$

where $\mu_{..} = \frac{\mu_{ij}}{ab} = \frac{\sum_i \mu_{i.}}{a}$ and $\sum_i \alpha_i = 0$

Individual effects of B is denoted as $\beta_{j(i)}$ where $j(i)$ indicates the j-th level of factor B is nested within the it-h level of factor A

$$\beta_{j(i)} = \mu_{ij} - \mu_{i.} = \mu_{ij} - \alpha_i - \mu_{..} \sum_j \beta_{j(i)} = 0, i = 1, ..., a$$

$\beta_{j(i)}$ is the **specific effect** of the jth level of factor B nested within the ith level of factor A. Hence,

$$\mu_{ij} \equiv \mu_{..} + \alpha_i + \beta_{j(i)} \equiv \mu_{..} + (\mu_{i.} - \mu_{..}) + (\mu_{ij} - \mu_{i.})$$

Model

$$Y_{ijk} = \mu_{..} + \alpha_i + \beta_{j(i)} + \epsilon_{ijk}$$

where

- Y_{ijk} response for the kth treatment when factor A is at the i-th level and factor B is at hte jth level (i = 1,..,a; j = 1,..,b; k = 1,..n)
- $\mu_{..}$ constant
- α_i constants subject to restriction $\sum_i \alpha_i = 0$
- $\beta_{j(i)}$ constants subject to restriction $\sum_j \beta_{j(i)} = 0$ for all i
- $\epsilon_{ijk} \sim iidN(0, \sigma^2)$

$$E(Y_{ijk}) = \mu_{..} + \alpha_i + \beta_{j(i)} var(Y_{ijk}) = \sigma^2$$

there is no interaction term in a nested model

ANOVA for Two-Factor Nested Designs

Least Squares and MLE estimates

Parameter	Estimator
$\mu_{..}$	$\bar{Y}_{...}$
α_i	$\bar{Y}_{i..} - \bar{Y}_{...}$
$\beta_{j(i)}$	$\bar{Y}_{ij.} - \bar{Y}_{i..}$
$\hat{Y}_{ijk}$	$\bar{Y}_{ij.}$

residual $e_{ijk} = Y_{ijk} - \bar{Y}_{ijk}$

$$SSTO = SSA + SSB(A) + SSE$$
$$\sum_i \sum_j \sum_k (Y_{ijk} - \bar{Y}_{...})^2 = bn \sum_i (\bar{Y}_{i..} - \bar{Y}_{...})^2 + n \sum_i \sum_j (\bar{Y}_{ij.} - \bar{Y}_{i..})^2 + \sum_i \sum_j \sum_k (Y_{ijk} - \bar{Y}_{ij.})^2$$

ANOVA Table

Source of				
Variation	SS	df	MS	E(MS)
Factor A	SSA	a-1	MSA	$\sigma^2 + bn\frac{\sum \alpha_i^2}{a-1}$
Factor B	SSB(A)	a(b-1)	MSB(A)	$\sigma^2 + n\frac{\vert\vert\vert\vert um \sum eta^2_{i)}}{a(b-1)}$
Error	SSE	ab(n-1)	MSE	σ^2
Total	SSTO	abn -1		

Tests For Factor Effects

$$H_0: \text{ All } \alpha_i = 0 H_a: \text{ not all } \alpha_i = 0$$

$F = \frac{MSA}{MSE} \sim f_{(1-\alpha;a-1,(n-1)ab)}$ reject if $F > f$

$$H_0: \text{ All } \beta_{j(i)} = 0 H_a: \text{ not all } \beta_{j(i)} = 0$$

$F = \frac{MSB(A)}{MSE} \sim f_{(1-\alpha;a(b-1),(n-1)ab)}$ reject $F > f$

Testing Factor Effect Contrasts

$L = \sum c_i \mu_i$ where $\sum c_i = 0$

$$\hat{L} = \sum c_i \bar{Y}_{i..} \hat{L} \pm t_{(1-\alpha/2;df)} s(\hat{L})$$

where $s^2(\hat{L}) = \sum c_i^2 s^2(\bar{Y}_{i..})$, where $s^2(\bar{Y}_{i..}) = \frac{MSE}{bn}, df = ab(n-1)$

Testing Treatment Means

$L = \sum c_i \mu_{.j}$ estimated by $\hat{L} = \sum c_i \bar{Y}_{ij}$ with confidence limits:

$$\hat{L} \pm t_{(1-\alpha/2;(n-1)ab)} s(\hat{L})$$

where

$$s^2(\hat{L}) = \frac{MSE}{n} \sum c_i^2$$

Unbalanced Nested Two-Factor Designs

If there are different number of levels of factor B for different levels of factor A, then the design is called **unbalanced**

The model

$$Y_{ijk} = \mu_{..} + \alpha_i + \beta_{j(i)} + \epsilon_{ijk} i = 1, 2; j = 1, .., b_i; k = 1, .., n_{ij} b_1 = 3, b_2 = 2, n_{11} = n_{13} = 2, n_{12} = 1, n_{21} = n_{22}$$

where $\alpha_1, \beta_{1(1)}, \beta_{2(1)}, \beta_{1(2)}$ are parameters. And constraints: $\alpha_2 = -\alpha_1, \beta_{3(1)} = -\beta_{1(1)} - \beta_{2(1)}, \beta_{2(2)} = -\beta_{1(2)}$

4 indicator variables

$$X_1 = \begin{cases} 1 & \text{if obs from school 1} \\ -1 & \text{if obs from school 2} \end{cases} \tag{20.4}$$

$$X_2 = \begin{cases} 1 & \text{if obs from instructor 1 in school 1} \\ -1 & \text{if obs from instructor 3 in school 1} \\ 0 & \text{otherwise} \end{cases} \tag{20.5}$$

$$X_3 = \begin{cases} 1 & \text{if obs from instructor 2 in school 1} \\ -1 & \text{if obs from instructor 3 in school 1} \\ 0 & \text{otherwise} \end{cases} \tag{20.6}$$

$$X_4 = \begin{cases} 1 & \text{if obs from instructor 1 in school 1} \\ -1 & \text{if obs from instructor 2 in school 1} \\ 0 & \text{otherwise} \end{cases} \tag{20.7}$$

Regression Full Model

$$Y_{ijk} = \mu_{..} + \alpha_1 X_{ijk1} + \beta_{1(1)} X_{ijk2} + \beta_{2(1)} X_{ijk3} + \beta_{1(2)} X_{ijk4} + \epsilon_{ijk}$$

Random Factor Effects

If

$$\alpha_1 \sim iidN(0, \sigma_\alpha^2) \beta_{j(i)} \sim iidN(0, \sigma_\beta^2)$$

Mean Square	Expected Mean Squares A fixed, B random	Expected Mean Squares A random, B random
MSA	$\sigma^2 + n\sigma^2_\beta + bn\frac{\sum \alpha_i^2}{a-1}$	$\sigma^2 + bn\sigma^2_\alpha + n\sigma^2_\beta$
MSB(A)	$\sigma^2 + n\sigma^2_\beta$	$\sigma^2 + n\sigma^2_\beta$
MSE	σ^2	σ^2

Test Statstics

Factor A	$\frac{MSA}{MSB(A)}$	$\frac{MSA}{MSB(A)}$
Factor B(A)	$\frac{MSB(A)}{MSE}$	$\frac{MSB(A)}{MSE}$

Another way to increase the precision of treatment comparisons by reducing variability is to use regression models to adjust for differences among experimental units (also known as **analysis of covariance**).

20.6 Single Factor Covariance Model

$$Y_{ij} = \mu_. + \tau_i + \gamma(X_{ij} - \bar{X}_{..}) + \epsilon_{ij}$$

for $i = 1, ..., r; j = 1, .., n_i$

where

- $\mu_.$ overall mean
- τ_i: fixed treatment effects ($\sum \tau_i = 0$)
- γ: fixed regression coefficient effect between X and Y
- X_{ij} covariate (not random)
- $\epsilon_{ij} \sim iidN(0, \sigma^2)$: random errors

If we just use γX_{ij} as the regression term (rather than $\gamma(X_{ij} - \bar{X}_{..})$), then $\mu_.$ is no longer the overall mean; thus we need to centered mean.

$$E(Y_{ij}) = \mu_. + \tau_i + \gamma(X_{ij} - \bar{X}_{..}) var(Y_{ij}) = \sigma^2$$

$Y_{ij} \sim N(\mu_{ij}, \sigma^2)$, where

$$\mu_{ij} = \mu_. + \tau_i + \gamma(X_{ij} - \bar{X}_{..}) \sum \tau_i = 0$$

Thus, the mean response (μ_{ij}) is a regression line with intercept $\mu_{.} + \tau_i$ and slope γ for each treatment i.

Assumption:

- All treatment regression lines have the same slope

- when treatment interact with covariate X (non-parallel slopes), covariance analysis is **not** appropriate. in which case we should use separate regression lines.

More complicated regression features (e.g., quadratic, cubic) or additional covariates e.g.,

$$Y_{ij} = \mu_{.} + \tau_i + \gamma_1(X_{ij1} - \bar{X}_{..2}) + \gamma_2(X_{ij2} - \bar{X}_{..2}) + \epsilon_{ij}$$

Regression Formulation

We can use indicator variables for treatments

$$l_1 = \begin{cases} 1 & \text{if case is from treatment 1} \\ -1 & \text{if case is from treatment r} \\ 0 & \text{otherwise} \end{cases} ...l_{r-1} = \begin{cases} 1 & \text{if case is from treatment r-1} \\ -1 & \text{if case is from treatment r} \\ 0 & \text{otherwise} \end{cases}$$

Let $x_{ij} = X_{ij} - \bar{X}_{..}$. the regression model is

$$Y_{ij} = \mu_{.} + \tau_1 l_{ij,1} + .. + \tau_{r-1} l_{ij,r-1} + \gamma x_{ij} + \epsilon_{ij}$$

where $I_{ij,1}$ is the indicator variable l_1 for the j-th case from treatment i. The treatment effect $\tau_1, ..\tau_{r-1}$ are just regression coefficients for the indicator variables.

We could use the same diagnostic tools for this case.

Inference

Treatment effects

$$H_0 : \tau_1 = \tau_2 = ... = 0 H_a : \text{not all } \tau_i = 0$$

$$\text{Full Model} : Y_{ij} = \mu_{.} + \tau_i + \gamma X_{ij} + \epsilon_{ij} \text{Reduced Model} : Y_{ij} = \mu_{.} + \gamma X_{ij} + \epsilon_{ij}$$

$$F = \frac{SSE(R) - SSE(F)}{(N-2) - (N-(r+1))} / \frac{SSE(F)}{N-(r+1)} \sim F_{(r-1, N-(r+1))}$$

If we are interested in comparisons of treatment effects.
For example, r - 3. We estimate $\tau_1, \tau_2, \tau_3 = -\tau_1 - \tau_2$

Comparison	Estimate	Variance of Estimator
$\tau_1 - \tau_2$	$\hat{\tau}_1 - \hat{\tau}_2$	$var(\hat{\tau}_1) + var(\hat{\tau}_2) - 2cov(\hat{\tau}_1 \hat{\tau}_2)$
$\tau_1 - \tau_3$	$2\hat{\tau}_1 + \hat{\tau}_2$	$4var(\hat{\tau}_1) + var(\hat{\tau}_2) - 4cov(\hat{\tau}_1 \hat{\tau}_2)$
$\tau_2 - \tau_3$	$\hat{\tau}_1 + 2\hat{\tau}_2$	$var(\hat{\tau}_1) + 4var(\hat{\tau}_2) - 4cov(\hat{\tau}_1 \hat{\tau}_2)$

Testing for Parallel Slopes

Example:

r = 3

$$Y_{ij} = \mu_. + \tau_1 I_{ij,1} + \tau_2 I_{ij,2} + \gamma X_{ij} + \beta_1 I_{ij,1} X_{ij} + \beta_2 I_{ij,2} X_{ij} + \epsilon_{ij}$$

where β_1, β_2: interaction coefficients.

$$H_0 : \beta_1 = \beta_2 = 0 H_a : \text{at least one} \beta \neq 0$$

If we can't reject H_0 using F-test then we have evidence that the slopes are parallel

Adjusted Means

The means in response after adjusting for the covariate effect

$$Y_{i.}(adj) = \bar{Y}_{i.} - \hat{\gamma}(\bar{X}_{i.} - \bar{X}_{..})$$

Chapter 21

Multivariate Methods

$y_1, ..., y_p$ are possibly correlated random variables with means $\mu_1, ..., \mu_p$

$$\mathbf{y} = \begin{pmatrix} y_1 \\ . \\ y_p \end{pmatrix}$$

$$E(\mathbf{y}) = \begin{pmatrix} \mu_1 \\ . \\ \mu_p \end{pmatrix}$$

Let $\sigma_{ij} = cov(y_i, y_j)$ for $i, j = 1, ..., p$

$$= (\sigma_{ij}) = \begin{pmatrix} \sigma_{11} & \sigma_{22} & ... & \sigma_{1p} \\ \sigma_{21} & \sigma_{22} & ... & \sigma_{2p} \\ . & . & . & . \\ \sigma_{p1} & \sigma_{p2} & ... & \sigma_{pp} \end{pmatrix}$$

where (symmetric) is the variance-covariance or dispersion matrix

Let $\mathbf{u}_{p \times 1}$ and $\mathbf{v}_{q \times 1}$ be random vectors with means μ_u and μ_v . Then

$${}_{uv} = cov(\mathbf{u}, \mathbf{v}) = E[(\mathbf{u} - \mu_u)(\mathbf{v} - \mu_v)']$$

in which ${}_{uv} \neq {}_{vu}$ and ${}_{uv} = {}'_{vu}$

Properties of Covariance Matrices

1. Symmetric $' =$

2. Non-negative definite $\mathbf{a}' \ \mathbf{a} \geq 0$ for any $\mathbf{a} \in R^p$, which is equivalent to eigenvalues of $\ $, $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p \geq 0$
3. $| \ | = \lambda_1 \lambda_2 ... \lambda_p \geq 0$ (**generalized variance**) (the bigger this number is, the more variation there is
4. $trace(\ ) = tr(\ ) = \lambda_1 + ... + \lambda_p = \sigma_{11} + ... + \sigma_{pp} =$ sum of variance (**total variance**)

Note:

- $\ $ is typically required to be positive definite, which means all eigenvalues are positive, and $\ $ has an inverse $\ ^{-1}$ such that $\ ^{-1} \ = \mathbf{I}_{p \times p} = \ \ ^{-1}$

Correlation Matrices

$$\rho_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}}$$

$$\mathbf{R} = \left(\begin{array}{cccc} \rho_{11} & \rho_{12} & \cdots & \rho_{1p} \\ \rho_{21} & \rho_{22} & \cdots & \rho_{2p} \\ . & . & . & . \\ \rho_{p1} & \rho_{p2} & \cdots & \rho_{pp} \end{array} \right)$$

where ρ_{ij} is the correlation, and $\rho_{ii} = 1$ for all i

Alternatively,

$$\mathbf{R} = [diag(\ )]^{-1/2} \ [diag(\ )]^{-1/2}$$

where $diag(\ )$ is the matrix which has the σ_{ii}'s on the diagonal and 0's elsewhere

and $\mathbf{A}^{1/2}$ (the square root of a symmetric matrix) is a symmetric matrix such as $\mathbf{A} = \mathbf{A}^{1/2}\mathbf{A}^{1/2}$

Equalities

Let

- $\mathbf{x}$ and $\mathbf{y}$ be random vectors with means μ_x and μ_y and variance -variance matrices $\ _x$ and $\ _y$.
- $\mathbf{A}$ and $\mathbf{B}$ be matrices of constants and $\mathbf{c}$ and $\mathbf{d}$ be vectors of constants

Then

- $E(\mathbf{Ay} + \mathbf{c}) = \mathbf{A}\mu_y + c$
- $var(\mathbf{Ay} + \mathbf{c}) = \mathbf{A}var(\mathbf{y})\mathbf{A}' = \mathbf{A} \ _{\mathbf{y}}\mathbf{A}'$
- $cov(\mathbf{Ay} + \mathbf{c}, \mathbf{By} + \mathbf{d}) = \mathbf{A} \ _{\mathbf{y}}\mathbf{B}'$
- $E(\mathbf{Ay} + \mathbf{Bx} + \mathbf{c}) = \mathbf{A} \ _{\mathbf{y}} + \mathbf{B} \ _{\mathbf{x}} + \mathbf{c}$

- $var(\mathbf{Ay}+\mathbf{Bx}+\mathbf{c}) = \mathbf{A}\ _{\mathbf{y}}\mathbf{A}' + \mathbf{B}\ _{\mathbf{x}}\mathbf{B}' + \mathbf{A}\ _{\mathbf{yx}}\mathbf{B}' + \mathbf{B}\ '_{\mathbf{yx}}\mathbf{A}'$

Multivariate Normal Distribution

Let $\mathbf{y}$ be a multivariate normal (MVN) random variable with mean μ and variance . Then the density of $\mathbf{y}$ is

$$f(\mathbf{y}) = \frac{1}{(2\pi)^{p/2}|\ |^{1/2}}\exp(-\frac{1}{2}(\mathbf{y}-\ )'\ ^{-\mathbf{1}}(\mathbf{y}-\ ))$$

$\mathbf{y} \sim N_p(\mu,\ )$

21.0.1 Properties of MVN

- Let $\mathbf{A}_{r\times p}$ be a fixed matrix. Then $\mathbf{Ay} \sim N_r(\mathbf{A}\ , \mathbf{A}\ \mathbf{A}')$. $r \leq p$ and all rows of $\mathbf{A}$ must be linearly independent to guarantee that $\mathbf{A}\ \mathbf{A}'$ is non-singular.
- Let $\mathbf{G}$ be a matrix such that $\ ^{-1} = \mathbf{GG}'$. Then $\mathbf{G}'\mathbf{y} \sim N_p(\mathbf{G}'\ , \mathbf{I})$ and $\mathbf{G}'(\mathbf{y}-\ ) \sim N_p(0, \mathbf{I})$
- Any fixed linear combination of $y_1, ..., y_p$ (say $\mathbf{c}'\mathbf{y}$) follows $\mathbf{c}'\mathbf{y} \sim N_1(\mathbf{c}'\ , \mathbf{c}'\ \mathbf{c})$
- Define a partition, $[\mathbf{y}_1', \mathbf{y}_2']'$ where
  - $\mathbf{y}_1$ is $p_1 \times 1$
  - $\mathbf{y}_2$ is $p_2 \times 1$,
  - $p_1 + p_2 = p$
  - $p_1, p_2 \geq 1$ Then

$$\begin{pmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{pmatrix} \sim N\left(\begin{pmatrix}\mu_1\\ \mu_2\end{pmatrix}, \begin{pmatrix} _{11} & _{12}\\ _{21} & _{22}\end{pmatrix}\right)$$

- The marginal distributions of $\mathbf{y}_1$ and $\mathbf{y}_2$ are $\mathbf{y}_1 \sim N_{p1}(\ _{\mathbf{1}},\ _{\mathbf{11}})$ and $\mathbf{y}_2 \sim N_{p2}(\ _{\mathbf{2}},\ _{\mathbf{22}})$
- Individual components $y_1, ..., y_p$ are all normally distributed $y_i \sim N_1(\mu_i, \sigma_{ii})$
- The conditional distribution of $\mathbf{y}_1$ and $\mathbf{y}_2$ is normal
  - $\mathbf{y}_1|\mathbf{y}_2 \sim N_{p1}(\ _{\mathbf{1}} + \ _{\mathbf{12}}\ ^{-\mathbf{1}}_{\mathbf{22}}(\mathbf{y_2} - \ _{\mathbf{2}}),\ _{\mathbf{11}} - \ _{\mathbf{12}}\ ^{-\mathbf{1}}_{\mathbf{22}}\ _{\mathbf{21}})$
    - $*$ In this formula, we see if we know (have info about) $\mathbf{y}_2$, we can re-weight $\mathbf{y}_1$'s mean, and the variance is reduced because we know more about $\mathbf{y}_1$ because we know $\mathbf{y}_2$
  - which is analogous to $\mathbf{y}_2|\mathbf{y}_1$. And $\mathbf{y}_1$ and $\mathbf{y}_2$ are independently distrusted only if $\ _{12} = 0$

- If $\mathbf{y} \sim N(\ , \ )$ and is positive definite, then $(\mathbf{y} - \ )' \ ^{-\mathbf{1}}(\mathbf{y} - \ ) \sim \chi^2_{(p)}$
- If $\mathbf{y}_i$ are independent $N_p(\ _i, \ _i)$ random variables, then for fixed matrices $\mathbf{A}_{i(m\times p)}$, $\sum_{i=1}^k \mathbf{A}_i\mathbf{y}_i \sim N_m(\sum_{i=1}^k \mathbf{A}_i \ _i, \sum_{i=1}^k \mathbf{A}_i \ _i\mathbf{A}_i)$

Multiple Regression

$$\begin{pmatrix} Y \\ \mathbf{x} \end{pmatrix} \sim N_{p+1}\left(\begin{bmatrix} \mu_y \\ \ _x \end{bmatrix}, \begin{bmatrix} \sigma^2_Y & \ _{yx} \\ \ _{yx} & \ _{xx} \end{bmatrix}\right)$$

The conditional distribution of Y given x follows a univariate normal distribution with

$$\begin{aligned} E(Y|\mathbf{x}) &= \mu_y + \ _{yx}\Sigma_{xx}^{-1}(\mathbf{x} - \mu_x) \\ &= \mu_y - \Sigma_{yx}\Sigma_{xx}^{-1}\mu_x + \Sigma_{yx}\Sigma_{xx}^{-1}\mathbf{x} \\ &= \beta_0 + \ '\mathbf{x} \end{aligned}$$

where $\beta = (\beta_1, ..., \beta_p)' = \ _{xx}^{-1} \ '_{yx}$ (e.g., analogous to $(\mathbf{x}'\mathbf{x})^{-\mathbf{1}}\mathbf{x}'\mathbf{y}$ but not the same if we consider Y_i and $\mathbf{x}_i$, $i = 1, .., n$ and use the empirical covariance formula: $var(Y|\mathbf{x}) = \sigma^2_Y - \ _{\mathbf{yx}} \ _{\mathbf{xx}}^{-\mathbf{1}} \ '_{\mathbf{yx}}$)

Samples from Multivariate Normal Populations

A random sample of size n, $\mathbf{y}_1, .., \mathbf{y}_n$ from $N_p(\ , \ )$. Then

- Since $\mathbf{y}_1, ..., \mathbf{y}_n$ are iid, their sample mean, $\bar{\mathbf{y}} = \sum_{i=1}^n \mathbf{y}_i/n \sim N_p(\ , \ /n)$. that is, $\bar{\mathbf{y}}$ is an unbiased estimator of
- The $p \times p$ sample variance-covariance matrix, $\mathbf{S}$ is $\mathbf{S} = \frac{1}{n-1}\sum_{i=1}^n(\mathbf{y}_i - \bar{\mathbf{y}})(\mathbf{y}_i - \bar{\mathbf{y}})' = \frac{1}{n-1}(\sum_{i=1}^n \mathbf{y}_i\mathbf{y}_i' - n\bar{\mathbf{y}}\bar{\mathbf{y}}')$
    - where $\mathbf{S}$ is symmetric, unbiased estimator of and has $p(p+1)/2$ random variables.
- $(n-1)\mathbf{S} \sim W_p(n-1, \ )$ is a Wishart distribution with n-1 degrees of freedom and expectation $(n-1)$. The Wishart distribution is a multivariate extension of the Chi-squared distribution.
- $\bar{\mathbf{y}}$ and $\mathbf{S}$ are independent
- $\bar{\mathbf{y}}$ and $\mathbf{S}$ are sufficient statistics. (All of the info in the data about and is contained in $\bar{\mathbf{y}}$ and $\mathbf{S}$, regardless of sample size).

Large Sample Properties

$\mathbf{y}_1, ..., \mathbf{y}_n$ are a random sample from some population with mean and variance-covariance matrix

- $\bar{\mathbf{y}}$ is a consistent estimator for μ
- $\mathbf{S}$ is a consistent estimator for

- **Multivariate Central Limit Theorem**: Similar to the univariate case, $\sqrt{n}(\bar{\mathbf{y}} - \mu)\dot{\sim}N_p(\mathbf{0}, \ )$ where n is large relative to p ($n \geq 25p$), which is equivalent to $\bar{\mathbf{y}}\dot{\sim}N_p(\mu, \ /n)$
- **Wald's Theorem**: $n(\bar{\mathbf{y}} - \mu)'\mathbf{S}^{-1}(\bar{\mathbf{y}} - \mu)$ when n is large relative to p.

Maximum Likelihood Estimation for MVN

Suppose iid $\mathbf{y}_1, ...\mathbf{y}_n \sim N_p(\mu, \ )$, the likelihood function for the data is

$$\begin{aligned} L(\mu, \ ) &= \prod_{j=1}^{n}\left(\frac{1}{(2\pi)^{p/2}|\ |^{1/2}}\exp(-\frac{1}{2}(\mathbf{y}_j - \mu)'\ ^{-1})(\mathbf{y}_j - \mu)\right) \\ &= \frac{1}{(2\pi)^{np/2}|\ |^{n/2}}\exp(-\frac{1}{2}\sum_{j=1}^{n}(\mathbf{y}_j - \mu)'\ ^{-1})(\mathbf{y}_j - \mu) \end{aligned}$$

Then, the MLEs are

$$\hat{\mu} = \bar{\mathbf{y}}$$

$$\hat{} = \frac{n-1}{n}\mathbf{S}$$

using derivatives of the log of the likelihood function with respect to μ and

Properties of MLEs

- Invariance: If $\hat{\theta}$ is the MLE of θ, then the MLE of $h(\theta)$ is $h(\hat{\theta})$ for any function h(.)
- Consistency: MLEs are consistent estimators, but they are usually biased
- Efficiency: MLEs are efficient estimators (no other estimator has a smaller variance for large samples)
- Asymptotic normality: Suppose that $\hat{\theta}_n$ is the MLE for θ based upon n independent observations. Then $\hat{\theta}_n\dot{\sim}N(\theta, \mathbf{H}^{-1})$
  - $\mathbf{H}$ is the Fisher Information Matrix, which contains the expected values of the second partial derivatives fo the log-likelihood function. the (i,j)th element of $\mathbf{H}$ is $-E(\frac{\partial^2 l(\ )}{\partial\theta_i\partial\theta_j})$
  - we can estimate $\mathbf{H}$ by finding the form determined above, and evaluate it at $\theta = \hat{\theta}_n$
- Likelihood ratio testing: for some null hypothesis, H_0 we can form a likelihood ratio test
  - The statistic is: $\Lambda = \frac{\max_{H_0} l(\ ,\ |\mathbf{Y})}{\max l(\mu,\ |\mathbf{Y})}$

- For large n, $-2\log\Lambda \sim \chi^2_{(v)}$ where v is the number of parameters in the unrestricted space minus the number of parameters under H_0

Test of Multivariate Normality

- Check univariate normality for each trait (X) separately
    - Can check Normality Assessment
    - The good thing is that if any of the univariate trait is not normal, then the joint distribution is not normal (see again Properties of MVN). If a joint multivariate distribution is normal, then the marginal distribution has to be normal.
    - However, marginal normality of all traits does not imply joint MVN
    - Easily rule out multivariate normality, but not easy to prove it
- Mardia's tests for multivariate normality
    - Multivariate skewness is

$$\beta_{1,p} = E[(\mathbf{y}- \quad)' \quad^{-1}(\mathbf{x}- \quad)]^3$$

    - where $\mathbf{x}$ and $\mathbf{y}$ are independent, but have the same distribution (note: β here is not regression coefficient)
    - Multivariate kurtosis is defined as
    -

$$\beta_{2,p} - E[(\mathbf{y}- \quad)' \quad^{-1}(\mathbf{x}- \quad)]^2$$

    - For the MVN distribution, we have $\beta_{1,p} = 0$ and $\beta_{2,p} = p(p+2)$
    - For a sample of size n, we can estimate

$$\hat{\beta}_{1,p} = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} g_{ij}^2$$

$$\hat{\beta}_{2,p} = \frac{1}{n}\sum_{i=1}^{n} g_{ii}^2$$

        - where $g_{ij} = (\mathbf{y}_i - \bar{\mathbf{y}})'\mathbf{S}^{-1}(\mathbf{y}_j - \bar{\mathbf{y}})$. Note: $g_{ii} = d_i^2$ where d_i^2 is the Mahalanobis distance
    - (MARDIA, 1970) shows for large n

$$\kappa_1 = \frac{n\hat{\beta}_{1,p}}{6} \dot\sim \chi^2_{p(p+1)(p+2)/6}$$

$$\kappa_2 = \frac{\hat{\beta}_{2,p} - p(p+2)}{\sqrt{8p(p+2)/n}} \sim N(0,1)$$

- Hence, we can use κ_1 and κ_2 to test the null hypothesis of MVN.
- When the data are non-normal, normal theory tests on the mean are sensitive to $\beta_{1,p}$, while tests on the covariance are sensitive to $\beta_{2,p}$

- Chi-square Q-Q plot
  - Let $\mathbf{y}_i, i = 1, ..., n$ be a random sample sample from $N_p(\ , \ )$
  - Then $\mathbf{z}_i = \ ^{-1/2}(\mathbf{y}_i - \ ), i = 1, ..., n$ are iid $N_p(\mathbf{0}, \mathbf{I})$. Thus, $d_i^2 = \mathbf{z}_i'\mathbf{z}_i \sim \chi^2_p, i = 1, ..., n$
  - plot the ordered d_i^2 values against the qualities of the χ^2_p distribution. When normality holds, the plot should approximately resemble a straight lien passing through the origin at a 45 degree
  - it requires large sample size (i.e., sensitive to sample size). Even if we generate data from a MVN, the tail of the Chi-square Q-Q plot can still be out of line.
- If the data are not normal, we can
  - ignore it
  - use nonparametric methods
  - use models based upon an approximate distirubiton (e.g., GLMM)
  - try performing a transformation

```
library(heplots)
```

```
## Warning: package 'heplots' was built under R version 4.0.5
```

```
library(ICSNP)
```

```
## Warning: package 'ICSNP' was built under R version 4.0.5

## Warning: package 'ICS' was built under R version 4.0.5
```

```
library(MVN)
```

```
## Warning: package 'MVN' was built under R version 4.0.5
```

```
library(tidyverse)

trees = read.table("images/trees.dat")
names(trees) <- c("Nitrogen","Phosphorous","Potassium","Ash","Height")
str(trees)
```

```
## 'data.frame':    26 obs. of  5 variables:
##  $ Nitrogen   : num  2.2 2.1 1.52 2.88 2.18 1.87 1.52 2.37 2.06 1.84 ...
##  $ Phosphorous: num  0.417 0.354 0.208 0.335 0.314 0.271 0.164 0.302 0.373 0.265 ..
##  $ Potassium  : num  1.35 0.9 0.71 0.9 1.26 1.15 0.83 0.89 0.79 0.72 ...
##  $ Ash        : num  1.79 1.08 0.47 1.48 1.09 0.99 0.85 0.94 0.8 0.77 ...
##  $ Height     : int  351 249 171 373 321 191 225 291 284 213 ...
```

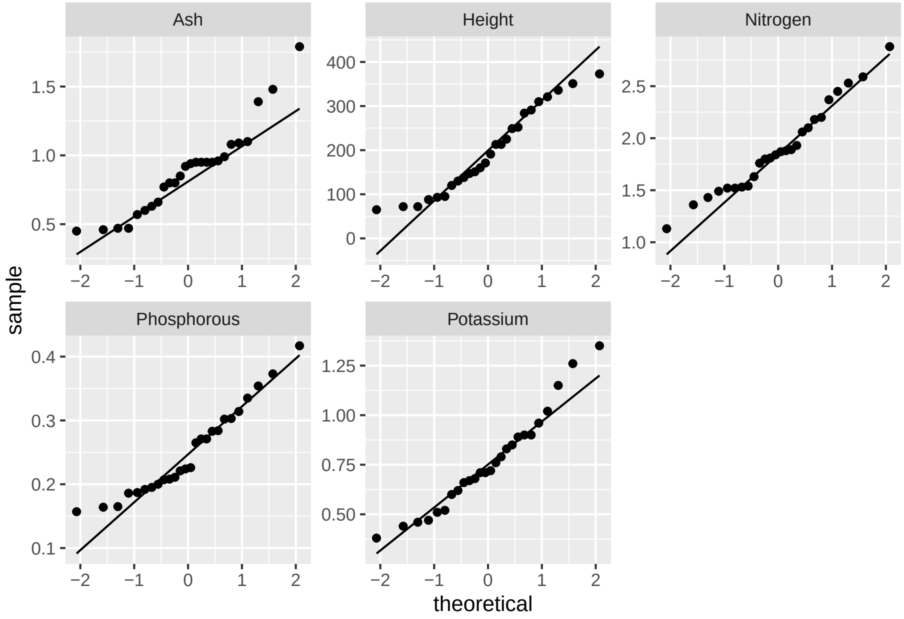
```
summary(trees)
```

```
##     Nitrogen      Phosphorous       Potassium           Ash        
##  Min.   :1.130   Min.   :0.1570   Min.   :0.3800   Min.   :0.4500  
##  1st Qu.:1.532   1st Qu.:0.1963   1st Qu.:0.6050   1st Qu.:0.6375  
##  Median :1.855   Median :0.2250   Median :0.7150   Median :0.9300  
##  Mean   :1.896   Mean   :0.2506   Mean   :0.7619   Mean   :0.8873  
##  3rd Qu.:2.160   3rd Qu.:0.2975   3rd Qu.:0.8975   3rd Qu.:0.9825  
##  Max.   :2.880   Max.   :0.4170   Max.   :1.3500   Max.   :1.7900  
##      Height     
##  Min.   : 65.0  
##  1st Qu.:122.5  
##  Median :181.0  
##  Mean   :196.6  
##  3rd Qu.:276.0  
##  Max.   :373.0
```

```
cor(trees, method = "pearson") # correlation matrix
```

```
##              Nitrogen Phosphorous Potassium       Ash    Height
## Nitrogen    1.0000000   0.6023902 0.5462456 0.6509771 0.8181641
## Phosphorous 0.6023902   1.0000000 0.7037469 0.6707871 0.7739656
## Potassium   0.5462456   0.7037469 1.0000000 0.6710548 0.7915683
## Ash         0.6509771   0.6707871 0.6710548 1.0000000 0.7676771
## Height      0.8181641   0.7739656 0.7915683 0.7676771 1.0000000
```

```
# qq-plot
gg <- trees %>%
    pivot_longer(everything(), names_to = "Var", values_to = "Value") %>%
    ggplot(aes(sample = Value)) +
    geom_qq() +
    geom_qq_line() +
    facet_wrap("Var", scales = "free")
gg
```



```
# Univariate normality
sw_tests <- apply(trees, MARGIN = 2, FUN = shapiro.test)
sw_tests
```

```
## $Nitrogen
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.96829, p-value = 0.5794
##
##
## $Phosphorous
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.93644, p-value = 0.1104
##
##
## $Potassium
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.95709, p-value = 0.3375
##
##
## $Ash
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.92071, p-value = 0.04671
##
##
## $Height
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.94107, p-value = 0.1424
```

```
# Kolmogorov-Smirnov test
ks_tests <- map(trees, ~ ks.test(scale(.x),"pnorm"))
```

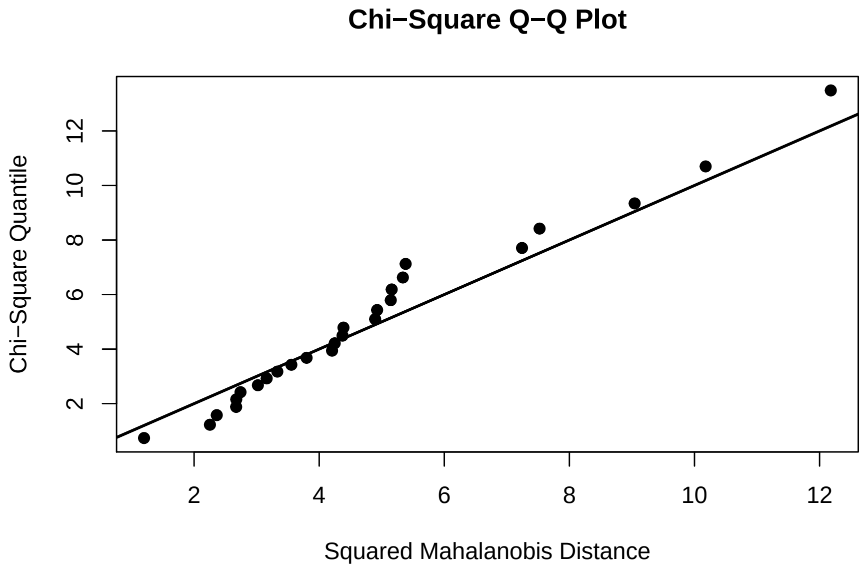
```
## Warning in ks.test(scale(.x), "pnorm"): ties should not be present for the
## Kolmogorov-Smirnov test

## Warning in ks.test(scale(.x), "pnorm"): ties should not be present for the
## Kolmogorov-Smirnov test

## Warning in ks.test(scale(.x), "pnorm"): ties should not be present for the
## Kolmogorov-Smirnov test

## Warning in ks.test(scale(.x), "pnorm"): ties should not be present for the
## Kolmogorov-Smirnov test

## Warning in ks.test(scale(.x), "pnorm"): ties should not be present for the
## Kolmogorov-Smirnov test
```

```
ks_tests
```

```
## $Nitrogen
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  scale(.x)
## D = 0.12182, p-value = 0.8351
## alternative hypothesis: two-sided
##
##
## $Phosphorous
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  scale(.x)
## D = 0.17627, p-value = 0.3944
## alternative hypothesis: two-sided
##
##
## $Potassium
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  scale(.x)
## D = 0.10542, p-value = 0.9348
## alternative hypothesis: two-sided
##
##
## $Ash
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  scale(.x)
## D = 0.14503, p-value = 0.6449
## alternative hypothesis: two-sided
##
##
## $Height
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  scale(.x)
## D = 0.1107, p-value = 0.9076
## alternative hypothesis: two-sided
```

```
# Mardia's test, need large sample size for power
mardia_test <-
    mvn(
        trees,
        mvnTest = "mardia",
        covariance = FALSE,
        multivariatePlot = "qq"
    )
```



```
mardia_test$multivariateNormality
```

```
##              Test          Statistic           p value Result
## 1 Mardia Skewness   29.7248528871795   0.72054426745778    YES
## 2 Mardia Kurtosis -1.67743173185383 0.0934580886477281    YES
## 3             MVN               <NA>               <NA>    YES
```

21.0.2 Mean Vector Inference

In the univariate normal distribution, we test $H_0 : \mu = \mu_0$ by using

$$T = \frac{\bar{y} - \mu_0}{s/\sqrt{n}} \sim t_{n-1}$$

under the null hypothesis. And reject the null if $|T|$ is large relative to $t_{(1-\alpha/2,n-1)}$ because it means that seeing a value as large as what we observed is rare if the null is true

Equivalently,

$$T^2 = \frac{(\bar{y} - \mu_0)^2}{s^2/n} = n(\bar{y} - \mu_0)(s^2)^{-1}(\bar{y} - \mu_0) \sim f_{(1,n-1)}$$

21.0.2.1 Natural Multivariate Generalization

$$H_0 : \quad = {}_0 H_a : \quad \neq {}_0$$

Define **Hotelling's** T^2 by

$$T^2 = n(\bar{\mathbf{y}} - {}_0)'\mathbf{S}^{-1}(\bar{\mathbf{y}} - {}_0)$$

which can be viewed as a generalized distance between $\bar{\mathbf{y}}$ and ${}_0$

Under the assumption of normality,

$$F = \frac{n-p}{(n-1)p}T^2 \sim f_{(p,n-p)}$$

and reject the null hypothesis when $F > f_{(1-\alpha,p,n-p)}$

- The T^2 test is invariant to changes in measurement units.
  - If $\mathbf{z} = \mathbf{C}\mathbf{y} + \mathbf{d}$ where $\mathbf{C}$ and $\mathbf{d}$ do not depend on $\mathbf{y}$, then $T^2(\mathbf{z}) - T^2(\mathbf{y})$
- The T^2 test can be derived as a **likelihood ratio** test of $H_0 : \mu = \mu_0$

21.0.2.2 Confidence Intervals

21.0.2.2.1 Confidence Region An "exact" $100(1-\alpha)\%$ confidence region for is the set of all vectors, $\mathbf{v}$, which are "close enough" to the observed mean vector, $\bar{\mathbf{y}}$ to satisfy

$$n(\bar{\mathbf{y}} - {}_0)'\mathbf{S}^{-1}(\bar{\mathbf{y}} - {}_0) \le \frac{(n-1)p}{n-p} f_{(1-\alpha,p,n-p)}$$

- $\mathbf{v}$ are just the mean vectors that are not rejected by the T^2 test when $\bar{\mathbf{y}}$ is observed.

In case that you have 2 parameters, the confidence region is a "hyper-ellipsoid".

In this region, it consists of all ${}_0$ vectors for which the T^2 test would not reject H_0 at significance level α

Even though the confidence region better assesses the joint knowledge concerning plausible values of , people typically include confidence statement about the individual component means. We'd like all of the separate confidence statements to hold **simultaneously** with a specified high probability. Simultaneous confidence intervals: intervals **against** any statement being incorrect

21.0.2.2.1.1 Simultaneous Confidence Statements

- Intervals based on a rectangular confidence region by projecting the previous region onto the coordinate axes:

$$\bar{y}_i \pm \sqrt{\frac{(n-1)p}{n-p} f_{(1-\alpha,p,n-p)} \frac{s_{ii}}{n}}$$

for all $i = 1, .., p$

which implied confidence region is conservative; it has at least $100(1-\alpha)\%$

Generally, simultaneous $100(1-\alpha)\%$ confidence intervals for all linear combinations , $\mathbf{a}$ of the elements of the mean vector are given by

$$\mathbf{a'\bar{y}} \pm \sqrt{\frac{(n-1)p}{n-p} f_{(1-\alpha,p,n-p)} \frac{\mathbf{a'Sa}}{n}}$$

- works for any arbitrary linear combination $\mathbf{a}' \ = a_1\mu_1 + ... + a_p\mu_p$, which is a projection onto the axis in the direction of $\mathbf{a}$
- These intervals have the property that the probability that at least one such interval does not contain the appropriate $\mathbf{a}' \ $ is no more than α
- These types of intervals can be used for "data snooping" (like Scheffe)

21.0.2.2.1.2 One μ at a time

- One at a time confidence intervals:

$$\bar{y}_i \pm t_{(1-\alpha/2,n-1}\sqrt{\frac{s_{ii}}{n}}$$

- Each of these intervals has a probability of $1-\alpha$ of covering the appropriate μ_i
- But they ignore the covariance structure of the p variables
- If we only care about k simultaneous intervals, we can use "one at a time" method with the Bonferroni correction.
- This method gets more conservative as the number of intervals k increases.

21.0.3 General Hypothesis Testing

21.0.3.1 One-sample Tests

$$H_0 : \mathbf{C} \ = \mathbf{0}$$

where

- $\mathbf{C}$ is a $c \times p$ matrix of rank c where $c \le p$

We can test this hypothesis using the following statistic

$$F = \frac{n-c}{(n-1)c}T^2$$

where $T^2 = n(\mathbf{C\bar{y}})'(\mathbf{CSC}')^{-1}(\mathbf{C\bar{y}})$

Example:

$$H_0 : \mu_1 = \mu_2 = ... = \mu_p$$

Equivalently,

$$\mu_1 - \mu_2 = 0 \vdots \mu_{p-1} - \mu_p = 0$$

a total of $p-1$ tests. Hence, we have $\mathbf{C}$ as the $p-1 \times p$ matrix

$$\mathbf{C} = \begin{pmatrix} 1 & -1 & 0 & ... & 0 \\ 0 & 1 & -1 & ... & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & 1 & -1 \end{pmatrix}$$

number of rows $= c = p - 1$

Equivalently, we can also compare all of the other means to the first mean. Then, we test $\mu_1 - \mu_2 = 0, \mu_1 - \mu_3 = 0, ..., \mu_1 - \mu_p = 0$, the $(p-1) \times p$ matrix $\mathbf{C}$ is

$$\mathbf{C} = \begin{pmatrix} -1 & 1 & 0 & ... & 0 \\ -1 & 0 & 1 & ... & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & ... & 0 & 1 \end{pmatrix}$$

The value of T^2 is invariant to these equivalent choices of $\mathbf{C}$

This is often used for **repeated measures designs**, where each subject receives each treatment once over successive periods of time (all treatments are administered to each unit).

Example:

Let y_{ij} be the response from subject i at time j for $i = 1, .., n, j = 1, ..., T$. In this case, $\mathbf{y}_i = (y_{i1}, ..., y_{iT})', i = 1, ..., n$ are a random sample from $N_T(\; , \;)$

Let $n = 8$ subjects, $T = 6$. We are interested in $\mu_1, .., \mu_6$

$$H_0 : \mu_1 = \mu_2 = ... = \mu_6$$

Equivalently,

$$\mu_1 - \mu_2 = 0 \mu_2 - \mu_3 = 0 ... \mu_5 - \mu_6 = 0$$

We can test orthogonal polynomials for 4 equally spaced time points. To test for example the null hypothesis that quadratic and cubic effects are jointly equal to 0, we would define $\mathbf{C}$

$$\mathbf{C} = \begin{pmatrix} 1 & -1 & -1 & 1 \\ -1 & 3 & -3 & 1 \end{pmatrix}$$

21.0.3.2 Two-Sample Tests

Consider the analogous two sample multivariate tests.

Example: we have data on two independent random samples, one sample from each of two populations

$$\mathbf{y}_{1i} \sim N_p(\ _{\mathbf{1}}, \ )\mathbf{y}_{2j} \sim N_p(\ _{\mathbf{2}}, \ )$$

We **assume**

- normality
- equal variance-covariance matrices
- independent random samples

We can summarize our data using the **sufficient statistics** $\bar{\mathbf{y}}_1, \mathbf{S}_1, \bar{\mathbf{y}}_2, \mathbf{S}_2$ with respective sample sizes, n_1, n_2

Since we assume that $\ _1 = \ _2 = \ $, compute a pooled estimate of the variance-covariance matrix on $n_1 + n_2 - 2$ df

$$\mathbf{S} = \frac{(n_1 - 1)\mathbf{S}_1 + (n_2 - 1)\mathbf{S}_2}{(n_1 - 1) + (n_2 - 1)}$$

$$H_0 : \ _1 = \ _2 H_a : \ _1 \neq \ _2$$

At least one element of the mean vectors is different

We use

- $\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2$ to estimate $\mu_1 - \mu_2$
- $\mathbf{S}$ to estimate

  Note: because we assume the two populations are independent, there is no covariance

  $cov(\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2) = var(\bar{\mathbf{y}}_1) + var(\bar{\mathbf{y}}_2) = \frac{\mathbf{1}}{n_1} + \frac{\mathbf{2}}{n_2} = \ (\frac{1}{n_1} + \frac{1}{n_2})$

Reject H_0 if

$$
\begin{aligned}
T^2 &= (\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2)'\{\mathbf{S}(\frac{1}{n_1} + \frac{1}{n_2})\}^{-1}(\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2) \\
&= \frac{n_1 n_2}{n_1 + n_2}(\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2)'\{\mathbf{S}\}^{-1}(\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2) \\
&\geq \frac{(n_1 + n_2 - 2)p}{n_1 + n_2 - p - 1} f_{(1-\alpha, n_1+n_2-p-1)}
\end{aligned}
$$

or equivalently, if

$$
F = \frac{n_1 + n_2 - p - 1}{(n_1 + n_2 - 2)p} T^2 \geq f_{(1-\alpha, p, n_1+n_2-p-1)}
$$

A $100(1-\alpha)\%$ confidence region for $\mu_1 - \mu_2$ consists of all vector δ which satisfy

$$
\frac{n_1 n_2}{n_1 + n_2}(\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2 - \quad)'\mathbf{S}^{-1}(\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2 - \quad) \leq \frac{(n_1 + n_2 - 2)p}{n_1 + n_2 - p - 1} f_{(1-\alpha, p, n_1+n_2-p-1)}
$$

The simultaneous confidence intervals for all linear combinations of $\mu_1 - \mu_2$ have the form

$$
\mathbf{a}'(\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2) \pm \sqrt{\frac{(n_1 + n_2 - 2)p}{n_1 + n_2 - p - 1} f_{(1-\alpha, p, n_1+n_2-p-1)}} \times \sqrt{\mathbf{a}'\mathbf{S}\mathbf{a}(\frac{1}{n_1} + \frac{1}{n_2})}
$$

Bonferroni intervals, for k combinations

$$
(\bar{y}_{1i} - \bar{y}_{2i}) \pm t_{(1-\alpha/2k, n_1+n_2-2)} \sqrt{(\frac{1}{n_1} + \frac{1}{n_2}) s_{ii}}
$$

21.0.3.3 Model Assumptions

If model assumption are not met

- Unequal Covariance Matrices
  - If $n_1 = n_2$ (large samples) there is little effect on the Type I error rate and power fo the two sample test
  - If $n_1 > n_2$ and the eigenvalues of $\ _1 \ _2^{-1}$ are less than 1, the Type I error level is inflated
  - If $n_1 > n_2$ and some eigenvalues of $\ _1 \ _2^{-1}$ are greater than 1, the Type I error rate is too small, leading to a reduction in power

- Sample Not Normal
  - Type I error level of the two sample T^2 test isn't much affect by moderate departures from normality if the two populations being sampled have similar distributions
  - One sample T^2 test is much more sensitive to lack of normality, especially when the distribution is skewed.
  - Intuitively, you can think that in one sample your distribution will be sensitive, but the distribution of the difference between two similar distributions will not be as sensitive.
  - Solutions:
    - Transform to make the data more normal
    - Large large samples, use the χ^2 (Wald) test, in which populations don't need to be normal, or equal sample sizes, or equal variance-covariance matrices
      - $H_0 : \mu_1 - \mu_2 = 0$ use $(\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2)'(\frac{1}{n_1}\mathbf{S}_1 + \frac{1}{n_2}\mathbf{S}_2)^{-1}(\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2) \dot{\sim} \chi^2_{(p)}$

21.0.3.3.1 Equal Covariance Matrices Tests With independent random samples from k populations of p-dimensional vectors. We compute the sample covariance matrix for each, $\mathbf{S}_i$, where $i = 1, ..., k$

$$H_0 : {}_1 = {}_2 = ... = {}_k = H_a : \text{at least 2 are different}$$

Assume H_0 is true, we would use a pooled estimate of the common covariance matrix,

$$\mathbf{S} = \frac{\sum_{i=1}^k (n_i - 1)\mathbf{S}_i}{\sum_{i=1}^k (n_i - 1)}$$

with $\sum_{i=1}^k (n_i - 1)$

21.0.3.3.1.1 Bartlett's Test (a modification of the likelihood ratio test). Define

$$N = \sum_{i=1}^k n_i$$

and (note: || are determinants here, not absolute value)

$$M = (N-k)\log|\mathbf{S}| - \sum_{i=1}^{k}(n_i - 1)\log|\mathbf{S}_i|$$

$$C^{-1} = 1 - \frac{2p^2+3p-1}{6(p+1)(k-1)}\{\sum_{i=1}^{k}(\frac{1}{n_i-1}) - \frac{1}{N-k}\}$$

- Reject H_0 when $MC^{-1} > \chi^2_{1-\alpha,(k-1)p(p+1)/2}$
- If not all samples are from normal populations, MC^{-1} has a distribution which is often shifted to the right of the nominal χ^2 distribution, which means H_0 is often rejected even when it is true (the Type I error level is inflated). Hence, it is better to test individual normality first, or then multivariate normality before you do Bartlett's test.

21.0.3.4 Two-Sample Repeated Measurements

- Define $\mathbf{y}_{hi} = (y_{hi1}, ..., y_{hit})'$ to be the observations from the i-th subject in the h-th group for times 1 through T
- Assume that $\mathbf{y}_{11}, ..., \mathbf{y}_{1n_1}$ are iid $N_t({}_1, \ )$ and that $\mathbf{y}_{21}, ..., \mathbf{y}_{2n_2}$ are iid $N_t({}_2, \ )$
- $H_0 : \mathbf{C}({}_1 - {}_2) = \mathbf{0}_c$ where $\mathbf{C}$ is a $c \times t$ matrix of rank c where $c \leq t$
- The test statistic has the form

$$T^2 = \frac{n_1 n_2}{n_1+n_2}(\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2)'\mathbf{C}'(\mathbf{CSC}')^{-1}\mathbf{C}(\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2)$$

where $\mathbf{S}$ is the pooled covariance estimate. Then,

$$F = \frac{n_1+n_2-c-1}{(n_1+n_2-2)c}T^2 \sim f_{(c,n_1+n_2-c-1)}$$

when H_0 is true

If the null hypothesis $H_0 : \mu_1 = \mu_2$ is rejected. A weaker hypothesis is that the profiles for the two groups are parallel.

$$\mu_{11} - \mu_{21} = \mu_{12} - \mu_{22} \vdots \mu_{1t-1} - \mu_{2t-1} = \mu_{1t} - \mu_{2t}$$

The null hypothesis matrix term is then

$H_0 : \mathbf{C}(\mu_1 - \mu_2) = \mathbf{0}_c$, where $c = t-1$ and

$$\mathbf{C} = \begin{pmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & -1 \end{pmatrix}_{(t-1)\times t}$$

```
# One-sample Hotelling's T^2 test
#  Create data frame
plants <- data.frame(
    y1 = c(2.11, 2.36, 2.13, 2.78, 2.17),
    y2 = c(10.1, 35.0, 2.0, 6.0, 2.0),
    y3 = c(3.4, 4.1, 1.9, 3.8, 1.7)
)

# Center the data with the hypothesized means and make a matrix
plants_ctr <- plants %>%
    transmute(y1_ctr = y1 - 2.85,
              y2_ctr = y2 - 15.0,
              y3_ctr = y3 - 6.0) %>%
    as.matrix()

# Use anova.mlm to calculate Wilks' lambda
onesamp_fit <- anova(lm(plants_ctr ~ 1), test = "Wilks")
onesamp_fit # can't reject the null of hypothesized vector of means
```

```
## Analysis of Variance Table
##
##             Df    Wilks approx F num Df den Df  Pr(>F)
## (Intercept)  1 0.054219   11.629      3      2 0.08022 .
## Residuals    4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Paired-Sample Hotelling's T^2 test
library(ICSNP)

#  Create data frame
waste <- data.frame(
    case = 1:11,
    com_y1 = c(6, 6, 18, 8, 11, 34, 28, 71, 43, 33, 20),
    com_y2 = c(27, 23, 64, 44, 30, 75, 26, 124, 54, 30, 14),
    state_y1 = c(25, 28, 36, 35, 15, 44, 42, 54, 34, 29, 39),
    state_y2 = c(15, 13, 22, 29, 31, 64, 30, 64, 56, 20, 21)
)

# Calculate the difference between commercial and state labs
waste_diff <- waste %>%
    transmute(y1_diff = com_y1 - state_y1,
              y2_diff = com_y2 - state_y2)
# Run the test
paired_fit <- HotellingsT2(waste_diff)
paired_fit # value T.2 in the output corresponds to the approximate F-value in the output from an
```

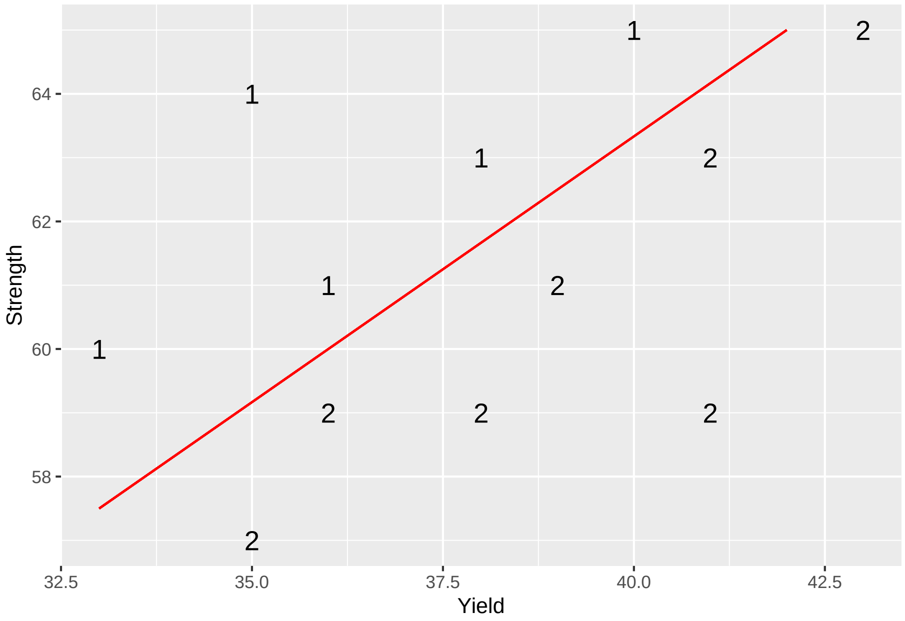
```
##
##  Hotelling's one sample T2-test
##
## data:  waste_diff
## T.2 = 6.1377, df1 = 2, df2 = 9, p-value = 0.02083
## alternative hypothesis: true location is not equal to c(0,0)
```

```
# reject the null that the two labs' measurements are equal
```

```
# Independent-Sample Hotelling's T^2 test with Bartlett's test

# Read in data
steel <- read.table("images/steel.dat")
names(steel) <- c("Temp", "Yield", "Strength")
str(steel)
```

```
## 'data.frame':    12 obs. of  3 variables:
##  $ Temp    : int  1 1 1 1 1 2 2 2 2 2 ...
##  $ Yield   : int  33 36 35 38 40 35 36 38 39 41 ...
##  $ Strength: int  60 61 64 63 65 57 59 59 61 63 ...
```

```
# Plot the data
ggplot(steel, aes(x = Yield, y = Strength)) +
    geom_text(aes(label = Temp), size = 5) +
    geom_segment(aes(
        x = 33,
        y = 57.5,
        xend = 42,
        yend = 65
    ), col = "red")
```



```
# Bartlett's test for equality of covariance matrices
# same thing as Box's M test in the multivariate setting
bart_test <- boxM(steel[, -1], steel$Temp)
bart_test # fail to reject the null of equal covariances
```

```
##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  steel[, -1]
## Chi-Sq (approx.) = 0.38077, df = 3, p-value = 0.9442
```

```
# anova.mlm
twosamp_fit <-
    anova(lm(cbind(Yield, Strength) ~ factor(Temp), data = steel), test = "Wilks")
twosamp_fit
```

```
## Analysis of Variance Table
##
##              Df    Wilks approx F num Df den Df    Pr(>F)
## (Intercept)   1 0.001177   3818.1      2      9 6.589e-14 ***
## factor(Temp)  1 0.294883     10.8      2      9  0.004106 **
## Residuals    10
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# ICSNP package
twosamp_fit2 <-
    HotellingsT2(cbind(steel$Yield, steel$Strength) ~ factor(steel$Temp))
twosamp_fit2
```

```
##
##  Hotelling's two sample T2-test
##
## data:  cbind(steel$Yield, steel$Strength) by factor(steel$Temp)
## T.2 = 10.76, df1 = 2, df2 = 9, p-value = 0.004106
## alternative hypothesis: true location difference is not equal to c(0,0)
```

```
# reject null. Hence, there is a difference in the means of the bivariate normal distributions
```

21.1 MANOVA

Multivariate Analysis of Variance

One-way MANOVA

Compare treatment means for h different populations

Population 1: $\mathbf{y}_{11}, \mathbf{y}_{12}, \dots, \mathbf{y}_{1n_1} \sim iddN_p(_1,)$

$\vdots$

Population h: $\mathbf{y}_{h1}, \mathbf{y}_{h2}, \dots, \mathbf{y}_{hn_h} \sim iddN_p(_h,)$

Assumptions

1. Independent random samples from h different populations
2. Common covariance matrices
3. Each population is multivariate **normal**

Calculate the summary statistics$\bar{\mathbf{y}}_i, \mathbf{S}$ and the pooled estimate of the covariance matrix $\mathbf{S}$

Similar to the univariate one-way ANVOA, we can use the effects model formulation $_i = + _i$, where

- $_i$ is the population mean for population i
- is the overall mean effect
- $_i$ is the treatment effect of the i-th treatment.

For the one-way model: $\mathbf{y}_{ij} = \mu + \tau_i + \epsilon_{ij}$ for $i = 1, .., h; j = 1, ..., n_i$ and $\epsilon_{ij} \sim N_p(\mathbf{0},)$

However, the above model is over-parameterized (i.e., infinite number of ways to define and the $_i$'s such that they add up to μ_i. Thus we can constrain by having

$$\sum_{i=1}^{h} n_i \tau_i = 0$$

or

$$_h = 0$$

The observational equivalent of the effects model is

$$\begin{aligned}\mathbf{y}_{ij} &= \bar{\mathbf{y}} + (\bar{\mathbf{y}}_i - \bar{\mathbf{y}}) + (\mathbf{y}_{ij} - \bar{\mathbf{y}}_i) \\ &= \text{overall sample mean} + \text{treatement effect} + \text{residual (under univariate ANOVA)}\end{aligned}$$

After manipulation

$$\sum_{i=1}^{h}\sum_{j=1}^{n_i}(\bar{\mathbf{y}}_{ij} - \bar{\mathbf{y}})(\bar{\mathbf{y}}_{ij} - \bar{\mathbf{y}})' = \sum_{i=1}^{h} n_i(\bar{\mathbf{y}}_i - \bar{\mathbf{y}})(\bar{\mathbf{y}}_i - \bar{\mathbf{y}})' + \sum_{i=1}^{h}\sum_{j=1}^{n_i}(\bar{\mathbf{y}}_{ij} - \bar{\mathbf{y}})(\bar{\mathbf{y}}_{ij} - \bar{\mathbf{y}}_i)'$$

LHS = Total corrected sums of squares and cross products (SSCP) matrix

RHS =

- 1st term = Treatment (or between subjects) sum of squares and cross product matrix (denoted H;B)
- 2nd term = residual (or within subject) SSCP matrix denoted (E;W)

Note:

$$\mathbf{E} = (n_1 - 1)\mathbf{S}_1 + ... + (n_h - 1)\mathbf{S}_h = (n - h)\mathbf{S}$$

MANOVA table

Table 21.1: MONOVA table

Source	SSCP	df
Treatment	$\mathbf{H}$	$h - 1$
Residual (error)	$\mathbf{E}$	$\sum_{i=1}^{h} n_i - h$
Total Corrected	$\mathbf{H} + \mathbf{E}$	$\sum_{i=1}^{h} n_i - 1$

$$H_0 : \tau_1 = \tau_2 = \cdots = \tau_h = \mathbf{0}$$

We consider the relative "sizes" of $\mathbf{E}$ and $\mathbf{H} + \mathbf{E}$

Wilk's Lambda

Define Wilk's Lambda

$$\Lambda^* = \frac{|\mathbf{E}|}{|\mathbf{H} + \mathbf{E}|}$$

Properties:

1. Wilk's Lambda is equivalent to the F-statistic in the univariate case
2. The exact distribution of Λ^* can be determined for especial cases.
3. For large sample sizes, reject H_0 if

$$-(\sum_{i=1}^{h} n_i - 1 - \frac{p+h}{2}) \log(\Lambda^*) > \chi^2_{(1-\alpha, p(h-1))}$$

21.1.1 Testing General Hypotheses

- h different treatments
- with the i-th treatment
- applied to n_i subjects that
- are observed for p repeated measures.

Consider this a p dimensional obs on a random sample from each of h different treatment populations.

$$\mathbf{y}_{ij} = \ + \ _i + \ _{ij}$$

for $i = 1, .., h$ and $j = 1, .., n_i$

Equivalently,

$$\mathbf{Y} = \mathbf{XB} +$$

where $n = \sum_{i=1}^{h} n_i$ and with restriction $\ _h = 0$

$$\mathbf{Y}_{(n\times p)} = \begin{bmatrix} \mathbf{y}'_{11} \\ \vdots \\ \mathbf{y}'_{1n_1} \\ \vdots \\ \mathbf{y}'_{hn_h} \end{bmatrix}, \mathbf{B}_{(h\times p)} = \begin{bmatrix} ' \\ ' \\ {}_1 \\ \vdots \\ ' \\ {}_{h-1} \end{bmatrix}, \ {}_{(n\times p)} = \begin{bmatrix} \epsilon'_{11} \\ \vdots \\ \epsilon'_{1n_1} \\ \vdots \\ \epsilon'_{hn_h} \end{bmatrix}$$

$$\mathbf{X}_{(n\times h)} = \begin{bmatrix} 1 & 1 & 0 & ... & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 1 & 0 & ... & 0 \\ \vdots & \vdots & \vdots & ... & \vdots \\ 1 & 0 & 0 & ... & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 0 & 0 & ... & 0 \end{bmatrix}$$

Estimation

$$\hat{\mathbf{B}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$$

Rows of $\mathbf{Y}$ are independent (i.e., $var(\mathbf{Y}) = \mathbf{I}_n \otimes \quad$, an $np \times np$ matrix, where $\otimes$ is the Kronecker product).

$$H_0 : \mathbf{LBM} = 0 H_a : \mathbf{LBM} \neq 0$$

where

- $\mathbf{L}$ is a $g \times h$ matrix of full row rank ($g \leq h$) = comparisons across groups
- $\mathbf{M}$ is a $p \times u$ matrix of full column rank ($u \leq p$) = comparisons across traits

The general treatment corrected sums of squares and cross product is

$$\mathbf{H} = \mathbf{M}'\mathbf{Y}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-\mathbf{1}}\mathbf{L}'[\mathbf{L}(\mathbf{X}'\mathbf{X})^{-\mathbf{1}}\mathbf{L}']^{-\mathbf{1}}\mathbf{L}(\mathbf{X}'\mathbf{X})^{-\mathbf{1}}\mathbf{X}'\mathbf{Y}\mathbf{M}$$

or for the null hypothesis $H_0 : \mathbf{LBM} = \mathbf{D}$

$$\mathbf{H} = (\mathbf{L}\hat{\mathbf{B}}\mathbf{M} - \mathbf{D})'[\mathbf{X}(\mathbf{X}'\mathbf{X})^{-\mathbf{1}}\mathbf{L}]^{-1}(\mathbf{L}\hat{\mathbf{B}}\mathbf{M} - \mathbf{D})$$

The general matrix of residual sums of squares and cross product

$$\mathbf{E} = \mathbf{M}'\mathbf{Y}'[\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-\mathbf{1}}\mathbf{X}']\mathbf{Y}\mathbf{M} = \mathbf{M}'[\mathbf{Y}'\mathbf{Y} - \hat{\mathbf{B}}'(\mathbf{X}'\mathbf{X})^{-\mathbf{1}}\hat{\mathbf{B}}]\mathbf{M}$$

We can compute the following statistic eigenvalues of $\mathbf{HE}^{-1}$

- Wilk's Criterion: $\Lambda^* = \frac{|\mathbf{E}|}{|\mathbf{H}+\mathbf{E}|}$. The df depend on the rank of $\mathbf{L}, \mathbf{M}, \mathbf{X}$
- Lawley-Hotelling Trace: $U = tr(\mathbf{H}\mathbf{E}^{-1})$
- Pillai Trace: $V = tr(\mathbf{H}(\mathbf{H}+\mathbf{E}^{-1})$
- Roy's Maximum Root: largest eigenvalue of $\mathbf{H}\mathbf{E}^{-1}$

If H_0 is true and n is large, $-(n-1-\frac{p+h}{2})\ln\Lambda^* \sim \chi^2_{p(h-1)}$. Some special values of p and h can give exact F-dist under H_0

```
# One-way MANOVA

library(car)
library(emmeans)
```

```
## Warning: package 'emmeans' was built under R version 4.0.5
```

```
library(profileR)
```

```
## Warning: package 'profileR' was built under R version 4.0.5
```

```
library(tidyverse)

## Read in the data
gpagmat <- read.table("images/gpagmat.dat")

## Change the variable names
names(gpagmat) <- c("y1", "y2", "admit")

## Check the structure
str(gpagmat)
```

```
## 'data.frame':    85 obs. of  3 variables:
##  $ y1   : num  2.96 3.14 3.22 3.29 3.69 3.46 3.03 3.19 3.63 3.59 ...
##  $ y2   : int  596 473 482 527 505 693 626 663 447 588 ...
##  $ admit: int  1 1 1 1 1 1 1 1 1 1 ...
```

```
## Plot the data
gg <- ggplot(gpagmat, aes(x = y1, y = y2)) +
    geom_text(aes(label = admit, col = as.character(admit))) +
    scale_color_discrete(name = "Admission",
                         labels = c("Admit", "Do not admit", "Borderline")) +
    scale_x_continuous(name = "GPA") +
    scale_y_continuous(name = "GMAT")

## Fit one-way MANOVA
oneway_fit <- manova(cbind(y1, y2) ~ admit, data = gpagmat)
summary(oneway_fit, test = "Wilks")
```

```
##           Df  Wilks approx F num Df den Df    Pr(>F)
## admit      1 0.6126   25.927      2     82 1.881e-09 ***
## Residuals 83
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# reject the null of equal multivariate mean vectors between the three admmission grou
```

```
# Repeated Measures MANOVA


## Create data frame
stress <- data.frame(
    subject = 1:8,
    begin = c(3, 2, 5, 6, 1, 5, 1, 5),
    middle = c(3, 4, 3, 7, 4, 7, 1, 2),
    final = c(6, 7, 4, 7, 6, 7, 3, 5)
)

# If independent = time with 3 levels -> univariate ANOVA (require sphericity assumpti
# If each level of indepednet time as a separate variable -> MANOVA (does not require 



## MANOVA
stress_mod <- lm(cbind(begin, middle, final) ~ 1, data = stress)
idata <-
    data.frame(time = factor(
        c("begin", "middle", "final"),
        levels = c("begin", "middle", "final")
    ))
repeat_fit <-
    Anova(
        stress_mod,
        idata = idata,
        idesign = ~ time,
        icontrasts = "contr.poly"
    )
summary(repeat_fit) # can't reject the null hypothesis of sphericity, hence univariate
```

```
##
## Type III Repeated Measures MANOVA Tests:
##
## ------------------------------------------
##
## Term: (Intercept)
##
```

```
##  Response transformation matrix:
##        (Intercept)
## begin            1
## middle           1
## final            1
##
## Sum of squares and products for the hypothesis:
##             (Intercept)
## (Intercept)        1352
##
## Multivariate Tests: (Intercept)
##                  Df test stat approx F num Df den Df     Pr(>F)
## Pillai            1  0.896552 60.66667      1      7 0.00010808 ***
## Wilks             1  0.103448 60.66667      1      7 0.00010808 ***
## Hotelling-Lawley  1  8.666667 60.66667      1      7 0.00010808 ***
## Roy               1  8.666667 60.66667      1      7 0.00010808 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ------------------------------------------
##
## Term: time
##
##  Response transformation matrix:
##               time.L     time.Q
## begin  -7.071068e-01  0.4082483
## middle -7.850462e-17 -0.8164966
## final   7.071068e-01  0.4082483
##
## Sum of squares and products for the hypothesis:
##           time.L   time.Q
## time.L 18.062500 6.747781
## time.Q  6.747781 2.520833
##
## Multivariate Tests: time
##                  Df test stat approx F num Df den Df   Pr(>F)
## Pillai            1 0.7080717 7.276498      2      6 0.024879 *
## Wilks             1 0.2919283 7.276498      2      6 0.024879 *
## Hotelling-Lawley  1 2.4254992 7.276498      2      6 0.024879 *
## Roy               1 2.4254992 7.276498      2      6 0.024879 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Univariate Type III Repeated-Measures ANOVA Assuming Sphericity
##
##             Sum Sq num Df Error SS den Df F value    Pr(>F)
## (Intercept) 450.67      1    52.00      7 60.6667 0.0001081 ***
## time         20.58      2    24.75     14  5.8215 0.0144578 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Mauchly Tests for Sphericity
##
##      Test statistic p-value
## time         0.7085 0.35565
##
##
## Greenhouse-Geisser and Huynh-Feldt Corrections
##  for Departure from Sphericity
##
##       GG eps Pr(>F[GG])
## time 0.77429    0.02439 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##         HF eps Pr(>F[HF])
## time 0.9528433 0.01611634
```

```
# we also see linear significant time effect, but no quadratic time effect


## Polynomial contrasts
# What is the reference for the marginal means?
ref_grid(stress_mod, mult.name = "time")
```

```
## 'emmGrid' object with variables:
##     1 = 1
##     time = multivariate response levels: begin, middle, final
```

```
# marginal means for the levels of time
contr_means <- emmeans(stress_mod, ~ time, mult.name = "time")
contrast(contr_means, method = "poly")
```

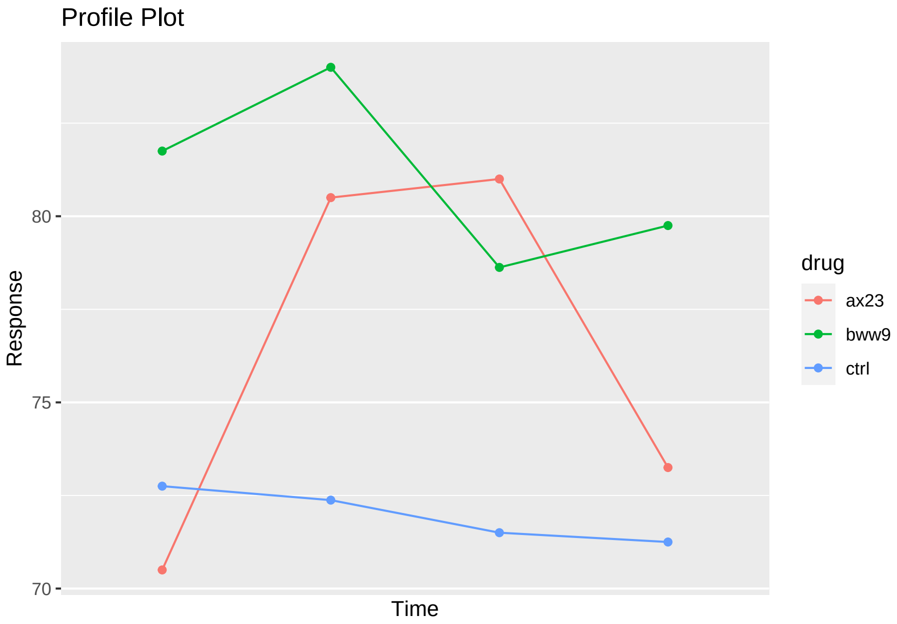
```
##  contrast  estimate    SE df t.ratio p.value
##  linear        2.12 0.766  7 2.773   0.0276
##  quadratic     1.38 0.944  7 1.457   0.1885
```

```
# MANOVA

## Read in Data
heart <- read.table("images/heart.dat")
names(heart) <- c("drug", "y1", "y2", "y3", "y4")
## Create a subject ID nested within drug
```

```
heart <- heart %>%
    group_by(drug) %>%
    mutate(subject = row_number()) %>%
    ungroup()
str(heart)
```

```
## tibble[,6] [24 x 6] (S3: tbl_df/tbl/data.frame)
##  $ drug   : chr [1:24] "ax23" "ax23" "ax23" "ax23" ...
##  $ y1     : int [1:24] 72 78 71 72 66 74 62 69 85 82 ...
##  $ y2     : int [1:24] 86 83 82 83 79 83 73 75 86 86 ...
##  $ y3     : int [1:24] 81 88 81 83 77 84 78 76 83 80 ...
##  $ y4     : int [1:24] 77 82 75 69 66 77 70 70 80 84 ...
##  $ subject: int [1:24] 1 2 3 4 5 6 7 8 1 2 ...
```

```
## Create means summary for profile plot, pivot longer for plotting with ggplot
heart_means <- heart %>%
    group_by(drug) %>%
    summarize_at(vars(starts_with("y")), mean) %>%
    ungroup() %>%
    pivot_longer(-drug, names_to = "time", values_to = "mean") %>%
    mutate(time = as.numeric(as.factor(time)))
gg_profile <- ggplot(heart_means, aes(x = time, y = mean)) +
    geom_line(aes(col = drug)) +
    geom_point(aes(col = drug)) +
    ggtitle("Profile Plot") +
    scale_y_continuous(name = "Response") +
    scale_x_discrete(name = "Time")
gg_profile
```



```
## Fit model
heart_mod <- lm(cbind(y1, y2, y3, y4) ~ drug, data = heart)
man_fit <- car::Anova(heart_mod)
summary(man_fit)
```

```
##
## Type II MANOVA Tests:
##
## Sum of squares and products for error:
##        y1      y2      y3     y4
## y1 641.00 601.750 535.250 426.00
## y2 601.75 823.875 615.500 534.25
## y3 535.25 615.500 655.875 555.25
## y4 426.00 534.250 555.250 674.50
##
## ------------------------------------------
##
## Term: drug
##
## Sum of squares and products for the hypothesis:
##        y1       y2       y3    y4
## y1 567.00 335.2500  42.7500 387.0
## y2 335.25 569.0833 404.5417 367.5
## y3  42.75 404.5417 391.0833 171.0
## y4 387.00 367.5000 171.0000 316.0
```

```
##
## Multivariate Tests: drug
##                  Df test stat  approx F num Df den Df     Pr(>F)
## Pillai            2  1.283456  8.508082      8     38 1.5010e-06 ***
## Wilks             2  0.079007 11.509581      8     36 6.3081e-08 ***
## Hotelling-Lawley  2  7.069384 15.022441      8     34 3.9048e-09 ***
## Roy               2  6.346509 30.145916      4     19 5.4493e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# reject the null hypothesis of no difference in means between treatments
```

reject the null hypothesis of no difference in means between treatments

```
## Contrasts
heart$drug <- factor(heart$drug)
L <- matrix(c(0, 2,
              1, -1,-1, -1), nrow = 3, byrow = T)
colnames(L) <- c("bww9:ctrl", "ax23:rest")
rownames(L) <- unique(heart$drug)
contrasts(heart$drug) <- L
contrasts(heart$drug)
```

```
##      bww9:ctrl ax23:rest
## ax23         0         2
## bww9         1        -1
## ctrl        -1        -1
```

```
# do not set contrast L if you do further analysis (e.g., Anova, lm)
# do M matrix instead

M <- matrix(c(1, -1, 0, 0,
              0, 1, -1, 0,
              0, 0, 1, -1), nrow = 4)
## update model to test contrasts
heart_mod2 <- update(heart_mod)
coef(heart_mod2)
```

```
##                  y1         y2        y3    y4
## (Intercept)   75.00 78.9583333 77.041667 74.75
## drugbww9:ctrl  4.50  5.8125000  3.562500  4.25
## drugax23:rest -2.25  0.7708333  1.979167 -0.75
```

```
# Hypothesis test for bww9 vs control after transformation M
# same as linearHypothesis(heart_mod, hypothesis.matrix = c(0,1,-1), P = M)
bww9vctrl <-
    car::linearHypothesis(heart_mod2,
                          hypothesis.matrix = c(0, 1, 0),
                      P = M)
bww9vctrl
```

```
##
##  Response transformation matrix:
##    [,1] [,2] [,3]
## y1    1    0    0
## y2   -1    1    0
## y3    0   -1    1
## y4    0    0   -1
##
## Sum of squares and products for the hypothesis:
##          [,1]   [,2]     [,3]
## [1,]  27.5625 -47.25  14.4375
## [2,] -47.2500  81.00 -24.7500
## [3,]  14.4375 -24.75   7.5625
##
## Sum of squares and products for error:
##          [,1]     [,2]    [,3]
## [1,]  261.375 -141.875  28.000
## [2,] -141.875  248.750 -19.375
## [3,]   28.000  -19.375 219.875
##
## Multivariate Tests:
##                  Df test stat approx F num Df den Df Pr(>F)
## Pillai            1 0.2564306 2.184141      3     19 0.1233
## Wilks             1 0.7435694 2.184141      3     19 0.1233
## Hotelling-Lawley  1 0.3448644 2.184141      3     19 0.1233
## Roy               1 0.3448644 2.184141      3     19 0.1233
```

```
bww9vctrl <-
    car::linearHypothesis(heart_mod,
                      hypothesis.matrix = c(0, 1, -1),
                      P = M)
bww9vctrl
```

```
##
##  Response transformation matrix:
##    [,1] [,2] [,3]
## y1    1    0    0
## y2   -1    1    0
## y3    0   -1    1
## y4    0    0   -1
##
## Sum of squares and products for the hypothesis:
##          [,1]   [,2]     [,3]
## [1,]  27.5625 -47.25  14.4375
## [2,] -47.2500  81.00 -24.7500
## [3,]  14.4375 -24.75   7.5625
##
## Sum of squares and products for error:
##          [,1]     [,2]    [,3]
## [1,]  261.375 -141.875  28.000
## [2,] -141.875  248.750 -19.375
## [3,]   28.000  -19.375 219.875
##
## Multivariate Tests:
##                  Df test stat approx F num Df den Df Pr(>F)
## Pillai            1 0.2564306 2.184141      3     19 0.1233
## Wilks             1 0.7435694 2.184141      3     19 0.1233
## Hotelling-Lawley  1 0.3448644 2.184141      3     19 0.1233
## Roy               1 0.3448644 2.184141      3     19 0.1233
```

there is no significant difference in means between the control and bww9 drug

```r
# Hypothesis test for ax23 vs rest after transformation M
axx23vrest <-
    car::linearHypothesis(heart_mod2,
                     hypothesis.matrix = c(0, 0, 1),
                     P = M)
axx23vrest
```

```
##
##  Response transformation matrix:
##    [,1] [,2] [,3]
## y1    1    0    0
## y2   -1    1    0
## y3    0   -1    1
## y4    0    0   -1
##
## Sum of squares and products for the hypothesis:
##           [,1]       [,2]      [,3]
## [1,]  438.0208  175.20833 -395.7292
## [2,]  175.2083   70.08333 -158.2917
## [3,] -395.7292 -158.29167  357.5208
##
## Sum of squares and products for error:
##          [,1]     [,2]    [,3]
## [1,]  261.375 -141.875  28.000
## [2,] -141.875  248.750 -19.375
## [3,]   28.000  -19.375 219.875
##
## Multivariate Tests:
```

```
##                   Df test stat approx F num Df den Df     Pr(>F)
## Pillai             1  0.855364 37.45483      3     19 3.5484e-08 ***
## Wilks              1  0.144636 37.45483      3     19 3.5484e-08 ***
## Hotelling-Lawley   1  5.913921 37.45483      3     19 3.5484e-08 ***
## Roy                1  5.913921 37.45483      3     19 3.5484e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
axx23vrest <-
    car::linearHypothesis(heart_mod,
                     hypothesis.matrix = c(2, -1, 1),
                     P = M)
axx23vrest
```

```
##
##  Response transformation matrix:
##    [,1] [,2] [,3]
## y1    1    0    0
## y2   -1    1    0
## y3    0   -1    1
## y4    0    0   -1
##
## Sum of squares and products for the hypothesis:
##           [,1]       [,2]      [,3]
## [1,]  402.5208  127.41667 -390.9375
## [2,]  127.4167   40.33333 -123.7500
## [3,] -390.9375 -123.75000  379.6875
##
## Sum of squares and products for error:
##          [,1]     [,2]    [,3]
## [1,]  261.375 -141.875  28.000
## [2,] -141.875  248.750 -19.375
## [3,]   28.000  -19.375 219.875
##
## Multivariate Tests:
##                   Df test stat approx F num Df den Df     Pr(>F)
## Pillai             1  0.842450 33.86563      3     19 7.9422e-08 ***
## Wilks              1  0.157550 33.86563      3     19 7.9422e-08 ***
## Hotelling-Lawley   1  5.347205 33.86563      3     19 7.9422e-08 ***
## Roy                1  5.347205 33.86563      3     19 7.9422e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

there is a significant difference in means between ax23 drug treatment and the rest of the treatments

21.1.2 Profile Analysis

Examine similarities between the treatment effects (between subjects), which is useful for longitudinal analysis. Null is that all treatments have the same average effect.

$$H_0 : \mu_1 = \mu_2 = \dots = \mu_h$$

Equivalently,

$$H_0 : \tau_1 = \tau_2 = \dots = \tau_h$$

The exact nature of the similarities and differences between the treatments can be examined under this analysis.

Sequential steps in profile analysis:

1. Are the profiles **parallel**? (i.e., is there no interaction between treatment and time)
2. Are the profiles **coincidental**? (i.e., are the profiles identical?)
3. Are the profiles **horizontal**? (i.e., are there no differences between any time points?)

If we reject the null hypothesis that the profiles are parallel, we can test

- Are there differences among groups within some subset of the total time points?
- Are there differences among time points in a particular group (or groups)?
- Are there differences within some subset of the total time points in a particular group (or groups)?

Example

- 4 times (p = 4)
- 3 treatments (h=3)

21.1.2.1 Parallel Profile

Are the profiles for each population identical expect for a mean shift?

$$H_0 : \mu_{11} - \mu_{21} - \mu_{12} - \mu_{22} = \dots = \mu_{1t} - \mu_{2t} \mu_{11} - \mu_{31} - \mu_{12} - \mu_{32} = \dots = \mu_{1t} - \mu_{3t} \dots$$

for $h-1$ equations

Equivalently,

$$H_0 : \mathbf{LBM} = \mathbf{0}$$

$$\mathbf{LBM} = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} \mu_{11} & \dots & \mu_{14} \\ \mu_{21} & \dots & \mu_{24} \\ \mu_{31} & \dots & \mu_{34} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} = \mathbf{0}$$

where this is the cell means parameterization of **B**

The multiplication of the first 2 matrices **LB** is

$$\begin{bmatrix} \mu_{11} - \mu_{21} & \mu_{12} - \mu_{22} & \mu_{13} - \mu_{23} & \mu_{14} - \mu_{24} \\ \mu_{11} - \mu_{31} & \mu_{12} - \mu_{32} & \mu_{13} - \mu_{33} & \mu_{14} - \mu_{34} \end{bmatrix}$$

which is the differences in treatment means at the same time

Multiplying by **M**, we get the comparison across time

$$\begin{bmatrix} (\mu_{11} - \mu_{21}) - (\mu_{12} - \mu_{22}) & (\mu_{11} - \mu_{21}) - (\mu_{13} - \mu_{23}) & (\mu_{11} - \mu_{21}) - (\mu_{14} - \mu_{24}) \\ (\mu_{11} - \mu_{31}) - (\mu_{12} - \mu_{32}) & (\mu_{11} - \mu_{31}) - (\mu_{13} - \mu_{33}) & (\mu_{11} - \mu_{31}) - (\mu_{14} - \mu_{34}) \end{bmatrix}$$

Alternatively, we can also use the effects parameterization

$$\mathbf{LBM} = \begin{bmatrix} 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} \mu' \\ \tau_1' \\ \tau_2' \\ \tau_3' \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} = \mathbf{0}$$

In both parameterizations, $rank(\mathbf{L}) = h - 1$ and $rank(\mathbf{M}) = p - 1$

We could also choose **L** and **M** in other forms

$$\mathbf{L} = \begin{bmatrix} 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

and

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \end{bmatrix}$$

and still obtain the same result.

21.1.2.2 Coincidental Profiles

After we have evidence that the profiles are parallel (i.e., fail to reject the parallel profile test), we can ask whether they are identical?

Given profiles are **parallel**, then if the sums of the components of μ_i are identical for all the treatments, then the profiles are **identical**.

$$H_0 : \mathbf{1'}_p \mu_1 = \mathbf{1'}_p \mu_2 = \dots = \mathbf{1'}_p \mu_h$$

Equivalently,

$$H_0 : \mathbf{LBM} = \mathbf{0}$$

where for the cell means parameterization

$$\mathbf{L} = \left[\begin{array}{ccc} 1 & 0 & -1 \\ 0 & 1 & -1 \end{array} \right]$$

and

$$\mathbf{M} = \left[\begin{array}{cccc} 1 & 1 & 1 & 1 \end{array} \right]'$$

multiplication yields

$$\left[\begin{array}{c} (\mu_{11} + \mu_{12} + \mu_{13} + \mu_{14}) - (\mu_{31} + \mu_{32} + \mu_{33} + \mu_{34}) \\ (\mu_{21} + \mu_{22} + \mu_{23} + \mu_{24}) - (\mu_{31} + \mu_{32} + \mu_{33} + \mu_{34}) \end{array} \right] = \left[\begin{array}{c} 0 \\ 0 \end{array} \right]$$

Different choices of **L** and **M** can yield the same result

21.1.2.3 Horizontal Profiles

Given that we can't reject the null hypothesis that all h profiles are the same, we can ask whether all of the elements of the common profile equal? (i.e., horizontal)

$$H_0 : \mathbf{LBM} = \mathbf{0}$$

$$\mathbf{L} = \left[\begin{array}{ccc} 1 & 0 & 0 \end{array} \right]$$

and

$$
\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \end{bmatrix}
$$

hence,

$$
\begin{bmatrix} (\mu_{11} - \mu_{12}) & (\mu_{12} - \mu_{13}) & (\mu_{13} + \mu_{14}) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}
$$

Note:

- If we fail to reject all 3 hypotheses, then we fail to reject the null hypotheses of both no difference between treatments and no differences between traits.

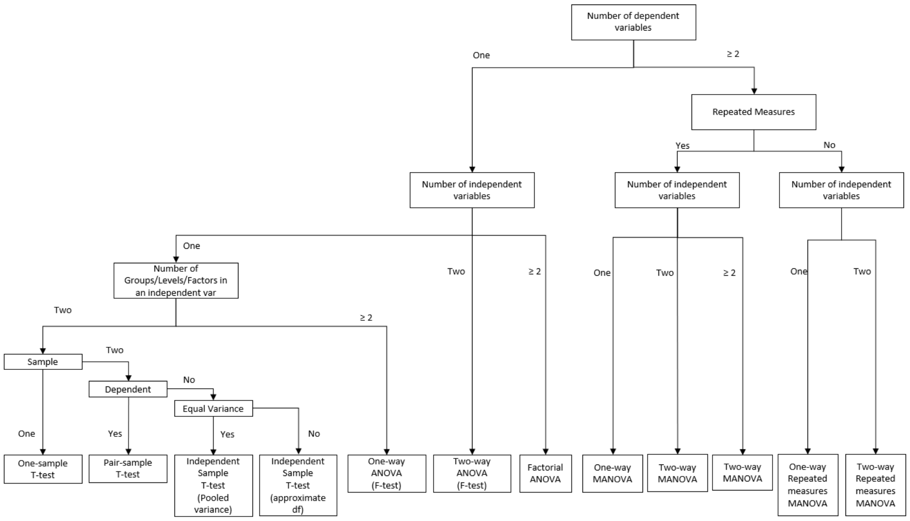
Test	Equivalent test for
Parallel profile	Interaction
Coincidental profile	main effect of between-subjects factor
Horizontal profile	main effect of repeated measures factor

```r
profile_fit <-
    pbg(
        data = as.matrix(heart[, 2:5]),
        group = as.matrix(heart[, 1]),
        original.names = TRUE,
        profile.plot = FALSE
    )
summary(profile_fit)
```

```
## Call:
## pbg(data = as.matrix(heart[, 2:5]), group = as.matrix(heart[,
##     1]), original.names = TRUE, profile.plot = FALSE)
##
## Hypothesis Tests:
## $`Ho: Profiles are parallel`
##   Multivariate.Test Statistic  Approx.F num.df den.df      p.value
## 1             Wilks 0.1102861 12.737599      6     38 7.891497e-08
## 2            Pillai 1.0891707  7.972007      6     40 1.092397e-05
## 3  Hotelling-Lawley 6.2587852 18.776356      6     36 9.258571e-10
## 4               Roy 5.9550887 39.700592      3     20 1.302458e-08
##
## $`Ho: Profiles have equal levels`
##             Df Sum Sq Mean Sq F value  Pr(>F)
## group        2  328.7  164.35   5.918 0.00915 **
## Residuals   21  583.2   27.77
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## $`Ho: Profiles are flat`
##          F df1 df2      p-value
## 1 14.30928   3  19 4.096803e-05
```

```
# reject null hypothesis of parallel profiles
# reject the null hypothesis of coincidental profiles
# reject the null hypothesis that the profiles are flat
```

21.1.3 Summary


Number of dependent variables
One
≥ 2
Repeated Measures
Yes
No
Number of independent variables
Number of independent variables
Number of independent variables
One
Two
≥ 2
One
Two
≥ 2
One
Two
Number of Groups/Levels/Factors in an independent var
Two
≥ 2
Sample
Two
Dependent
No
Equal Variance
One
Yes
Yes
No
One-sample T-test
Pair-sample T-test
Independent Sample T-test (Pooled variance)
Independent Sample T-test (approximate df)
One-way ANOVA (F-test)
Two-way ANOVA (F-test)
Factorial ANOVA
One-way MANOVA
Two-way MANOVA
Two-way MANOVA
One-way Repeated measures MANOVA
Two-way Repeated measures MANOVA

21.2 Principal Components

- Unsupervised learning
- find important features
- reduce the dimensions of the data set
- “decorrelate” multivariate vectors that have dependence.
- uses eigenvector/eigvenvalue decomposition of covariance (correlation) matrices.

According to the “spectral decomposition theorem”, if $_{p\times p}$ i s a positive semi-definite, symmetric, real matrix, then there exists an orthogonal matrix $\mathbf{A}$ such that $\mathbf{A}'\ \mathbf{A} = \Lambda$ where Λ is a diagonal matrix containing the eigenvalues

$$= \begin{pmatrix} \lambda_1 & 0 & ... & 0 \\ 0 & \lambda_2 & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & \lambda_p \end{pmatrix}$$

$$\mathbf{A} = \begin{pmatrix} \mathbf{a}_1 & \mathbf{a}_2 & ... & \mathbf{a}_p \end{pmatrix}$$

the i-th column of $\mathbf{A}$, $\mathbf{a}_i$, is the i-th $p \times 1$ eigenvector of   that corresponds to the eigenvalue, λ_i , where $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p$. Alternatively, express in matrix decomposition:

$$= \mathbf{A} \quad \mathbf{A}'$$

$$= \mathbf{A} \begin{pmatrix} \lambda_1 & 0 & ... & 0 \\ 0 & \lambda_2 & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & \lambda_p \end{pmatrix} \mathbf{A}' = \sum_{i=1}^{p} \lambda_i \mathbf{a}_i \mathbf{a}_i'$$

where the outer product $\mathbf{a}_i \mathbf{a}_i'$ is a $p \times p$ matrix of rank 1.

For example,

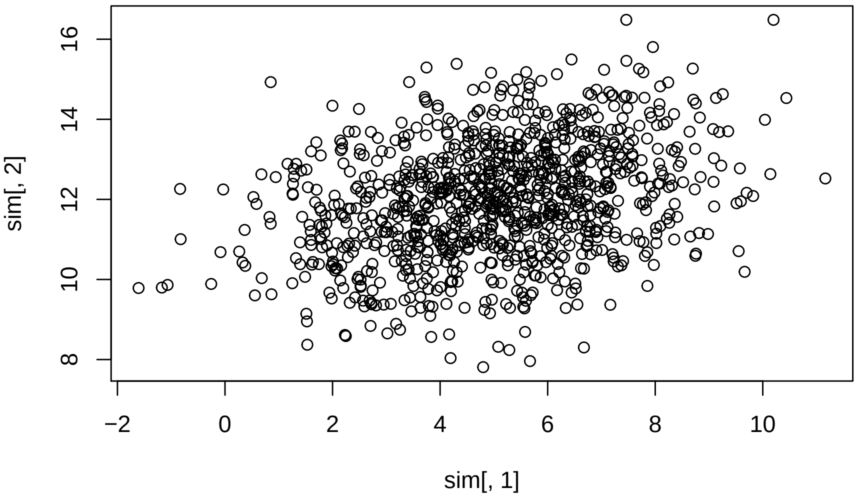
$\mathbf{x} \sim N_2(\; , \;)$

$$= \begin{pmatrix} 5 \\ 12 \end{pmatrix}; \quad = \begin{pmatrix} 4 & 1 \\ 1 & 2 \end{pmatrix}$$

```
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
mu = as.matrix(c(5,12))
Sigma = matrix(c(4,1,1,2),nrow = 2, byrow = T)
sim <- mvrnorm(n = 1000, mu = mu, Sigma = Sigma)
plot(sim[,1],sim[,2])
```



Here,

$$\mathbf{A} = \begin{pmatrix} 0.9239 & -0.3827 \\ 0.3827 & 0.9239 \end{pmatrix}$$

Columns of **A** are the eigenvectors for the decomposition

Under matrix multiplication ($\mathbf{A}' \ \mathbf{A}$ or $\mathbf{A}'\mathbf{A}$), the off-diagonal elements equal to 0

Multiplying data by this matrix (i.e., projecting the data onto the orthogonal axes); the distriubiton of the resulting data (i.e., “scores”) is

$$N_2(\mathbf{A}' \ , \mathbf{A}' \ \mathbf{A}) = N_2(\mathbf{A}' \ , \ )$$

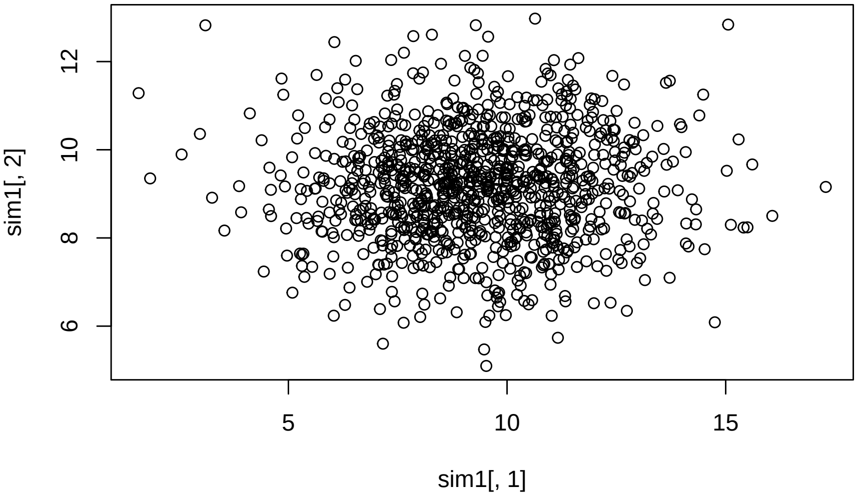
Equivalently,

$$\mathbf{y} = \mathbf{A}'\mathbf{x} \sim N \left[\begin{pmatrix} 9.2119 \\ 9.1733 \end{pmatrix}, \begin{pmatrix} 4.4144 & 0 \\ 0 & 1.5859 \end{pmatrix} \right]$$

```
A_matrix = matrix(c(0.9239,-0.3827,0.3827,0.9239),nrow = 2, byrow = T)
t(A_matrix) %*% A_matrix
```

```
##          [,1]     [,2]
## [1,] 1.000051 0.000000
## [2,] 0.000000 1.000051
```

```
sim1 <- mvrnorm(n = 1000, mu = t(A_matrix) %*% mu, Sigma = t(A_matrix) %*% Sigma %*% A_
plot(sim1[,1],sim1[,2])
```



No more dependence in the data structure, plot

Notes:

- The i-th eigenvalue is the variance of a linear combination of the elements of $\mathbf{x}$; $var(y_i) = var(\mathbf{a_i'}\mathbf{x}) = \lambda_i$
- The values on the transformed set of axes (i.e., the y_i's) are called the scores. These are the orthogonal projections of the data onto the "new principal component axes
- Variances of y_1 are greater than those for any other possible projection

Covariance matrix decomposition and projection onto orthogonal axes = PCA

21.2.1 Population Principal Components

$p \times 1$ vectors $\mathbf{x}_1, \dots, \mathbf{x}_n$ which are iid with $var(\mathbf{x}_i) =$

- The first PC is the linear combination $y_1 = \mathbf{a}_1'\mathbf{x} = a_{11}x_1 + \dots + a_{1p}x_p$ with $\mathbf{a}_1'\mathbf{a}_1 = 1$ such that $var(y_1)$ is the maximum of all linear combinations of $\mathbf{x}$ which have unit length
- The second PC is the linear combination $y_1 = \mathbf{a}_2'\mathbf{x} = a_{21}x_1 + \dots + a_{2p}x_p$ with $\mathbf{a}_2'\mathbf{a}_2 = 1$ such that $var(y_1)$ is the maximum of all linear combinations of $\mathbf{x}$

which have unit length and uncorrelated with y_1 (i.e., $cov(\mathbf{a}_1'\mathbf{x}, \mathbf{a}_2'\mathbf{x}) = 0$

- continues for all y_i to y_p

$\mathbf{a}_i$'s are those that make up the matrix $\mathbf{A}$ in the symmetric decomposition $\mathbf{A}'\ \mathbf{A} =$, where $var(y_1) = \lambda_1, \dots, var(y_p) = \lambda_p$ And the total variance of $\mathbf{x}$ is

$$
\begin{aligned}
var(x_1) + \dots + var(x_p) = tr(\Sigma) &= \lambda_1 + \dots + \lambda_p \\
&= var(y_1) + \dots + var(y_p)
\end{aligned}
$$

Data Reduction

To reduce the dimension of data from p (original) to k dimensions without much "loss of information", we can use properties of the population principal components

- Suppose $\approx \sum_{i=1}^k \lambda_i \mathbf{a}_i \mathbf{a}_i'$. Even thought the true variance-covariance matrix has rank p , it can be be well approximate by a matrix of rank k (k <p)
- New "traits" are linear combinations of the measured traits. We can attempt to make meaningful interpretation fo the combinations (with orthogonality constraints).
- The proportion of the total variance accounted for by the j-th principal component is

$$
\frac{var(y_j)}{\sum_{i=1}^p var(y_i)} = \frac{\lambda_j}{\sum_{i=1}^p \lambda_i}
$$

- The proportion of the total variation accounted for by the first k principal components is $\frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^p \lambda_i}$
- Above example , we have $4.4144/(4+2) = .735$ of the total variability can be explained by the first principal component

21.2.2 Sample Principal Components

Since is unknown, we use

$$
\mathbf{S} = \frac{1}{n-1} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})'
$$

Let $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \dots \geq \hat{\lambda}_p \geq 0$ be the eigenvalues of $\mathbf{S}$ and $\hat{\mathbf{a}}_1, \hat{\mathbf{a}}_2, \dots, \hat{\mathbf{a}}_p$ denote the eigenvectors of $\mathbf{S}$

Then, the i-th sample principal component score (or principal component or score) is

$$\hat{y}_{ij} = \sum_{k=1}^{p} \hat{a}_{ik} x_{kj} = \hat{\mathbf{a}}_i' \mathbf{x}_j$$

Properties of Sample Principal Components

- The estimated variance of $y_i = \hat{\mathbf{a}}_i' \mathbf{x}_j$ is $\hat{\lambda}_i$
- The sample covariance between $\hat{y}_i$ and $\hat{y}_{i'}$ is 0 when $i \neq i'$
- The proportion of the total sample variance accounted for by the i-th sample principal component is $\frac{\hat{\lambda}_i}{\sum_{k=1}^{p} \hat{\lambda}_k}$
- The estimated correlation between the i-th principal component score and the l-th attribute of $\mathbf{x}$ is

$$r_{x_l, \hat{y}_i} = \frac{\hat{a}_{il} \sqrt{\lambda_i}}{\sqrt{s_{ll}}}$$

- The correlation coefficient is typically used to interpret the components (i.e., if this correlation is high then it suggests that the l-th original trait is important in the i-th principle component). According to (Johnson and Wichern, 1988, pp.433-434), $r_{x_l, \hat{y}_i}$ only measures the univariate contribution of an individual X to a component Y without taking into account the presence of the other X's. Hence, some prefer $\hat{a}_{il}$ coefficient to interpret the principal component.
- $r_{x_l, \hat{y}_i}; \hat{a}_{il}$ are referred to as "loadings"

To use k principal components, we must calculate the scores for each data vector in the sample

$$\mathbf{y}_j = \begin{pmatrix} y_{1j} \\ y_{2j} \\ \vdots \\ y_{kj} \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{a}}_1' \mathbf{x}_j \\ \hat{\mathbf{a}}_2' \mathbf{x}_j \\ \vdots \\ \hat{\mathbf{a}}_k' \mathbf{x}_j \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{a}}_1' \\ \hat{\mathbf{a}}_2' \\ \vdots \\ \hat{\mathbf{a}}_k' \end{pmatrix} \mathbf{x}_j$$

Issues:

- Large sample theory exists for eigenvalues and eigenvectors of sample covariance matrices if inference is necessary. But we do not do inference with PCA, we only use it as exploratory or descriptive analysis.
- PC is not invariant to changes in scale (Exception: if all trait are resecaled by multiplying by the same constant, such as feet to inches).
  - PCA based on the correlation matrix $\mathbf{R}$ is different than that based on the covariance matrix

- PCA for the correlation matrix is just rescaling each trait to have unit variance
- Transform $\mathbf{x}$ to $\mathbf{z}$ where $z_{ij} = (x_{ij} - \bar{x}_i)/\sqrt{s_{ii}}$ where the denominator affects the PCA
- After transformation, $cov(\mathbf{z}) = \mathbf{R}$
- PCA on $\mathbf{R}$ is calculated in the same way as that on $\mathbf{S}$ (where $\hat{\lambda}_1 + \dots + \hat{\lambda}_p = p$)
- The use of $\mathbf{R}, \mathbf{S}$ depends on the purpose of PCA.
  - If the scale of the observations if different, covariance matrix is more preferable. but if they are dramatically different, analysis can still be dominated by the large variance traits.
- How many PCs to use can be guided by
  - Scree Graphs: plot the eigenvalues against their indices. Look for the "elbow" where the steep decline in the graph suddenly flattens out; or big gaps.
  - minimum Percent of total variation (e.g., choose enough components to have 50% or 90%). can be used for interpretations.
  - Kaiser's rule: use only those PC with eigenvalues larger than 1 (applied to PCA on the correlation matrix) - ad hoc
  - Compare to the eigenvalue scree plot of data to the scree plot when the data are randomized.

21.2.3 Application

PCA on the covariance matrix is usually not preferred due to the fact that PCA is not invariant to changes in scale. Hence, PCA on the correlation matrix is more preferred

The eigvenvectors may differ by a multiplication of -1 for different implementation, but same interpretation.

```r
library(tidyverse)
## Read in and check data
stock <- read.table("images/stock.dat")
names(stock) <- c("allied", "dupont", "carbide", "exxon", "texaco")
str(stock)
```

```
## 'data.frame':    100 obs. of  5 variables:
##  $ allied : num  0 0.027 0.1228 0.057 0.0637 ...
##  $ dupont : num  0 -0.04485 0.06077 0.02995 -0.00379 ...
##  $ carbide: num  0 -0.00303 0.08815 0.06681 -0.03979 ...
##  $ exxon  : num  0.0395 -0.0145 0.0862 0.0135 -0.0186 ...
##  $ texaco : num  0 0.0435 0.0781 0.0195 -0.0242 ...
```

```
## Covariance matrix of data
cov(stock)
```

```
##               allied       dupont      carbide        exxon       texaco
## allied  0.0016299269 0.0008166676 0.0008100713 0.0004422405 0.0005139715
## dupont  0.0008166676 0.0012293759 0.0008276330 0.0003868550 0.0003109431
## carbide 0.0008100713 0.0008276330 0.0015560763 0.0004872816 0.0004624767
## exxon   0.0004422405 0.0003868550 0.0004872816 0.0008023323 0.0004084734
## texaco  0.0005139715 0.0003109431 0.0004624767 0.0004084734 0.0007587370
```

```
## Correlation matrix of data
cor(stock)
```

```
##            allied    dupont   carbide     exxon    texaco
## allied  1.0000000 0.5769244 0.5086555 0.3867206 0.4621781
## dupont  0.5769244 1.0000000 0.5983841 0.3895191 0.3219534
## carbide 0.5086555 0.5983841 1.0000000 0.4361014 0.4256266
## exxon   0.3867206 0.3895191 0.4361014 1.0000000 0.5235293
## texaco  0.4621781 0.3219534 0.4256266 0.5235293 1.0000000
```

```
# cov(scale(stock)) # give the same result

## PCA with covariance
cov_pca <- prcomp(stock) # uses singular value decomposition for calculation and an N
# alternatively, princomp can do PCA via spectral decomposition, but it has worse nume

# eigen values
cov_results <- data.frame(eigen_values = cov_pca$sdev ^ 2)
cov_results %>%
    mutate(proportion = eigen_values / sum(eigen_values),
           cumulative = cumsum(proportion)) # first 2 PCs account for 73% variance in
```

```
##   eigen_values proportion cumulative
## 1 0.0035953867 0.60159252  0.6015925
## 2 0.0007921798 0.13255027  0.7341428
## 3 0.0007364426 0.12322412  0.8573669
## 4 0.0005086686 0.08511218  0.9424791
## 5 0.0003437707 0.05752091  1.0000000
```

```
# eigen vectors
cov_pca$rotation # prcomp calls rotation
```

```
##               PC1         PC2        PC3         PC4         PC5
## allied  0.5605914  0.73884565 -0.1260222  0.28373183 -0.20846832
## dupont  0.4698673 -0.09286987 -0.4675066 -0.68793190  0.28069055
## carbide 0.5473322 -0.65401929 -0.1140581  0.50045312 -0.09603973
## exxon   0.2908932 -0.11267353  0.6099196 -0.43808002 -0.58203935
## texaco  0.2842017  0.07103332  0.6168831  0.06227778  0.72784638
```

```
# princomp calls loadings.

# first PC = overall average
# second PC compares Allied to Carbide

## PCA with correlation
#same as scale(stock) %>% prcomp
cor_pca <- prcomp(stock, scale = T)



# eigen values
cor_results <- data.frame(eigen_values = cor_pca$sdev ^ 2)
cor_results %>%
    mutate(proportion = eigen_values / sum(eigen_values),
           cumulative = cumsum(proportion))
```

```
##   eigen_values proportion cumulative
## 1    2.8564869 0.57129738  0.5712974
## 2    0.8091185 0.16182370  0.7331211
## 3    0.5400440 0.10800880  0.8411299
## 4    0.4513468 0.09026936  0.9313992
## 5    0.3430038 0.06860076  1.0000000
```

```
# first egiven values corresponds to less variance than PCA based on the covariance matrix

# eigen vectors
cor_pca$rotation
```

```
##               PC1        PC2        PC3        PC4        PC5
## allied  0.4635405 -0.2408499  0.6133570 -0.3813727  0.4532876
## dupont  0.4570764 -0.5090997 -0.1778996 -0.2113068 -0.6749814
## carbide 0.4699804 -0.2605774 -0.3370355  0.6640985  0.3957247
## exxon   0.4216770  0.5252647 -0.5390181 -0.4728036  0.1794482
## texaco  0.4213291  0.5822416  0.4336029  0.3812273 -0.3874672
```

```
# interpretation of PC2 is different from above: it is a comparison of Allied, Dupont and Carbid
```

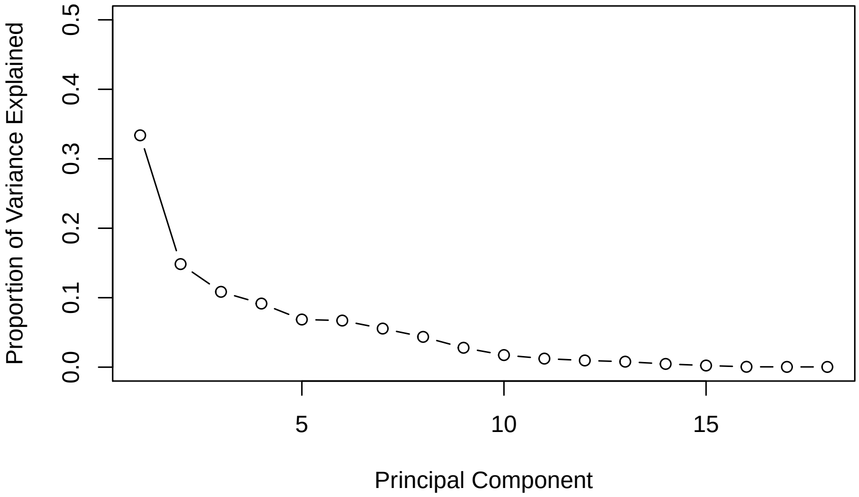
Covid Example

To reduce collinearity problem in this dataset, we can use principal components as regressors.

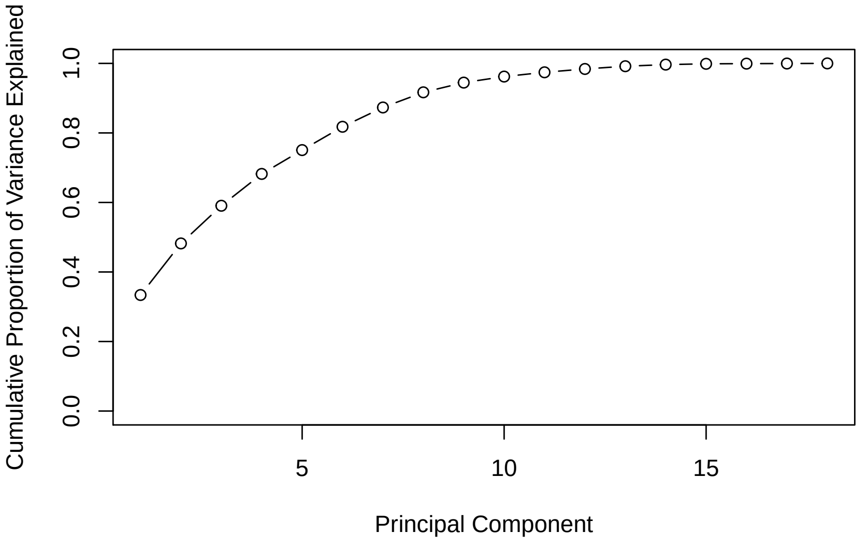
```
load('images/MOcovid.RData')
covidpca <- prcomp(ndat[,-1],scale = T,center = T)

covidpca$rotation[,1:2]
```

```
##                                                      PC1         PC2
## X..Population.in.Rural.Areas                  0.32865838  0.05090955
## Area..sq..miles.                              0.12014444 -0.28579183
## Population.density..sq..miles.               -0.29670124  0.28312922
## Literacy.rate                                -0.12517700 -0.08999542
## Families                                     -0.25856941  0.16485752
## Area.of.farm.land..sq..miles.                 0.02101106 -0.31070363
## Number.of.farms                              -0.03814582 -0.44809679
## Average.value.of.all.property.per.farm..dollars. -0.05410709  0.14404306
## Estimation.of.rurality..                     -0.19040210  0.12089501
## Male..                                        0.02182394 -0.09568768
## Number.of.Physcians.per.100.000              -0.31451606  0.13598026
## average.age                                   0.29414708  0.35593459
## X0.4.age.proportion                          -0.11431336 -0.23574057
## X20.44.age.proportion                        -0.32802128 -0.22718550
## X65.and.over.age.proportion                   0.30585033  0.32201626
## prop..White..nonHisp                          0.35627561 -0.14142646
## prop..Hispanic                               -0.16655381 -0.15105342
## prop..Black                                  -0.33333359  0.24405802
```

```
# Variability of each principal component: pr.var
pr.var <- covidpca$sdev ^ 2
# Variance explained by each principal component: pve
pve <- pr.var / sum(pr.var)
plot(
    pve,
    xlab = "Principal Component",
    ylab = "Proportion of Variance Explained",
    ylim = c(0, 0.5),
    type = "b"
)
```



```
plot(
    cumsum(pve),
    xlab = "Principal Component",
    ylab = "Cumulative Proportion of Variance Explained",
    ylim = c(0, 1),
    type = "b"
)
```



```
# the first six principe account for around 80% of the variance.


#using base lm function for PC regression
pcadat <- data.frame(covidpca$x[, 1:6])
pcadat$y <- ndat$Y
pcr.man <- lm(log(y) ~ ., pcadat)
mean(pcr.man$residuals ^ 2)
```

```
## [1] 0.03453371
```

```
#comparison to lm w/o prin comps
lm.fit <- lm(log(Y) ~ ., data = ndat)
mean(lm.fit$residuals ^ 2)
```

```
## [1] 0.02335128
```

MSE for the PC-based model is larger than regular regression, because models with a large degree of collinearity can still perform well.

`pcr` function in `pls` can be used for fitting PC regression (it will select the optimal number of components in the model).

21.3 Factor Analysis

Purpose

- Using a few linear combinations of underlying unobservable (latent) traits, we try to describe the covariance relationship among a large number of measured traits
- Similar to PCA, but factor analysis is **model based**

More details can be found on PSU stat or UMN stat

Let $\mathbf{y}$ be the set of p measured variables

$E(\mathbf{y}) =$

$var(\mathbf{y}) =$

We have

$$
\begin{aligned}
\mathbf{y} - \quad &= \mathbf{L}\mathbf{f} + \epsilon \\
&= \begin{pmatrix} l_{11}f_1 + l_{12}f_2 + \dots + l_{tm}f_m \\ \vdots \\ l_{p1}f_1 + l_{p2}f_2 + \dots + l_{pm}f_m \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_p \end{pmatrix}
\end{aligned}
$$

where

- $\mathbf{y} - \quad =$ the p centered measurements
- $\mathbf{L} = p \times m$ matrix of factor loadings
- $\mathbf{f} =$ unobserved common factors for the population
- $\quad =$ random errors (i.e., variation that is not accounted for by the common factors).

We want m (the number of factors) to be much smaller than p (the number of measured attributes)

Restrictions on the model

- $E(\epsilon) = \mathbf{0}$
- $var(\epsilon) = \Psi_{p \times p} = diag(\psi_1, \dots, \psi_p)$
- $\ , \mathbf{f}$ are independent
- Additional assumption could be $E(\mathbf{f}) = \mathbf{0}, var(\mathbf{f}) = \mathbf{I}_{m \times m}$ (known as the orthogonal factor model) , which imposes the following covariance structure on $\mathbf{y}$

$$
\begin{aligned}
var(\mathbf{y}) = \quad &= var(\mathbf{L}\mathbf{f} + \quad) \\
&= var(\mathbf{L}\mathbf{f}) + var(\epsilon) \\
&= \mathbf{L}var(\mathbf{f})\mathbf{L}' + \\
&= \mathbf{L}\mathbf{I}\mathbf{L}' + \\
&= \mathbf{L}\mathbf{L}' +
\end{aligned}
$$

Since is diagonal, the off-diagonal elements of $\mathbf{LL}'$ are σ_{ij}, the co variances in , which means $cov(y_i, y_j) = \sum_{k=1}^m l_{ik} l_{jk}$ and the covariance of $\mathbf{y}$ is completely determined by the m factors ($m << p$)

$var(y_i) = \sum_{k=1}^m l_{ik}^2 + \psi_i$ where ψ_i is the **specific variance** and the summation term is the i-th **communality** (i.e., portion of the variance of the i-th variable contributed by the m common factors ($h_i^2 = \sum_{k=1}^m l_{ik}^2$)

The factor model is only uniquely determined up to an orthogonal transformation of the factors.

Let $\mathbf{T}_{m \times m}$ be an orthogonal matrix $\mathbf{TT}' = \mathbf{T}'\mathbf{T} = \mathbf{I}$ then

$$
\begin{aligned}
\mathbf{y} - \quad &= \mathbf{Lf} + \epsilon \\
&= \mathbf{LTT}'\mathbf{f} + \epsilon \\
&= \mathbf{L}^*(\mathbf{T}'\mathbf{f}) + \epsilon \quad \text{where } \mathbf{L}^* = \mathbf{LT}
\end{aligned}
$$

and

$$
\begin{aligned}
&= \mathbf{LL}' + \\
&= \mathbf{LTT}'\mathbf{L} + \\
&= (\mathbf{L}^*)(\mathbf{L}^*)' +
\end{aligned}
$$

Hence, any orthogonal transformation of the factors is an equally good description of the correlations among the observed traits.

Let $\mathbf{y} = \mathbf{Cx}$, where $\mathbf{C}$ is any diagonal matrix, then $\mathbf{L}_y = \mathbf{CL}_x$ and $_y = \mathbf{C} \, _x\mathbf{C}$

Hence, we can see that factor analysis is also invariant to changes in scale

21.3.1 Methods of Estimation

To estimate $\mathbf{L}$

1. Principal Component Method
2. Principal Factor Method
3. 21.3.1.3

21.3.1.1 Principal Component Method

Spectral decomposition

$$
\begin{aligned}
&= \lambda_1 \mathbf{a}_1 \mathbf{a}_1' + \cdots + \lambda_p \mathbf{a}_p \mathbf{a}_p' \\
&= \mathbf{A}\ \mathbf{A}' \\
&= \sum_{k=1}^{m} \lambda + k \mathbf{a}_k \mathbf{a}_k' + \sum_{k=m+1}^{p} \lambda_k \mathbf{a}_k \mathbf{a}_k' \\
&= \sum_{k=1}^{m} l_k l_k' + \sum_{k=m+1}^{p} \lambda_k \mathbf{a}_k \mathbf{a}_k'
\end{aligned}
$$

where $l_k = \mathbf{a}_k \sqrt{\lambda_k}$ and the second term is not diagonal in general.

Assume

$$\psi_i = \sigma_{ii} - \sum_{k=1}^{m} l_{ik}^2 = \sigma_{ii} - \sum_{k=1}^{m} \lambda_i a_{ik}^2$$

then

$$\approx \mathbf{L}\mathbf{L}' +$$

To estimate $\mathbf{L}$ and Ψ , we use the expected eigenvalues and eigenvectors from $\mathbf{S}$ or $\mathbf{R}$

- The estimated factor loadings don't change as the number of actors increases
- The diagonal elements of $\hat{\mathbf{L}}\hat{\mathbf{L}}' + \hat{}$ are equal to the diagonal elements of $\mathbf{S}$ and $\mathbf{R}$, but the covariances may not be exactly reproduced
- We select m so that the off-diagonal elements close to the values in $\mathbf{S}$ (or to make the off-diagonal elements of $\mathbf{S} - \hat{\mathbf{L}}\hat{\mathbf{L}}' + \hat{}$ small)

21.3.1.2 Principal Factor Method

Consider modeling the correlation matrix, $\mathbf{R} = \mathbf{L}\mathbf{L}' +$. Then

$$\mathbf{L}\mathbf{L}' = \mathbf{R} - \quad = \begin{pmatrix} h_1^2 & r_{12} & \dots & r_{1p} \\ r_{21} & h_2^2 & \dots & r_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ r_{p1} & r_{p2} & \dots & h_p^2 \end{pmatrix}$$

where $h_i^2 = 1 - \psi_i$ (the communality)

Suppose that initial estimates are available for the communalities, $(h_1^*)^2, (h_2^*)^2, \dots, (h_p^*)^2$, then we can regress each trait on all the others, and then use the r^2 as h^2

The estimate of $\mathbf{R}-\ $ at step k is

$$(\mathbf{R}-\ )_k = \begin{pmatrix} (h_1^*)^2 & r_{12} & \dots & r_{1p} \\ r_{21} & (h_2^*)^2 & \dots & r_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ r_{p1} & r_{p2} & \dots & (h_p^*)^2 \end{pmatrix} = \mathbf{L}_k^*(\mathbf{L}_k^*)'$$

where

$$\mathbf{L}_k^* = (\sqrt{\hat{\lambda}_1^* \hat{\mathbf{a}}_1^*, \dots \hat{\lambda}_m^* \hat{\mathbf{a}}_m^*})$$

and

$$\hat{\psi}_{i,k}^* = 1 - \sum_{j=1}^{m} \hat{\lambda}_i^* (\hat{a}_{ij}^*)^2$$

we used the spectral decomposition on the estimated matrix $(\mathbf{R}-\ )$ to calculate the $\hat{\lambda}_i^* s$ and the $\hat{\mathbf{a}}_i^* s$

After updating the values of $(\hat{h}_i^*)^2 = 1 - \hat{\psi}_{i,k}^*$ we will use them to form a new $\mathbf{L}_{k+1}^*$ via another spectral decomposition. Repeat the process

Notes:

- The matrix $(\mathbf{R}-\ )_k$ is not necessarily positive definite
- The principal component method is similar to principal factor if one considers the initial communalities are $h^2 = 1$
- if m is too large, some communalities may become larger than 1, causing the iterations to terminate. To combat, we can
    - fix any communality that is greater than 1 at 1 and then continues.
    - continue iterations regardless of the size of the communalities. However, results can be outside fo the parameter space.

21.3.1.3 Maximum Likelihood Method

Since we need the likelihood function, we make the additional (critical) assumption that

- $\mathbf{y}_j \sim N(\ ,\ )$ for $j = 1, .., n$
- $\mathbf{f} \sim N(\mathbf{0}, \mathbf{I})$
- $\epsilon_j \sim N(\mathbf{0},\ )$

and restriction

- $\mathbf{L}' \ {}^{-1}\mathbf{L} =$ where is a diagonal matrix. (since the factor loading matrix is not unique, we need this restriction).

Notes:

- Finding MLE can be computationally expensive
- we typically use other methods for exploratory data analysis
- LIkelihood ratio tests could be used for testing hypotheses in this framework (i.e., Confirmatory Factor Analysis)

21.3.2 Factor Rotation

$\mathbf{T}_{m\times m}$ is an orthogonal matrix that has the property that

$$\hat{\mathbf{L}}\hat{\mathbf{L}}' + \hat{} = \hat{\mathbf{L}}^*(\hat{\mathbf{L}}^*)' + \hat{}$$

where $\mathbf{L}^* = \mathbf{L}\mathbf{T}$

This means that estimated specific variances and communalities are not altered by the orthogonal transformation.

Since there are an infinite number of choices for $\mathbf{T}$, some selection criterion is necessary

For example, we can find the orthogonal transformation that maximizes the objective function

$$\sum_{j=1}^{m}[\frac{1}{p}\sum_{i=1}^{p}(\frac{l_{ij}^{*2}}{h_i})^2 - \{\frac{\gamma}{p}\sum_{i=1}^{p}(\frac{l_{ij}^{*2}}{h_i})^2\}^2]$$

where $\frac{l_{ij}^{*2}}{h_i}$ are "scaled loadings", which gives variables with small communalities more influence.

Different choices of γ in the objective function correspond to different orthogonal rotation found in the literature;

1. Varimax $\gamma = 1$ (rotate the factors so that each of the p variables should have a high loading on only one factor, but this is not always possible).
2. Quartimax $\gamma = 0$
3. Equimax $\gamma = m/2$
4. Parsimax $\gamma = \frac{p(m-1)}{p+m-2}$
5. Promax: non-orthognal or olique transformations
6. Harris-Kaiser (HK): non-orthognal or olique transformations

21.3.3 Estimation of Factor Scores

Recall

$$(\mathbf{y}_j - \ ) = \mathbf{L}_{p\times m}\mathbf{f}_j + \epsilon_j$$

If the factor model is correct then

$$var(\epsilon_j) = \ = diag(\psi_1, \dots, \psi_p)$$

Thus we could consider using weighted least squares to estimate $\mathbf{f}_j$, the vector of factor scores for the j-th sampled unit by

$$\begin{aligned} \hat{\mathbf{f}} &= (\mathbf{L}'\ ^{-1}\mathbf{L})^{-1}\mathbf{L}'\ ^{-1}(\mathbf{y}_j - \ ) \\ &\approx (\mathbf{L}'\ ^{-1}\mathbf{L})^{-1}\mathbf{L}'\ ^{-1}(\mathbf{y}_j - \bar{\mathbf{y}}) \end{aligned}$$

21.3.3.1 The Regression Method

Alternatively, we can use the regression method to estimate the factor scores

Consider the joint distribution of $(\mathbf{y}_j - \ )$ and $\mathbf{f}_j$ assuming multivariate normality, as in the maximum likelihood approach. then,

$$\begin{pmatrix} \mathbf{y}_j - \ \\ \mathbf{f}_j \end{pmatrix} \sim N_{p+m}\left(\begin{bmatrix} \mathbf{LL}' + \ & \mathbf{L} \\ \mathbf{L}' & \mathbf{I}_{m\times m} \end{bmatrix}\right)$$

when the m factor model is correct

Hence,

$$E(\mathbf{f}_j|\mathbf{y}_j - \ ) = \mathbf{L}'(\mathbf{LL}' + \ )^{-1}(\mathbf{y}_j - \ )$$

notice that $\mathbf{L}'(\mathbf{LL}' + \ )^{-1}$ is an $m \times p$ matrix of regression coefficients

Then, we use the estimated conditional mean vector to estimate the factor scores

$$\hat{\mathbf{f}}_j = \hat{\mathbf{L}}'(\hat{\mathbf{L}}\hat{\mathbf{L}}' + \hat{}\ )^{-1}(\mathbf{y}_j - \bar{\mathbf{y}})$$

Alternatively, we could reduce the effect of possible incorrect determination fo the number of factors m by using $\mathbf{S}$ as a substitute for $\hat{\mathbf{L}}\hat{\mathbf{L}}' + \hat{}$ then

$$\hat{\mathbf{f}}_j = \hat{\mathbf{L}}'\mathbf{S}^{-1}(\mathbf{y}_j - \bar{\mathbf{y}})$$

where $j = 1, \dots, n$

21.3.4 Model Diagnostic

- Plots
- Check for outliers (recall that $\mathbf{f}_j \sim iidN(\mathbf{0}, \mathbf{I}_{m\times m})$)
- Check for multivariate normality assumption
- Use univariate tests for normality to check the factor scores
- **Confirmatory Factor Analysis**: formal testing of hypotheses about loadings, use MLE and full/reduced model testing paradigm and measures of model fit

21.3.5 Application

In the `psych` package,

- h2 = the communalities
- u2 = the uniqueness
- com = the complexity

```
library(psych)
```

```
##
## Attaching package: 'psych'

## The following object is masked from 'package:lavaan':
##
##     cor2cov

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha

## The following object is masked from 'package:car':
##
##     logit
```

```
library(tidyverse)
## Load the data from the psych package
data(Harman.5)
Harman.5
```

```
##        population schooling employment professional housevalue
## Tract1       5700      12.8       2500          270      25000
## Tract2       1000      10.9        600           10      10000
## Tract3       3400       8.8       1000           10       9000
## Tract4       3800      13.6       1700          140      25000
## Tract5       4000      12.8       1600          140      25000
## Tract6       8200       8.3       2600           60      12000
## Tract7        1200      11.4        400           10      16000
## Tract8        9100      11.5       3300           60      14000
## Tract9        9900      12.5       3400          180      18000
## Tract10       9600      13.7       3600          390      25000
## Tract11       9600       9.6       3300           80      12000
## Tract12       9400      11.4       4000          100      13000
```

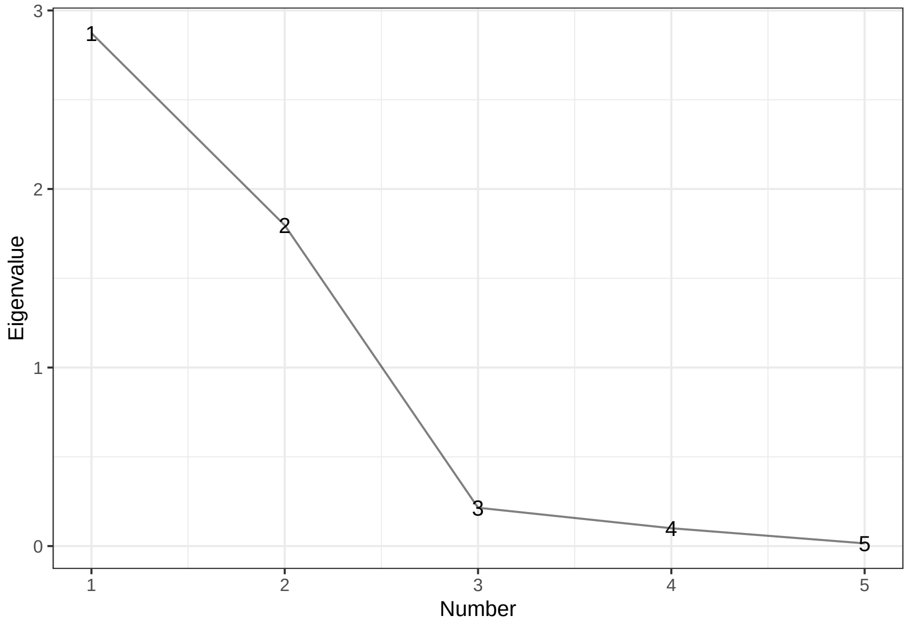
```
# Correlation matrix
cor_mat <- cor(Harman.5)
cor_mat
```

```
##               population  schooling employment professional housevalue
## population   1.00000000 0.00975059  0.9724483    0.4388708 0.02241157
## schooling    0.00975059 1.00000000  0.1542838    0.6914082 0.86307009
## employment   0.97244826 0.15428378  1.0000000    0.5147184 0.12192599
## professional 0.43887083 0.69140824  0.5147184    1.0000000 0.77765425
## housevalue   0.02241157 0.86307009  0.1219260    0.7776543 1.00000000
```

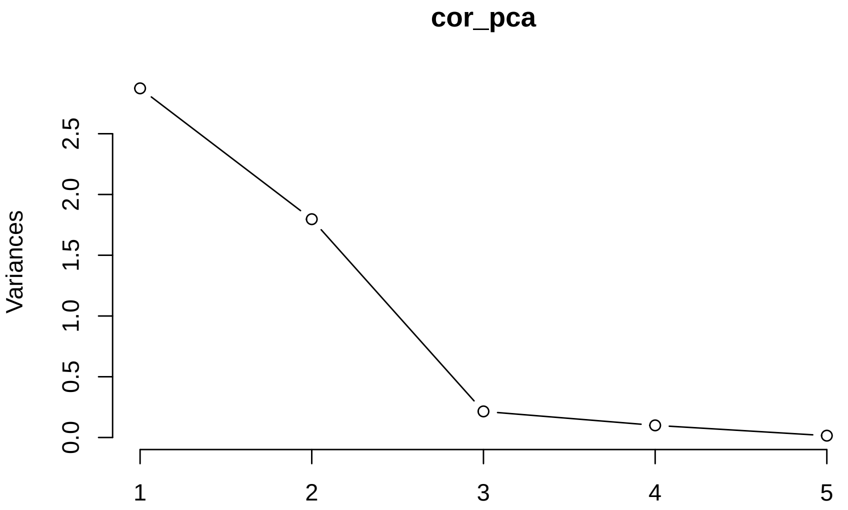
```
## Principal Component Method with Correlation
cor_pca <- prcomp(Harman.5, scale = T)
# eigen values
cor_results <- data.frame(eigen_values = cor_pca$sdev ^ 2)

cor_results <- cor_results %>%
    mutate(
        proportion = eigen_values / sum(eigen_values),
        cumulative = cumsum(proportion),
        number = row_number()
    )
cor_results
```

```
##   eigen_values  proportion cumulative number
## 1   2.87331359 0.574662719  0.5746627      1
## 2   1.79666009 0.359332019  0.9339947      2
## 3   0.21483689 0.042967377  0.9769621      3
## 4   0.09993405 0.019986811  0.9969489      4
## 5   0.01525537 0.003051075  1.0000000      5
```

```
# Scree plot of Eigenvalues
scree_gg <- ggplot(cor_results, aes(x = number, y = eigen_values)) +
    geom_line(alpha = 0.5) +
    geom_text(aes(label = number)) +
    scale_x_continuous(name = "Number") +
    scale_y_continuous(name = "Eigenvalue") +
    theme_bw()
scree_gg
```



```r
screeplot(cor_pca, type = 'lines')
```



```r
## Keep 2 factors based on scree plot and eigenvalues
factor_pca <- principal(Harman.5, nfactors = 2, rotate = "none")
factor_pca
```

```
## Principal Components Analysis
## Call: principal(r = Harman.5, nfactors = 2, rotate = "none")
## Standardized loadings (pattern matrix) based upon correlation matrix
##               PC1   PC2   h2    u2 com
## population   0.58  0.81 0.99 0.012 1.8
## schooling    0.77 -0.54 0.89 0.115 1.8
## employment   0.67  0.73 0.98 0.021 2.0
## professional 0.93 -0.10 0.88 0.120 1.0
## housevalue   0.79 -0.56 0.94 0.062 1.8
##
##                        PC1  PC2
## SS loadings           2.87 1.80
## Proportion Var        0.57 0.36
## Cumulative Var        0.57 0.93
## Proportion Explained  0.62 0.38
## Cumulative Proportion 0.62 1.00
##
## Mean item complexity =  1.7
## Test of the hypothesis that 2 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.03
##  with the empirical chi square  0.29  with prob <  0.59
##
## Fit based upon off diagonal values = 1
```

```
# factor 1 = overall socioeconomic health
# factor 2 = contrast of the population and employment against school and house value


## Ssquared multiple correlation (SMC) prior, no rotation
factor_pca_smc <- fa(
    Harman.5,
    nfactors = 2,
    fm = "pa",
    rotate = "none",
    SMC = TRUE
)
```

```
## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
## The estimated weights for the factor scores are probably incorrect. Try a
## different factor score estimation method.
```

```
## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, : An
## ultra-Heywood case was detected. Examine the results carefully
```

```
factor_pca_smc
```

```
## Factor Analysis using method =  pa
## Call: fa(r = Harman.5, nfactors = 2, rotate = "none", SMC = TRUE, fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##               PA1   PA2   h2      u2 com
## population   0.62  0.78 1.00 -0.0027 1.9
## schooling    0.70 -0.53 0.77  0.2277 1.9
## employment   0.70  0.68 0.96  0.0413 2.0
## professional 0.88 -0.15 0.80  0.2017 1.1
## housevalue   0.78 -0.60 0.96  0.0361 1.9
##
##                        PA1  PA2
## SS loadings           2.76 1.74
## Proportion Var        0.55 0.35
## Cumulative Var        0.55 0.90
## Proportion Explained  0.61 0.39
## Cumulative Proportion 0.61 1.00
##
## Mean item complexity =  1.7
## Test of the hypothesis that 2 factors are sufficient.
##
## The degrees of freedom for the null model are  10  and the objective function was  6.38 with C
## The degrees of freedom for the model are 1  and the objective function was  0.34
##
## The root mean square of the residuals (RMSR) is  0.01
## The df corrected root mean square of the residuals is  0.03
##
## The harmonic number of observations is  12 with the empirical chi square  0.02  with prob <  0
## The total number of observations was  12  with Likelihood Chi Square =  2.44  with prob <  0.1
##
## Tucker Lewis Index of factoring reliability =  0.596
## RMSEA index =  0.336  and the 90 % confidence intervals are  0 0.967
## BIC =  -0.04
## Fit based upon off diagonal values = 1
```

```
## SMC prior, Promax rotation
factor_pca_smc_pro <- fa(
    Harman.5,
    nfactors = 2,
    fm = "pa",
    rotate = "Promax",
    SMC = TRUE
)
```

```
## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
```

```
## The estimated weights for the factor scores are probably incorrect. Try a
## different factor score estimation method.

## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
## An ultra-Heywood case was detected. Examine the results carefully
```

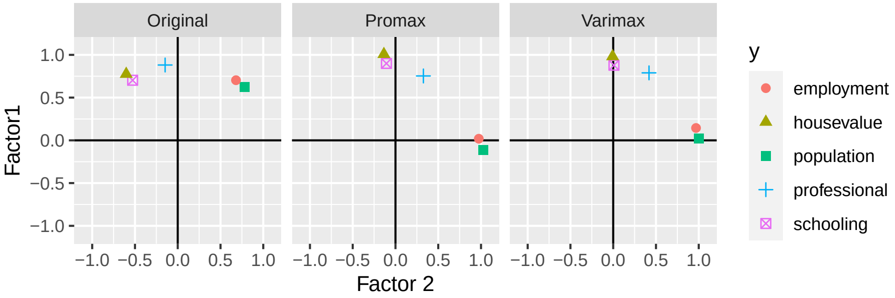
```
factor_pca_smc_pro
```

```
## Factor Analysis using method =  pa
## Call: fa(r = Harman.5, nfactors = 2, rotate = "Promax", SMC = TRUE,
##     fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                PA1   PA2   h2      u2 com
## population   -0.11  1.02 1.00 -0.0027 1.0
## schooling     0.90 -0.11 0.77  0.2277 1.0
## employment    0.02  0.97 0.96  0.0413 1.0
## professional  0.75  0.33 0.80  0.2017 1.4
## housevalue    1.01 -0.14 0.96  0.0361 1.0
##
##                        PA1  PA2
## SS loadings           2.38 2.11
## Proportion Var        0.48 0.42
## Cumulative Var        0.48 0.90
## Proportion Explained  0.53 0.47
## Cumulative Proportion 0.53 1.00
##
##  With factor correlations of
##      PA1  PA2
## PA1 1.00 0.25
## PA2 0.25 1.00
##
## Mean item complexity =  1.1
## Test of the hypothesis that 2 factors are sufficient.
##
## The degrees of freedom for the null model are  10  and the objective function was  6
## The degrees of freedom for the model are 1  and the objective function was  0.34
##
## The root mean square of the residuals (RMSR) is  0.01
## The df corrected root mean square of the residuals is  0.03
##
## The harmonic number of observations is  12 with the empirical chi square  0.02  with
## The total number of observations was  12  with Likelihood Chi Square =  2.44  with p
##
## Tucker Lewis Index of factoring reliability =  0.596
## RMSEA index =  0.336  and the 90 % confidence intervals are  0 0.967
## BIC =  -0.04
## Fit based upon off diagonal values = 1
```

```
## SMC prior, varimax rotation
factor_pca_smc_var <- fa(
    Harman.5,
    nfactors = 2,
    fm = "pa",
    rotate = "varimax",
    SMC = TRUE
)
```

```
## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
## The estimated weights for the factor scores are probably incorrect. Try a
## different factor score estimation method.

## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
## An ultra-Heywood case was detected. Examine the results carefully
```

```
## Make a data frame of the loadings for ggplot2
factors_df <-
    bind_rows(
        data.frame(
            y = rownames(factor_pca_smc$loadings),
            unclass(factor_pca_smc$loadings)
        ),
        data.frame(
            y = rownames(factor_pca_smc_pro$loadings),
            unclass(factor_pca_smc_pro$loadings)
        ),
        data.frame(
            y = rownames(factor_pca_smc_var$loadings),
            unclass(factor_pca_smc_var$loadings)
        ),
        .id = "Rotation"
    )
flag_gg <- ggplot(factors_df) +
    geom_vline(aes(xintercept = 0)) +
    geom_hline(aes(yintercept = 0)) +
    geom_point(aes(
        x = PA2,
        y = PA1,
        col = y,
        shape = y
    ), size = 2) +
    scale_x_continuous(name = "Factor 2", limits = c(-1.1, 1.1)) +
    scale_y_continuous(name = "Factor1", limits = c(-1.1, 1.1)) + facet_wrap("Rotation",
                                                                             labeller(
                                                                                "1" =
                                                                            ))) +
    coord_fixed(ratio = 1) # make aspect ratio of each facet 1
flag_gg
```



```
# promax and varimax did a good job to assign trait to a particular factor

factor_mle_1 <- fa(
    Harman.5,
    nfactors = 1,
    fm = "mle",
    rotate = "none",
    SMC = TRUE
)
factor_mle_1
```

```
## Factor Analysis using method =  ml
## Call: fa(r = Harman.5, nfactors = 1, rotate = "none", SMC = TRUE, fm = "mle")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                ML1    h2     u2 com
## population    0.97 0.950 0.0503   1
## schooling     0.14 0.021 0.9791   1
## employment    1.00 0.995 0.0049   1
## professional 0.51 0.261 0.7388   1
## housevalue    0.12 0.014 0.9864   1
```

```
##
##                  ML1
## SS loadings    2.24
## Proportion Var 0.45
##
## Mean item complexity =  1
## Test of the hypothesis that 1 factor is sufficient.
##
## The degrees of freedom for the null model are  10  and the objective function was  6.38 with C
## The degrees of freedom for the model are 5  and the objective function was  3.14
##
## The root mean square of the residuals (RMSR) is  0.41
## The df corrected root mean square of the residuals is  0.57
##
## The harmonic number of observations is  12 with the empirical chi square  39.41  with prob <
## The total number of observations was  12  with Likelihood Chi Square =  24.56  with prob <  0.
##
## Tucker Lewis Index of factoring reliability =  0.022
## RMSEA index =  0.564  and the 90 % confidence intervals are  0.374 0.841
## BIC =  12.14
## Fit based upon off diagonal values = 0.5
## Measures of factor score adequacy
##                                                    ML1
## Correlation of (regression) scores with factors   1.00
## Multiple R square of scores with factors          1.00
## Minimum correlation of possible factor scores     0.99
```

```
factor_mle_2 <- fa(
    Harman.5,
    nfactors = 2,
    fm = "mle",
    rotate = "none",
    SMC = TRUE
)
factor_mle_2
```

```
## Factor Analysis using method =  ml
## Call: fa(r = Harman.5, nfactors = 2, rotate = "none", SMC = TRUE, fm = "mle")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                ML2  ML1   h2    u2 com
## population   -0.03 1.00 1.00 0.005 1.0
## schooling     0.90 0.04 0.81 0.193 1.0
## employment    0.09 0.98 0.96 0.036 1.0
## professional  0.78 0.46 0.81 0.185 1.6
## housevalue    0.96 0.05 0.93 0.074 1.0
##
```

```
##                         ML2  ML1
## SS loadings           2.34 2.16
## Proportion Var        0.47 0.43
## Cumulative Var        0.47 0.90
## Proportion Explained  0.52 0.48
## Cumulative Proportion 0.52 1.00
##
## Mean item complexity =  1.1
## Test of the hypothesis that 2 factors are sufficient.
##
## The degrees of freedom for the null model are  10  and the objective function was
## The degrees of freedom for the model are 1  and the objective function was  0.31
##
## The root mean square of the residuals (RMSR) is  0.01
## The df corrected root mean square of the residuals is  0.05
##
## The harmonic number of observations is  12 with the empirical chi square  0.05  wit
## The total number of observations was  12  with Likelihood Chi Square =  2.22  with
##
## Tucker Lewis Index of factoring reliability =  0.658
## RMSEA index =  0.307  and the 90 % confidence intervals are  0 0.945
## BIC =  -0.26
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy
##                                                    ML2  ML1
## Correlation of (regression) scores with factors   0.98 1.00
## Multiple R square of scores with factors          0.95 1.00
## Minimum correlation of possible factor scores     0.91 0.99
```

```r
factor_mle_3 <- fa(
    Harman.5,
    nfactors = 3,
    fm = "mle",
    rotate = "none",
    SMC = TRUE
)
factor_mle_3
```

```
## Factor Analysis using method =  ml
## Call: fa(r = Harman.5, nfactors = 3, rotate = "none", SMC = TRUE, fm = "mle")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                ML2  ML1   ML3   h2     u2 com
## population   -0.12 0.98 -0.11 0.98 0.0162 1.1
## schooling     0.89 0.15  0.29 0.90 0.0991 1.3
## employment    0.00 1.00  0.04 0.99 0.0052 1.0
## professional  0.72 0.52 -0.10 0.80 0.1971 1.9
## housevalue    0.97 0.13 -0.09 0.97 0.0285 1.1
##
##                        ML2  ML1  ML3
## SS loadings           2.28 2.26 0.11
## Proportion Var        0.46 0.45 0.02
## Cumulative Var        0.46 0.91 0.93
## Proportion Explained  0.49 0.49 0.02
## Cumulative Proportion 0.49 0.98 1.00
##
## Mean item complexity =  1.2
## Test of the hypothesis that 3 factors are sufficient.
##
## The degrees of freedom for the null model are  10  and the objective function was  6.38 with C
## The degrees of freedom for the model are -2  and the objective function was  0
##
## The root mean square of the residuals (RMSR) is  0
## The df corrected root mean square of the residuals is  NA
##
## The harmonic number of observations is  12 with the empirical chi square  0  with prob <  NA
## The total number of observations was  12  with Likelihood Chi Square =  0  with prob <  NA
##
## Tucker Lewis Index of factoring reliability =  1.318
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy
##                                                    ML2  ML1  ML3
## Correlation of (regression) scores with factors   0.99 1.00 0.82
## Multiple R square of scores with factors          0.98 1.00 0.68
## Minimum correlation of possible factor scores     0.96 0.99 0.36
```

The output info for the null hypothesis of no common factors is in the statement "The degrees of freedom for the null model .."

The output info for the null hypothesis that number of factors is sufficient is in the statement "The total number of observations was …"

One factor is not enough, two is sufficient, and not enough data for 3 factors (df of -2 and NA for p-value). Hence, we should use 2-factor model.

21.4 Discriminant Analysis

Suppose we have two or more different populations from which observations could come from. Discriminant analysis seeks to determine which of the possible population an observation comes from while making as few mistakes as possible

Notation

SImilar to MANOVA, let $\mathbf{y}_{j1}, \mathbf{y}_{j2}, \dots, \mathbf{y}_{in_j} \sim iid f_j(\mathbf{y})$ for $j = 1, \dots, h$

Let $f_j(\mathbf{y})$ be the density function for population j . Note that each vector $\mathbf{y}$ contain measurements on all p traits

1. Assume that each observation is from one of h possible populations.
2. We want to form a discriminant rule that will allocate an observation $\mathbf{y}$ to population j when $\mathbf{y}$ is in fact from this population

21.4.1 Known Populations

The maximum likelihood discriminant rule for assigning an observation $\mathbf{y}$ to one of the h populations allocates $\mathbf{y}$ to the population that gives the largest likelihood to $\mathbf{y}$

Consider the likelihood for a single observation $\mathbf{y}$, which has the form $f_j(\mathbf{y})$ where j is the true population.

Since j is unknown, to make the likelihood as large as possible, we should choose the value j which causes $f_j(\mathbf{y})$ to be as large as possible

Consider a simple univariate example. Suppose we have data from one of two binomial populations.

- The first population has $n = 10$ trials with success probability $p = .5$
- The second population has $n = 10$ trials with success probability $p = .7$
- to which population would we assign an observation of $y = 7$
- Note:
  - $f(y = 7|n = 10, p = .5) = .117$
  - $f(y = 7|n = 10, p = .7) = .267$ where $f(.)$ is the binomial likelihood.
  - Hence, we choose the second population

Another example

We have 2 populations, where

- First population: $N(\mu_1, \sigma_1^2)$
- Second population: $N(\mu_2, \sigma_2^2)$

The likelihood for a single observation is

$$f_j(y) = (2\pi\sigma_j^2)^{-1/2} \exp\{-\frac{1}{2}(\frac{y-\mu_j}{\sigma_j})^2\}$$

Consider a likelihood ratio rule

$$
\begin{aligned}
\Lambda &= \frac{\text{likelihood of y from pop 1}}{\text{likelihood of y from pop 2}} \\
&= \frac{f_1(y)}{f_2(y)} \\
&= \frac{\sigma_2}{\sigma_1} \exp\{-\frac{1}{2}[(\frac{y-\mu_1}{\sigma_1})^2 - (\frac{y-\mu_2}{\sigma_2})^2]\}
\end{aligned}
$$

Hence, we classify into

- pop 1 if $\Lambda > 1$
- pop 2 if $\Lambda < 1$
- for ties, flip a coin

Another way to think:

we classify into population 1 if the "standardized distance" of y from μ_1 is less than the "standardized distance" of y from μ_2 which is referred to as a **quadratic discriminant rule**.

(Significant simplification occurs in th special case where $\sigma_1 = \sigma_2 = \sigma^2$)

Thus, we classify into population 1 if

$$
(y-\mu_2)^2 > (y-\mu_1)^2
$$

or

$$
|y-\mu_2| > |y-\mu_1|
$$

and

$$
-2\log(\Lambda) = -2y\frac{(\mu_1-\mu_2)}{\sigma^2} + \frac{(\mu_1^2-\mu_2^2)}{\sigma^2} = \beta y + \alpha
$$

Thus, we classify into population 1 if this is less than 0.

Discriminant classification rule is linear in y in this case.

21.4.1.1 Multivariate Expansion

Suppose that there are 2 populations

- $N_p(_1, _1)$
- $N_p(_2, _2)$

$$
\begin{aligned}
-2\log(\frac{f_1(\mathbf{x})}{f_2(\mathbf{x})}) &= \log|\ _1| + (\mathbf{x}-\ _1)'\ _1^{-1}(\mathbf{x}-\ _1) \\
&\quad - [\log|\ _2| + (\mathbf{x}-\ _2)'\ _2^{-1}(\mathbf{x}-\ _2)]
\end{aligned}
$$

Again, we classify into population 1 if this is less than 0, otherwise, population 2. And like the univariate case with non-equal variances, this is a quadratic discriminant rule.

And if the covariance matrices are equal: $_1 =\ _2 =\ _1$ classify into population 1 if

$$
(\ _1-\ _2)'\ ^{-1}\mathbf{x} - \frac{1}{2}(\ _1-\ _2)'\ ^{-1}(\ _1-\ _2) \geq 0
$$

This linear discriminant rule is also referred to as **Fisher's linear discriminant function**

By assuming the covariance matrices are equal, we assume that the shape and orientation fo the two populations must be the same (which can be a strong restriction)

When $_1,\ _2,\ $ are known, the probability of misclassification can be determined:

$$
\begin{aligned}
P(2|1) &= P(\text{calssify into pop 2}|\text{ x is from pop 1}) \\
&= P((\ _1-\ _2)'\ ^{-1}\mathbf{x} \leq \frac{1}{2}(\ _1-\ _2)'\ ^{-1}(\ _1-\ _2)|\mathbf{x} \sim N(\mu_1,\ ) \\
&= \Phi(-\frac{1}{2}\delta)
\end{aligned}
$$

where

- $\delta^2 = (\ _1-\ _2)'\ ^{-1}(\ _1-\ _2)$
- Φ is the standard normal cdf

Suppose there are h possible populations, which are distributed as $N_p(\ _p,\ )$. Then, the maximum likelihood (linear) discriminant rule allocates $\mathbf{y}$ to population j where j minimizes the squared Mahalanobis distance

$$
(\mathbf{y}-\ _j)'\ ^{-1}(\mathbf{y}-\ _j)
$$

Bayes Discriminant Rules

If we know that population j has prior probabilities π_j (assume $\pi_j > 0$) we can form the Bayes discriminant rule.

This rule allocates an observation $\mathbf{y}$ to the population for which $\pi_j f_j(\mathbf{y})$ is maximized.

Note:

- **Maximum likelihood discriminant rule** is a special case of the **Bayes discriminant rule**, where it sets all the $\pi_j = 1/h$

Optimal Properties of Bayes Discriminant Rules

- let p_{ii} be the probability of correctly assigning an observation from population i
- then one rule (with probabilities p_{ii}) is as good as another rule (with probabilities p'_{ii}) if $p_{ii} \geq p'_{ii}$ for all $i = 1, \dots, h$
- The first rule is better than the alternative if $p_{ii} > p'_{ii}$ for at least one i.
- A rule for which there is no better alternative is called admissible
- Bayes Discriminant Rules are admissible
- If we utilized prior probabilities, then we can form the posterior probability of a correct allocation, $\sum_{i=1}^h \pi_i p_{ii}$
- Bayes Discriminant Rules have the largest possible posterior probability of correct allocation with respect to the prior
- These properties show that Bayes Discriminant rule is our best approach.

Unequal Cost

- We want to consider the cost misallocation Define c_{ij} to be the cost associated with allocation a member of population j to population i.
- Assume that
  - $c_{ij} > 0$ for all $i \neq j$
  - $c_{ij} = 0$ if $i = j$
- We could determine the expected amount of loss for an observation allocated to population i as $\sum_j c_{ij} p_{ij}$ where the $p_{ij}s$ are the probabilities of allocating an observation from population j into population i
- We want to minimize the amount of loss expected for our rule. Using a Bayes Discrimination, allocate $\mathbf{y}$ to the population j which minimizes $\sum_{k \neq j} c_{ij} \pi_k f_k(\mathbf{y})$
- We could assign equal probabilities to each group and get a maximum likelihood type rule. here, we would allocate $\mathbf{y}$ to population j which minimizes $\sum_{k \neq j} c_{jk} f_k(\mathbf{y})$

Example:

Two binomial populations, each of size 10, with probabilities $p_1 = .5$ and $p_2 = .7$

And the probability of being in the first population is .9

However, suppose the cost of inappropriately allocating into the first population is 1 and the cost of incorrectly allocating into the second population is 5.

In this case, we pick population 1 over population 2

In general, we consider two regions, R_1 and R_2 associated with population 1 and 2:

$$R_1 : \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} \geq \frac{c_{12}\pi_2}{c_{21}\pi_1}$$

$$R_2 : \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} < \frac{c_{12}\pi_2}{c_{21}\pi_1}$$

where c_{12} is the cost of assigning a member of population 2 to population 1.

21.4.1.2 Discrimination Under Estimation

Suppose we know the form of the distributions for populations of interests, but we still have to estimate the parameters.

Example:

we know the distributions are multivariate normal, but we have to estimate the means and variances

21.4.1.3 The maximum likelihood discriminant rule allocates an observation y to population j when j maximizes the function

$$f_j(\mathbf{y}|\hat{\theta})$$

where $\hat{\theta}$ are the maximum likelihood estimates of the unknown parameters

For instance, we have 2 multivariate normal populations with distinct means, but common variance covariance matrix

MLEs for $_1$ and $_2$ are $\bar{\mathbf{y}}_1$ and $\bar{\mathbf{y}}_2$and common is $\mathbf{S}$.

Thus, an estimated discriminant rule could be formed by substituting these sample values for the population values

21.4.2 Probabilities of Misclassification

When the distribution are exactly known, we can determine the misclassification probabilities exactly. however, when we need to estimate the population parameters, we have to estimate the probability of misclassification

- Naive method

  - Plugging the parameters estimates into the form for the misclassification probabilities results to derive at the estimates of the misclassification probability.
  - But this will tend to be optimistic when the number of samples in one or more populations is small.
- Resubstitution method
  - Use the proportion of the samples from population i that would be allocated to another population as an estimate of the misclassification probability
  - But also optimistic when the number of samples is small
- Jack-knife estimates:
  - The above two methods use observation to estimate both parameters and also misclassification probabilities based upon the discriminant rule
  - Alternatively, we determine the discriminant rule based upon all of the data except the k-th observation from the j-th population
  - then, determine if the k-th observation would be misclassified under this rule
  - perform this process for all n_j observation in population j . An estimate fo the misclassficaiton probability would be the fraction of n_j observations which were misclassified
  - repeat the process for other $i \neq j$ populations
  - This method is more reliable than the others, but also computationally intensive
- Cross-Validation

Summary

Consider the group-specific densities $f_j(\mathbf{x})$ for multivariate vector $\mathbf{x}$.

Assume equal misclassification costs, the Bayes classification probability of $\mathbf{x}$ belonging to the j-th population is

$$p(j|\mathbf{x}) = \frac{\pi_j f_j(\mathbf{x})}{\sum_{k=1}^{h} \pi_k f_k(\mathbf{x})}$$

$j = 1, \dots, h$

where there are h possible groups.

We then classify into the group for which this probability of membership is largest

Alternatively, we can write this in terms of a **generalized squared distance** formation

$$D_j^2(\mathbf{x}) = d_j^2(\mathbf{x}) + g_1(j) + g_2(j)$$

where

- $d_j^2(\mathbf{x}) = (\mathbf{x} - {}_j)'\mathbf{V}_j^{-1}(\mathbf{x} - {}_j)$ is the squared Mahalanobis distance from $\mathbf{x}$ to the centroid of group j, and
  - $\mathbf{V}_j = \mathbf{S}_j$ if the within group covariance matrices are not equal
  - $\mathbf{V}_j = \mathbf{S}_p$ if a pooled covariance estimate is appropriate

and

$$g_1(j) = \begin{cases} \ln|\mathbf{S}_j| & \text{within group covariances are not equal} \\ 0 & \text{pooled covariance} \end{cases}$$

$$g_2(j) = \begin{cases} -2\ln\pi_j & \text{prior probabilities are not equal} \\ 0 & \text{prior probabilities are equal} \end{cases}$$

then, the posterior probability of belonging to group j is

$$p(j|\mathbf{x}) = \frac{\exp(-.5D_j^2(\mathbf{x}))}{\sum_{k=1}^h \exp(-.5D_k^2(\mathbf{x}))}$$

where $j = 1, \dots, h$

and $\mathbf{x}$ is classified into group j if $p(j|\mathbf{x})$ is largest for $j = 1, \dots, h$ (or, $D_j^2(\mathbf{x})$ is smallest).

21.4.3 Unknown Populations/ Nonparametric Discrimination

When your multivariate data are not Gaussian, or known distributional form at all, we can use the following methods

21.4.3.1 Kernel Methods

We approximate $f_j(\mathbf{x})$ by a kernel density estimate

$$\hat{f}_j(\mathbf{x}) = \frac{1}{n_j}\sum_{i=1}^{n_j} K_j(\mathbf{x} - \mathbf{x}_i)$$

where

- $K_j(.)$ is a kernel function satisfying $\int K_j(\mathbf{z})d\mathbf{z} = 1$
- $\mathbf{x}_i$, $i = 1, \dots, n_j$ is a random sample from the j-th population.

Thus, after finding $\hat{f}_j(\mathbf{x})$ for each of the h populations, the posterior probability of group membership is

$$p(j|\mathbf{x}) = \frac{\pi_j \hat{f}_j(\mathbf{x})}{\sum_{k-1}^h \pi_k \hat{f}_k(\mathbf{x})}$$

where $j = 1, \dots, h$

There are different choices for the kernel funciton:

- Uniform
- Normal
- Epanechnikov
- Biweight
- Triweight

We these kernels, we have to pick the "radius" (or variance, width, window width, bandwidth) of the kernel, which is a smoothing parameter (the larger the radius, the more smooth the kernel estimate of the density).

To select the smoothness parameter, we can use the following method

If we believe the populations were close to multivariate normal, then

$$R = (\frac{4/(2p+1)}{n_j})^{1/(p+1}$$

But since we do not know for sure, we might choose several different values and select one that vies the best out of sample or cross-validation discrimination.

Moreover, you also have to decide whether to use different kernel smoothness for different populations, which is similar to the individual andpooled covariances in the classical methodology.

21.4.3.2 Nearest Neighbor Methods

The nearest neighbor (also known as k-nearest neighbor) method performs the classification of a new observation vector based on the group membership of its nearest neighbors. In practice, we find

$$d_{ij}^2(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x}, \mathbf{x}_i) V_j^{-1} (\mathbf{x}, \mathbf{x}_i)$$

which is the distance between the vector $\mathbf{x}$ and the i-th observation in group j

We consider different choices for $\mathbf{V}_j$

For example,

$$\mathbf{V}_j = \mathbf{S}_p \mathbf{V}_j = \mathbf{S}_j \mathbf{V}_j = \mathbf{I} \mathbf{V}_j = diag(\mathbf{S}_p)$$

We find the k observations that are closest to $\mathbf{x}$ (where users pick k). Then we classify into the most common population, weighted by the prior.

21.4.3.3 Modern Discriminant Methods

Note:

Logistic regression (with or without random effects) is a flexible model-based procedure for classification between two populations.

The extension of logistic regression to the multi-group setting is polychotomous logistic regression (or, mulinomial regression).

The machine learning and pattern recognition are growing with strong focus on nonlienar discriminant analysis methods such as:

- radial basis function networks
- support vector machines
- multiplayer perceptrons (neural networks)

The general framework

$$g_j(\mathbf{x}) = \sum_{l=1}^{m} w_{jl} \phi_l(\mathbf{x};\ _l) + w_{j0}$$

where

- $j = 1, \dots, h$
- m nonlinear basis functions ϕ_l, each of which has n_m parameters given by $\theta_l = \{\theta_{lk} : k = 1, \dots, n_m\}$

We assign $\mathbf{x}$ to the j-th population if $g_j(\mathbf{x})$ is the maximum for all $j = 1, \dots, h$

Development usually focuses on the choice and estimation of the basis functions, ϕ_l and the estimation of the weights w_{jl}

[More details](Statistical Pattern Recognition | Wiley Online Books

)

21.4.4 Application

```
library(class)
```

```
##
## Attaching package: 'class'
```

```
## The following object is masked from 'package:reshape':
##
##     condense
```

```
library(klaR)
```

```
## Warning: package 'klaR' was built under R version 4.0.5
```

```
library(MASS)
library(tidyverse)

## Read in the data
crops <- read.table("images/crops.txt")
names(crops) <- c("crop", "y1", "y2", "y3", "y4")
str(crops)
```

```
## 'data.frame':    36 obs. of  5 variables:
##  $ crop: chr  "Corn" "Corn" "Corn" "Corn" ...
##  $ y1  : int  16 15 16 18 15 15 12 20 24 21 ...
##  $ y2  : int  27 23 27 20 15 32 15 23 24 25 ...
##  $ y3  : int  31 30 27 25 31 32 16 23 25 23 ...
##  $ y4  : int  33 30 26 23 32 15 73 25 32 24 ...
```

```
## Read in test data
crops_test <- read.table("images/crops_test.txt")
names(crops_test) <- c("crop", "y1", "y2", "y3", "y4")
str(crops_test)
```

```
## 'data.frame':    5 obs. of  5 variables:
##  $ crop: chr  "Corn" "Soybeans" "Cotton" "Sugarbeets" ...
##  $ y1  : int  16 21 29 54 32
##  $ y2  : int  27 25 24 23 32
##  $ y3  : int  31 23 26 21 62
##  $ y4  : int  33 24 28 54 16
```

21.4.4.1 LDA

Default prior is proportional to sample size and `lda` and `qda` do not fit a constant or intercept term

```
## Linear discriminant analysis
lda_mod <- lda(crop ~ y1 + y2 + y3 + y4,
               data = crops)
lda_mod
```

```
## Call:
## lda(crop ~ y1 + y2 + y3 + y4, data = crops)
##
## Prior probabilities of groups:
##     Clover      Corn     Cotton   Soybeans Sugarbeets
##  0.3055556  0.1944444  0.1666667  0.1666667  0.1666667
##
## Group means:
##                  y1       y2       y3       y4
## Clover     46.36364 32.63636 34.18182 36.63636
## Corn       15.28571 22.71429 27.42857 33.14286
## Cotton     34.50000 32.66667 35.00000 39.16667
## Soybeans   21.00000 27.00000 23.50000 29.66667
## Sugarbeets 31.00000 32.16667 20.00000 40.50000
##
## Coefficients of linear discriminants:
##              LD1          LD2         LD3          LD4
## y1 -6.147360e-02  0.009215431 -0.02987075 -0.014680566
## y2 -2.548964e-02  0.042838972  0.04631489  0.054842132
## y3  1.642126e-02 -0.079471595  0.01971222  0.008938745
## y4  5.143616e-05 -0.013917423  0.05381787 -0.025717667
##
## Proportion of trace:
##    LD1    LD2    LD3    LD4
## 0.7364 0.1985 0.0576 0.0075
```

```
## Look at accuracy on the training data
lda_fitted <- predict(lda_mod,newdata = crops)
# Contingency table
lda_table <- table(truth = crops$crop, fitted = lda_fitted$class)
lda_table
```

```
##             fitted
## truth         Clover Corn Cotton Soybeans Sugarbeets
##   Clover           6    0      3        0          2
##   Corn             0    6      0        1          0
##   Cotton           3    0      1        2          0
##   Soybeans         0    1      1        3          1
##   Sugarbeets       1    1      0        2          2
```

```
# accuracy of 0.5 is just random (not good)

## Posterior probabilities of membership
crops_post <- cbind.data.frame(crops,
                               crop_pred = lda_fitted$class,
                               lda_fitted$posterior)
crops_post <- crops_post %>%
    mutate(missed = crop != crop_pred)
head(crops_post)
```

```
##   crop y1 y2 y3 y4 crop_pred     Clover      Corn    Cotton  Soybeans
## 1 Corn 16 27 31 33      Corn 0.08935164 0.4054296 0.1763189 0.2391845
## 2 Corn 15 23 30 30      Corn 0.07690181 0.4558027 0.1420920 0.2530101
## 3 Corn 16 27 27 26      Corn 0.09817815 0.3422454 0.1365315 0.3073105
## 4 Corn 18 20 25 23      Corn 0.10521511 0.3633673 0.1078076 0.3281477
## 5 Corn 15 15 31 32      Corn 0.05879921 0.5753907 0.1173332 0.2086696
## 6 Corn 15 32 32 15  Soybeans 0.09723648 0.3278382 0.1318370 0.3419924
##   Sugarbeets missed
## 1 0.08971545  FALSE
## 2 0.07219340  FALSE
## 3 0.11573442  FALSE
## 4 0.09546233  FALSE
## 5 0.03980738  FALSE
## 6 0.10109590   TRUE
```

```
# posterior shows that posterior of corn membershp is much higher than the prior

## LOOCV
# leave-one-out cross validation for linear discriminant analysis
# cannot run the prdecit function using the object with CV = TRUE because it returns the wihtin s
lda_cv <- lda(crop ~ y1 + y2 + y3 + y4,
              data = crops, CV = TRUE)
# Contingency table
lda_table_cv <- table(truth = crops$crop, fitted = lda_cv$class)
lda_table_cv
```

```
##             fitted
## truth        Clover Corn Cotton Soybeans Sugarbeets
##   Clover          4    3      1        0          3
##   Corn            0    4      1        2          0
##   Cotton          3    0      0        2          1
##   Soybeans        0    1      1        3          1
##   Sugarbeets      2    1      0        2          1
```

```
## Predict the test data
lda_pred <- predict(lda_mod, newdata = crops_test)

## Make a contingency table with truth and most likely class
table(truth=crops_test$crop, predict=lda_pred$class)
```

```
##             predict
## truth        Clover Corn Cotton Soybeans Sugarbeets
##   Clover          0    0      1        0          0
##   Corn            0    1      0        0          0
##   Cotton          0    0      0        1          0
##   Soybeans        0    0      0        1          0
##   Sugarbeets      1    0      0        0          0
```

LDA didn't do well on both within sample and out-of-sample data.

21.4.4.2 QDA

```
## Quadratic discriminant analysis
qda_mod <- qda(crop ~ y1 + y2 + y3 + y4,
               data = crops)

## Look at accuracy on the training data
qda_fitted <- predict(qda_mod, newdata = crops)
# Contingency table
qda_table <- table(truth = crops$crop, fitted = qda_fitted$class)
qda_table
```

```
##               fitted
## truth        Clover Corn Cotton Soybeans Sugarbeets
##   Clover          9    0      0        0          2
##   Corn            0    7      0        0          0
##   Cotton          0    0      6        0          0
##   Soybeans        0    0      0        6          0
##   Sugarbeets      0    0      1        1          4
```

```
## LOOCV
qda_cv <- qda(crop ~ y1 + y2 + y3 + y4,
              data = crops, CV = TRUE)
# Contingency table
qda_table_cv <- table(truth = crops$crop, fitted = qda_cv$class)
qda_table_cv
```

```
##               fitted
## truth        Clover Corn Cotton Soybeans Sugarbeets
##   Clover          9    0      0        0          2
##   Corn            3    2      0        0          2
##   Cotton          3    0      2        0          1
##   Soybeans        3    0      0        2          1
##   Sugarbeets      3    0      1        1          1
```

```
## Predict the test data
qda_pred <- predict(qda_mod, newdata = crops_test)
## Make a contingency table with truth and most likely class
```

```
table(truth = crops_test$crop, predict = qda_pred$class)
```

```
##             predict
## truth        Clover Corn Cotton Soybeans Sugarbeets
##   Clover          1    0      0        0          0
##   Corn            0    1      0        0          0
##   Cotton          0    0      1        0          0
##   Soybeans        0    0      0        1          0
##   Sugarbeets      0    0      0        0          1
```

21.4.4.3 KNN

knn uses design matrices of the features.

```
## Design matrices
X_train <- crops %>%
    dplyr::select(-crop)
X_test <- crops_test %>%
    dplyr::select(-crop)
Y_train <- crops$crop
Y_test <- crops_test$crop

## Nearest neighbors with 2 neighbors
knn_2 <- knn(X_train, X_train, Y_train, k = 2)
table(truth = Y_train, fitted = knn_2)
```

```
##             fitted
## truth        Clover Corn Cotton Soybeans Sugarbeets
##   Clover          7    0      3        0          1
##   Corn            0    5      0        2          0
##   Cotton          1    0      5        0          0
##   Soybeans        0    0      0        4          2
##   Sugarbeets      0    0      2        1          3
```

```
## Accuracy
mean(Y_train==knn_2)
```

```
## [1] 0.6666667
```

```
## Performance on test data
knn_2_test <- knn(X_train, X_test, Y_train, k = 2)
table(truth = Y_test, predict = knn_2_test)
```

```
##             predict
## truth        Clover Corn Cotton Soybeans Sugarbeets
##   Clover          1    0      0        0          0
##   Corn            0    1      0        0          0
##   Cotton          0    0      0        0          1
##   Soybeans          0    0      0        1          0
##   Sugarbeets        0    0      0        0          1
```

```
## Accuracy
mean(Y_test==knn_2_test)
```

```
## [1] 0.8
```

```
## Nearest neighbors with 3 neighbors
knn_3 <- knn(X_train, X_train, Y_train, k = 3)
table(truth = Y_train, fitted = knn_3)
```

```
##             fitted
## truth        Clover Corn Cotton Soybeans Sugarbeets
##   Clover          7    0      3        1          0
##   Corn            0    5      1        1          0
##   Cotton          1    0      4        0          1
##   Soybeans        0    0      1        3          2
##   Sugarbeets      0    0      0        3          3
```

```
## Accuracy
mean(Y_train==knn_3)
```

```
## [1] 0.6111111
```

```
## Performance on test data
knn_3_test <- knn(X_train, X_test, Y_train, k = 3)
table(truth = Y_test, predict = knn_3_test)
```

```
##             predict
## truth        Clover Corn Cotton Soybeans Sugarbeets
##   Clover          1    0      0        0          0
##   Corn            0    1      0        0          0
##   Cotton          0    0      1        0          0
##   Soybeans        0    0      0        1          0
##   Sugarbeets      0    0      0        0          1
```

```
## Accuracy
mean(Y_test==knn_3_test)
```

```
## [1] 1
```

21.4.4.4 Stepwise

Stepwise discriminant analysis using the `stepclass` in function in the `klaR` package.

```
step <- stepclass(
    crop ~ y1 + y2 + y3 + y4,
    data = crops,
    method = "qda",
    improvement = 0.15
)
```

```
##  `stepwise classification', using 10-fold cross-validated correctness rate of method qda'.

## 36 observations of 4 variables in 5 classes; direction: both

## stop criterion: improvement less than 15%.

## correctness rate: 0.43333;  in: "y1";  variables (1): y1
##
##  hr.elapsed min.elapsed sec.elapsed
##        0.00        0.00        0.22
```

```
step$process
```

```
##    step var varname result.pm
## 0 start   0      -- 0.0000000
## 1    in   1      y1 0.4333333
```

```
step$performance.measure
```

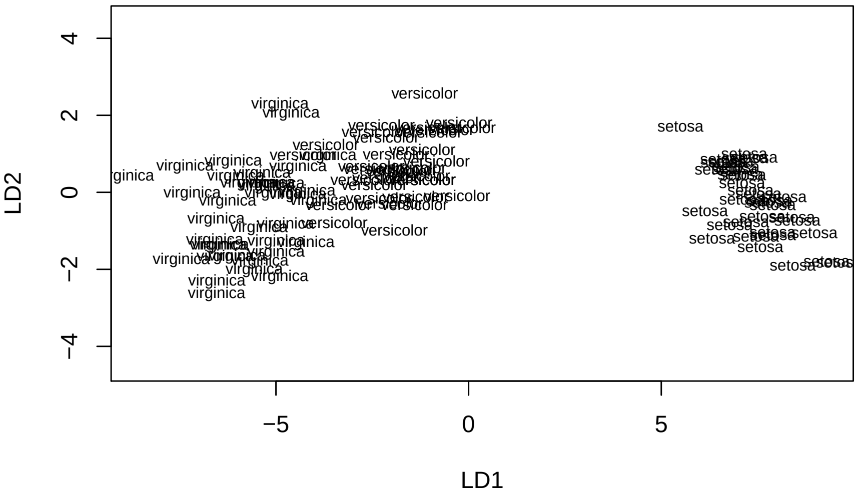
```
## [1] "correctness rate"
```

Iris Data

```
library(dplyr)
data('iris')
set.seed(1)
samp <-
    sample.int(nrow(iris), size = floor(0.70 * nrow(iris)), replace = F)

train.iris <- iris[samp,] %>% mutate_if(is.numeric,scale)
test.iris <- iris[-samp,] %>% mutate_if(is.numeric,scale)

library(ggplot2)
iris.model <- lda(Species ~ ., data = train.iris)
#pred
pred.lda <- predict(iris.model, test.iris)
table(truth = test.iris$Species, prediction = pred.lda$class)
```

```
##             prediction
## truth        setosa versicolor virginica
##   setosa         15          0         0
##   versicolor      0         17         0
##   virginica       0          0        13
```

```
plot(iris.model)
```



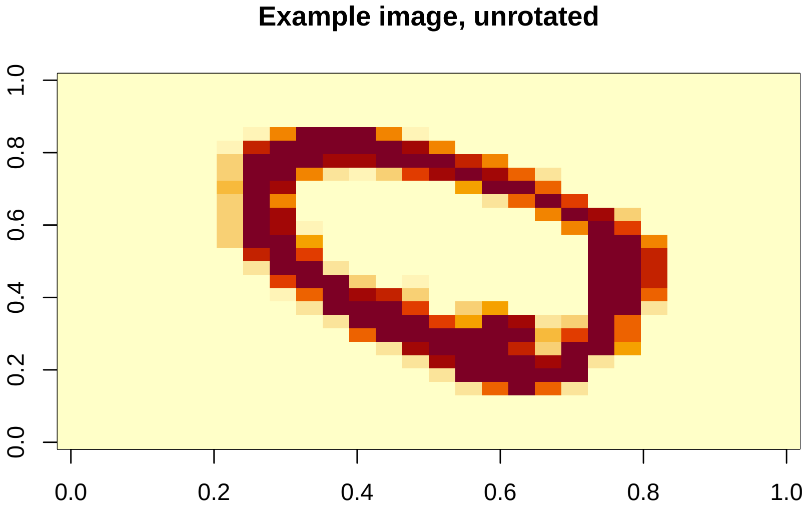
```r
iris.model.qda <- qda(Species~.,data=train.iris)
#pred
pred.qda <- predict(iris.model.qda,test.iris)
table(truth=test.iris$Species,prediction=pred.qda$class)
```

```
##             prediction
## truth        setosa versicolor virginica
##   setosa         15          0         0
##   versicolor      0         16         1
##   virginica       0          0        13
```

21.4.4.5 PCA with Discriminant Analysis

we can use both PCA for dimension reduction in discriminant analysis

```r
zeros <- as.matrix(read.table("images/mnist0_train_b.txt"))
nines <- as.matrix(read.table("images/mnist9_train_b.txt"))
train <- rbind(zeros[1:1000, ], nines[1:1000, ])
train <- train / 255 #divide by 255 per notes (so ranges from 0 to 1)
train <- t(train) #each column is an observation
image(matrix(train[, 1], nrow = 28), main = 'Example image, unrotated')
```



```
test <- rbind(zeros[2501:3000, ], nines[2501:3000, ])
test <- test / 255
test <- t(test)
y.train <- c(rep(0, 1000), rep(9, 1000))
y.test <- c(rep(0, 500), rep(9, 500))


library(MASS)
pc <- prcomp(t(train))
train.large <- data.frame(cbind(y.train, pc$x[, 1:10]))
large <- lda(y.train ~ ., data = train.large)
#the test data set needs to be constucted w/ the same 10 princomps
test.large <- data.frame(cbind(y.test, predict(pc, t(test))[, 1:10]))
pred.lda <- predict(large, test.large)
table(truth = test.large$y.test, prediction = pred.lda$class)
```

```
##      prediction
## truth   0   9
##     0 491   9
##     9   5 495
```

```
large.qda <- qda(y.train~.,data=train.large)
#prediction
pred.qda <- predict(large.qda,test.large)
table(truth=test.large$y.test,prediction=pred.qda$class)
```

```
##      prediction
## truth   0   9
##     0 493   7
##     9   3 497
```

21.5 Cluster Analysis

Chapter 22

Causality

After all of the mumbo jumbo that we have learned so far, I want to now talk about the concept of causality.
We usually say that correlation is not causation. Then, what is causation?
One of my favorite books has explained this concept beautifully (Mackenzie and Pearl, 2018). And I am just going to quickly summarize the gist of it from my understanding. I hope that it can give you an initial grasp on the concept so that later you can continue to read up and develop a deeper understanding.

It's important to have a deep understanding regarding the method research. However, one needs to be aware of its limitation and compliment with conceptual understanding. The aspect of concepts is typically referred in statistics when as expert knowledge. As mentioned in various sections throughout the book, we see that we need to ask experts for number as our baseline or visit literature to gain insight from past research.

Here, we dive in a more conceptual side statistical analysis as a whole, regardless of particular approach.

Chapter 23

Report

Structure

- Exploratory analysis
  - plots
  - preliminary results
  - interesting structure/features in the data
  - outliers
- Model
  - Assumptions
  - Why this model/ How is this model the best one?
  - Consideration: interactions, collinearity, dependence
- Model Fit
  - How well does it fit?
  - Are the model assumptions met?
    - Residual analysis
- Inference/ Prediction
  - Are there different way to support your inference?
- Conclusion
  - Recommendation
  - Limitation of the analysis
  - How to correct those in the future

This chapter is based on the `jtools` package. More information can be found here.

23.1 One summary table

```
library(jtools)
data(movies)
fit <- lm(metascore ~ budget + us_gross + year, data = movies)
summ(fit)
```

Observations	831 (10 missing obs. deleted)
Dependent variable	metascore
Type	OLS linear regression

F(3,827)	26.23
R^2	0.09
Adj. R^2	0.08

	Est.	S.E.	t val.	p
(Intercept)	52.06	139.67	0.37	0.71
budget	-0.00	0.00	-5.89	0.00
us_gross	0.00	0.00	7.61	0.00
year	0.01	0.07	0.08	0.94

Standard errors: OLS

```
summ(fit, scale = TRUE, vifs = TRUE, part.corr = TRUE, confint = TRUE, pvals = FALSE) #
```

Observations	831 (10 missing obs. deleted)
Dependent variable	metascore
Type	OLS linear regression

F(3,827)	26.23
R^2	0.09
Adj. R^2	0.08

	Est.	2.5%	97.5%	t val.	VIF	partial.r	part.r
(Intercept)	63.01	61.91	64.11	112.23	NA	NA	NA
budget	-3.78	-5.05	-2.52	-5.89	1.31	-0.20	-0.20
us_gross	5.28	3.92	6.64	7.61	1.52	0.26	0.25
year	0.05	-1.18	1.28	0.08	1.24	0.00	0.00

Standard errors: OLS; Continuous predictors are mean-centered and scaled by 1 s.d.

```r
#obtain clsuter-robust SE
data("PetersenCL", package = "sandwich")
fit2 <- lm(y ~ x, data = PetersenCL)
summ(fit2, robust = "HC3", cluster = "firm")
```

Observations	5000
Dependent variable	y
Type	OLS linear regression

F(1,4998)	1310.74
R^2	0.21
Adj. R^2	0.21

	Est.	S.E.	t val.	p
(Intercept)	0.03	0.07	0.44	0.66
x	1.03	0.05	20.36	0.00

Standard errors: Cluster-robust, type = HC3

Model to Equation

```r
# install.packages("equatiomatic")
fit <- lm(metascore ~ budget + us_gross + year, data = movies)
# show the theoretical model
equatiomatic::extract_eq(fit)
```

```
## Registered S3 methods overwritten by 'broom':
##   method            from
##   tidy.glht         jtools
##   tidy.summary.glht jtools
```

$$\text{metascore} = \alpha + \beta_{1}(\text{budget}) + \beta_{2}(\text{us\_gross}) + \beta_{3}(\text{year}) + \epsilon$$

```r
# display the actual coefficients
equatiomatic::extract_eq(fit, use_coefs = TRUE)
```

$$\widehat{\text{metascore}} = 52.06 + 0(\text{budget}) + 0(\text{us\_gross}) + 0.01(\text{year})$$

23.2 Model Comparison

```r
fit <- lm(metascore ~ log(budget), data = movies)
fit_b <- lm(metascore ~ log(budget) + log(us_gross), data = movies)
fit_c <- lm(metascore ~ log(budget) + log(us_gross) + runtime, data = movies)
coef_names <- c("Budget" = "log(budget)", "US Gross" = "log(us_gross)",
                "Runtime (Hours)" = "runtime", "Constant" = "(Intercept)")
export_summs(fit, fit_b, fit_c, robust = "HC3", coefs = coef_names)
```

```
## Warning in checkMatrixPackageVersion(): Package version inconsistency detected.
## TMB was built with Matrix version 1.2.18
## Current Matrix version is 1.3.2
## Please re-install 'TMB' from source using install.packages('TMB', type = 'source')
```

Table 23.1

	Model 1	Model 2	Model 3
Budget	-2.43 ***	-5.16 ***	-6.70 ***
	(0.44)	(0.62)	(0.67)
US Gross		3.96 ***	3.85 ***
		(0.51)	(0.48)
Runtime (Hours)			14.29 ***
			(1.63)
Constant	105.29 ***	81.84 ***	83.35 ***
	(7.65)	(8.66)	(8.82)
N	831	831	831
R2	0.03	0.09	0.17

Standard errors are heteroskedasticity robust. *** $p < 0.001$; ** $p < 0.01$; * $p < 0.05$.

Another package is `stargazer`

```r
library("stargazer")
```

```
##
## Please cite as:
## Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics
## R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```
stargazer(attitude)
```

```
##
## % Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fa
## % Date and time: Mon, May 10, 2021 - 5:18:55 PM
## \begin{table}[!htbp] \centering
##   \caption{}
##   \label{}
## \begin{tabular}{@{\extracolsep{5pt}}lccccccc}
## \\[-1.8ex]\hline
## \hline \\[-1.8ex]
## Statistic & \multicolumn{1}{c}{N} & \multicolumn{1}{c}{Mean} & \multicolumn{1}{c}{St. Dev.} &
## \hline \\[-1.8ex]
## rating & 30 & 64.633 & 12.173 & 40 & 58.8 & 71.8 & 85 \\
## complaints & 30 & 66.600 & 13.315 & 37 & 58.5 & 77 & 90 \\
## privileges & 30 & 53.133 & 12.235 & 30 & 45 & 62.5 & 83 \\
## learning & 30 & 56.367 & 11.737 & 34 & 47 & 66.8 & 75 \\
## raises & 30 & 64.633 & 10.397 & 43 & 58.2 & 71 & 88 \\
## critical & 30 & 74.767 & 9.895 & 49 & 69.2 & 80 & 92 \\
## advance & 30 & 42.933 & 10.289 & 25 & 35 & 47.8 & 72 \\
## \hline \\[-1.8ex]
## \end{tabular}
## \end{table}
```

```
## 2 OLS models
linear.1 <- lm(rating ~ complaints + privileges + learning + raises + critical,data = attitude)
linear.2 <- lm(rating ~ complaints + privileges + learning, data = attitude)
## create an indicator dependent variable, and run a probit model
attitude$high.rating <- (attitude$rating > 70)
probit.model <-
    glm(
        high.rating ~ learning + critical + advance,
        data = attitude,
        family = binomial(link = "probit")
    )
stargazer(linear.1,
          linear.2,
          probit.model,
          title = "Results",
          align = TRUE)
```

```
##
## % Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fa
## % Date and time: Mon, May 10, 2021 - 5:18:55 PM
## % Requires LaTeX packages: dcolumn
## \begin{table}[!htbp] \centering
##   \caption{Results}
##   \label{}
## \begin{tabular}{@{\extracolsep{5pt}}lD{.}{.}{-3} D{.}{.}{-3} D{.}{.}{-3} }
## \\[-1.8ex]\hline
## \hline \\[-1.8ex]
##  & \multicolumn{3}{c}{\textit{Dependent variable:}} \\
## \cline{2-4}
## \\[-1.8ex] & \multicolumn{2}{c}{rating} & \multicolumn{1}{c}{high.rating} \\
## \\[-1.8ex] & \multicolumn{2}{c}{\textit{OLS}} & \multicolumn{1}{c}{\textit{probit}}
## \\[-1.8ex] & \multicolumn{1}{c}{(1)} & \multicolumn{1}{c}{(2)} & \multicolumn{1}{c}
## \hline \\[-1.8ex]
##  complaints & 0.692^{***} & 0.682^{***} &  \\
##   & (0.149) & (0.129) &  \\
##   & & & \\
##  privileges & -0.104 & -0.103 &  \\
##   & (0.135) & (0.129) &  \\
##   & & & \\
##  learning & 0.249 & 0.238^{*} & 0.164^{***} \\
##   & (0.160) & (0.139) & (0.053) \\
##   & & & \\
##  raises & -0.033 &  &  \\
##   & (0.202) &  &  \\
##   & & & \\
##  critical & 0.015 &  & -0.001 \\
##   & (0.147) &  & (0.044) \\
##   & & & \\
##  advance &  &  & -0.062 \\
##   &  &  & (0.042) \\
##   & & & \\
##  Constant & 11.011 & 11.258 & -7.476^{**} \\
##   & (11.704) & (7.318) & (3.570) \\
##   & & & \\
## \hline \\[-1.8ex]
## Observations & \multicolumn{1}{c}{30} & \multicolumn{1}{c}{30} & \multicolumn{1}{c}
## R$^{2}$ & \multicolumn{1}{c}{0.715} & \multicolumn{1}{c}{0.715} &  \\
## Adjusted R$^{2}$ & \multicolumn{1}{c}{0.656} & \multicolumn{1}{c}{0.682} &  \\
## Log Likelihood &  &  & \multicolumn{1}{c}{-9.087} \\
## Akaike Inf. Crit. &  &  & \multicolumn{1}{c}{26.175} \\
## Residual Std. Error & \multicolumn{1}{c}{7.139 (df = 24)} & \multicolumn{1}{c}{6.863
## F Statistic & \multicolumn{1}{c}{12.063$^{***}$ (df = 5; 24)} & \multicolumn{1}{c}{2
## \hline
## \hline \\[-1.8ex]
## \textit{Note:}  & \multicolumn{3}{r}{$^{*}$p$<$0.1; $^{**}$p$<$0.05; $^{***}$p$<$0.0
## \end{tabular}
## \end{table}
```

```
# Latex
stargazer(
    linear.1,
    linear.2,
    probit.model,
    title = "Regression Results",
    align = TRUE,
    dep.var.labels = c("Overall Rating", "High Rating"),
    covariate.labels = c(
        "Handling of Complaints",
        "No Special Privileges",
        "Opportunity to Learn",
        "Performance-Based Raises",
        "Too Critical",
        "Advancement"
    ),
    omit.stat = c("LL", "ser", "f"),
    no.space = TRUE
)
```

```
# ASCII text output
stargazer(
    linear.1,
    linear.2,
    type = "text",
    title = "Regression Results",
    dep.var.labels = c("Overall Rating", "High Rating"),
    covariate.labels = c(
        "Handling of Complaints",
        "No Special Privileges",
        "Opportunity to Learn",
        "Performance-Based Raises",
        "Too Critical",
        "Advancement"
    ),
    omit.stat = c("LL", "ser", "f"),
    ci = TRUE,
    ci.level = 0.90,
    single.row = TRUE
)
```

```
##
## Regression Results
## ========================================================================
##                                         Dependent variable:
##                           -----------------------------------------------
##                                           Overall Rating
##                                     (1)                     (2)
## ------------------------------------------------------------------------
## Handling of Complaints   0.692*** (0.447, 0.937) 0.682*** (0.470, 0.894)
## No Special Privileges    -0.104 (-0.325, 0.118)  -0.103 (-0.316, 0.109)
## Opportunity to Learn      0.249 (-0.013, 0.512)   0.238* (0.009, 0.467)
## Performance-Based Raises -0.033 (-0.366, 0.299)
## Too Critical              0.015 (-0.227, 0.258)
## Advancement              11.011 (-8.240, 30.262) 11.258 (-0.779, 23.296)
## ------------------------------------------------------------------------
## Observations                        30                      30
## R2                                 0.715                   0.715
## Adjusted R2                        0.656                   0.682
## ========================================================================
## Note:                                        *p<0.1; **p<0.05; ***p<0.01
```

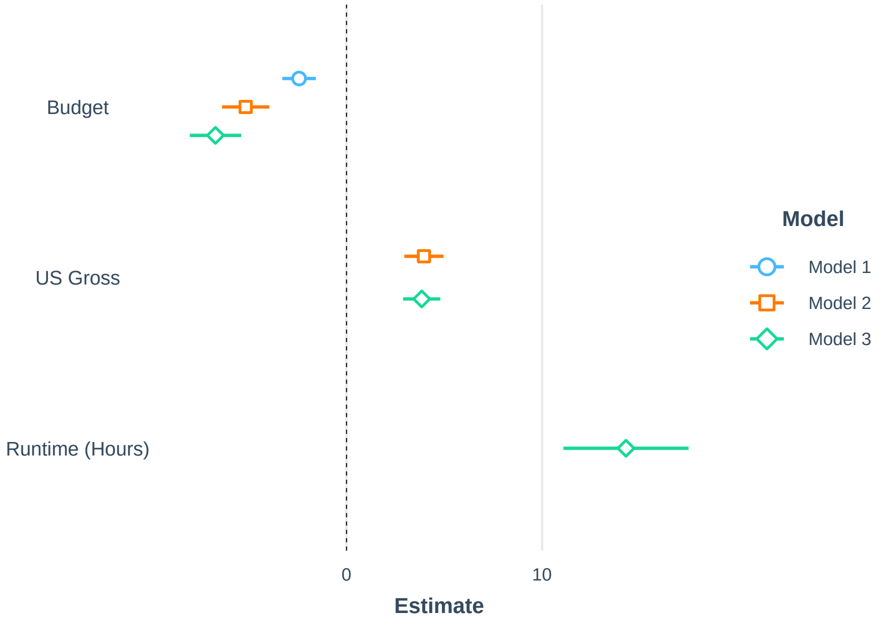
```
stargazer(
    linear.1,
    linear.2,
    probit.model,
    title = "Regression Results",
    align = TRUE,
    dep.var.labels = c("Overall Rating", "High Rating"),
    covariate.labels = c(
        "Handling of Complaints",
        "No Special Privileges",
        "Opportunity to Learn",
        "Performance-Based Raises",
        "Too Critical",
        "Advancement"
    ),
    omit.stat = c("LL", "ser", "f"),
    no.space = TRUE
)
```

Correlation Table

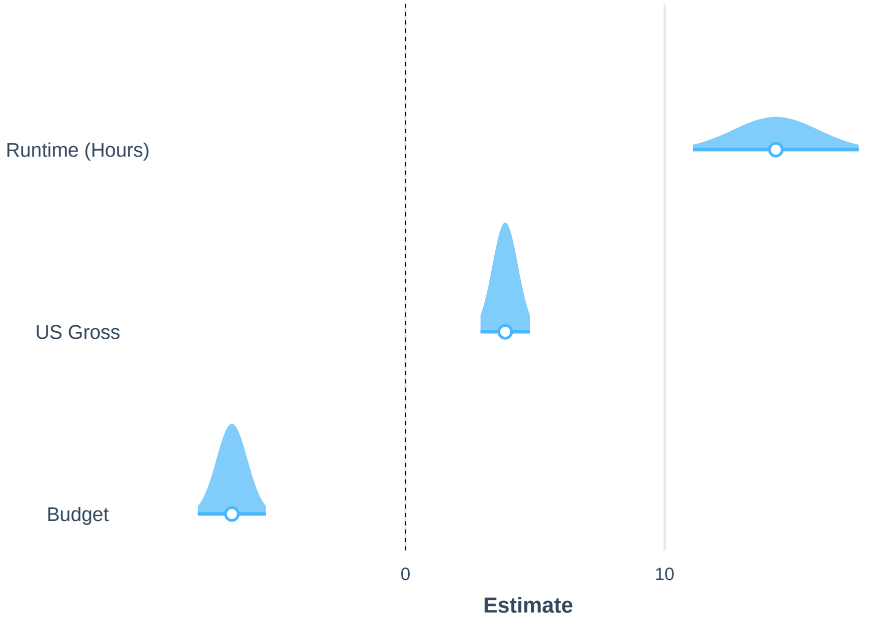
```
correlation.matrix <- cor(attitude[,c("rating","complaints","privileges")])
stargazer(correlation.matrix, title="Correlation Matrix")
```

23.3 Changes in an estimate

```
coef_names <- coef_names[1:3] # Dropping intercept for plots
plot_summs(fit, fit_b, fit_c, robust = "HC3", coefs = coef_names)
```



```r
plot_summs(fit_c, robust = "HC3", coefs = coef_names, plot.distributions = TRUE)
```



Appendix A

Appendix

A.1 Git

Cheat Sheet

Cheat Sheet in different languages

Learn Git

Interactive Cheat Sheet

Ultimate Guide of Git and GitHub for R user

- Setting up Git: `git config` with `--global` option to configure user name, email, editor, etc.
- Creating a repository: `git init` to initialize a repo. Git stores all of its repo data in the `.git` directory.
- Tracking changes:
    - `git status` shows the status of the repo
        - File are stored in the project's working directory (which users see)
        - The staging area (where the next commit is being built)
        - local repo is where commits are permanently recorded
    - `git add` put files in the staging area
    - `git commit` saves the staged content as a new commit in the local repo.
        - `git commit -m "your own message"` to give a messages for the purpose of your commit.

- History
  - `git diff` shows differences between commits
  - `git checkout` recovers old version of fields
    - `git checkout HEAD` to go to the last commit
    - `git checkout <unique ID of your commit>` to go to such commit
- Ignoring
  - `.gitignore` file tells Git what files to ignore
  - `cat . gitignore *.dat results/` ignore files ending with "dat" and folder "results".
- Remotes in GitHub
  - A local git repo can be connected to one or more remote repos.
  - Use the HTTPS protocol to connect to remote repos
  - `git push` copies changes from a local repo to a remote repo
  - `git pull` copies changes from a remote repo to a local repo
- Collaborating
  - `git clone` copies remote repo to create a local repo with a remote called `origin` automatically set up
- Branching
  - `git check - b <new-branch-name`
  - `git checkout master` to switch to master branch.
- Conflicts
  - occur when 2 or more people change the same lines of the same file
  - the version control system does not allow to overwrite each other's changes blindly, but highlights conflicts so that they can be resolved.
- Licensing
  - People who incorporate General Public License (GPL'd) software into their won software must make their software also open under the GPL license; most other open licenses do not require this.
  - The Creative Commons family of licenses allow people to mix and match requirements and restrictions on attribution, creation of derivative works, further sharing and commercialization.
- Citation:

  - Add a CITATION file to a repo to explain how you want others to cite your work.

- Hosting
  - Rules regarding intellectual property and storage of sensitive info apply no matter where code and data are hosted.

A.2 Short-cut

These are shortcuts that you probably you remember when working with R. Even though it might take a bit of time to learn and use them as your second nature, but they will save you a lot of time.
Just like learning another language, the more you speak and practice it, the more comfortable you are speaking it.

function	short-cut
navigate folders in console	`" " + tab`
pull up short-cut cheat sheet	`ctrl + shift + k`
go to file/function (everything in your project)	`ctrl + .`
search everything	`cmd + shift + f`
navigate between tabs	`Crtl + shift + .`
type function faster	`snip + shift + tab`
type faster	`use tab for fuzzy match`
`cmd + up`	
`ctrl + .`	

Sometimes you can't stage a folder because it's too large. In such case, use `Terminal` pane in Rstudio then type `git add -A` to stage all changes then commit and push like usual.

A.3 Function short-cut

apply one function to your data to create a new variable: `mutate(mod=map(data,function))`
instead of using `i in 1:length(object)`: `for (i in seq_along(object))`
apply multiple function: `map_dbl`
apply multiple function to multiple variables:`map2`
`autoplot(data)` plot times series data
`mod_tidy = linear(reg) %>% set_engine('lm') %>% fit(price ~ ., data=data)` fit lm model. It could also fit other models (stan, spark, glmnet, keras)

- Sometimes, data-masking will not be able to recognize whether you're calling from environment or data variables. To bypass this, we use `.data$variable` or `.env$variable`. For example `data %>% mutate(x=.env$variable/.data$variable`

- Problems with data-masking:
  - Unexpected masking by data-var: Use `.data` and `.env` to disambiguate
  - Data-var cant get through:
  - Tunnel data-var with {{}} + Subset `.data` with [[]]
- Passing Data-variables through arguments

```
library("dplyr")
mean_by <- function(data,by,var){
    data %>%
        group_by({{{by}}}) %>%
        summarise("{{var}}":=mean({{var}})) # new name for each var will be created by
}

mean_by <- function(data,by,var){
    data %>%
        group_by({{{by}}}) %>%
        summarise("{var}":=mean({{var}})) # use single {} to glue the string, but hard
}
```

- Trouble with selection:

```
library("purrr")
name <- c("mass","height")
starwars %>% select(name) # Data-var. Here you are referring to variable named "name"

starwars %>% select(all_of((name))) # use all_of() to disambiguate when

averages <- function(data,vars){ # take character vectors with all_of()
    data %>%
        select(all_of(vars)) %>%
        map_dbl(mean,na.rm=TRUE)
}

x = c("Sepal.Length","Petal.Length")
iris %>% averages(x)


# Another way
averages <- function(data,vars){ # Tunnel selectiosn with {{}}
```

```
    data %>%
        select({{vars}}) %>%
        map_dbl(mean,na.rm=TRUE)
}

x = c("Sepal.Length","Petal.Length")
iris %>% averages(x)
```

A.4 Citation

include a citation by `[@Farjam_2015]`

cite packages used in this session

`package=ls(sessionInfo()$loadedOnly) for (i in package){print(toBibtex(citation(i)))}`

```
package=ls(sessionInfo()$loadedOnly)
for (i in package){
    print(toBibtex(citation(i)))
    }
```

Appendix B

Bookdown cheat sheet

```
# to see non-scientific notation a result
format(12e-17, scientific = FALSE)
```

```
## [1] "0.00000000000000012"
```

B.1 Operation

R commands to do derivatives of a defined function Taking derivatives in R involves using the `expression`, `D`, and `eval` functions. You wrap the function you want to take the derivative of in expression(), then use D, then eval as follows.

simple example

```
#define a function
f=expression(sqrt(x))

#take the first derivative
df.dx=D(f,'x')
df.dx
```

```
## 0.5 * x^-0.5
```

```
#take the second derivative
d2f.dx2=D(D(f,'x'),'x')
d2f.dx2
```

```
## 0.5 * (-0.5 * x^-1.5)
```

Evaluate
* The first argument passed to eval is the expression you want to evaluate *

the second is a list containing the values of all quantities that are not defined elsewhere.

```
#evaluate the function at a given x
eval(f,list(x=3))
```

```
## [1] 1.732051
```

```
#evaluate the first derivative at a given x
eval(df.dx,list(x=3))
```

```
## [1] 0.2886751
```

```
#evaluate the second derivative at a given x
eval(d2f.dx2,list(x=3))
```

```
## [1] -0.04811252
```

B.2 Math Expresssion/ Syntax

Full list

Aligning equations

```
\begin{aligned}
a & = b \\
X &\sim {\sf Norm}(10, 3) \\
5 & \le 10
\end{aligned}
```

$$\begin{aligned}
a &= b \\
X &\sim \mathsf{Norm}(10, 3) \\
5 &\le 10
\end{aligned}$$

Syntax	Notation
Math	
`$\pm$`	$\pm$
`$\ge$`	$\ge$
`$\le$`	$\le$
`$\neq$`	$\neq$
`$\equiv$`	$\equiv$
`$^\circ$`	$^\circ$
`$\times$`	$\times$
`$\cdot$`	$\cdot$
`$\leq$`	$\leq$
`$\geq$`	$\geq$
`\propto`	$\propto$

Syntax	Notation
`$\subset$`	$\subset$
`$\subseteq$`	$\subseteq$
`$\leftarrow$`	$\leftarrow$
`$\rightarrow$`	$\rightarrow$
`$\Leftarrow$`	$\Leftarrow$
`$\Rightarrow$`	$\Rightarrow$
`$\approx$`	$\approx$
`$\mathbb{R}$`	$\mathbb{R}$
`$\sum_{n=1}^{10} n^2$`	$\sum_{n=1}^{10} n^2$
`$$\sum_{n=1}^{10} n^2$$`	$$\sum_{n=1}^{10} n^2$$
`$x^{n}$`	x^{n}
`$x_{n}$`	x_{n}
`$\overline{x}$`	$\overline{x}$
`$\hat{x}$`	$\hat{x}$
`$\tilde{x}$`	$\tilde{x}$
`\check{}`	$\check{}$
`\underset{\gamma}{\operatorname{argmin}}`	$\underset{\gamma}{\operatorname{argmin}}$
`$\frac{a}{b}$`	$\frac{a}{b}$
`$\frac{a}{b}$`	$\frac{a}{b}$
`$\displaystyle \frac{a}{b}$`	$\displaystyle \frac{a}{b}$
`$\binom{n}{k}$`	$\binom{n}{k}$
`$x_{1} + x_{2} + \cdots + x_{n}$`	$x_{1} + x_{2} + \cdots + x_{n}$
`$x_{1}, x_{2}, \dots, x_{n}$`	$x_{1}, x_{2}, \dots, x_{n}$
`\mathbf{x} = \langle x_{1}, x_{2}, \dots, x_{n}\rangle$`	$\mathbf{x} = \langle x_{1}, x_{2}, \dots, x_{n}\rangle$
`$x \in A$`	$x \in A$
`$\|A\|$`	$\|A\|$
`$x \in A$`	$x \in A$
`$x \subset B$`	$x \subset B$
`$x \subseteq B$`	$x \subseteq B$
`$A \cup B$`	$A \cup B$
`$A \cap B$`	$A \cap B$
`$X \sim {\sf Binom}(n, \pi)$`	$X \sim \mathsf{Binom}(n, \pi)$
`$\mathrm{P}(X \le x) = {\tt pbinom}(x, n, \pi)$`	$\mathrm{P}(X \le x) = \texttt{pbinom}(x, n, \pi)$
`$P(A \mid B)$`	$P(A \mid B)$
`$\mathrm{P}(A \mid B)$`	$\mathrm{P}(A \mid B)$
`$\{1, 2, 3\}$`	$\{1, 2, 3\}$
`$\sin(x)$`	$\sin(x)$

Syntax	Notation
`$\log(x)$`	$\log(x)$
`$\int_{a}^{b}$`	$\int_{a}^{b}$
`$\left(\int_{a}^{b} f(x) \; dx\right)$`	$\left(\int_{a}^{b} f(x) \; dx\right)$
`$\left[\int_{\-infty}^{\infty} f(x) \; dx\right]$`	$\left[\int_{-\infty}^{\infty} f(x) \; dx\right]$
`$\left. F(x) \right\|_{a}^{b}$`	$\left. F(x) \right\|_{a}^{b}$
`$\sum_{x = a}^{b} f(x)$`	$\sum_{x = a}^{b} f(x)$
`$\prod_{x = a}^{b} f(x)$`	$\prod_{x = a}^{b} f(x)$
`$\lim_{x \to \infty} f(x)$`	$\lim_{x \to \infty} f(x)$
`$\displaystyle \lim_{x \to \infty} f(x)$`	$\displaystyle \lim_{x \to \infty} f(x)$
Greek Letters	
`$\alpha A$`	αA
`$\beta B$`	βB
`$\gamma \Gamma$`	$\gamma \Gamma$
`$\delta \Delta$`	$\delta \Delta$
`$\epsilon \varepsilon E$`	$\epsilon \varepsilon E$
`$\zeta Z \sigma \,\!$`	$\zeta Z \sigma$
`$\eta H$`	ηH
`$\theta \vartheta \Theta$`	$\theta \vartheta \Theta$
`$\iota I$`	ιI
`$\kappa K$`	κK
`$\lambda \Lambda$`	$\lambda \Lambda$
`$\mu M$`	μM
`$\nu N$`	νN
`$\xi\Xi$`	$\xi\Xi$
`$o O$`	$o O$
`$\pi \Pi$`	$\pi \Pi$
`$\rho\varrho P$`	$\rho\varrho P$
`$\sigma \Sigma$`	$\sigma \Sigma$
`$\tau T$`	τT
`$\upsilon \Upsilon$`	$\upsilon \Upsilon$
`$\phi \varphi \Phi$`	$\phi \varphi \Phi$
`$\chi X$`	χX
`$\psi \Psi$`	$\psi \Psi$
`$\omega \Omega$`	$\omega \Omega$
`$\cdot$`	$\cdot$
`$\cdots$`	$\cdots$
`$\ddots$`	$\ddots$
`$\ldots$`	$\ldots$

Limit `P(\lim_{n\to \infty}\bar{X}_n =\mu) =1`

$$P(\lim_{n\to \infty}\bar{X}_n =\mu) =1$$

Matrices

```
$$\begin{array}
{rrr}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{array}
$$
```

$$\begin{array}{rrr}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{array}$$

```
$$\mathbf{X} = \left[\begin{array}
{rrr}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{array}\right]
$$
```

$$\mathbf{X} = \left[\begin{array}{rrr}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{array}\right]$$

Aligning Equations

Aligning Equations with Comments

```
\begin{aligned}
    3+x &=4 && \text{(Solve for} x \text{.)}\\
    x &=4-3 && \text{(Subtract 3 from both sides.)}\\
    x &=1   && \text{(Yielding the solution.)}
\end{aligned}
```

$$\begin{aligned}
3+x &=4 && \text{(Solve for} x \text{.)}\\
x &=4-3 && \text{(Subtract 3 from both sides.)}\\
x &=1 && \text{(Yielding the solution.)}
\end{aligned}$$

B.2.1 Statistics Notation

```
$$
f(y|N,p) = \frac{N!}{y!(N-y)!}\cdot p^y \cdot (1-p)^{N-y} = {{N}\choose{y}} \cdot p^y \
$$
```

$$f(y|N,p) = \frac{N!}{y!(N-y)!}\cdot p^y \cdot (1-p)^{N-y} = {{N}\choose{y}} \cdot p^y \cdot (1-p)^{N-y}$$

```
\begin{cases}
\frac{1}{b-a}&\text{for $x\in[a,b]$}\\
0&\text{otherwise}\\
\end{cases}
```

$$\begin{cases}
\frac{1}{b-a}&\text{for $x\in[a,b]$}\\
0&\text{otherwise}\\
\end{cases}$$

B.3 Table

```
+---------------+---------------+--------------------+
| Fruit         | Price         | Advantages         |
+===============+===============+====================+
| *Bananas*     | $1.34         | - built-in wrapper |
|               |               | - bright color     |
+---------------+---------------+--------------------+
| Oranges       | $2.10         | - cures scurvy     |
|               |               | - **tasty**        |
+---------------+---------------+--------------------+
```

Fruit	Price	Advantages
Bananas	$1.34	- built-in wrapper - bright color
Oranges	$2.10	- cures scurvy - **tasty**

```
(\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y}
```

$(\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y}$

Bibliography

Ahrens, H. and Pincus, R. (1981). On two measures of unbalancedness in a one-way model and their relation to efficiency. *Biometrical Journal*, 23(3):227–235.

Amemiya, T. (1984). Tobit models: A survey. *Journal of Econometrics*, 24(1-2):3–61.

Amemiya, T. and MaCurdy, T. E. (1986). Instrumental-variable estimation of an error-components model. *Econometrica*, 54(4):869.

Anderson, T. W. and Darling, D. A. (1952). Asymptotic theory of certain ”goodness of fit” criteria based on stochastic processes. *The Annals of Mathematical Statistics*, 23(2):193–212.

Balestra, P. and Varadharajan-Krishnakumar, J. (1987). Full information estimations of a system of simultaneous equations with error component structure. *Econometric Theory*, 3(2):223–246.

Baltagi, B. H. (1981). Simultaneous equations with error components. *Journal of Econometrics*, 17(2):189–200.

Baltagi, B. H. and Li, Q. (1991). A joint test for serial correlation and random individual effects. *Statistics & Probability Letters*, 11(3):277–280.

Baltagi, B. H. and Li, Q. (1995). Testing AR(1) against MA(1) disturbances in an error component model. *Journal of Econometrics*, 68(1):133–151.

Bareinboim, E., Tian, J., and Pearl, J. (2014). Proceedings of the twenty-eighth aaai conference on artificial intelligence.

Bates, D. M. and Watts, D. G. (1980). Relative curvature measures of nonlinearity. *Journal of the Royal Statistical Society: Series B (Methodological)*, 42(1):1–16.

Bates, D. M. and Watts, D. G. (1981). A relative offset orthogonality convergence criterion for nonlinear least squares. *Technometrics*, 23(2):179.

Beale, E. M. L. and Little, R. J. A. (1975). Missing values in multivariate analysis. *Journal of the Royal Statistical Society: Series B (Methodological)*, 37(1):129–145.

Bendel, R. B. and Afifi, A. A. (1977). Comparison of stopping rules in forward “stepwise” regression. *Journal of the American Statistical Association*, 72(357):46–53.

Bera, A. K. and Jarque, C. M. (1981). Efficient tests for normality, homoscedasticity and serial independence of regression residuals. *Economics Letters*, 7(4):313–318.

Bera, A. K., Sosa-Escudero, W., and Yoon, M. (2001). Tests for the error component model in the presence of local misspecification. *Journal of Econometrics*, 101(1):1–23.

Bodner, T. E. (2008). What improves with increased missing data imputations? *Structural Equation Modeling: A Multidisciplinary Journal*, 15(4):651–675.

BREUSCH, T. S. (1978). TESTING FOR AUTOCORRELATION IN DYNAMIC LINEAR MODELS∗. *Australian Economic Papers*, 17(31):334–355.

Breusch, T. S., Mizon, G. E., and Schmidt, P. (1989). Efficient estimation using panel data. *Econometrica*, 57(3):695.

Breusch, T. S. and Pagan, A. R. (1980). The lagrange multiplier test and its applications to model specification in econometrics. *The Review of Economic Studies*, 47(1):239.

Bruin, J. (2011). newtest: command to compute new test @ONLINE.

Ebbes, P., Wedel, M., B?ckenholt, U., and Steerneman, T. (2005). Solving and testing for regressor-error (in)dependence when no instrumental variables are available: With new evidence for the effect of education on income. *Quantitative Marketing and Economics*, 3(4):365–392.

EJD, Agresti, A., and Finlay, B. (1998). Statistical methods for the social sciences. *Journal of the American Statistical Association*, 93(442):844.

Faraway, J. J. (2016). *Extending the Linear Model with R.* Chapman and Hall/CRC.

Fox, J. (1991). *Maximum-Likelihood Methods, Score Tests, and Constructed Variables.*

Fuller, W. A. and Battese, G. E. (1974). Estimation of linear models with crossed-error structure. *Journal of Econometrics*, 2(1):67–78.

Furnival, G. M. and Wilson, R. W. (1974). Regressions by leaps and bounds. *Technometrics*, 16(4):499–511.

Glasser, M. (1964). Linear regression analysis with missing observations among the independent variables. *Journal of the American Statistical Association*, 59(307):834–844.

Godfrey, L. G. (1978). Testing against general autoregressive and moving average error models when the regressors include lagged dependent variables. *Econometrica*, 46(6):1293.

Gourieroux, C., Holly, A., and Monfort, A. (1982). Likelihood ratio test, wald test, and kuhn-tucker test in linear models with inequality constraints on the regression parameters. *Econometrica*, 50(1):63.

Gourieroux, C. and Monfort, A. (1981). On the problem of missing data in linear models. *The Review of Economic Studies*, 48(4):579.

Greene, W. H. (1990). *Econometric Analysis*.

Haitovsky, Y. (1968). Missing data in regression analysis. *Journal of the Royal Statistical Society: Series B (Methodological)*, 30(1):67–82.

Hausman, J. A. (1978). Specification tests in econometrics. *Econometrica*, 46(6):1251.

Heckman, J. J. (1976). The common structure of statistical models of truncation, sample selection and limited dependent variables and a simple estimator for such models. *Annals of Economic and Social Measurement*, 5(4):475–492.

Henderson, C. R. (1950). Estimation of genetic parameters. *Annals of Mathematical Statistics*, 21:309–310.

Honda, Y. (1985). Testing the error components model with non-normal disturbances. *The Review of Economic Studies*, 52(4):681.

HURVICH, C. M. and TSAI, C.-L. (1989). Regression and time series model selection in small samples. *Biometrika*, 76(2):297–307.

Johnson, A. R. and Wichern, D. W. (1988). Applied multivariate statistical analysis. *Biometrics*, 44(3):920.

Jones, M. P. (1996). Indicator and stratification methods for missing explanatory variables in multiple linear regression. *Journal of the American Statistical Association*, 91(433):222–230.

Kim, J.-S. and Frees, E. W. (2007). Multilevel modeling with correlated effects. *Psychometrika*, 72(4):505–533.

King, G., Honaker, J., Joseph, A., and Scheve, K. (2001). Analyzing incomplete political science data: An alternative algorithm for multiple imputation. *American Political Science Review*, 95(1):49–69.

King, M. L. and Wu, P. X. (1997). Locally optimal one-sided tests for multiparameter hypotheses. *Econometric Reviews*, 16(2):131–156.

Koenker, R. (1996). *Quantile Regression*.

Laird, N. M. and Ware, J. H. (1982). Random-effects models for longitudinal data. *Biometrics*, 38(4):963.

Laurent, R. T. S. and Cook, R. D. (1992). Leverage and superleverage in nonlinear regression. *Journal of the American Statistical Association*, 87(420):985.

Lewbel, A. (1997). Constructing instruments for regressions with measurement error when no additional data are available, with an application to patents and r&d. *Econometrica*, 65(5):1201.

Lewbel, A. (2012). Using heteroscedasticity to identify and estimate mismeasured and endogenous regressor models. *Journal of Business & Economic Statistics*, 30(1):67–80.

LIANG, K. and ZEGER, S. (1986). Longitudinal data analysis using generalized linear models. *Biometrika*, 73(1):13–22.

Little, R. J. A. (1992). Regression with missing x's: A review. *Journal of the American Statistical Association*, 87(420):1227.

Little, R. J. A. and Smith, P. J. (1987). Editing and imputation for quantitative survey data. *Journal of the American Statistical Association*, 82(397):58–68.

Looney, S. W. and Gulledge, T. R. (1985). Use of the correlation coefficient with normal probability plots. *The American Statistician*, 39(1):75.

Mackenzie, D. and Pearl, J. (2018). *The Book of Why: The New Science of Cause and Effect*. ISBN 978-1541698963.

Magel, R. C. and Hertsgaard, D. (1987). A collinearity diagnostic for nonlinear regression. *Communications in Statistics - Simulation and Computation*, 16(1):85–97.

MARDIA, K. V. (1970). Measures of multivariate skewness and kurtosis with applications. *Biometrika*, 57(3):519–530.

Marsaglia, G. and Marsaglia, J. (2004). Evaluating the anderson-darling distribution. *Journal of Statistical Software*, 9(2).

McCullagh, P. and Nelder, J. (2019). An outline of generalized linear models. In *Generalized Linear Models*, pages 21–47. Routledge.

Mroz, T. A. (1987). The sensitivity of an empirical model of married women's hours of work to economic and statistical assumptions. *Econometrica*, 55(4):765.

Nelder, J. A. and Wedderburn, R. W. M. (1972). Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3):370.

Nerlove, M. (1971). A note on error components models. *Econometrica*, 39(2):383.

Park, S. and Gupta, S. (2012). Handling endogenous regressors by joint estimation using copulas. *Marketing Science*, 31(4):567–586.

Rubin, D. B. (1996). Multiple imputation after 18+ years. *Journal of the American Statistical Association*, 91(434):473–489.

Satterthwaite, F. E. (1946). An approximate distribution of estimates of variance components. *Biometrics Bulletin*, 2(6):110.

Schabenberger, O. and Pierce, F. J. (2001). *Contemporary Statistical Models for the Plant and Soil Sciences*. CRC Press.

Shapiro, S. S. and Francia, R. S. (1972). An approximate analysis of variance test for normality. *Journal of the American Statistical Association*, 67(337):215–216.

Shapiro, S. S. and Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591.

Silverman, B. (1969). *Density Estimation for Statistics and Data Analysis*.

Stephens, M. A. (1974). EDF statistics for goodness of fit and some comparisons. *Journal of the American Statistical Association*, 69(347):730–737.

Stock, J. H. and Yogo, M. (2005). Testing for weak instruments in linear IV regression. In *Identification and Inference for Econometric Models*, pages 80–108. Cambridge University Press.

Swamy, P. A. V. B. and Arora, S. S. (1972). The exact finite sample properties of the estimators of coefficients in the error components regression models. *Econometrica*, 40(2):261.

Vach, W. (1994). *Logistic Regression with Missing Values in the Covariates*. Springer New York.

von Hippel, P. T. (2009). 8. how to impute interactions, squares, and other transformed variables. *Sociological Methodology*, 39(1):265–291.

Wallace, T. D. and Hussain, A. (1969). The use of error components models in combining cross section with time series data. *Econometrica*, 37(1):55.

Yu, K., Lu, Z., and Stander, J. (2003). Quantile regression: Applications and current research areas. *Journal of the Royal Statistical Society*, 52:331–350.