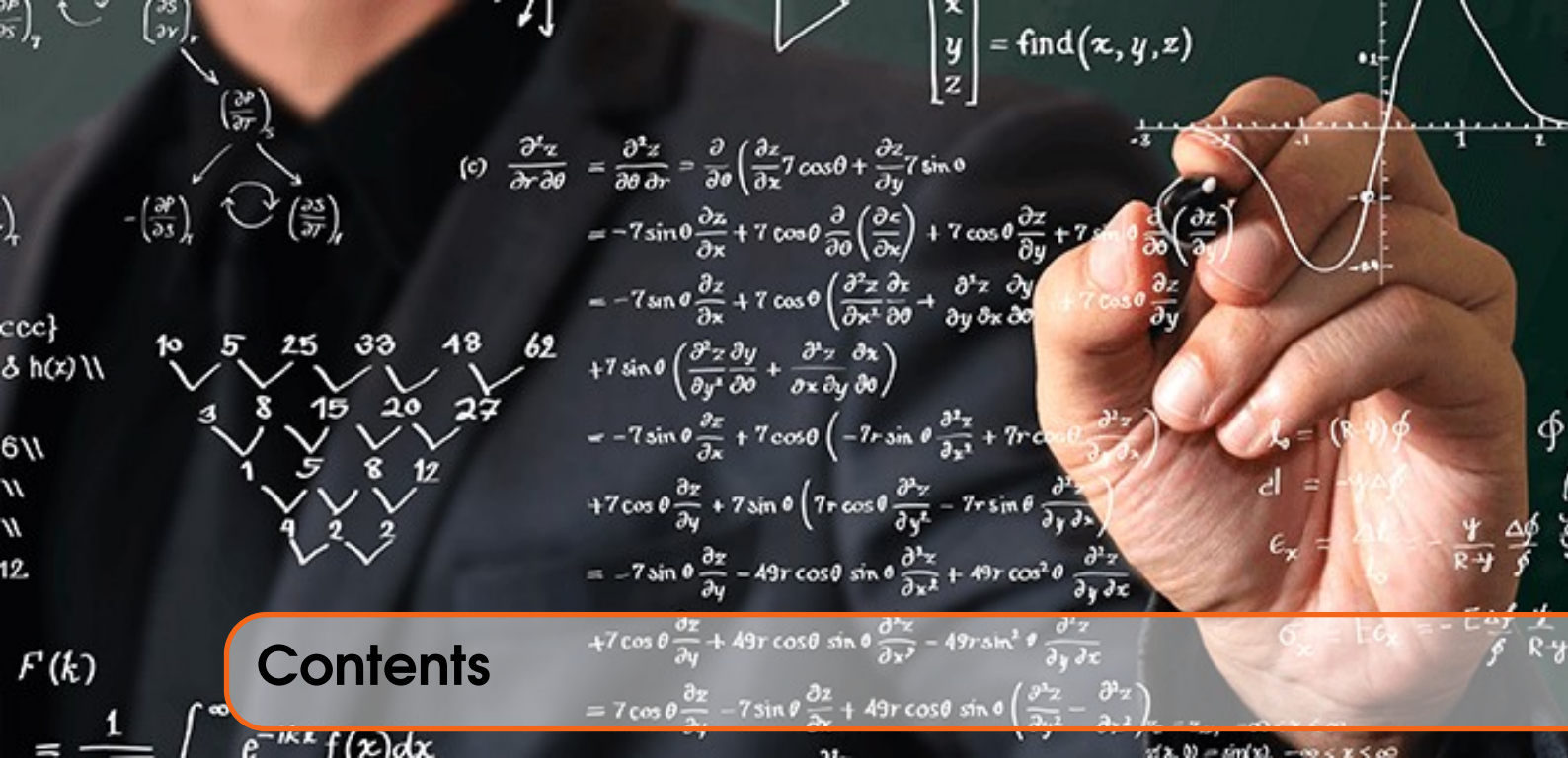
STAT 331 Course Notes

University of Waterloo

The One And Only
Waterloo 76er

Bill Zhuo



Contents

I STAT331 Main Content

1 Introduction . . . 7

1.1 Regression Model 7

1.2 Simple Linear Regression 8

1.2.1 Least Square Estimation . . . 9

1.2.2 Inference on Simple Linear Regression . . . 10

1.3 Prediction for SLR 13

1.3.1 Prediction Estimator and Error Estimator . . . 13

1.3.2 $(1-\alpha)$ Prediction Interval . . . 14

1.3.3 R Demo - Florange . . . 16

2 Multiple Linear Regression (MLR) . . . 19

2.1 Multiple Linear Regression 19

2.2 Linear Algebra Preparation 20

2.3 Multivariate Normal Distribution (MVN) 21

2.4 Inference for MLR 22

2.4.1 Residuals in MLR . . . 23

2.4.2 Inference for $\hat{\beta}$ . . . 23

2.5 Prediction for MLR 26

2.5.1 R Demo - Rocket . . . 28

2.6 Categorical Predicators in MLR 31

2.7 Analysis of Variance (ANOVA) 33

2.7.1 Summarize in ANOVA Table . . . 35

2.8 Hypothesis Testing Based on F-Distribution 36
2.8.1 R Demo - ANOVA F-Test . . . 41
2.9 Multicollinearity in Regression 45
2.9.1 R Demo - Multicollinearity . . . 47
2.10 Model Selection 51
2.10.1 Examples of Metrics for Model Selection . . . 51
2.10.2 Search Strategies . . . 55
2.10.3 R Demo - Model Selection . . . 56
2.11 Model Assumption Check 59
2.11.1 Model Assumption Check Options . . . 59
2.11.2 R Demo - Model Assumption Check . . . 62
2.11.3 Addressing Model Assumption Problems . . . 66
2.11.4 R Demo - Addressing Model Assumption Problems . . . 68
2.12 Effect of Indivudal Observations 69
2.12.1 Outliers . . . 69
2.12.2 Detection/Characterization of Outliers . . . 70
2.13.1 R Demo - Effect of Individual Observations . . . 74
2.14 Predictive Performance of MLR 76
2.15.1 R Demo - Cross-Validation . . . 78
2.16 More on K-Fold CV 79
2.16.1 R Demo - Beyond Model Selection . . . 81

3 General Linear Models (GLM) . . . 85
3.1 Introduction 85
3.2 La Dernière Classe 87
3.2.1 Project Tips/Q&A . . . 87
3.2.2 Residual (Sur)realism . . . 88
3.2.3 R Demo - Residual (Sur)realism . . . 89

STAT331 Main Content

1 Introduction 7

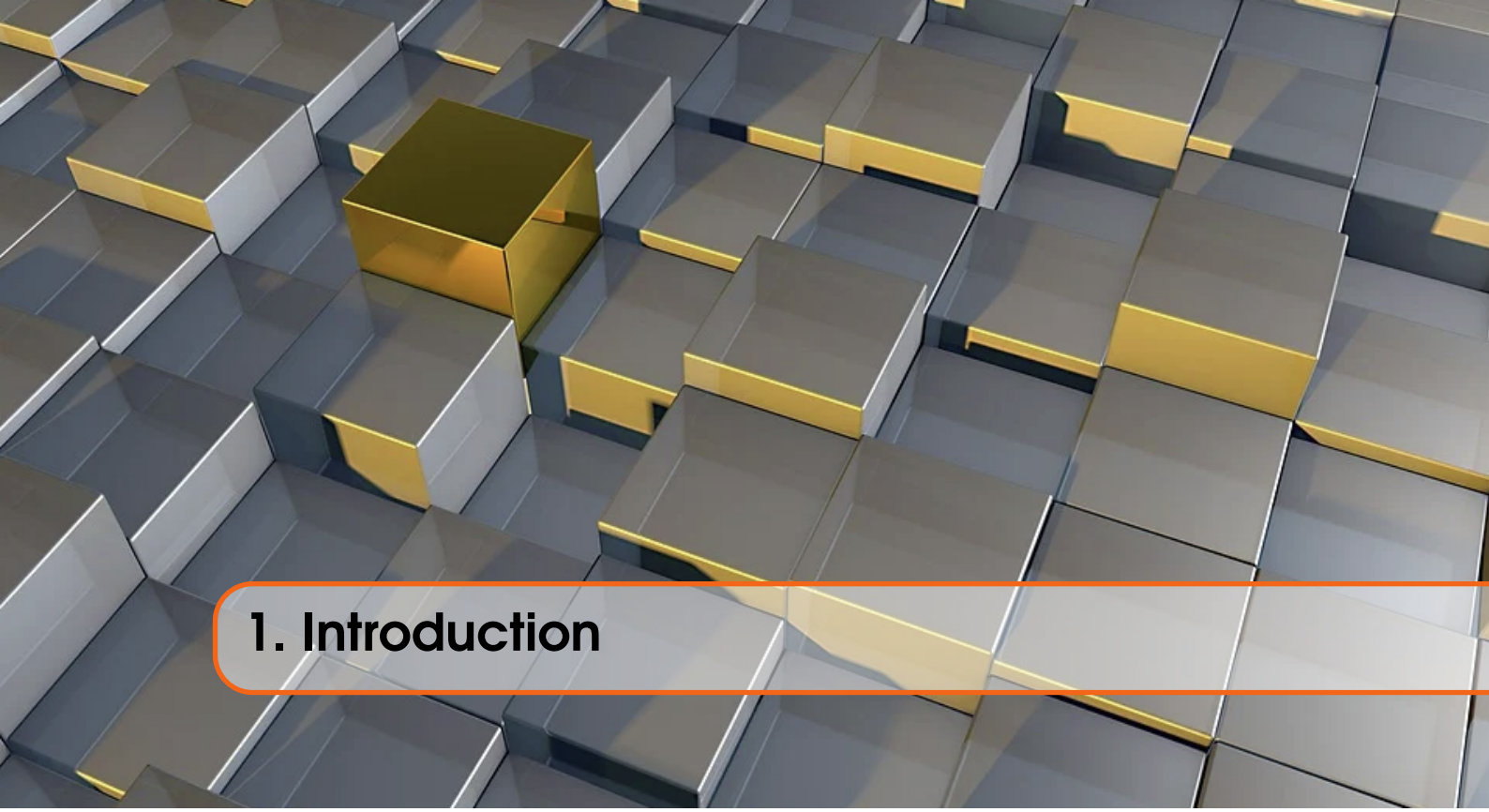
1.1 Regression Model
1.2 Simple Linear Regression
1.3 Prediction for SLR

2 Multiple Linear Regression (MLR) 19

2.1 Multiple Linear Regression
2.2 Linear Algebra Preparation
2.3 Multivariate Normal Distribution (MVN)
2.4 Inference for MLR
2.5 Prediction for MLR
2.6 Categorical Predicators in MLR
2.7 Analysis of Variance (ANOVA)
2.8 Hypothesis Testing Based on F-Distribution
2.9 Multicollinearity in Regression
2.10 Model Selection
2.11 Model Assumption Check
2.12 Effect of Indivudal Observations
2.14 Predictive Performance of MLR
2.16 More on K-Fold CV

3 General Linear Models (GLM) 85

3.1 Introduction
3.2 La Dernière Classe



1. Introduction

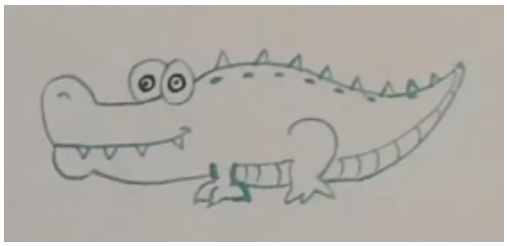
1.1 Regression Model

Definition 1.1.1 — Response and Explanatory Variables. Regression modelling infers the relationship between:

1. **Response (or dependent) variable**: the variable of primary interest, usually denoted by Y.
2. **Explanatory (or independent) varaible(s)**: sometimes can be called covariates, predictors, or features depends on the fields. These are the variables that potentially impact response

$$(x_1, x_2, \ldots, x_p)$$

■ **Example 1.1 — Alligator in Florida.**



Cute Alligator Drawn by Prof.Wong

We can analyze some alligator data, I mean why not?

1. Length (m)
2. Male/Female
3. Mass in stomach of (ewwww):
   (a) Fish
   (b) Invertibrates (small stuffs)
   (c) Reptiles (like another cute turtle)
   (d) Bird
   (e) Other

We shall let length to be our response variable Y and all the others as explanatory variables. ■

How do we make them related? Aha, functions!

Definition 1.1.2 — Regression Problem. We imagine we can explain Y in terms of $(x_1, x_2, \ldots, x_p)$ using some function so that

$$Y = f(x_1, \ldots, x_p)$$

So the task is to find such a f.

For the sake of STAT331, Applied Linear Model, it makes sense to have ...

Definition 1.1.3 — Linear Regression Problem. Linear Regression model assumes that

$$Y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \varepsilon$$

where

1. Y is the value of the response
2. $x_1, \cdots, x_p$ are the values of p explanatory variables, which **fixed constants** that do not have probability distributions.
3. $\beta_0, \beta_1, \cdots, \beta_p$ are the model parameters, in particular,
   (a) β_0 is called the intercept, interpreted as the expected value of Y given all $x_j = 0$ for $j = 1, \cdots, p$.
   (b) β_js quantify the effect of x_j on y.
4. ε is some random error: well, all models are wrong, but we can make the best use of it, cannot be correct all the time. So, this ε is a random variable a distribution, assumed in this course to be,
$$\varepsilon \sim N(0, \sigma^2)$$

R Okay, the only randomness of Y comes from ε, so what we are really looking at is just a normal random variable. We need to compute its mean and variance to characterize it.

$$\mathbb{E}(Y) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$
$$\mathbf{Var}(Y) = \mathbf{Var}(\varepsilon) = \sigma^2$$

thus,

$$Y \sim N(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p, \sigma^2)$$

1.2 Simple Linear Regression

Definition 1.2.1 — Simple Linear Regression. A linear model with response variable Y and one explanatory variable X.

$$Y = \beta_0 + \beta_1 x + \varepsilon$$

We are usually given data points $(x_i, y_i), i = 1, \cdots, n$. We can do a scatter plot

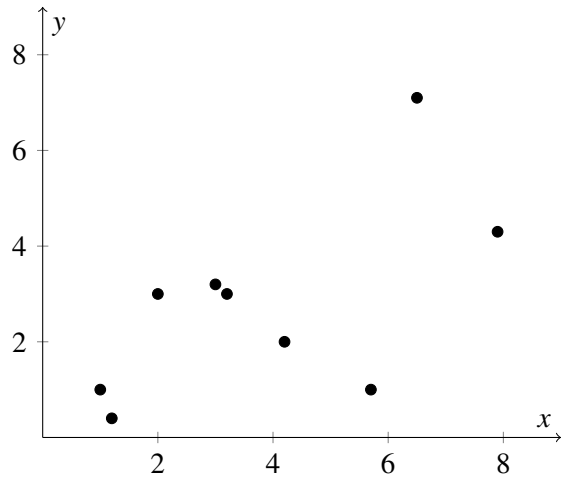


Figure 1.2.1: Scatter Plot

to visualize if there is a linear relationship between X and Y. We can also look at the correlation directly. Recall from probability theory, we have ...

Definition 1.2.2 — Correlation. If X, Y are r.v.s then

$$\rho = \mathbf{Corr}(X,Y) = \frac{\mathbf{Cov}(X,Y)}{\mathbf{SD}(X)\mathbf{SD}(Y)}$$

But we only have discrete data pairs, what can we do? Based on the data (x_i, y_i), we can estimate the sample correlation.

Definition 1.2.3 — Sample Correlation. For given bivariate data pairs $(x_i, y_i), i = 1, \cdots, n$, we have the sample correlation to be

$$r = \frac{\frac{1}{n-1}\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\frac{1}{n-1}\sum_{i=1}^n (x_i - \bar{x})^2}\sqrt{\frac{1}{n-1}\sum_{i=1}^n (y_i - \bar{y})^2}} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

where we shall denote, for later convenience,

$$S_{xy} = \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}), S_{xx} = \sum_{i=1}^n (x_i - \bar{x})^2, S_{yy} = \sum_{i=1}^n (y_i - \bar{y})^2$$

so, we can rewrite the sample correlation to be

$$r = \frac{S_{xy}}{\sqrt{S_{xx} S_{yy}}}$$

R

1. Sample correlation measures strength and direction of **linear** relationship.
2. $-1 \leq r \leq 1$, which can be shown by Cauchy-Schwarz inequality
3. $|r| \approx 1$ implies strong linear relationship
4. $|r| \approx 0$ means lack of linear relationship
5. $r > 0$ implies positive linear relationship
6. $r < 0$ implies negative linear relationship

but we cannot predict Y from x.

For data $(x_i, y_i), i = 1, \cdots, n$, we have the simple linear model to be

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

where $\varepsilon_i \overset{\text{i.i.d}}{\sim} N(0, \sigma^2)$. This setup means

$$Y_i \overset{\text{indep}}{\sim} N(\underbrace{\beta_0 + \beta_1 x_i}_{\mu_i = \mathbb{E}(Y_i)}, \underbrace{\sigma^2}_{\mathbf{Var}(Y_i)})$$

they are for sure independent but not necessarily identically distributed. Well, it seems like we need to find β_i. How to do it?

1.2.1 Least Square Estimation

Intuitively, we want the following distance squared to be small as a sum.

$$\min_{\beta_0, \beta_1} \sum_{i=1}^n [y_i - (\beta_0 + \beta_1 x_i)]^2 =: S(\beta_0, \beta_1)$$

This is exactly called the least square estimation of β_0, β_1. One may ask, "why square? why not any other even degrees or just absolute values?". Recall from STAT231, the least square estimations of β_0, β_1 are in fact the maximum likelihood estimates (MLE) of β_0, β_1 as well when ε_i are iid Normal. Since this S is a convex function, if we find a critical point by differentiation, it would give us the minimum.

$$\frac{\partial S}{\partial \beta_0} = 2\sum [y_i - (\beta_0 + \beta_1 x_i)](-1)$$
$$\frac{\partial S}{\partial \beta_1} = 2\sum [y_i - (\beta_0 + \beta_1 x_i)](-x_i)$$

we can set $\frac{\partial S}{\partial \beta_0} = \frac{\partial S}{\partial \beta_1} = 0$ simultaneously. The process is elementary algebra, we shall skip it here. We have the result to be

$$\begin{cases} \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \\ \hat{\beta}_1 = \frac{S_{xy}}{S_{xx}} \end{cases}$$

we call $\hat{\mu}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ to be the **fitted value** and $e_i = y_i - \hat{\mu}_i$ is called the **residual**.

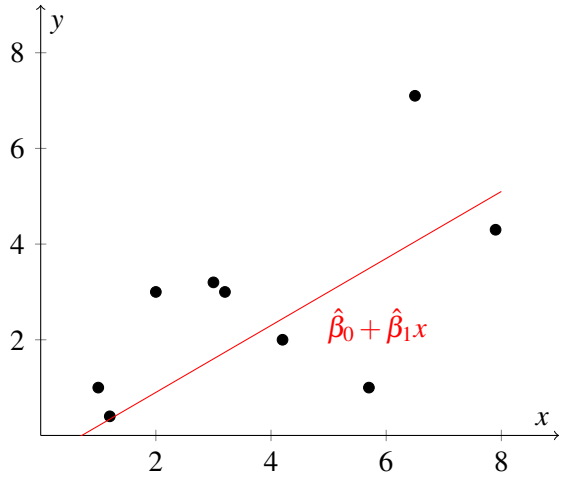


Figure 1.2.2: Simple Linear Regression

1.2.2 Inference on Simple Linear Regression

Recall the simple linear regression model with

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

. It has the regression line $y = \hat{\beta}_0 + \hat{\beta}_1 x$. The interpretations are:

1. $\hat{\beta}_0$ is the estimate of expected response when $x = 0$ (but not meaningful if outside range of x_i's in data)
2. $\hat{\beta}_1$ is the estimate of expected change in response of one unit increase in x
3. σ^2 is the variability around line

How to Estimate σ^2?

We know that

$$\varepsilon_i = Y_i - (\beta_0 + \beta_1 x_i)$$

as the random variable. And we have the residual (actual data)

$$e_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)$$

. The intuition is that we are going to use the variability of our residuals to estimate σ^2.
We use

$$\hat{\sigma}^2 = \frac{\sum(e_i - \hat{e})^2}{n-2} = \frac{\sum e_i^2}{n-2}$$

which looks like the sample variance of e_i's. The equation is true since

$$\bar{e} = \bar{y} - (\hat{\beta}_0 + \hat{\beta}_1 \bar{x}) = 0$$

by the computations of $\hat{\beta}_0, \hat{\beta}_1$. We shall denote **SS(Res)** $= \sum e_i^2$ as the sum of squared residuals.

R **Why do we have $n-2$ in the denominator?** For now, without looking at the linear algebra behind the curtain, we think of in a "degrees of freedom" way where $d.f. = n - $#estimate parameters. Here we have β_0, β_1. Bingo!

Another way we can look at this is the fact that $\tilde{\sigma}^2 = \frac{\sum \varepsilon_i^2}{n-2}$ is an unbiased estimator of σ^2.

For inference purposes, we want to answer whether there is a statistically significant relationship?

Theorem 1 — Linear combination of normals. Suppose $Y_i \sim N(\mu_i, \sigma_i^2)$ are all independent, then

$$\sum a_i Y_i \sim N\left(\sum a_i \mu_i, \sum a_i^2 \sigma_i^2\right)$$

for any constants a_i.

Let's try to interpret $\hat{\beta}_1$ as a random variable. We note that

$$\hat{\beta}_1 = \frac{\sum(x_i - \bar{x})(Y_i - \bar{Y})}{\sum(x_i - \bar{x})^2} = \frac{\sum(x_i - \bar{x})Y_i}{\sum(x_i - \bar{x})x_i}$$

we see that the random variables are $Y_i, \bar{Y}$ and $x_i, \bar{x}$ are constants. This can be viewed as a linear combination of normals, thus $\hat{\beta}_1$ is normal. So, $\hat{\beta}_1 = \sum a_i Y_i$ where $a_i = \frac{x_i - \bar{x}}{\sum x_i(x - \bar{x})}$. Then,

1. **Expectation**:

$$\begin{aligned}
\mathbb{E}(\hat{\beta}_1) &= \sum a_i \mathbb{E}(Y_i) \\
&= \frac{\sum(x_i - \bar{x})(\beta_0 + \beta_1 x_i)}{\sum x_i(x_i - \bar{x})} \\
&= \frac{\sum \beta_0(x_i - \bar{x}) + \beta_1 \sum x_i(x_i - \bar{x})}{\sum x_i(x_i - \bar{x})} \\
&= \beta_1
\end{aligned}$$

   This means our estimator is unbiased.

2. **Variance**:

$$\begin{aligned}
\mathbf{Var}\left(\hat{\beta}_1\right) &= \sum a_i^2 \mathbf{Var}(Y_i) = \sum a_i^2 \sigma^2 \\
&= \sigma^2 \frac{\sum(x_i - \bar{x})^2}{\left(\sum x_i(x_i - \bar{x})\right)^2} \\
&= \sigma^2 \frac{\sum(x_i - \bar{x})^2}{\left(\sum(x_i - \bar{x})^2\right)^2} \\
&= \frac{\sigma^2}{S_{xx}}
\end{aligned}$$

Thus,

$$\hat{\beta}_1 \sim N\left(\beta_1, \frac{\sigma^2}{S_{xx}}\right)$$

Similarly, we have

$$\hat{\beta}_0 \sim N\left(\beta_0, \sigma^2\left(\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}}\right)\right)$$

Again, $\hat{\beta}_0$ is also unbiased.

Exercise 1.1 Show

$$\hat{\beta}_0 \sim N\left(\beta_0, \sigma^2\left(\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}}\right)\right)$$

■

Then, by standardization, we have

$$\frac{\hat{\beta}_1 - \beta_1}{\sigma/\sqrt{S_{xx}}} \sim N(0,1)$$

where σ is unknown, we need to estimate it with $\hat{\sigma}$, then

$$\frac{\hat{\beta}_1 - \beta_1}{\hat{\sigma}/\sqrt{S_{xx}}} \sim t_{n-2}$$

where $\mathbf{SE}(\hat{\beta}_1) = \hat{\sigma}/\sqrt{S_{xx}}$ =:standard error. **Why t_{n-2}?**

Definition 1.2.4 — Student-t Distribution. We define a random variable

$$T = \frac{Z}{\sqrt{U/k}}$$

where $Z \sim N(0,1)$ and $U \sim \chi^2_k$ and they are independent. Then, define

$$T \sim t_k$$

to be the student-t distribution with k degrees of freedom.

R A useful fact here: given a SLR model

$$\frac{\hat{\sigma}^2(n-2)}{\sigma^2} = \frac{\mathbf{SS(Res)}}{\sigma^2} \sim \chi^2_{n-2}$$

Now, we are ready to crack this problem

Theorem 2

$$\frac{\hat{\beta}_1 - \beta_1}{\hat{\sigma}/\sqrt{S_{xx}}} \sim t_{n-2}$$

Proof.

$$\frac{\hat{\beta}_1 - \beta_1}{\hat{\sigma}/\sqrt{S_{xx}}} = \frac{\frac{\hat{\beta}_1 - \beta_1}{\sigma/\sqrt{S_{xx}}}}{\sqrt{\frac{\hat{\sigma}^2(n-2)}{\sigma^2} \cdot \frac{1}{n-2}}} \sim t_{n-2}$$

we note that $\frac{\hat{\beta}_1-\beta_1}{\sigma/\sqrt{S_{xx}}} \sim N(0,1)$, $\frac{\hat{\sigma}^2(n-2)}{\sigma^2} \sim \chi^2_{n-2}$. One can show that these two independent. ■

Confidence Interval for β_1

We note that geometrically,

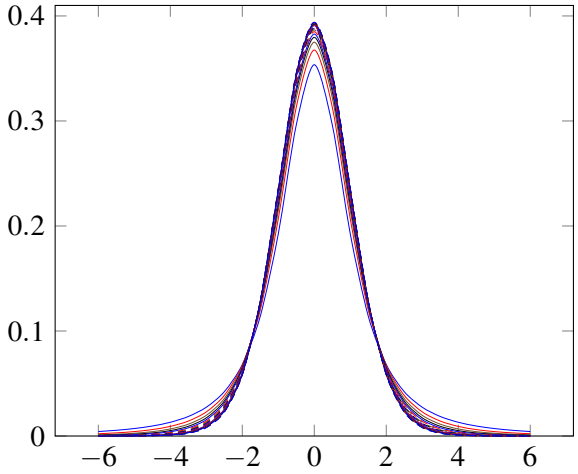


Figure 1.2.3: Student-t Distributions

the student-t distribution is symmetric. Thus, our confidence interval should be constructed in a symmetric fashion as well.

Definition 1.2.5 — $(1-\alpha)$ CI for β_1.

$$\hat{\beta}_1 \pm c\mathbf{SE}(\hat{\beta}_1)$$

where c is the $1-\frac{\alpha}{2}$ quantile of t_{n-2}, which means

$$\mathbb{P}(|T| \leq c) = 1-\alpha \text{ or } \mathbb{P}(T \leq c) = 1-\frac{\alpha}{2}$$

where $T \sim t_{n-2}$.

Hypothesis Testing on β_1

Our null hypothesis is $H_0 : \beta_1 = 0$ vs. $H_A : \beta_1 \neq 0$. If H_0 is true, then

$$\frac{\hat{\beta}_1-\beta_1}{\mathbf{SE}(\hat{\beta}_1)} = \frac{\hat{\beta}_1}{\mathbf{SE}(\hat{\beta}_1)} \sim t_{n-2}$$

so, calculate $t = \frac{\hat{\beta}_1}{\mathbf{SE}(\hat{\beta}_1)}$. Reject the H_0 at level α if $|t| > c$ where c is $1-\frac{\alpha}{2}$ quantile of t_{n-2}. Equivalently, we can calculate the

$$\text{p-value} = \mathbb{P}(|T| \geq |t|) = 2\mathbb{P}(T \geq |t|)$$

1.3 Prediction for SLR

1.3.1 Prediction Estimator and Error Estimator

Suppose we want to predict the response y for a given **new** value of x, say $x = x_0$. Then, SLR model says $Y_0 \sim N(\beta_0+\beta_1 x_0, \sigma^2)$ is our r.v. for response when $x = x_0$. And the fitted model predicts value of y to be $\hat{y}_0 = \hat{\beta}_0 + \hat{\beta}_1 x_0$.

As a r.v., $\hat{Y}_0 = \hat{\beta}_0 + \hat{\beta}_1 x_0$ where $\hat{\beta}_0, \hat{\beta}_1$ are random variables from last section. We know that $\hat{\beta}_0, \hat{\beta}_1$ are unbiased,

$$\mathbb{E}(\hat{Y}_0) = \beta_0 + \beta_1 x_0 = \mathbb{E}(Y_0)$$

This implies that $\hat{Y}_0$ is the unbiased estimator of the mean of Y_0.
For variance, by expressing $\hat{Y}_0$ as a linear combination of Y_i, one can show that

$$\mathbf{Var}(\hat{Y}_0) = \sigma^2 \left(\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{S_{xx}} \right)$$

Thus,

$$\hat{Y}_0 \sim N\left(\beta_0 + \beta_1 x_0, \sigma^2 \left(\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{S_{xx}} \right) \right)$$

Moreover, the r.v. for our **prediction error** is $Y_0 - \hat{Y}_0$.

R We note that $Y_0, \hat{Y}_0$ are independent since $\hat{Y}_0$ is a function of sampling r.v.s $Y_1, \cdots, Y_n$ while Y_0 is something completely unknown and new.

Now, we look at the expectation,

$$\mathbb{E}(Y_0 - \hat{Y}_0) = 0$$

and variance

$$\mathbf{Var}(Y_0 - \hat{Y}_0) = \mathbf{Var}(Y_0) + \mathbf{Var}(\hat{Y}_0) = \sigma^2 + \sigma^2 \left(\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{S_{xx}} \right)$$

Again, we have a linear combination of independent normals, so

$$Y_0 - \hat{Y}_0 \sim N\left(0, \sigma^2 \left(1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{S_{xx}} \right) \right)$$

R The intuition for prediction error that is being composed of 2 terms.

$$\underbrace{\sigma^2}_{(1)} + \underbrace{\sigma^2 \left(\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{S_{xx}} \right)}_{(2)}$$

2 sources of uncertainty:

1. (1) is the random error of the new observation
2. (2) is the random error coming from the estimation of β_0, β_1

We need to note that prediction may not make sense of x_0 outside range of x_i's in data. (Interpolation vs. exapolation)

1.3.2 $(1-\alpha)$ Prediction Interval

We note that

$$\frac{Y_0 - \hat{Y}_0}{\hat{\sigma}\sqrt{1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{S_{xx}}}} \sim t_{n-2}$$

Theorem 3 — $(1-\alpha)$ Prediction Interval for y_0.

$$\hat{y}_0 \pm c\hat{\sigma}\sqrt{1+\frac{1}{n}+\frac{(x_0-\bar{x})^2}{S_{xx}}}$$

where c is the $1-\frac{\alpha}{2}$ quantile of t_{n-2}.

■ **Example 1.2 — Orange Production (2018) in Florida.** Let x denote the arces of orange fields and y denote the number of boxes of oranges (1000s). (x_i, y_i) recorded for each of 25 Florida counties.

Cute Oranges Drawn by Prof.Wong

We have the following summary statistics:

1. $r = 0.964$
2. $\bar{x} = 16133$
3. $\bar{y} = 1798$
4. $S_{xx} = 1.245 \times 10^{10}$
5. $S_{xy} = 1.453 \times 10^{9}$

Let's do some calculations!

1. $\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}} = 0.1167$. This is the estimation of expected number of boxes produced to be approximately 117 higher per additional acre.
2. $\hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x} = -85.3$. Not meaningful to interpret since no county has $x = 0$.

Now suppose **SS(Res)** $= 1.31 \times 10^7$.

1. $\hat{\sigma}^2 = \frac{\sum e_i^2}{n-2} = \frac{1.31\times 10^7}{25-2} = 5.7 \times 10^5$
2. $\mathbf{SE}(\hat{\beta}_1) = \frac{\hat{\sigma}}{\sqrt{S_{xx}}} = 0.00676$
3. To test $H_0 : \beta_1 = 0$, we calculate

$$t = \frac{\hat{\beta}_1 - 0}{\mathbf{SE}(\hat{\beta}_1)} = \frac{0.1167}{0.00676} \approx 17.3$$

   e.g. the 97.5% quantile of t_{23} is 2.07. Since $17.3 > 2.07$, we think there is strong evidence against the null hypothesis since it is very unlikely to see this for a random draw from t_{23}. We reject H_0 at 5% level. We conclude that there is a significant linear relationship between acres and oranges produced.
4. 95% CI for β_1 is

$$0.1167 \pm 2.07 \times 0.00676$$

   by the correspondence between HT and CI, this CI does not contain 0.
5. p-value for $H_0 : \beta_1 = 0$ is $\mathbb{P}(|t_{23}| \geq 17.3) = 2\mathbb{P}(t_{23} \geq 17.3) \approx 1.2 \times 10^{-14}$.
6. Predict the number of boxes (1000s) produced if we had 10000 arces to grow oranges.

$$\hat{\beta}_0 + \hat{\beta}_1 x_0 = -85.3 + 0.1167(10000) \approx 1082$$

7. 95% PI for y_0 is

$$1082 \pm 2.07\sqrt{5.7 \times 10^3}\sqrt{1+\frac{1}{25}+\frac{6133^2}{1.245 \times 10^{10}}}$$

■

We are not trying to establish causation!

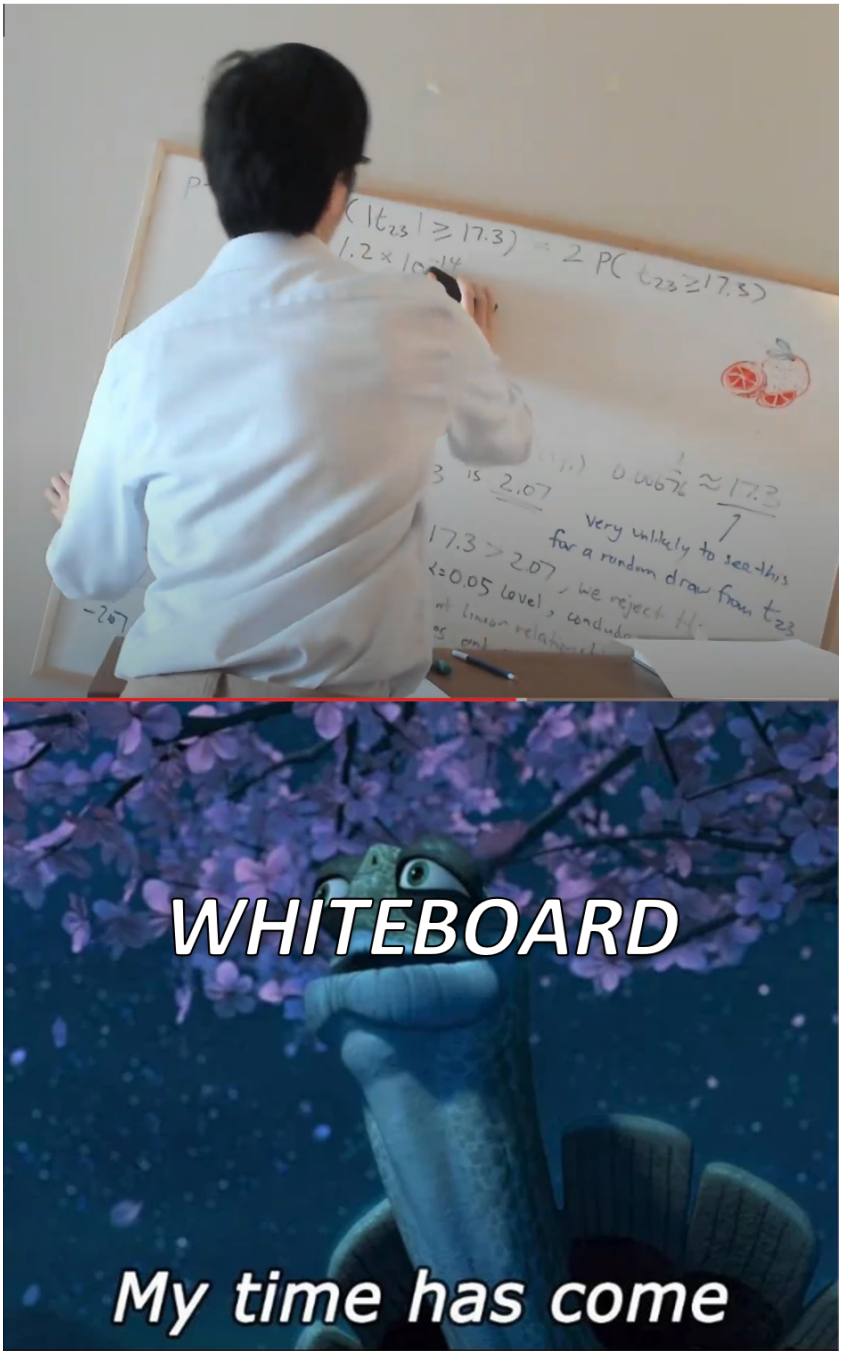


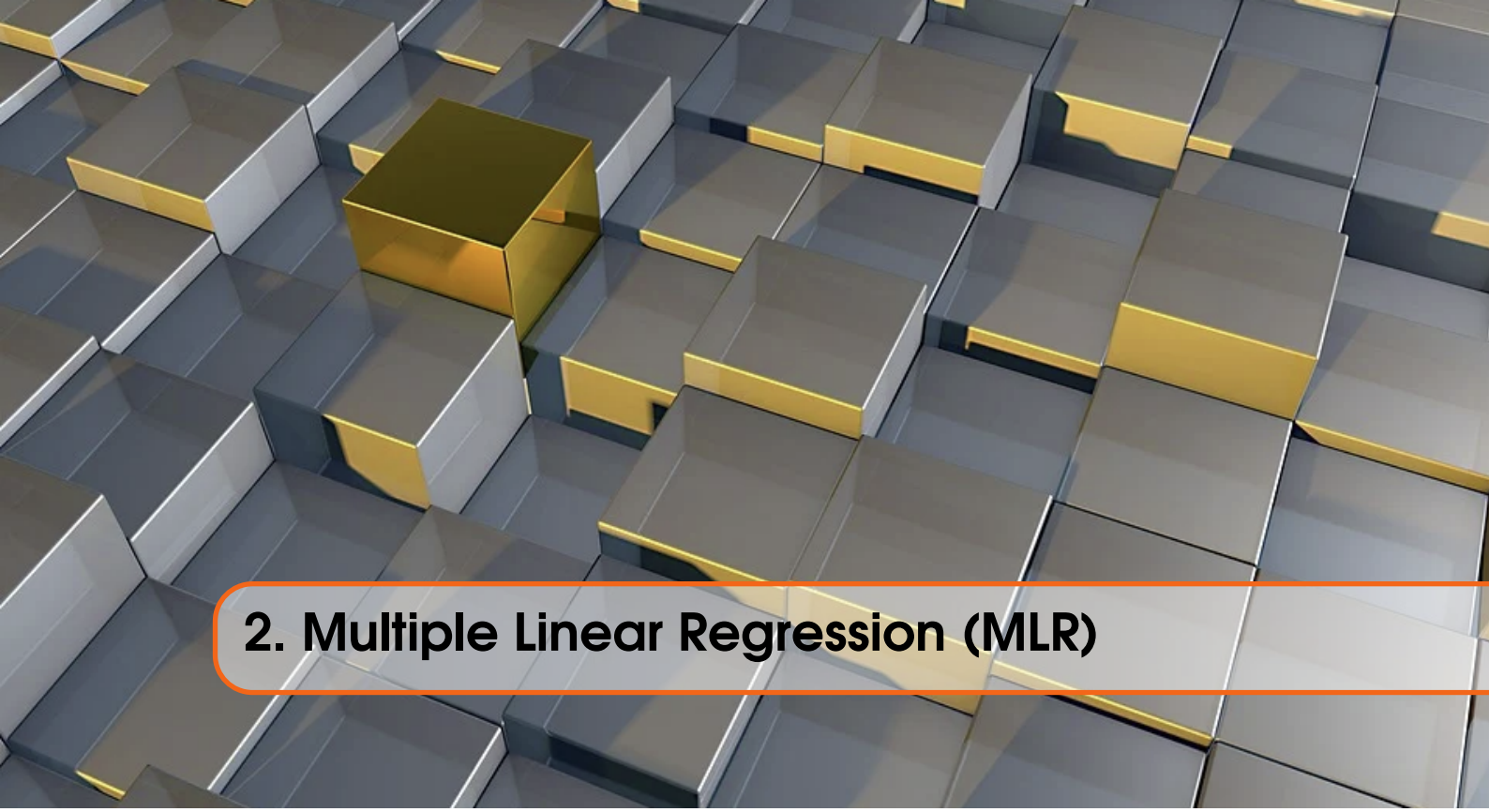
Figure 1.3.1: Whiteboard was almost destroyed

1.3.3 R Demo - Florange

```
%### STAT 331 simple linear regression demo

dat <- read.csv("florange.csv")
```

```
4
5  head(dat)
6       county boxes acres
7  1   Brevard    51   696
8  2 Charlotte   821 13447
9  3   Collier  2088 29351
10 4    DeSoto  7688 66365
11 5    Glades   368  5396
12 6    Hardee  5306 43126
13
14 # Scatterplot
15 plot(dat$acres,dat$boxes)
16
17 # Summary statistics calculation examples
18 r <- cor(dat$acres,dat$boxes)
19 xbar <- mean(dat$acres)
20 ybar <- mean(dat$boxes)
21 sd_x <- sd(dat$acres)
22 sd_y <- sd(dat$boxes)
23
24 # Manual calculation examples
25 Sxx <- sum( (dat$acres - xbar)^2 )
26 Sxy <- sum( (dat$acres - xbar) * (dat$boxes - ybar) )
27
28 # R's "lm" function fits linear models
29 lm.1 <- lm(dat$boxes~dat$acres)
30 summary(lm.1)
31
32 Call:
33 lm(formula = dat$boxes ~ dat$acres)
34
35 Residuals:
36      Min       1Q   Median       3Q      Max
37 -2470.81    -6.17    71.72   106.46  1677.32
38
39 Coefficients:
40              Estimate Std. Error t value Pr(>|t|)
41 (Intercept) -85.391989 186.178031  -0.459    0.651
42 dat$acres     0.116717   0.006761  17.263 1.16e-14 ***
43 ---
44 Signif. codes:
45 0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
46
47 Residual standard error: 754.4 on 23 degrees of freedom
48 Multiple R-squared:  0.9284,    Adjusted R-squared:  0.9252
49 F-statistic:   298 on 1 and 23 DF,  p-value: 1.164e-14
50
51 # Fitted values
52 lm.1$fitted.values
53
54          1           2           3           4           5           6
55  -4.157016 1484.100381 3340.366215 7660.526148  544.412490 4948.141734
56          7           8           9          10          11          12
57 7122.811315  -36.137451 6612.758392  250.869446  657.978049  529.005857
58         13          14          15          16          17          18
59  908.219117 1608.170463   12.066636  144.540335  234.879228   34.126133
60         19          20          21          22          23          24
61  656.227296  115.711257 7323.681128  864.683708    7.281242  -57.846797
62         25
63  -23.415307
64
```

```
# Residuals
lm.1$residuals

          1           2           3           4           5
  55.157016 -663.100381 -1252.366215   27.473852 -176.412490
          6           7           8           9          10
 357.858266 -2470.811315   75.137451 1230.241608   53.130554
         11          12          13          14          15
 162.021951   37.994143 -267.219117   -6.170463   56.933364
         16          17          18          19          20
 106.459665  -99.879228   73.873867  160.772704   89.288743
         21          22          23          24          25
1677.318872  546.316292   71.718758   79.846797   74.415307

# Manual calculation of sigma^2 estimate
sum(lm.1$residuals^2) / 23
# or sigma estimate
sqrt(sum(lm.1$residuals^2) / 23)

[1] 569124.3

# t distribution values
qt(0.975,23)
(1-pt(17.263,23))*2

[1] 754.4033

# Discussion
# - is sigma plausibly the same for all values of y? --> appears to be
    violated, can consider
# taking the log
# - are the error terms plausibly independent? (e.g., does knowing one e_i
    help predict
# e_j for a different county?)
```



2. Multiple Linear Regression (MLR)

I mean, come on, you really think this world is simple enough that your little SLR gonna solve all the problems? Wake up XD

2.1 Multiple Linear Regression

Definition 2.1.1 — Multiple Linear Regression Model (MLR). Consider p explanatory variables, develop linear relationship between response y and $x_1, \cdots, x_p$.
Now, our data of n observations are consisted of response and p explanatory variables, $(y_i, x_{i1}, \cdots, x_{ip})$.
Then, model:

$$Y_i \sim N(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}, \sigma^2)$$

where $\mathbb{E}(Y_i) = \mu_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$ or

$$Y_i = \mu_i + \varepsilon_i$$

where $\varepsilon_i \overset{\text{i.i.d}}{\sim} N(0, \sigma^2)$.

We can change the model into a vector/matrix format.

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} \beta_0 + \beta_1 x_{11} + \cdots + \beta_p x_{1p} \\ \beta_0 + \beta_1 x_{21} + \cdots + \beta_p x_{2p} \\ \vdots \\ \beta_0 + \beta_1 x_{n1} + \cdots + \beta_p x_{np} \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

which we can write as

$$\vec{Y} = X\vec{\beta} + \vec{\varepsilon}$$

where

$$\vec{Y} = \underbrace{\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}}_{n\times 1}, \quad X = \underbrace{\begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}}_{n\times(p+1)}, \quad \vec{\beta} = \underbrace{\begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}}_{(p+1)\times 1} \quad \vec{\varepsilon} = \underbrace{\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}}_{n\times 1}$$

2.2 Linear Algebra Preparation

Definition 2.2.1 — Random Vector. We call $\vec{Y} = (Y_1, \cdots, Y_n)^\top$ a random vector. Then, analogously, we have

1. Mean vector $n \times 1$

$$\mathbb{E}\left(\vec{Y}\right) = \begin{bmatrix} \mathbb{E}(Y_1) \\ \mathbb{E}(Y_2) \\ \vdots \\ \mathbb{E}(Y_n) \end{bmatrix}$$

2. Covariance matrix $n \times n$

$$\mathbf{Var}\left(\vec{Y}\right) = \begin{bmatrix} \mathbf{Var}(Y_1) & \mathbf{Cov}(Y_1, Y_2) & \cdots & \mathbf{Cov}(Y_1, Y_n) \\ \mathbf{Cov}(Y_2, Y_1) & \mathbf{Var}(Y_2) & \cdots & \mathbf{Cov}(Y_2, Y_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Cov}(Y_n, Y_1) & \mathbf{Cov}(Y_n, Y_2) & \cdots & \mathbf{Var}(Y_n) \end{bmatrix}$$

R Some apparent observations:

1. $\mathbf{Var}\left(\vec{Y}\right)$ is a symmetric matrix
2. From linear algebra, we know that the covariance matrix is Hermitian and positive semi-definite. i.e.
$$\vec{a}^\top \mathbf{Var}\left(\vec{Y}\right)\vec{a} \geq 0, \forall \vec{a} \in \mathbb{R}^n$$
3. Just like covariance original definition, we have
$$\mathbf{Var}\left(\vec{Y}\right) = \mathbb{E}\left[(\vec{Y} - \mathbb{E}(\vec{Y}))(\vec{Y} - \mathbb{E}(\vec{Y}))^\top\right]$$
this is usually proved in STAT330.

Proposition 2.2.1 — Properties of Random Vectors. Let $\vec{a}$ be a $1 \times n$ scalar vector and A is a $n \times n$ scalar matrix.

1. Mean vector
$$\mathbb{E}\left(\vec{a}\vec{Y}\right) = \vec{a}\mathbb{E}\left(\vec{Y}\right)$$
and
$$\mathbb{E}\left(A\vec{Y}\right) = A\mathbb{E}\left(\vec{Y}\right)$$
2. Covariance matrix
$$\mathbf{Var}\left(\vec{a}\vec{Y}\right) = \vec{a}\mathbf{Var}\left(\vec{Y}\right)\vec{a}^\top$$
and
$$\mathbf{Var}\left(A\vec{Y}\right) = A\mathbf{Var}\left(\vec{Y}\right)A^\top$$

Proof. First part on mean vector is quite immediate. Just multiply things out. We shall show

$$\mathbf{Var}\left(A\vec{Y}\right) = A\mathbf{Var}\left(\vec{Y}\right)A^\top$$

$$\begin{aligned}\mathbf{Var}\left(A\vec{Y}\right) &= \mathbb{E}\left[(A\vec{Y}-\mathbb{E}(A\vec{Y}))(A\vec{Y}-\mathbb{E}(A\vec{Y}))^\top\right]\\ &= \mathbb{E}\left[A(\vec{Y}-\mathbb{E}(\vec{Y}))(\vec{Y}-\mathbb{E}(\vec{Y}))^\top A^\top\right]\\ &= A\mathbb{E}\left[(\vec{Y}-\mathbb{E}(\vec{Y}))(\vec{Y}-\mathbb{E}(\vec{Y}))^\top\right]A^\top\\ &= A\mathbf{Var}\left(\vec{Y}\right)A^\top\end{aligned}$$

The punchline here is $(AB)^\top = B^\top A^\top$. ■

■ **Example 2.1 — Numerical Example - Sanity Check.** Consider a random vector $\vec{Y} = (Y_1, Y_2, Y_3)^\top$ and suppose

$$\mathbb{E}\left(\vec{Y}\right) = \begin{bmatrix}3\\1\\2\end{bmatrix}, \quad \mathbf{Var}\left(\vec{Y}\right) = \begin{bmatrix}4 & \frac{1}{2} & -2\\ \frac{1}{2} & 1 & 0\\ -2 & 0 & 3\end{bmatrix}, \quad \vec{a} = \begin{bmatrix}1 & -1 & 2\end{bmatrix}, \quad A = \begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{bmatrix}$$

Compute $\mathbb{E}(\vec{a}\vec{Y}), \mathbf{Var}\left(\vec{a}\vec{Y}\right), \mathbb{E}(A\vec{Y}), \mathbf{Var}\left(A\vec{Y}\right)$

1.
$$\mathbb{E}(\vec{a}\vec{Y}) = \vec{a}\mathbb{E}(\vec{Y}) = \begin{bmatrix}1 & -1 & 2\end{bmatrix}\begin{bmatrix}3\\1\\2\end{bmatrix} = 1(3)-1(1)+2(2) = 6$$

2.
$$\mathbf{Var}\left(\vec{a}\vec{Y}\right) = \vec{a}\mathbf{Var}\left(\vec{Y}\right)\vec{a}^\top = \begin{bmatrix}1 & -1 & 2\end{bmatrix}\begin{bmatrix}4 & \frac{1}{2} & -2\\ \frac{1}{2} & 1 & 0\\ -2 & 0 & 3\end{bmatrix}\begin{bmatrix}1\\-1\\2\end{bmatrix} = 8$$

■

Exercise 2.1 Finish the calculations in the example above. ■

2.3 Multivariate Normal Distribution (MVN)

Definition 2.3.1 — Multivariate Normal Distribution (MVN). We say that $\vec{Y} \sim \mathbf{MVN}(\vec{\mu}, \Sigma)$ (where $\vec{Y} = (Y_1, \cdots, Y_n)^\top$, $\vec{\mu}$ is the mean vector, and Σ is the corresponding covariance matrix) if it has the joint density function,

$$f(\vec{y};\vec{\mu},\Sigma) = \frac{1}{(2\pi)^{n/2}\underbrace{|\Sigma|^{1/2}}_{\text{determinant}}}\exp\left\{-\frac{1}{2}(\vec{y}-\vec{\mu})\underbrace{\Sigma^{-1}}_{\text{inverse}}(\vec{y}-\vec{\mu})^\top\right\}$$

R Compare to the 1D case wher we have

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

we can see they are really similar.

Proposition 2.3.1 — Properties of MVN. Let $\vec{Y} = (Y_1, \cdots, Y_n)^\top \sim \mathbf{MVN}(\vec{\mu}, \Sigma)$, $\vec{a}$ be a $1 \times n$ constant vector, and A is an $n \times n$ constant matrix.

1. **Linear transformations of MVN is still MVN:**
$$\vec{a}\vec{Y} \sim \mathbf{MVN}\left(\vec{a}\vec{\mu}, \vec{a}\Sigma\vec{a}^\top\right)$$
$$A\vec{Y} \sim \mathbf{MVN}\left(A\vec{\mu}, A\Sigma A^\top\right)$$
2. **Marginal distribution of Y_i is normal:** specifically,
$$Y_i \sim N(\mu_i, \Sigma_{ii})$$
where μ_i is the $i-th$ element in the mean vector and Σ_{ii} is the $i-$th entry on the diagonal of Σ. In fact, any subset of Y_i's is still MVN (lower dimension)
3. **Conditionals are also MVN:** If $\vec{y}$ and $\vec{x}$ are jointly MVN with $\Sigma_{yx} \neq 0$, then $\vec{Y}|\vec{X} = \vec{x}$ is a MVN.
4. **Uncorrelation implies independence:**
$$\underbrace{\mathbf{Cov}(Y_i, Y_j)}_{\text{uncorrolated}} \iff \underbrace{Y_i \perp\!\!\!\perp Y_j}_{independent}$$
In general, independence always implies uncorrelation. But normals are magical, the converse is true here. (proved in STAT330 or STAT240)

2.4 Inference for MLR

We can also write a vector/matrix for the random error vector as follow:

$$\vec{\varepsilon} \sim \mathbf{MVN}\left(\underbrace{\begin{bmatrix}0\\0\\\vdots\\0\end{bmatrix}}_{\vec{\mu}}, \underbrace{\begin{bmatrix}\sigma^2 & 0 & & \\ 0 & \sigma^2 & & \\ & & \ddots & \\ & & & \sigma^2\end{bmatrix}}_{\Sigma}\right)$$
$$\sim \mathbf{MVN}\left(\vec{0}, \sigma^2 I_{n\times n}\right)$$

from the property of MVN, we know that ε_i's are independent. Therefore, we have

$$\vec{Y} \sim \mathbf{MVN}\left(X\vec{\beta}, \sigma^2 I\right)$$

Again, to obtain $\vec{\beta}$, we shall use least square method.

Least Square Method:
Define the cost function S as

$$S(\beta_0, \beta_1, \cdots, \beta_p) = \sum_{y=1}^{n} (y_i - \underbrace{(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip})}_{\mathbb{E}(Y_i)=\mu_i})^2$$

we take derivatives w.r.t β_j,

$$\frac{\partial S}{\partial \beta_0} = \sum_{i=1}^{n} 2(y_i - \mu_i)(-1)$$
$$\frac{\partial S}{\partial \beta_j} = \sum_{i=1}^{n} 2(y_i - \mu_i)(-x_{ij}), j \in \{1, \cdots, p\}$$

Overall, we have $p+1$ equations set $\frac{\partial S}{\partial \beta_j} = 0, j \in \{0, \cdots, p\}$. First, note that

$$X = \underbrace{\begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}}_{n \times (p+1)} = \begin{bmatrix} | & | & \cdots & | \\ \vec{1} & \vec{x}_1 & \cdots & \vec{x}_p \\ | & | & \cdots & | \end{bmatrix}$$

Then,

$$\begin{cases} \sum_{i=1}^n (y_i - \mu_i) = 0 \\ \sum_{i=1}^n (y_i - \mu_i) x_{ij} = 0 \end{cases} \iff \begin{cases} \vec{1}^\top (\vec{y} - \vec{\mu}) = 0 \\ \vec{x}_j^\top (\vec{y} - \vec{\mu}) = 0 \end{cases}, j \in \{1, \cdots, p\}$$

Combining these $p+1$ equations, we have

$$X^\top (\vec{y} - \vec{\mu}) = 0$$

to give us the least-square solution for $\vec{\beta}$. Note that

$$\left\{ \begin{aligned} X^\top (\vec{y} - \vec{\mu}) = 0 &\iff X^\top (\vec{y} - X\vec{\beta}) = 0 \\ &\iff X^\top \vec{y} - X^\top X \vec{\mu} = 0 \\ &\iff X^\top X \vec{\beta} = X^\top \vec{y} \end{aligned} \right\} \implies \vec{\beta} = (X^\top X)^{-1} X^\top \vec{y}$$

so least-square estimate is given by $\hat{\vec{\beta}} = (X^\top X)^{-1} X^\top \vec{y}$ assuming $X^\top X$ is invertible (full rank of $p+1$ or say the column vectors of the matrix are linearly independent)

R **Why is it usually the case that $X^\top X$ is invertible?** If your explanatory variables are not linearly independent, then you can let other variables do the work by linear combinations. This is why we check the correlation of predictors later on.

2.4.1 Residuals in MLR

Definition 2.4.1 — Residuals. Define residuals to be $e_i = y_i - \underbrace{\left(\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_p x_{ip}\right)}_{\text{fitted value } \hat{\mu}_i}$.

Equivalently, we can write

$$\hat{\vec{\mu}} = X\hat{\vec{\beta}}, \vec{e} = \vec{y} - \hat{\vec{\mu}}$$

Estimate σ^2 based on e_i's

The result is

$$\hat{\sigma}^2 = \frac{\textbf{SS(Res)}}{n-(p+1)} = \frac{\sum_{i=1}^n e_i^2}{n-p-1} = \frac{\vec{e}^\top \vec{e}}{n-p-1}$$

and likewise, as a r.v., we have

$$\frac{(n-p-1)\hat{\sigma}^2}{\sigma^2} \sim \chi^2_{n-p-1}$$

2.4.2 Inference for $\hat{\vec{\beta}}$

From a r.v. perspective, we have

$$\hat{\vec{\beta}} = (\hat{\beta}_0, \cdots, \hat{\beta}_p)^\top = (X^\top X)^{-1} X^\top \vec{Y}$$

note that $\hat{\vec{\beta}}$ is a constant matrix times a random MVN vector (linear transformation of $\vec{Y}$). Since

$$\vec{Y} \sim \mathbf{MVN}\left(X\vec{\beta}, \sigma^2 I\right)$$

so,

1. mean vector

$$\mathbb{E}\left(\hat{\vec{\beta}}\right) = \mathbb{E}\left[(X^\top X)^{-1}X^\top \vec{Y}\right] = (X^\top X)^{-1}X^\top \mathbb{E}\left(\vec{Y}\right) = (X^\top X)^{-1}X^\top X\vec{\beta} = \vec{\beta}$$

   Again, this is an unbiased estimator component-wise.
2. covariance matrix

$$\begin{aligned}\mathbf{Var}\left(\hat{\vec{\beta}}\right) &= \mathbf{Var}\left((X^\top X)^{-1}X^\top \vec{Y}\right) \\ &= (X^\top X)^{-1}X^\top \mathbf{Var}\left(\vec{Y}\right)\left[(X^\top X)^{-1}X^\top\right]^\top \\ &= (X^\top X)^{-1}X^\top \sigma^2 I X(X^\top X)^{-1} \\ &= \sigma^2(X^\top X)^{-1}\end{aligned}$$

Thus,

$$\hat{\vec{\beta}} \sim \mathbf{MVN}\left(\vec{\beta}, \sigma^2 \underbrace{(X^\top X)^{-1}}_{V}\right)$$

Inference on β_j

From previous knowledge on MVN's marginals, we know that

$$\hat{\beta}_j \sim N(\beta_j, \sigma^2 V_{jj})$$

Then, we are back in our 1D business,

$$Z = \frac{\hat{\beta}_j - \beta_j}{\sigma\sqrt{V_{jj}}} \sim N(0,1), \qquad T = \frac{\hat{\beta}_j - \beta_j}{\hat{\sigma}\sqrt{V_{jj}}} \sim t_{n-p-1}$$

Theorem 4 — $(1-\alpha)$ CI for β_j.

$$\hat{\beta}_j \pm c\mathbf{SE}\left(\hat{\beta}_j\right)$$

where c is $1 - \frac{\alpha}{2}$ quantile of t_{n-p-1}.

Theorem 5 — Hypothesis test on $H_0 : \beta_j = 0, H_A : \beta_j \neq 0$. The t-statistics is

$$t = \frac{\hat{\beta}_j}{\hat{\sigma}\sqrt{V_{jj}}}$$

reject at level α if $|t| > c$. Corresponding p-value is

$$2\mathbb{P}(T \geq |t|), T \sim t_{n-p-1}$$

Interpretation of $\hat{\vec{\beta}}$

Fitted MLR model says $\widehat{\mathbb{E}(Y)} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p$, the estimate of expected response, $\hat{\beta}_0$ is the estimate of the expected response when all the explanatory variables are 0. And $\hat{\beta}_j$ is the estimated change in the expected response for a unit increase in x_j while holding other explanatory variables constant.

■ **Example 2.2 — The NASA Rocket.**

Cute Rocket Drawn by Prof.Wong

We are given the following summary statistics:

1. $n = 12$
2. $$\hat{\vec{\beta}} = \begin{bmatrix} 473.6 \\ 16.7 \\ -1.09 \end{bmatrix} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \end{bmatrix}$$
3. $$\hat{\sigma}^2 = \frac{\sum_{i=1}^{12} e_i^2}{12-2-1} = 2.655$$

Interpretation of $\hat{\vec{\beta}}$:

1. $\hat{\beta}_1$ is the estimated change in expected thrust is 16.7 when changing small to large nozzle, while holding other variables constant
2. $\hat{\beta}_2$ is the estimated thrust decrease by 1.09 on average for a unit increase in propellant ratio, while holding other variables constant.

Suppose further that $\mathbf{SE}\left(\hat{\beta}_2\right) = 0.94$. Then,

1. t-statistics for testing $H_0 : \beta_2 = 0, H_A : \beta_2 \neq 0$ is $t = \frac{-1.09}{0.94} = -1.16$.
2. Corresponding p-value is

$$2\mathbb{P}(T \geq 1.16) \underset{2*(1-pt(1.16,9))}{=} 0.275 > 0.05$$

Thus, do not reject H_0 at $\alpha = 0.05$ level. Propellant ratio does not significantly influence thrust.

■

We summarize everything we have seen so far a bit: $\vec{Y} = X\vec{\beta} + \vec{\varepsilon} \sim \mathbf{MVN}(X\vec{\beta}, \sigma^2 I)$ and

$$\underbrace{\hat{\vec{\beta}}}_{\text{estimate}} = (X^\top X)^{-1} X^\top \vec{Y}, \qquad \underbrace{\hat{\vec{\mu}}}_{\text{fitted value}} = X\hat{\vec{\beta}}, \qquad \underbrace{\vec{e}}_{\text{residuals}} = \vec{y} - \hat{\vec{\mu}}$$

Geometric Interpretation of Data

$$\underbrace{X}_{\text{constant}} = \underbrace{\begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}}_{n \times (p+1)} = \begin{bmatrix} | & | & \cdots & | \\ \vec{1} & \vec{x}_1 & \cdots & \vec{x}_p \\ | & | & \cdots & | \end{bmatrix}, \quad \underbrace{\vec{y}}_{\text{values of responses}} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n$$

Recall that $\text{span}(X) = \left\{ b_0\vec{1} + b_1\vec{x}_1 + \cdots + b_p\vec{x}_p : b_i \in \mathbb{R} \right\}$ which is a subspace of $\mathbb{R}^n$. By the assumption of MLR, $\text{rank}(X) = p + 1 \leq n$. Moreover, this span represents all possible vectors of

values $X\vec{b}$, $\vec{=}(b_0, b_1, \cdots, b_p)^\top$. Generally, $\vec{y} \in \text{span}(X)$, since the linear model is an approximation and $\vec{\varepsilon}$ represents the variability not explained by the model.

Intuitively, makes sense to choose an estimate $\hat{\vec{\beta}}$ so that $X\hat{\vec{\beta}}$ is as close to $\vec{y}$ as possible. How to see this from a picture?

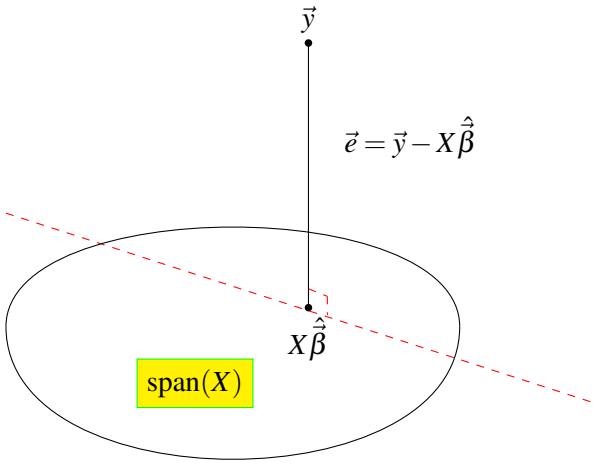


Figure 2.4.1: $\vec{e}$ is orthogonal to span(X)

This means $\vec{e}$ is orthogonal to columns of X:

$$\left.\begin{aligned} \vec{1}^\top\left(\vec{y} - \hat{\vec{\mu}}\right) &= 0 \\ \vec{x}_1^\top\left(\vec{y} - \hat{\vec{\mu}}\right) &= 0 \\ &\vdots \\ \vec{x}_p^\top\left(\vec{y} - \hat{\vec{\mu}}\right) &= 0 \end{aligned}\right\} \text{same as LS equations}$$

2.5 Prediction for MLR

Definition 2.5.1 — Hat Matrix. Define the hat matrix to be $H = X(X^\top X)^{-1}X^\top$.

Proposition 2.5.1 — Properties of H. 1. H is symmetric $H = H^\top$

Proof.

$$\left(X(X^\top X)^{-1}X^\top\right)^\top = X(X^\top X)^{-1}X^\top$$

■

2. H is idempotent $H^2 = H$

Proof.

$$(X(X^\top X)^{-1}X^\top)(X(X^\top X)^{-1}X^\top) = X(X^\top X)^{-1}(X^\top X)(X^\top X)^{-1}X^\top = H$$

■

3. $I - H$ is also symmetric and idempotent.

Now, let's view $\hat{\vec{\mu}}$ and $\vec{e}$ as random vectors. Then,

$$\hat{\vec{\mu}} = X\hat{\vec{\beta}} = \underbrace{X(X^\top X)^{-1}X^\top}_{=H}\vec{Y} = H\vec{Y}$$
$$\vec{e} = \vec{Y} - \hat{\vec{\mu}} = I\vec{Y} - H\vec{Y} = (I-H)\vec{Y}$$

then,

$$\mathbb{E}(\hat{\vec{\mu}}) = \mathbb{E}(H\vec{Y}) = H\mathbb{E}(\vec{Y}) = X(X^\top X)^{-1}X^\top \underbrace{X\vec{\beta}}_{\mathbb{E}(\vec{Y})} = X\vec{\beta}$$

$$\mathbf{Var}\left(\hat{\vec{\mu}}\right) = \mathbf{Var}\left(H\vec{Y}\right) = H\mathbf{Var}\left(\vec{Y}\right)H^\top = H\sigma^2 I H^\top = \sigma^2 H$$

meanwhile,

$$\mathbb{E}(\vec{e}) = \mathbb{E}(\vec{Y}) - \mathbb{E}(H\vec{Y}) = X\vec{\beta} - X\vec{\beta} = 0$$
$$\mathbf{Var}(\vec{e}) = (I-H)\mathbf{Var}(Y)(I-H)^\top = \sigma^2(I-H)$$

Distributions of $\hat{\vec{\mu}}, \vec{e}$

Since $\hat{\vec{\mu}}$ and $\vec{e}$ are linear transformations of $\vec{Y}$.

$$\hat{\vec{\mu}} \sim \mathbf{MVN}\left(X\vec{\beta}, \sigma^2 H\right), \quad \vec{e} \sim \mathbf{MVN}\left(0, \sigma^2(I-H)\right)$$

Prediction

Suppose we want to predict response for

$$\vec{x}_0 = \left[\underbrace{1}_{\text{for intercept}} \quad x_{01} \quad x_{02} \quad \cdots \quad x_{0p}\right] \qquad 1 \times (p+1)$$

Let Y_0 representing response associated with $\vec{x}_0$. By MLR, we have

$$Y_0 \sim N\left(\beta_0 + \beta_1 x_{01} + \cdots + \beta_p x_{0p}, \sigma^2\right)$$

so we predict the value $\hat{y}_0 = \underbrace{\hat{\beta}_0 + \hat{\beta}_1 x_{01} + \cdots \hat{\beta}_p x_{0p}}_{\text{estimated mean res given } x_{oi}} = \vec{x}_0\hat{\vec{\beta}}$. And corresponding distribution has

$$\mathbb{E}\left(\hat{Y}_0\right) = \vec{x}_0\mathbb{E}(\hat{\vec{\beta}}) = \vec{x}_0\vec{\beta}$$

and

$$\mathbf{Var}\left(\hat{Y}_0\right) = \vec{x}_0\mathbf{Var}\left(\hat{\vec{\beta}}\right)\vec{x}_0^\top = \vec{x}_0\sigma^2\left(X^\top X\right)^{-1}\vec{x}_0^\top$$

thus,

$$\hat{Y}_0 \sim N\left(\vec{x}_0\vec{\beta}, \vec{x}_0\sigma^2\left(X^\top X\right)^{-1}\vec{x}_0^\top\right)$$

Then, similar to previous results, we have

$$\frac{\hat{Y}_0 - \vec{x}_0\vec{\beta}}{\sigma\sqrt{\vec{x}_0(X^\top X)^{-1}\vec{x}_0^\top}} \sim N(0,1) \qquad \frac{\hat{Y}_0 - \vec{x}_0\vec{\beta}}{\hat{\sigma}\sqrt{\vec{x}_0(X^\top X)^{-1}\vec{x}_0^\top}} \sim t_{n-(p+1)}$$

Theorem 6 — $(1-\alpha)$ CI for the mean response given $\vec{x}_0$.

$$\hat{y}_0 \pm c\hat{\sigma}\sqrt{\vec{x}_0(X^\top X)^{-1}\vec{x}_0^\top}$$

where c is $1-\frac{\alpha}{2}$ quantile of t_{n-p-1}.

Definition 2.5.2 — Prediction Error. We assume that Y_0 and $\hat{Y}_0$ are independent since $\hat{Y}_0$ is a function of $Y_1, \cdots, Y_n$. Then the prediction error is $Y_0 - \hat{Y}_0$ which is a normal random variable.

1.
$$\mathbb{E}(Y_0 - \hat{Y}_0) = \vec{x}_0\vec{\beta} - \vec{x}_0\vec{\beta} = 0$$

2.
$$\mathbf{Var}\left(Y_0 - \hat{Y}_0\right) = \sigma^2 + \sigma^2\left(\vec{x}_0\left(X^\top X\right)^{-1}\vec{x}_0^\top\right)$$

so,

$$Y_0 - \hat{Y}_0 \sim N\left(0, \sigma^2\left(1 + \vec{x}_0\left(X^\top X\right)^{-1}\vec{x}_0^\top\right)\right)$$

Theorem 7 — $(1-\alpha)$ PI for y_0.

$$\hat{y}_0 \pm c\hat{\sigma}\sqrt{1 + \vec{x}_0(X^\top X)^{-1}\vec{x}_0^\top}$$

where c is $1-\frac{\alpha}{2}$ quantile of t_{n-p-1}.

R The intuition about PI being wider than the CI for mean is that estimating average is "easier" than individual response.

2.5.1 R Demo - Rocket

```
%## NASA rocket data example

## From: R.S. Jankovsky, T.D. Smith, A.J. Pavli (1999). "High-Area-Ratio Rocket
## Nozzle at High Combustion Chamber Pressure-Experimental and Analytical
## Validation".

# setwd(...) first if your CSV file is somewhere else
rocket <- read.csv(file="rocket.csv")
rocket

# Scatter plots
par(mfrow = c(1,2))
plot(rocket$nozzle, rocket$thrust, ylab="Thrust", xlab="Nozzle size (1 = large)")
plot(rocket$propratio, rocket$thrust, ylab="Thrust", xlab="Propellant to fuel ratio")
```

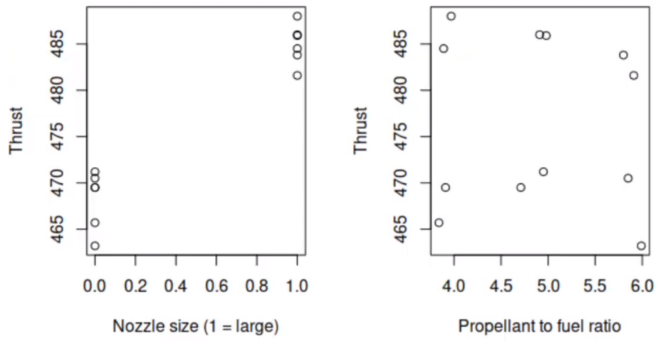


Figure 2.5.1: Output Scatter Plot

```
%# Fit MLR using lm
m1 <- lm(thrust ~ nozzle + propratio, data = rocket)
summary(m1)
```

```
%Call:
lm(formula = thrust ~ nozzle + propratio, data = rocket)

Residuals:
    Min      1Q  Median      3Q     Max
-3.8459 -1.7555  0.5934  1.2906  3.3008

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 473.6039     4.7158 100.430 4.88e-15
nozzle       16.7383     1.5329  10.919 1.71e-06
propratio    -1.0948     0.9414  -1.163    0.275

(Intercept) ***
nozzle      ***
propratio
---
Signif. codes:
0 *** 0.001 ** 0.01 * 0.05 . 0.1  1

Residual standard error: 2.655 on 9 degrees of freedom
Multiple R-squared:  0.9303,    Adjusted R-squared:  0.9148
F-statistic: 60.05 on 2 and 9 DF,  p-value: 6.238e-06
```

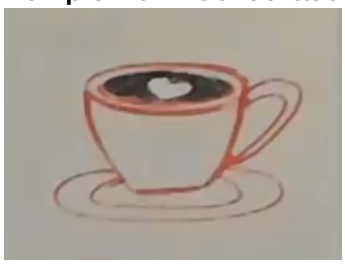
```
%# Manual beta estimates
X <- cbind(rep(1, 12), rocket$nozzle, rocket$propratio) # X matrix
y <- matrix(rocket$thrust, ncol = 1) # response vector
beta_hat <- solve(t(X) %*% X) %*% t(X) %*% y
## t() is transpose and solve() is inverse
beta_hat
           [,1]
[1,] 473.603924
[2,]  16.738319
[3,]  -1.094822

# Manual sigma estimate
mu_hat <- X %*% beta_hat # fitted values
e <- y - mu_hat # residuals
sigma_hat <- sqrt((t(e) %*% e) / 9) # Note n-p-1 = 12-2-1 = 9
sigma_hat
       [,1]
```

```
[1,] 2.6545

sigma_hat <- sqrt( sum(e^2) / 9) # equivalent
sigma_hat
[1] 2.6545

# Covariance matrix of beta_hat
vcov(m1)
            (Intercept)      nozzle   propratio
(Intercept)   22.238325 -1.02316688 -4.32080608
nozzle        -1.023167  2.34987593 -0.03102117
propratio     -4.320806 -0.03102117  0.88631920

sqrt(diag(vcov(m1))) # SEs of individual betas
(Intercept)      nozzle   propratio
  4.7157528   1.5329305   0.9414453

# Manual
se_beta <- sigma_hat * sqrt(diag(solve(t(X) %*% X)))
se_beta
[1] 4.7157528 1.5329305 0.9414453

# Estimate the mean response for units with small nozzle and propellant
    ratio 5.5
# include a 95% CI
predict(object = m1, newdata = data.frame(nozzle = 0, propratio = 5.5),
        interval = "confidence", level = 0.95)
       fit      lwr      upr
1 467.5824 464.7929 470.3719

# Manual calculation
x0 <- matrix(c(1, 0, 5.5), nrow = 1)
y0_hat <- x0 %*% beta_hat
se_mu0 <- sigma_hat * sqrt(x0 %*% solve(t(X) %*% X) %*% t(x0))
crit_val <- qt(0.975,9)
ci_lo <- y0_hat - crit_val*se_mu0
ci_hi <- y0_hat + crit_val*se_mu0
c(y0_hat, ci_lo, ci_hi)
[1] 467.5824 464.7929 470.3719

# Predict the value of the response for a unit with small nozzle and
    propellant ratio 5.5
# include a 95% PI
predict(object = m1, newdata = data.frame(nozzle = 0, propratio = 5.5),
        interval = "prediction", level = 0.95)
       fit      lwr      upr
1 467.5824 460.9612 474.2036

# Manual calculation
x0 <- matrix(c(1, 0, 5.5), nrow = 1)
y0_hat <- x0 %*% beta_hat
se_y0 <- sigma_hat * sqrt(1+ x0 %*% solve(t(X) %*% X) %*% t(x0))
crit_val <- qt(0.975,9)
pi_lo <- y0_hat - crit_val*se_y0
pi_hi <- y0_hat + crit_val*se_y0
c(y0_hat, pi_lo, pi_hi)
[1] 467.5824 460.9612 474.2036
```

2.6 Categorical Predicators in MLR

So far, our statisitical inferences have been conducted on numerical explanatory variates. What about and categorical explanatory variates? Consider a research on the probability of a person having COVID-19 and one of the explanatory variate is coughing symptom. In this case, we can code it as 1 and 0 in a binary fashion. Sometimes the categorical variate is ordinal, with an implicit order. For example, the size of the houses can be categorized into small, median, and large sizes. What we can do is to convert the them into indicator variables or treat them as numericals if it makes sense to do so.

■ **Example 2.3 — Coffee Quality Institute (2018).**



Cute Cappuccino Drawn by Prof.Wong

We are given the following variables:

	Acidity	Method	⋯	Flavour
1	8.7	Washed-wet		⋯
2	8.3	Washed-wet		⋯
3	8.2	Natural-dry		⋯
4	8.4	Semi-washed/puped		⋯
⋮	⋮	⋮		⋮

How to set up X?

Naive Try:

We can try

$$x_{i2} = \begin{cases} 0 & \text{dry} \\ 1 & \text{semi} \\ 2 & \text{wet} \end{cases}$$

but a clear drawback here is that we are imposing an ordering or size on a not-ordered categorical. This is not generally appropriate unless we think the response is linear according to this scheme.

Indicator/Dummy Variables Try:

We introduce a more flexible approach. In fact, this can be related to the notion of one-hot encoding of categorical data in the realm of machine learning.

$$x_{i2} = \begin{cases} 1 & \text{semi} \\ 0 & \text{otherwise} \end{cases}, \quad x_{i3} = \begin{cases} 1 & \text{wet} \\ 0 & \text{otherwise} \end{cases}, \cdots$$

we can look at the resulting X,

$$X = \begin{bmatrix} 1 & 8.7 & 0 & 1 & \cdots \\ 1 & 8.3 & 0 & 1 & \cdots \\ 1 & 8.2 & 0 & 0 & \cdots \\ 1 & 8.4 & 1 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

R **Why don't we define x_{i4} for "dry"?**

Suppose that we do,

$$x_{i4} = \begin{cases} 1 & \text{dry} \\ 0 & \text{otherwise} \end{cases}$$

and the resulting X is,

$$X = \begin{bmatrix} 1 & 8.7 & 0 & 1 & 0 & \cdots \\ 1 & 8.3 & 0 & 1 & 0 & \cdots \\ 1 & 8.2 & 0 & 0 & 1 & \cdots \\ 1 & 8.4 & 1 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \end{matrix} \\ \vec{1} \quad \vec{x}_1 \quad \vec{x}_2 \quad \vec{x}_3 \quad \vec{x}_4$$

But now we have linearly dependent column vectors, namely $\vec{x}_4 = \vec{1} - \vec{x}_2 - \vec{x}_3$. Linear independence is important for us to conduct MLR. In other words, there is no new information by defining x_{i4} for the dryness.

Suppose for now, that we restrict ourselves with these explanatory variates. We can model this MLR as

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \varepsilon_i$$

Interpretations:

1. Mean flavour if acidity is x_{01} and method is dry is $\beta_0 + \beta_1 x_{01}$.
2. Mean flavour if acidity is x_{01} and method is wet is $\beta_0 + \beta_1 x_{01} + \beta_3$.
3. Mean flavour if acidity is x_{01} and method is semi is $\beta_0 + \beta_1 x_{01} + \beta_2$.
4. β_2 represents the difference between semi and dry in the expected response while holding other variables constant.
5. β_3 represents the difference between wet and dry in the expected response while holding other variables constant.
6. $\beta_2 - \beta_3$ represents the difference between semi and wet in the expected response while holding other variables constant.

From prior knowledge on MLR, we have

$$\hat{\vec{\beta}} \sim \mathbf{MVN}\left(\vec{\beta}, \sigma^2 V\right)$$

where $V = (X^\top X)^{-1}$. We know that $\hat{\beta}_j \sim N(\beta_j, \sigma^2 V_{jj})$ and $\mathbf{SE}\left(\hat{\beta}_j\right) = \hat{\sigma}\sqrt{V_{jj}}$ We have no problem constructing CI for β_2, β_3, what about $\beta_2 - \beta_3$? Well, let's explore:

$$\begin{aligned} \mathbf{Var}\left(\hat{\beta}_2 - \hat{\beta}_3\right) &= \mathbf{Var}\left(\hat{\beta}_2\right) + \mathbf{Var}\left(\hat{\beta}_3\right) - 2\mathbf{Cov}\left(\hat{\beta}_2, \hat{\beta}_3\right) \\ &= \sigma^2 V_{22} + \sigma^2 V_{33} - 2\sigma^2 V_{23} \end{aligned}$$

thus, $\mathbf{SE}\left(\hat{\beta}_2 - \hat{\beta}_3\right) = \hat{\sigma}\sqrt{V_{22} + V_{33} - 2V_{23}}$. Now, we can construct CI for $\beta_2 - \beta_3$. ∎

Let's generalize the observation that we have seen in the example above.

Theorem 8 For k categorical explanatory variables, we need $k - 1$ indicator variables to encode.

R Okay, now, we are comfortable with all types of explanatory variables. Numerical, categorical, ordinal, complex, etc, you name it. What if our response variable is bound to be categorical like, you gonna be perfect in STAT331 or nah.

Figure 2.6.1: Clair de Lune, C. Debussy

2.7 Analysis of Variance (ANOVA)

Everyboby asks what is the model, but nobody asks how is the model.

—Model Depression Association

Essentially, we want to know how well does regression model fit response variable.

Theorem 9 — ANOVA Decomposition.

$$\begin{aligned}\underbrace{\textbf{SS(Total)}}_{\text{Total sum of squares}} &= \sum_{i=1}^{n}(y_i-\bar{y})^2 \\ &= \underbrace{\textbf{SS(Reg)}}_{\text{Regression sum of squares}} + \underbrace{\textbf{SS(Res)}}_{\text{residual sum of squares}} \\ &= \underbrace{\sum_{i=1}^{n}(\hat{\mu}_i-\bar{y})^2}_{SS(Reg)} + \underbrace{\sum_{i=1}^{n}(y_i-\hat{\mu}_i)^2}_{SS(Res)}\end{aligned}$$

R

1. Recall from Assignment 1, the **SS(Total)**is closely related to sample variance of $\{y_1,\cdots,y_n\}$, which is $\frac{\textbf{SS(Total)}}{n-1}$.
2. **SS(Reg)** represents the variation explained by the model
3. **SS(Res)** represents the variable **not** explained by the model
4. we proved this result using the fact that

$$y_i-\bar{y}=y_i-\hat{\mu}_i+\hat{\mu}_i-\bar{y}$$

Visualization (We are going to draw a picture. *2000 years later*)

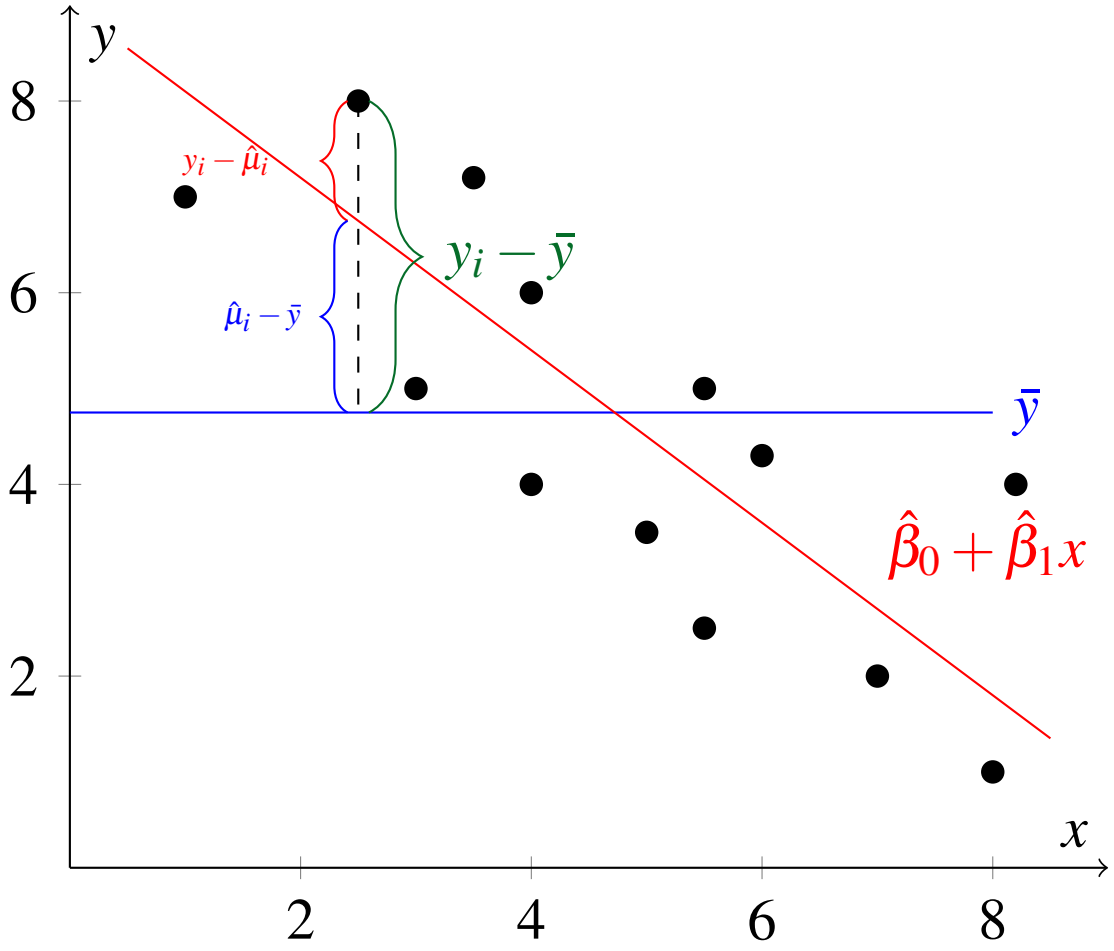


Figure 2.7.1: SLR Revisit

$$y_i - \bar{y} = \underbrace{y_i - \hat{\mu}_i}_{\text{difference between the observation and the fitted}} + \underbrace{\hat{\mu}_i - \bar{y}}_{\text{difference between the fitted and the average}}$$

R When regression fits data well, y_i tends to be closer to $\hat{\mu}_i$. $\bar{y}$ in the picture is like the regression line with $\beta_1 = 0$.

MLR Version

$$\begin{aligned}\sum_{i=1}^{n}(\hat{\mu}_i - \bar{y})(y_i - \hat{\mu}_i) &= \sum_{i=1}^{n}(\hat{\mu}_i - \bar{y})e_i \\ &= \sum_{i=1}^{n} e_i\hat{\mu}_i - \bar{y}\sum_{i=1}^{n} e_i \\ &= \hat{\mu}^\top \vec{e} - \bar{y}\vec{1}^\top \vec{e}\end{aligned}$$

Recall that $\vec{1}^\top \vec{e} = 0$ is one of the LS equations. Moreover, $\hat{\vec{\mu}} = X\hat{\vec{\beta}}$ is in span(X). $\vec{e}$ is orthogonal to span(X), so $\hat{\vec{\mu}}^\top \vec{e} = 0$. Thus,

$$\sum_{i=1}^{n} (\hat{\mu}_i - \bar{y})(y_i - \hat{\mu}_i) = 0$$

this, again, leads to the ANOVA decomposition.

2.7.1 Summarize in ANOVA Table

Source	**df**	**SS**	Mean Square	F-Statistic
Regression	p	**SS(Reg)**	**SS(Reg)**$/p$	**MS(Reg)/MS(Res)**
Residual	$n-p-1$	**SS(Res)**	**SS(Res)**$/(n-p-1) = \hat{\sigma}^2$	
Total	$n-1$	**SS(Total)**		

Table 2.7.1: ANOVA Table

R

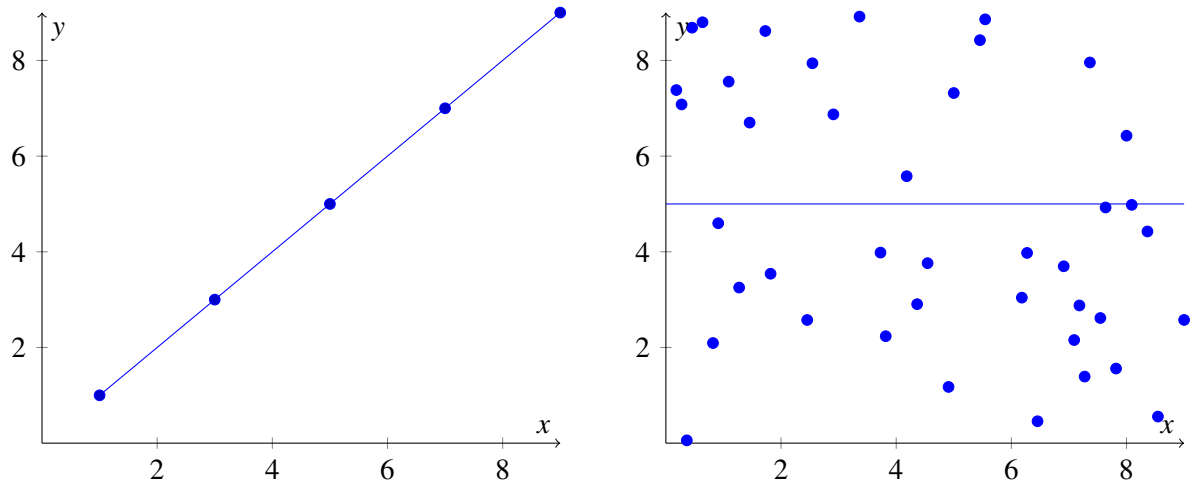
1. Note that both df and SS do add up to totals.
2. Mean square takes corresponding SS divided by its df.
3. The F-statistic will be used to test, for example, overall significance of regression (later)

Definition 2.7.1 — Coefficient of Determination.

$$R^2 = \frac{\textbf{SS(Reg)}}{\textbf{SS(Total)}} = 1 - \frac{\textbf{SS(Res)}}{\textbf{SS(Total)}}$$

clearly, $0 \le R^2 \le 1$.

R^2 is the proportion of variation in the response variable that is explained by the regression model. Larger R^2 means fitting values are closer to observations y_i, smaller the **SS(Res)**.
Furthermore, recall from Assignment 1, in SLR, R^2 is equivalent to the square of sample correlation between x and y based on $(x_1, y_1), \cdots, (x_n, y_n)$. In SLR,



(a) $R^2 = 1$ (b) $R^2 = 0$

Figure 2.7.2: Pictures for SLR are easy to draw

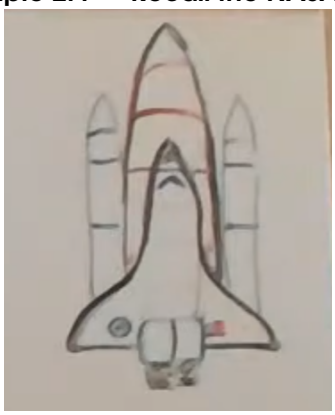
1. For $R^2 = 1$, we will have

$$\textbf{SS(Res)} = 0, \qquad e_i = 0, \qquad \hat{\mu}_i = y_i$$

2. For $R^2 = 0$, we will have

$$\textbf{SS(Reg)} = 0, \qquad \hat{\mu}_i = \bar{y}, \qquad \hat{\beta}_0 = \bar{y}, \qquad \hat{\beta}_1 = 0$$

■ **Example 2.4 — Recall the NASA Rocket.**



Cute Rocket Drawn by Prof.Wong

Given that we have 2 predictors, nozzle size and propellant ratio, and 12 observations, we have the following ANOVA table:

Source	df	SS	MS	F
Reg	2	846.2	423.1	60
Res	9	63.42	$\hat{\sigma}^2 = 7.05$	
Tot	11	909.62		

Table 2.7.2: ANOVA Table

According to this ANOVA table, we see that, for the response thrust,

$$R^2 = \frac{846.2}{909.62} \approx 0.93$$

The regression model with nozzle size and propellant ratio explains 93% of variation in thrust response. ■

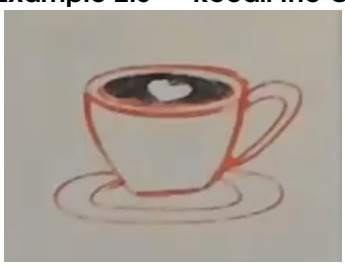
2.8 Hypothesis Testing Based on F-Distribution

So far, we have been able to test $H_0 : \beta_j = 0$ vs. $H_A : \beta_j \neq 0$ involving individual parameters using the t-distribution. Now, we consider HT of the form

$$H_0 : A\vec{\beta} = \vec{0}$$

where A is the **matrix of constraints** specifying linear combinations of parameters. This gives us more flexibility in terms of HT possibilities.

■ **Example 2.5 — Recall the Coffee Quality Institute (2018).**



Cute Cappuccino Drawn by Prof.Wong

We are given the following variables:

	Acidity	Method	$\cdots$	Flavour
1	8.7	Washed-wet		$\cdots$
2	8.3	Washed-wet		$\cdots$
3	8.2	Natural-dry		$\cdots$
4	8.4	Semi-washed/puped		$\cdots$
$\vdots$	$\vdots$	$\vdots$		$\vdots$

We define the flavour response, Y_i, using

$$Y_i = \beta_0 + \beta_1 \underbrace{x_{i1}}_{\text{acidity}} + \beta_2 \underbrace{x_{i2}}_{\begin{cases}1 & \text{if semi}\\ 0 & \text{otherwise}\end{cases}} + \beta_3 \underbrace{x_{i3}}_{\begin{cases}1 & \text{if wet}\\ 0 & \text{otherwise}\end{cases}} + \varepsilon_i$$

we call this the **"full model"**. We have some **"reduced"** models by setting A:

1. $H_0 : \beta_1 = \beta_2 = \beta_3 = 0$ vs H_A : at least one of $\beta_1, \beta_2, \beta_0$ not 0: If H_0 is true, the model reduced to

$$Y_i = \beta_0 + \varepsilon_i$$

This HT tests the **overall significance of regression**, whether any of the predictors impact response. What is our A here?

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

here we have 3 constraints.

2. $H_0 : \beta_2 = \beta_3 = 0$ vs H_A : at least one of β_2, β_3 is not 0: If H_0 is true, then the model is reduced to

$$Y_i = \beta_0 + \beta_1 x_{i1} + \varepsilon_i$$

this HT is testing whether the reduced model with only acidity is plausible. The A in this case is

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

here we have 2 constraints.

3. $H_0 : \beta_2 - \beta_3 = 0$ vs. $H_A : \beta_2 \neq \beta_3$: if H_0 is true, the reduced model is

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 \underbrace{(x_{i2} + x_{i3})}_{\begin{cases} 1 & \text{if semi/wet} \\ 0 & \text{otherwise} \end{cases}} + \varepsilon_i$$

this HT is testing whether wet/semi methods have the same impact on the response holding acidity constant. The A in this case is

$$A = \begin{bmatrix} 0 & 0 & 1 & -1 \end{bmatrix}$$

here we have 1 constraint.

■

In general, with l constraints, A is a $l \times (p+1)$ matrix with rank l. Recall that span(X) = $\left\{\beta_0 \vec{1} + \beta_1 \vec{x}_1 + \cdots \beta_p \vec{x}_p\right\}$. This can be considered as the unconstrained subspace (the columnn space of X). The constraints imposed by A that we have seen so far can be encoded as

$$\text{span}_A(X) := \left\{\beta_0 \vec{1} + \beta_1 \vec{x}_1 + \cdots \beta_p \vec{x}_p : A\vec{\beta} = 0\right\}$$

which is a subspace of span(X).

Definition 2.8.1 — Span(X) with constraint A on $\vec{\beta}$.

$$\text{span}_A(X) := \left\{\beta_0 \vec{1} + \beta_1 \vec{x}_1 + \cdots \beta_p \vec{x}_p : A\vec{\beta} = 0\right\}$$

Visualization of span$_A(X)$

Let $\hat{\vec{\mu}}_A$ denote the fitted values from fitting the reduced model.

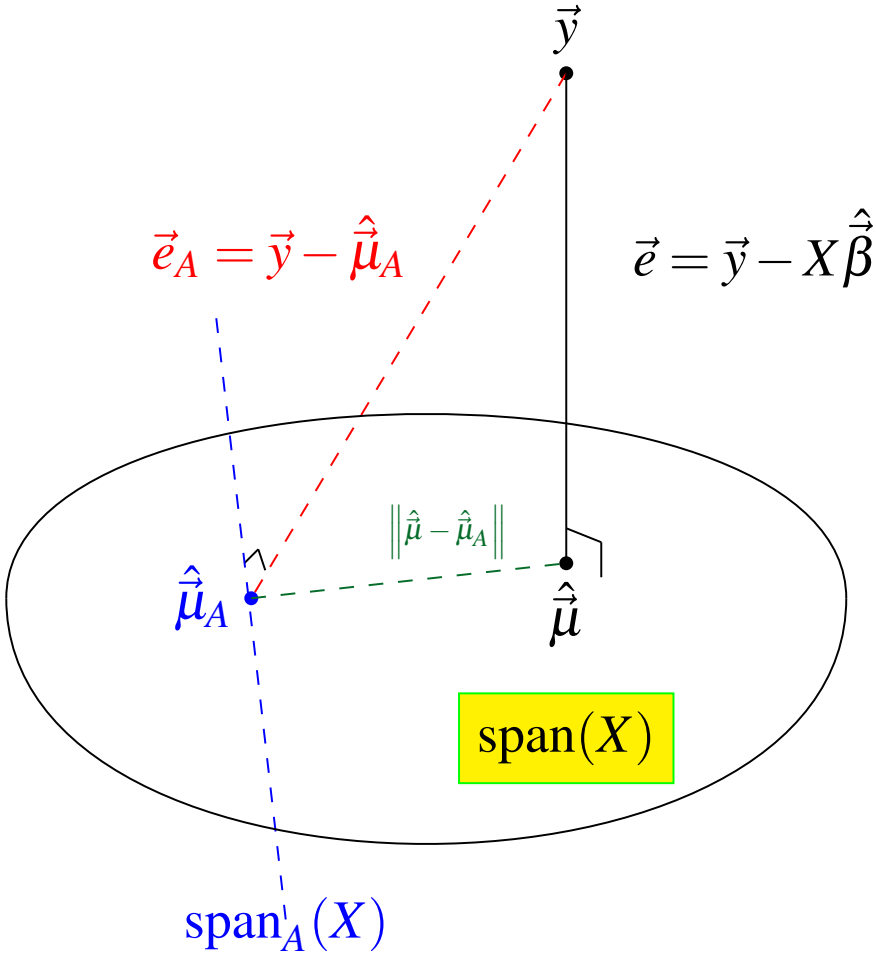


Figure 2.8.1: $\vec{e}_A$ is orthogonal to $\text{span}_A(X)$

as we can see in the picture, $\vec{e}_A$ is the residual if we fit model with $A\vec{\beta} = 0$. If $H_0 : A\vec{\beta} = \vec{0}$ is true, then $\hat{\vec{\mu}}$ and $\hat{\vec{\mu}}_A$ should be close, which means model makes similar predictions whether we set $A\vec{\beta} = \vec{0}$ or not when fitting the model. If H_0 is plausible, look at the vector different $\left\|\hat{\vec{\mu}} - \hat{\vec{\mu}}_A\right\|$.

Definition 2.8.2 — Euclidean Norm (L_2 Norm).

$$\left\|\hat{\vec{\mu}} - \hat{\vec{\mu}}_A\right\| = \sqrt{\left(\hat{\vec{\mu}} - \hat{\vec{\mu}}_A\right)^\top \left(\hat{\vec{\mu}} - \hat{\vec{\mu}}_A\right)}$$

If this norm is "large", it is evidence against H_0. If it is "small" close to 0, then it is evidence for the H_0.

By Pythagoras,

$$\left\|\vec{y} - \hat{\vec{\mu}}_A\right\|^2 = \left\|\vec{y} - \hat{\vec{\mu}}\right\|^2 + \left\|\hat{\vec{\mu}} - \hat{\vec{\mu}}_A\right\|^2$$

equivalently,

$$\|\vec{e}_A\|^2 = \|\vec{e}\|^2 + \left\|\hat{\vec{\mu}} - \hat{\vec{\mu}}_A\right\|^2$$

Normal people: *use proof by contradiction to show $\sqrt{2}$ is irrational*

Pythagoras, casually murdering Hippasus:

moreover, we can write $\|\vec{e}_A\|^2 = \vec{e}_A^\top \vec{e}_A$ and $\|\vec{e}\|^2 = \vec{e}^\top \vec{e}$. If you look at these expression very closely for a couple of seconds. Okay, you have done that. We can see that $\vec{e}^\top \vec{e}$ is the sum of squared residuals. Let us denote $\textbf{SS(Res)}_A = \vec{e}_A^\top \vec{e}_A$ and $\textbf{SS(Res)} = \vec{e}^\top \vec{e}$. Finally, we have

$$\left\|\hat{\vec{\mu}} - \hat{\vec{\mu}}_A\right\|^2 = \textbf{SS(Res)}_A - \textbf{SS(Res)} \geq 0$$

, which is the **additional sum of squares explained by full model vs. reduced one with constraints** A.

Practical Implications

1. **SS(Res)** cannot decrease when constraints applied.
2. Equivalently, the full model always has a smaller (or equal) **SS(Res)** for a fixed **SS(Total)** and thus higher R^2 compared to a reduced model.

This thing really satisfies everything that we need for a **discrepancy measure/test statistic** as soon as we make it a random variable.

We use this intuition to define the following test statistic:

Definition 2.8.3 — F-Statistic.

$$F = \frac{(\textbf{SS(Res)}_A - \textbf{SS(Res)})/l}{\textbf{SS(Res)}/(n-p-1)} = \frac{(\textbf{SS(Res)}_A - \textbf{SS(Res)})/l}{\hat{\sigma}^2}$$

where $\hat{\sigma}^2$ is for the full model and A is a constraint matrix of rank l.

It has a corresponding distribution.

Definition 2.8.4 — F-Distribution. If $U \sim \chi^2_a$ and $V \sim \chi^2_b$ are independent, then

$$\frac{U/a}{V/b} \sim F_{a,b}$$

In this context, we have the following facts when H_0 is true:

1. We have seen that (in MLR)

$$V = \frac{\hat{\sigma}^2(n-p-1)}{\sigma^2} \sim \chi^2_{n-p-1}$$

2. Moreover,

$$U = \frac{\left\|\hat{\vec{\mu}} - \hat{\vec{\mu}}_A\right\|^2}{\sigma^2} \sim \chi^2_l$$

3. U, V are independent when H_0 is true.

Therefore,

$$F = \frac{\frac{\left\|\hat{\vec{\mu}} - \hat{\vec{\mu}}_A\right\|^2}{\sigma^2} \cdot \frac{1}{l}}{\frac{\hat{\sigma}^2(n-p-1)}{\sigma^2} \cdot \frac{1}{n-p-1}} \sim F_{l,n-p-1}$$

when H_0 is true.

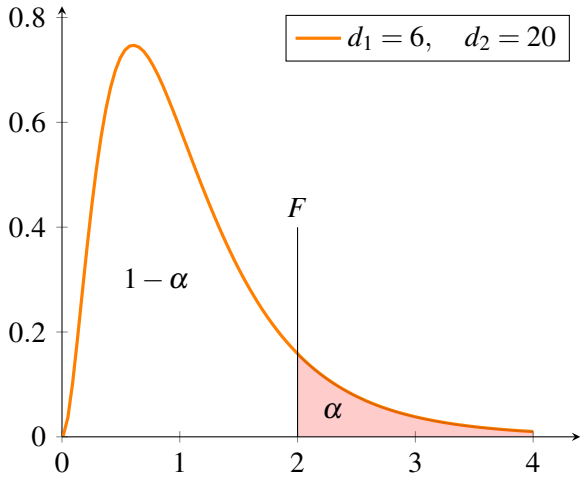


Figure 2.8.2: $F_{a,b}$ Distribution PDF

Theorem 10 — F-Test. We reject $H_0 : A\vec{\beta} = 0$ at level α if

$$F > (1-\alpha) \text{ quantile of } F_{l,n-p-1}$$

and p-value is $\mathbb{P}(Y \geq F)$ where $Y \sim F_{l,n-p-1}$.

Relation to t-test

Say $Y \sim t_a$, then we can write

$$Y = \frac{Z}{\sqrt{U/a}}$$

where $Z \sim N(0,1)$ and $U \sim \chi^2_a$ independently. Note that

$$Y^2 = \frac{Z^2/1}{U/a} \sim F_{1,a}$$

where $Z^2 \sim \chi^2_1$ by definition. Thus, if HT has only 1 constraint, then F-test is equivalent to t-test of the same hypothesis.

Special Case: Overall Test of Significance

Are any predictors related to response?

where $H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0$ vs. H_A : at least one of $\beta_i \neq 0$. The constraint matrix is

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} = \begin{bmatrix} \vec{0} & I_{p\times p} \end{bmatrix}$$

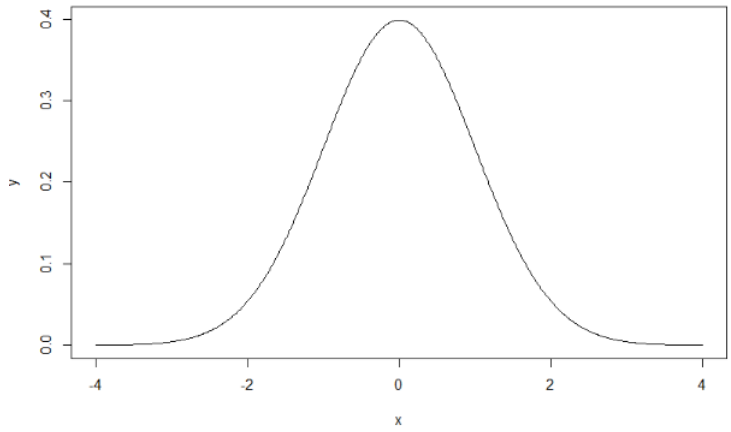
If H_0 is true, then $Y_i = \beta_0 + \varepsilon_i, Y_i \sim N(\beta_0, \sigma^2)$. We fit the reduced model as follow, i.e. estimating β_0 using least square, $\min \sum_{i=1}^n (y_i - \beta_0)^2$. It can be shown that $\hat{\beta}_0 = \bar{y}$. Thus,

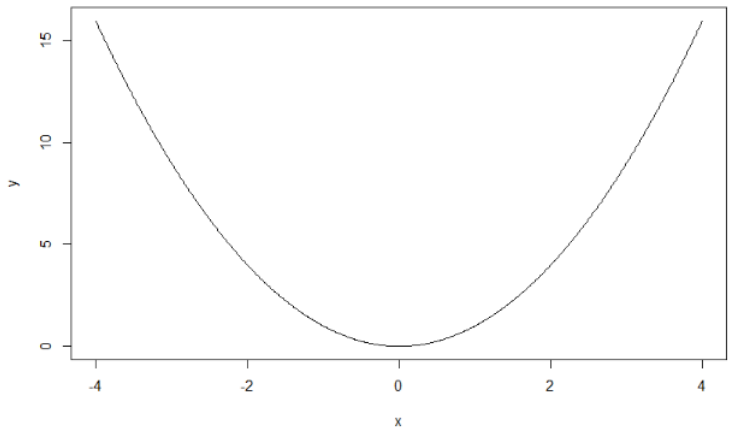
$$\textbf{SS(Res)}_A = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - \hat{\mu}_i)^2 = \sum_{i=1}^n (y_i - \bar{y})^2 = \textbf{SS(Total)}$$

Then, $F = \frac{(\textbf{SS(Total)} - \textbf{SS(Res)})/p}{\textbf{SS(Res)}/(n-p-1)} = \frac{\textbf{SS(Reg)}/p}{\textbf{SS(Res)}/(n-p-1)} = \frac{\textbf{MS(Reg)}}{\textbf{MS(Res)}} \sim F_{p,n-p-1}$. This is exactly the F-statistics shown in the ANOVA table.

2.8.1 R Demo - ANOVA F-Test

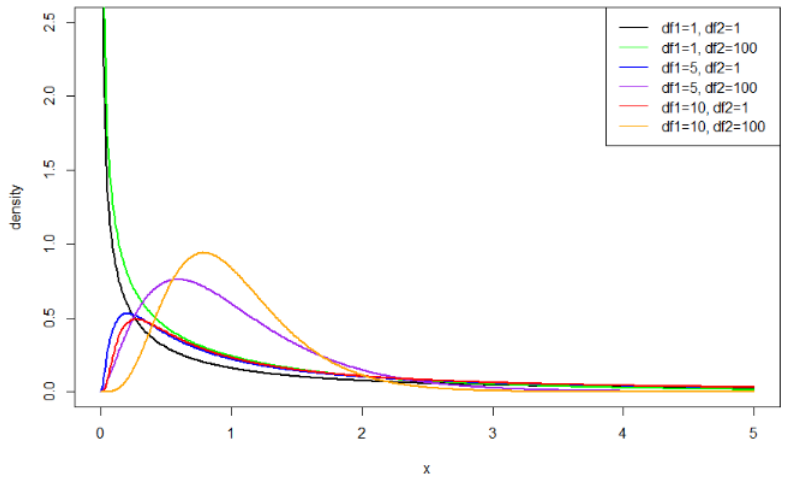
```
%## R demo for Oct 19
## Plotting functions and histograms, F distribution,
## ANOVA tables, F tests, MLR with categorical variables

# Plotting functions (e.g., probability density functions)
x <- seq(-4,4,0.01)  # grid of x values to evaluate
y <- dnorm(x,0,1)    # function that evaluates normal PDF with mean 0 and
    SD 1
plot(x,y, type="l")

y <- x^2
plot(x,y, type="l")
```



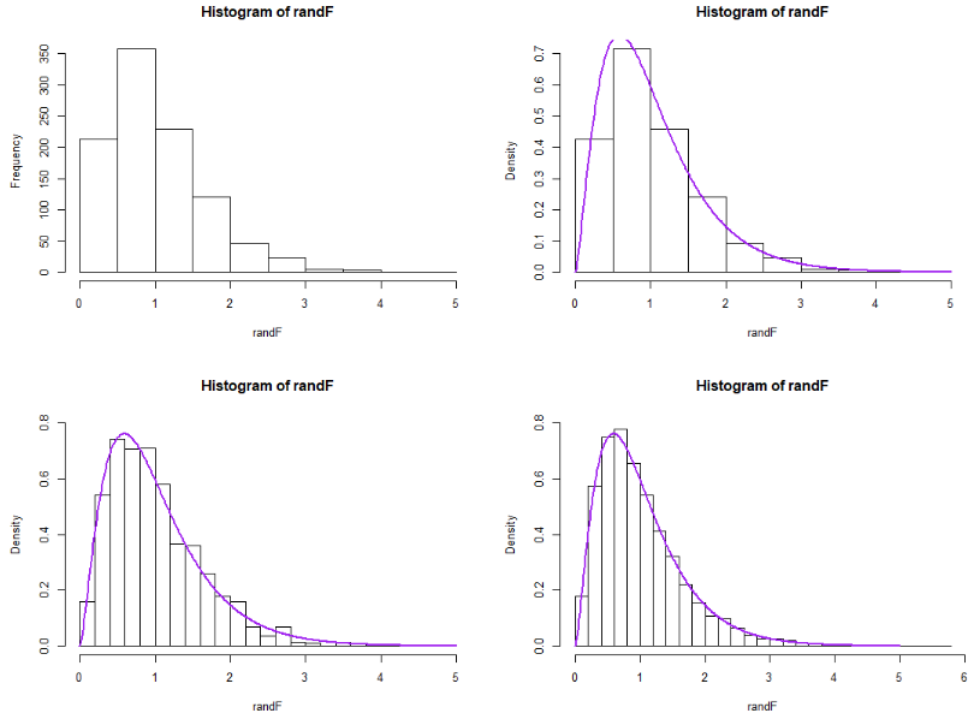



```
%# F distribution examples
x <- seq(0,5,0.01)
plot(x, y = df(x, df1 = 1, df2 = 1), type = "l", xlab = "x", ylab = "
    density")
plot(x, y = df(x, df1 = 1, df2 = 1), type = "l", col = "black", xlab = "x",
     ylab = "density", ylim = c(0,2.5), lwd=2)
lines(x, y = df(x, df1 = 1, df2 = 100), type = "l", col = "green", lwd=2)
lines(x, y = df(x, df1 = 5, df2 = 1), type = "l", col = "blue", lwd=2)
lines(x, y = df(x, df1 = 5, df2 = 100), type = "l", col = "purple", lwd=2)
```

```
lines(x, y = df(x, df1 = 10, df2 = 1), type = "l", col = "red", lwd=2)
lines(x, y = df(x, df1 = 10, df2 = 100), type = "l", col = "orange", lwd=2)
legend("topright", legend = c("df1=1, df2=1","df1=1, df2=100","df1=5, df2=1
    ",
                              "df1=5, df2=100","df1=10, df2=1","df1=10, df2=100
    "),
       lty = 1, col = c("black", "green", "blue", "purple", "red", "orange"
    ))
```




```
%# Random numbers from F distribution
set.seed(12345678)
randF <- rf(1000, 5, 100)
hist(randF)
hist(randF, freq=FALSE)
# add true density
lines(x, y = df(x, df1 = 5, df2 = 100), type = "l", col = "purple", lwd=2)
# set y-axis limits and more detailed histogram bins
hist(randF, freq=FALSE, ylim=c(0,0.8), breaks=25)
lines(x, y = df(x, df1 = 5, df2 = 100), type = "l", col = "purple", lwd=2)

randF <- rf(10000, 5, 100)
hist(randF, freq=FALSE, ylim=c(0,0.8), breaks=25)
lines(x, y = df(x, df1 = 5, df2 = 100), type = "l", col = "purple", lwd=2)
```




```
%## Revisit rocket example
rocket <- read.csv(file="rocket.csv")
m1 <- lm(thrust ~ nozzle + propratio, data = rocket)
summary(m1)

Call:
lm(formula = thrust ~ nozzle + propratio, data = rocket)

Residuals:
    Min      1Q  Median      3Q     Max
-3.8459 -1.7555  0.5934  1.2906  3.3008

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 473.6039     4.7158 100.430 4.88e-15 ***
nozzle       16.7383     1.5329  10.919 1.71e-06 ***
propratio    -1.0948     0.9414  -1.163    0.275
---
Signif. codes:
0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 2.655 on 9 degrees of freedom
Multiple R-squared:  0.9303,    Adjusted R-squared:  0.9148
F-statistic: 60.05 on 2 and 9 DF,  p-value: 6.238e-06

anova(m1)  # compare with ANOVA table on board Oct 5

Analysis of Variance Table

Response: thrust
           Df Sum Sq Mean Sq  F value    Pr(>F)
nozzle      1 836.67  836.67 118.7377 1.743e-06 ***
propratio   1   9.53    9.53   1.3524    0.2748
Residuals   9  63.42    7.05
```

```
---
Signif. codes:
0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

anova(m1)$`Sum Sq`

[1] 836.670000   9.529332  63.417335

sum(anova(m1)$`Sum Sq`[1:2])

[1] 846.1993

SSRes <- anova(m1)$`Sum Sq`[3]

# Test of overall significance
m_red <- lm(thrust ~ 1, data = rocket)
summary(m_red)

Call:
lm(formula = thrust ~ 1, data = rocket)

Residuals:
     Min       1Q   Median
-13.4167  -7.1167  -0.2167
      3Q      Max
  8.2333  11.3833

Coefficients:
            Estimate
(Intercept)  476.617
            Std. Error
(Intercept)      2.625
            t value Pr(>|t|)
(Intercept)   181.6   <2e-16

(Intercept) ***
---
Signif. codes:
  0 *** 0.001 ** 0.01
  * 0.05 . 0.1   1

Residual standard error: 9.094 on 11 degrees of freedom

anova(m_red)

Analysis of Variance Table

Response: thrust
          Df Sum Sq Mean Sq
Residuals 11 909.62  82.692
          F value Pr(>F)
Residuals

SSRes_A <- anova(m_red)$`Sum Sq`[1]

# F-statistic
l <- 2
n <- nrow(rocket)
p <- 2
Fstat <- ((SSRes_A - SSRes)/l) / (SSRes / (n-p-1))
pval <- 1 - pf(Fstat, df1 = l, df2 = n-p-1)
```

```r


## Coffee example  (Coffee Quality Institute, 2018)
coffee <- read.csv("coffee_arabica.csv")
head(coffee)

mfull <- lm(Flavor~ factor(Processing.Method) + Aroma + Aftertaste +
    Body + Acidity + Balance + Sweetness + Uniformity + Moisture, dat=
    coffee)
summary(mfull)
anova(mfull)
SSRes <- anova(mfull)$`Sum Sq`[10]

## Reduced model without Uniformity and Moisture (beta9=beta10=0):
m_red <- lm(Flavor~ factor(Processing.Method) + Aroma + Aftertaste +
              Body + Acidity + Balance + Sweetness, dat=coffee)
summary(m_red)
anova(m_red)
SSRes_A <- anova(m_red)$`Sum Sq`[8]

## F-statistic
l <- 2
n <- nrow(coffee)
p <- 10
Fstat <- ((SSRes_A - SSRes)/l) / (SSRes / (n-p-1))
pval <- 1 - pf(Fstat, df1 = l, df2 = n-p-1)

## Reduced model without Uniformity and Moisture and
## setting effect of Dry = Semi (beta1=beta9=beta10=0)
coffee$method2 <- ifelse(coffee$Processing.Method %in%
                 c('Natural / Dry', 'Semi-washed / Semi-pulped'), 0, 1) # 1
    = wet, 0 otherwise
coffee$wet <- ifelse(coffee$Processing.Method == 'Washed / Wet', 0, 1) # 1
    = semi/dry, 0 otherwise

m_red2 <- lm(Flavor~ method2 + Aroma + Aftertaste +
               Body + Acidity + Balance + Sweetness, dat=coffee)
summary(m_red2)
anova(m_red2)
SSRes_A <- anova(m_red2)$`Sum Sq`[8]

## F-statistic
l <- 3
n <- nrow(coffee)
p <- 10
Fstat <- ((SSRes_A - SSRes)/l) / (SSRes / (n-p-1))
pval <- 1 - pf(Fstat, df1 = l, df2 = n-p-1)
```

2.9 Multicollinearity in Regression

Definition 2.9.1 — Multicollinearity. Occurs when some explanatory variables have a strong linear relationship amongst themselves.

For example, this might occur exactly

$$\vec{x}_3 = \alpha_0\vec{1} + \alpha_1\vec{x}_1 + \alpha_2\vec{x}_2$$

in which case the columns of X would be linearly dependent and $X^\top X$ does not have an inverse. Practically, there is no new information including $\vec{x}_3$ when $\vec{x}_1, \vec{x}_2$ are in the model. Sometimes, approximately,

$$\vec{x}_3 \approx \alpha_0\vec{1} + \alpha_1\vec{x}_1 + \alpha_2\vec{x}_2$$

in which case the columns of X are close to be linearly dependent. This causes the $\mathbf{Var}\left(\hat{\beta}_j\right)$ terms to be inflated. In turn, this leads to inaccurate CIs and conclusions of HTs for the parameters. In practice, $\mathbf{SE}\left(\vec{\beta}_j\right)$ when fitting models can change drastically when adding/removing variables from model.

■ **Example 2.6 — Hockey (Very Canadian) - Exact Multicollinearity.** In NHL, we have

$$\textbf{Goals} + \textbf{Assists} = \textbf{Points}$$

we want to predict a forward's salary. If we let

$$\begin{aligned} x_1 &= \textbf{Goals} \\ x_2 &= \textbf{Assists} \\ x_3 &= \textbf{Points} \\ &\vdots \end{aligned}$$

then, we have an **exact multicollinearity**. ■

■ **Example 2.7 — Python in Florida - Approximate Multicollinearity.**

Overized Python Drawn by Prof.Wong

Python in my computer

The response variable is the fat content.

1. x_1 =mass
2. x_2 =overall length
3. x_3 =snout-to-vent length

Conceptually, x_2 and x_3 are highly correlated since both of them are measuring the lengths. If we include all variables in MLR. This will lead to inflated $\mathbf{SE}\left(\hat{\beta}_2\right)$ and $\mathbf{SE}\left(\hat{\beta}_3\right)$. ■

How to detect multicollinearity?

1. If two predictors are related:
   (a) Pairwise scatterplot/scatterplot matrix: all possible pairs of scaterplots between $y, x_1, x_2, \cdots, x_p$
   (b) Correlation matrix: all pairwise correlations
2. In general, for multicollinearity among > 2 predictors:

Definition 2.9.2 — Variance Inflation Factor (VIF). For each predictors $x_1, \cdots, x_p$, we have $j \in \{1, \cdots, p\}$,

$$\mathbf{VIF}_j = \frac{\mathbf{Var}\left(\hat{\beta}_j\right)}{\mathbf{Var}\left(\hat{\beta}_j^*\right)}$$

where $\hat{\beta}_j$ is the estimate of β_j with all predictors in the model and $\hat{\beta}_j^*$ is the estimate of β_j based on only x_j in the regression.

Theorem 11

$$\mathbf{VIF}_j \geq 1$$

Proof. Leave as an exercise for the readers. Margin too small to fit. ■

This can be viewed in terms of R^2: fit the MLR of x_j in terms of other predictors.

$$X_{ij} = \alpha_0 + \alpha_1 x_{i1} + \cdots + \alpha_{j-1} x_{i,j-1} + \alpha_{j+1} x_{i,j+1} + \cdots + \alpha_p x_p$$

and computes R^2 for this model, call it R_j^2. The intuition is that, if R_j^2 is close to 1, x_j is strongly related linearly to other predictors. It can be shown that

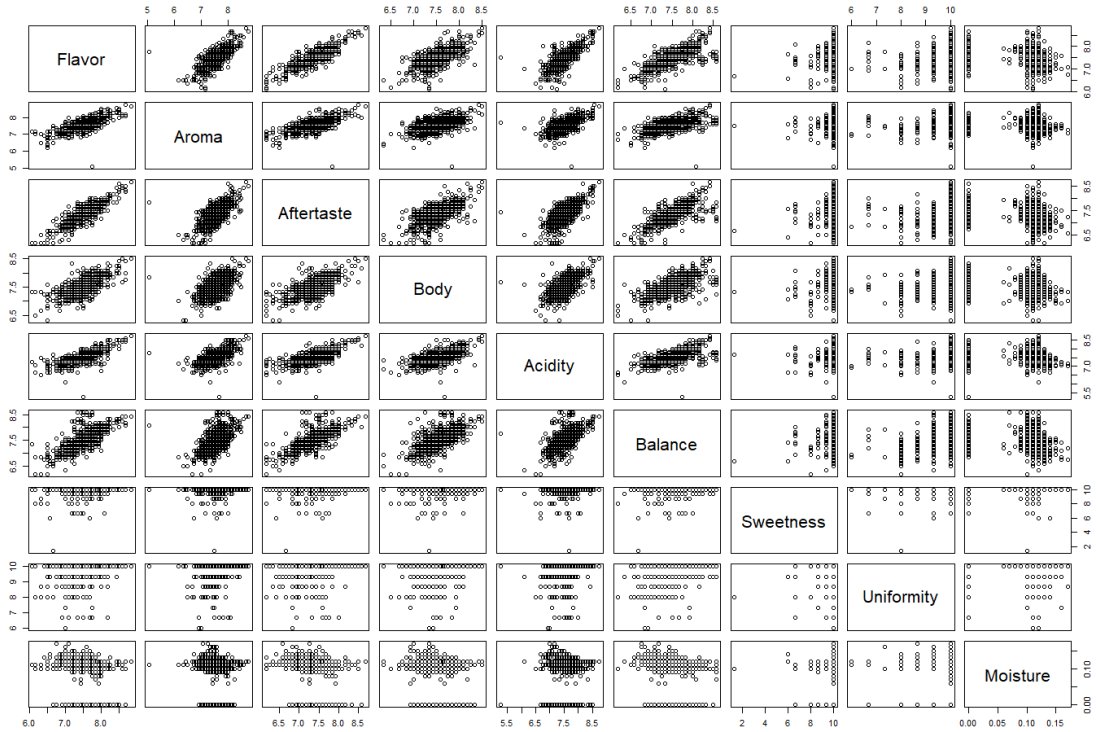
$$\mathbf{VIF}_j = \frac{1}{1 - R_j^2}$$

This gives a proof of Theorem 11 but not in detail. **Rule of Thumb:** if $\mathbf{VIF}_j \geq 10 \iff R_j^2 \geq 0.9$, we have serious multicollinearity. The procedure is:

(a) Remove predictors with largest **VIF**, if it exceeds 10
(b) Repeat until no more multicollinearity.

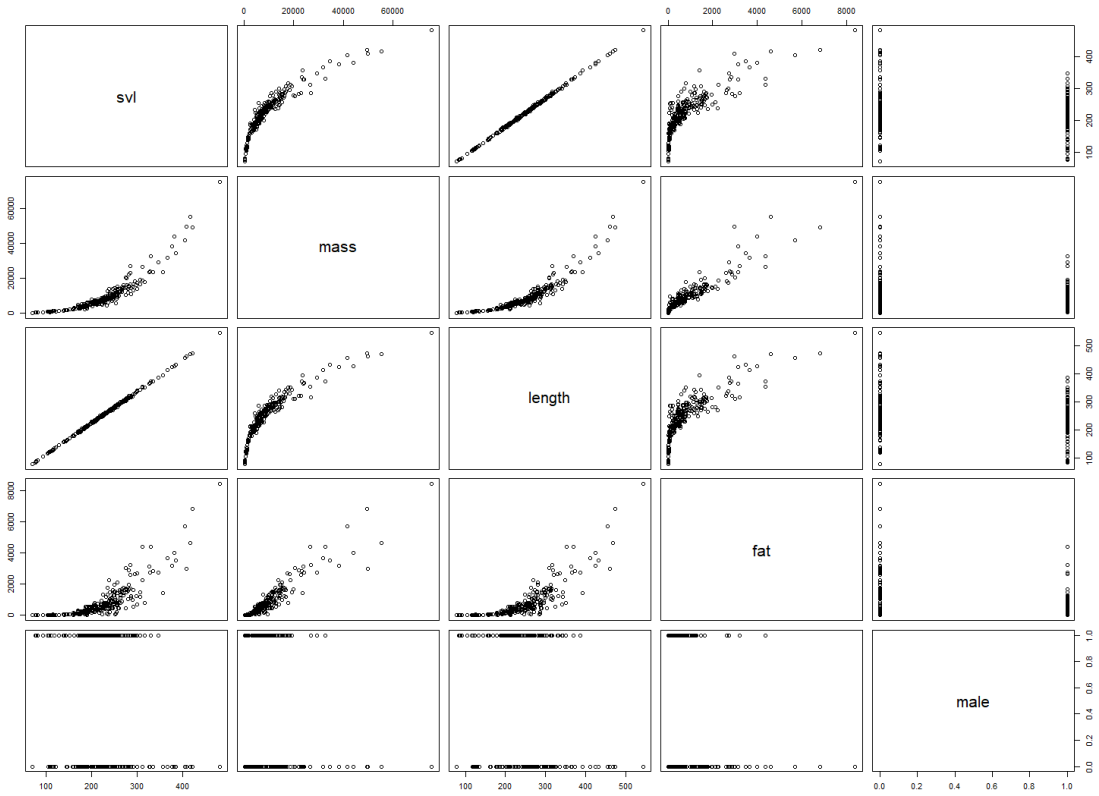
2.9.1 R Demo - Multicollinearity

```
%## Coffee example  (Coffee Quality Institute, 2018) continued
coffee <- read.csv("coffee_arabica.csv")

cor(coffee) # doesn't work as there's a categorical variable
cor(coffee[,-1]) # e.g., remove first column
pairs(coffee[,-1])
pairs(~ Flavor + Aroma + Aftertaste + Body +
        Acidity + Balance + Sweetness + Uniformity + Moisture, data=coffee)
```




```
%# Code our own indicators, so that we can more easily interpret VIFs
coffee$wet <- ifelse(coffee$Processing.Method == 'Washed / Wet', 1, 0) # 1 = wet, 0 otherwise
coffee$semi <- ifelse(coffee$Processing.Method == 'Semi-washed / Semi-pulped', 1, 0) # 1 = semi/dry, 0 otherwise

mfull <- lm(Flavor~ wet + semi + Aroma + Aftertaste +
      Body + Acidity + Balance + Sweetness + Uniformity + Moisture, dat=coffee)

wet_reg <- lm(wet ~ semi + Aroma + Aftertaste +
                Body + Acidity + Balance + Sweetness + Uniformity + Moisture, dat=coffee)
r2_wet <- summary(wet_reg)$r.squared
VIF_wet <- 1 / (1 - r2_wet)

Aroma_reg <- lm(Aroma ~ wet + semi + Aftertaste +
      Body + Acidity + Balance + Sweetness + Uniformity + Moisture, dat=coffee)
r2_Aroma <- summary(Aroma_reg)$r.squared
VIF_Aroma <- 1 / (1 - r2_Aroma)

Aftertaste_reg <- lm(Aftertaste ~ wet + semi + Aroma +
       Body + Acidity + Balance + Sweetness + Uniformity + Moisture, dat=coffee)
r2_Aftertaste <- summary(Aftertaste_reg)$r.squared
VIF_Aftertaste <- 1 / (1 - r2_Aftertaste)

library(car)
# install.packages("car")  if you don't have the library
vif(mfull) # vif function in the "car" library

       wet       semi      Aroma Aftertaste
```

```
  1.236212   1.178004   2.085382   3.449479
      Body    Acidity    Balance  Sweetness
  2.317728   2.232210   3.002813   1.159602
Uniformity   Moisture
  1.209901   1.086101

## Python in FL everglades example (2017)
## Sex, length, total mass, fat mass, and specimen condition data for
## 248 Burmese pythons (Python bivittatus) collected in the Florida
    Everglades

python <- read.csv("FLpython.csv")
head(python)

  sex  svl mass length    fat
1   F 70.0  186   77.5  6.000
2   M 76.0  310   83.8 11.000
3   M 77.0  260   86.1  6.000
4   M 78.0  262   87.1  8.000
5   M 81.0  306   91.1  4.000
6   M 93.5  605  104.6 18.959

python$male <- ifelse(python$sex == 'M', 1, 0) # 1 = M, 0 =F

cor(python[,-1])

               svl       mass     length        fat       male
svl      1.0000000  0.8843022  0.9994935  0.8098652 -0.1602418
mass     0.8843022  1.0000000  0.8858256  0.9419114 -0.2190993
length   0.9994935  0.8858256  1.0000000  0.8114658 -0.1593512
fat      0.8098652  0.9419114  0.8114658  1.0000000 -0.2933111
male    -0.1602418 -0.2190993 -0.1593512 -0.2933111  1.0000000

pairs(python[,-1])
```




```
%mpf <- lm(fat ~ male+svl+mass+length, data = python)
summary(mpf)

Call:
lm(formula = fat ~ male + svl + mass + length, data = python)

Residuals:
     Min       1Q   Median       3Q      Max
-2445.77  -137.41    -5.29   110.00  1527.27

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.021e+02  1.331e+02   1.518    0.130
male        -1.971e+02  4.732e+01  -4.165 4.32e-05 ***
svl         -3.370e+00  1.125e+01  -0.300    0.765
mass         1.178e-01  5.302e-03  22.210  < 2e-16 ***
length       1.594e+00  1.010e+01   0.158    0.875
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 360.9 on 243 degrees of freedom
Multiple R-squared:  0.897,     Adjusted R-squared:  0.8953
F-statistic:   529 on 4 and 243 DF,  p-value: < 2.2e-16

vif(mpf)

       male         svl        mass      length
   1.058699  994.546545    4.813078 1007.484200

# remove "length" based on VIF
mpf2 <- lm(fat ~ male+mass+svl, data = python)
```

```
summary(mpf2)

Call:
lm(formula = fat ~ male + mass + svl, data = python)

Residuals:
     Min       1Q   Median       3Q      Max
-2444.44  -137.38    -6.66   109.22  1530.81

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  204.09840  132.30121   1.543   0.1242
male        -196.71705   47.16396  -4.171 4.22e-05 ***
mass           0.11788    0.00524  22.495  < 2e-16 ***
svl           -1.59841    0.76433  -2.091   0.0375 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 360.2 on 244 degrees of freedom
Multiple R-squared:  0.897,     Adjusted R-squared:  0.8957
F-statistic: 708.2 on 3 and 244 DF,  p-value: < 2.2e-16

vif(mpf2)

    male     mass      svl
1.056139 4.720065 4.611903
```

2.10 Model Selection

Given p explanatory variables, we want to find the subset $k \leq p$ explanatory variables (reduced model) that gives us the "best" model:

1. **Goodness of fit**
2. **Interpretability**
3. **Predictive performance**

Some related concepts:

1. **F-test:** compares between 2 specific models, where test adequacy of a reduced model (subset, nested) relative to the full model.
2. **Multicollinearity:** this can affect interpretability of $\hat{\beta}_j$. The usual interpretation is

   > holding other variables constant does not really work when x_j is strongly correlated with other predictors

3. R^2 : proportion of variability in response explained by regression model.

   R Always increases when adding more variables.

4. $\hat{\sigma}^2$: the estimated residual variance, used for prediction. We want to this to be small to get a good prediction performance

Two key ingredients of model selection:

1. **Metric (or criterion)** for comparing different models with potentially different number of predictors
2. **Selection/search strategy**, which models to fit?

2.10.1 Examples of Metrics for Model Selection

Adjusted R^2

Definition 2.10.1 — Adjusted R^2.

$$R^2_{adj} = 1 - \frac{\textbf{SS(Res)}/(n-k-1)}{\textbf{SS(Total)}/(n-1)}$$

where k is the number of model predictors.

R We have seen that

$$R^2 = 1 - \frac{\textbf{SS(Res)}}{\textbf{SS(Total)}}$$

In particular, $\frac{\textbf{SS(Res)}}{n-k-1}$ is the estimated $\hat{\sigma}^2$ for model with k predictors and $\frac{\textbf{SS(Total)}}{n-1}$ is the sample variance of response y_i. An alternative form is

$$R^2_{adj} = 1 - \frac{n-1}{n-k-1}(1-R^2) = 1 - \left(1 + \frac{k}{n-k-1}\right)(1-R^2) = R^2 - (1-R^2)\frac{k}{n-k-1}$$

Why do we do this? R^2_{adj} accounts for the number of variables in the model, penalizes inclusion of unimportant predictors (like someone once said, all predictors are equal, some are more eQuAL). This means **SS(Res)** decreases not as much when adding such variables. While R^2 always increases with more predictors (now, you know how to hack your R^2 presentation, or your friends do. So when someone brings you a result with 0.99 R^2, the first thing you do is to check whether he/she graduated from the University of Toronto), but R^2_{adj} can decrease if **SS(Res)**'s change is small.

While R^2_{adj} loses usual interpretation of R^2, but can be used as a measure of "goodness of fit" (oh my goodness) and model selection criterion. Higher your R^2_{adj} is, better the subset of predictors.

■ **Example 2.8 — Numerical Example - R^2_{adj}.** Given $n = 25$, **SS(Total)** $= 20$, and $p = 6$ predictors. Suppose we are considering a subset of $k = 4$ predictors, and find:

	Reduced $k=4$	Full $p=6$
SS(Total)	20	20
SS(Res)	10	9.8
Calculate R^2	0.5	$\frac{10.2}{20} = 0.51$
Calculate R^2_{adj}	$1 - \frac{10/20}{20/24} = 0.4$	$1 - \frac{9.8/18}{20/24} \approx 0.347$
$\hat{\sigma}^2$	$\frac{10}{20} = 0.5$	$\frac{9.8}{20} \approx 0.544$

Here $n-k-1$ is the degrees of freedom of the residuals in the reduced model and $n-p-1$ is the degrees of freedom of residuals in the full model. ■

R We can see that $R^2_{adj} < R^2$, and as $n \to \infty$, we do we have $R^2_{adj} \to R^2$. (But no one lives in the infinity world except Thanos)

Moreover, the model with higher R^2_{adj} has lower $\hat{\sigma}^2$. Thus, this is a reasonable metric for model selection.

Akaike Information Criterion (AIC)

Definition 2.10.2 — Akaike Information Criterion (AIC). Let n be the sample size, q be the number of parameters (In MLR, $q = k+1+1$ consisted of k predictors, intercept β_0 and σ^2).

$$\textbf{AIC} = 2q - 2\ln L(\hat{\theta})$$

where $L(\hat{\theta})$ is the likelihood function evaluated at $\hat{\theta}$, parameter estimates.

R One thing you might recall from STAT231 is that the least square estimates for $\vec{\beta}$ are equivalent to the MLE under usual Normal assumption on $\vec{\varepsilon}$.

R We can consider $2q$ as a penalty for including more predictors. With more parameters, $L(\hat{\theta})$ will increase but will be offsetted by penalty $2q$. A model with lower AIC is preferred. Note that we cannot interpret AIC by itself but we compare it amoung different models.

Bayesian Information Criterion (BIC)

Definition 2.10.3 — Bayesian Information Criterion (BIC). Similar to AIC setup, but more strongly penalizes inclusion of more variables,

$$\mathbf{BIC} = q\ln(n) - 2\ln L(\hat{\theta})$$

where n is the sample size.

R

1. R^2_{adj}, **AIC**, **BIC** are all based on comparing fitted models (explanatory power of model).
2. All have penalties to try to prevent overfitting (i.e. having too many variables, modeling *spurious* relationships that are actually noise)

> **Spurious:** adj. not being what it purports to be; false or fake.
>
> *a proof of your GRE/GMAT worthiness*

Mean-Square prediction Error (MSPE)

Consider predictive performance of model on new data. (i.e. data not used in the fitting models)

> Is model generalizable to new data?

Overfitting models tend to have high prediction error. We can do this by using cross-validation schemes.

R This is pretty much the revelation for people who started doing data science before learning statistics.

Too many to test

So far, we have 4 examples of metrics/criteria for comparing models: Imagine we have p predictors:

$$\left.\begin{array}{c}\binom{p}{1} = p \text{ regression models with 1 predictor} \\ \binom{p}{2} \text{ regression models with 2 predictors} \\ \vdots \\ \binom{p}{p} = 1 \text{ regression models with p predictors}\end{array}\right\} = \sum_{j=1}^{p} \binom{p}{j} = 2^p - 1 = \mathcal{O}(2^p)$$

This is computationally infeasible (exponential complexity, not polynomial complexity). Well, if we have a quantum computer, we could do it theoretically.

Philosophical Question

Occam's Razor Principle: simplest explanation is most likely the right one.

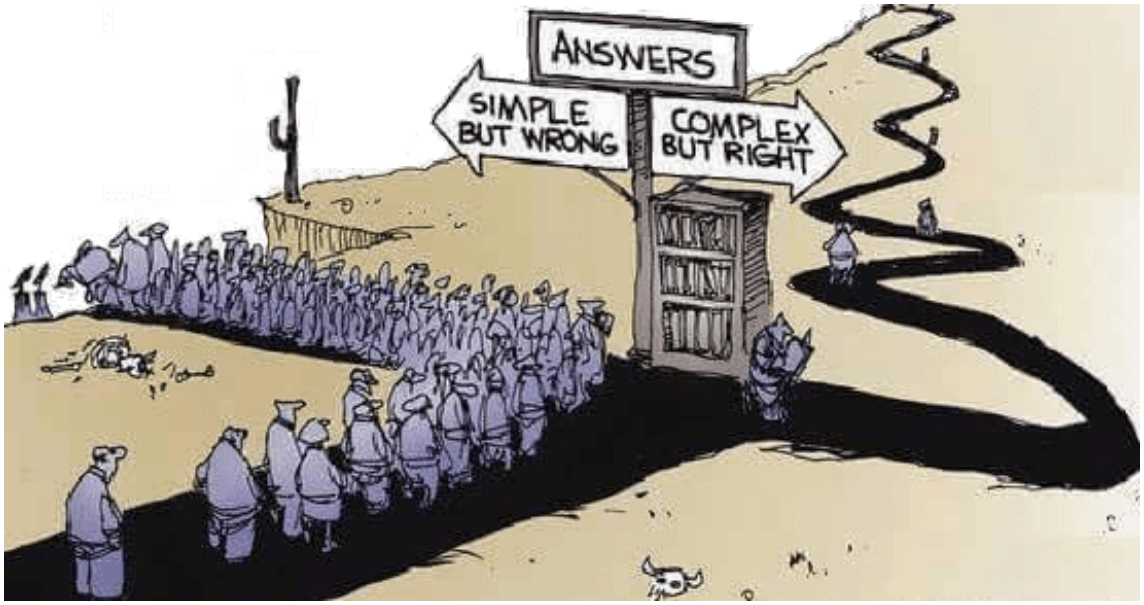


Figure 2.10.1: What if Occam is wrong and he is an impostor?

2.10.2 Search Strategies

1. **All possible regressions:** with p predictors, $\sum_{j=0}^{p}\binom{p}{j} = 2^p$ is number of potential models to fit

   R We can for sure find the optimal model, but computationally intensive or not even feasible (with large p) with a classical machine. We want to find a "good" (useful) model in reasonable computational time (not necessarily optimal). Many strategies focus on adding/removing variables at a time sequentially.

2. **Forward Selection:** add one variable at a time to the model
   (a) Start with model that only has intercept β_0
   (b) Fit p simple linear regression models $\vec{y} = \beta_0\vec{1} + \beta_1\vec{x}_j + \vec{\varepsilon}, j = 1, \cdots, p$
   (c) Pick the test of p models with 1 predictor according to the chosen criterion, and add that variable x_j to the model
   (d) Fit $p-1$ models containing x_j and another variable.
      i. If none of $p-1$ models improves criterion, STOP
      ii. pick best of $p-1$ models according to the criterion, so we have 2 variables in the model
   (e) Continue adding 1 variable at a time until no more variables improve the criterion
   (f) Final model is the one with the best criterion after we STOP

   R Note that this is much faster than all possible regressions, since the max number of models to fit is

   $$p + (p-1) + (p-1) + \cdots + 2 + 1 = \frac{p(p+1)}{2} = \mathcal{O}(p^2) < \mathcal{O}(2^p)$$

3. **Backward Elimination:** Start with full model that has all p predictors
   (a) Fit p models that result from removing one variable from the regression (i.e. each has $p-1$ variables)
   (b) Pick the best of p models according to the criterion and eliminate such x_j from the model

(c) Fit $p-1$ models that remove x_j and one other variable from the model
   i. Continue until removing additional variable does not improve

R This has the same computational complexity to the forward selection scheme.

4. **Foward-Backward:** allows individual variables to be both added/removed
   (a) Start as a forward selection
   (b) If we have k variables in the model:
      i. Backwards: fit k models with $k-1$ variables. If any of the these improve criterion, remove the variable that improves the most
      ii. Forwards: fit $p-k$ models with $k+1$ variables. If any of those improve criterion, add variable that improve the most

R These are the basic "stepwise" selection methods to get a "good" (useful) model. Many other more sophisticated procedures available (stochastic search, LASSO)
We have assumed that $n > p$ (else $X^\top X$ is not invertible), which is a *classical regression* setting. More specialized methods needed if number of predictors is larger than sample size.

2.10.3 R Demo - Model Selection

```
%## Coffee example  (Coffee Quality Institute, 2018) continued
coffee <- read.csv("coffee_arabica.csv")

mfull <- lm(Flavor~ factor(Processing.Method) + Aroma + Aftertaste +
    Body + Acidity + Balance + Sweetness + Uniformity + Moisture, dat=
   coffee)
summary(mfull)$adj.r.squared
[1] 0.8073297

AIC(mfull)
[1] -1087.524

BIC(mfull)
[1] -1027.282

library(leaps)
all_regs <- regsubsets(Flavor ~ ., data = coffee, nvmax = 10, nbest = 2^10,
     really.big = TRUE)
all_regs_summ <- summary(all_regs)
all_regs_summ$which
all_regs_summ$adjr2
all_regs_summ$bic

# Organize results according to number of variables in model
p <- 10
k <- c(rep(1, choose(p,1)),
           rep(2, choose(p,2)),
           rep(3, choose(p,3)),
           rep(4, choose(p,4)),
           rep(5, choose(p,5)),
           rep(6, choose(p,6)),
           rep(7, choose(p,7)),
           rep(8, choose(p,8)),
           rep(9, choose(p,9)),
           rep(10, choose(p,10)))
boxplot(all_regs_summ$adjr2 ~ k, xlab = "Number of predictors", ylab =
   expression(R[adj]^2), ylim = c(0,1))
abline(h = c(0,1), lty = 2, col = "red")
```

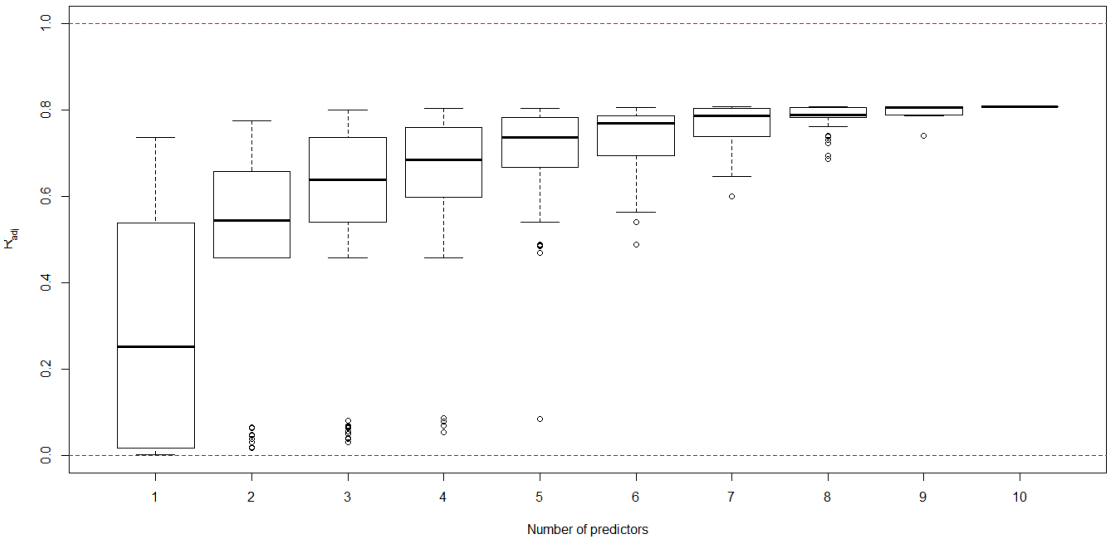


Figure 2.10.2: Adjusted R^2

```
%boxplot(all_regs_summ$bic ~ k, xlab = "Number of predictors", ylab = "BIC"
    )
```

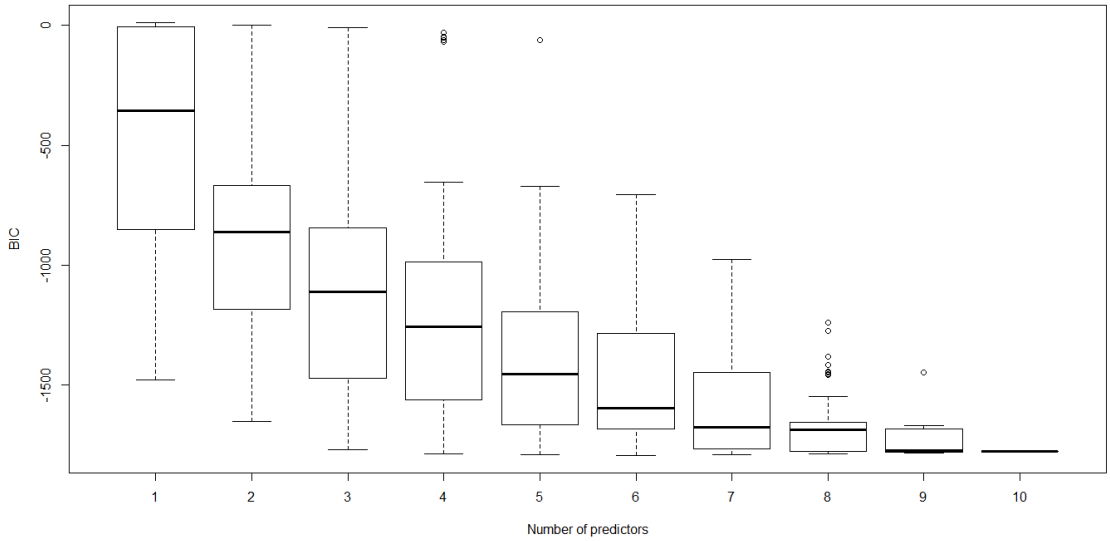


Figure 2.10.3: **BIC**

```
%max(all_regs_summ$adjr2)
[1] 0.8075027

bestR2adj <- which.max(all_regs_summ$adjr2)
min(all_regs_summ$bic)
[1] -1793.389

bestBIC <- which.min(all_regs_summ$bic)
```

```
9
10 # Find out which predictors in those models
11 all_regs_summ$which[bestR2adj,]
12 all_regs_summ$which[bestBIC,]
13
14
15 coffee$wet <- ifelse(coffee$Processing.Method == 'Washed / Wet', 1, 0) # 1
       = wet, 0 otherwise
16 coffee$semi <- ifelse(coffee$Processing.Method == 'Semi-washed / Semi-
       pulped', 1, 0) # 1 = semi/dry, 0 otherwise
17 coffee$Processing.Method <- NULL
18
19 m_bestr2adj <- lm(Flavor~ wet + Aroma + Aftertaste +
20             Body + Acidity + Balance + Sweetness + Uniformity + Moisture,
       dat=coffee)
21 summary(m_bestr2adj)
22 AIC(m_bestr2adj)
23 BIC(m_bestr2adj)
24
25 m_bestBIC <- lm(Flavor~ wet + Aroma + Aftertaste +
26                   Body + Acidity + Sweetness , dat=coffee)
27 summary(m_bestBIC)
28 AIC(m_bestBIC)
29 BIC(m_bestBIC)
30
31 # Let's also try stepwise methods
32 library(MASS)
33
34 # Full model and empty model with just intercept
35 full <- lm(Flavor ~ ., data = coffee)
36 empty <- lm(Flavor ~ 1, data = coffee)
37
38 # default stepAIC uses AIC criterion
39 stepAIC(object = empty, scope = list(upper = full, lower = empty),
       direction = "forward")
40
41 # Let's get stepAIC to use BIC by specifying the penalty k = log(n)
42 # Forward
43 stepAIC(object = empty, scope = list(upper = full, lower = empty),
       direction = "forward", k = log(nrow(coffee)))
44 m_f <- stepAIC(object = empty, scope = list(upper = full, lower = empty),
       direction = "forward", trace = 0, k = log(nrow(coffee)))
45 summary(m_f)
46
47 # Backward
48 stepAIC(object = full, scope = list(upper = full, lower = empty), direction
       = "backward", k = log(nrow(coffee)))
49 m_b <- stepAIC(object = full, scope = list(upper = full, lower = empty),
       direction = "backward", trace = 0, k = log(nrow(coffee)))
50 summary(m_b)
51
52 # Forward-backward
53 stepAIC(object = empty, scope = list(upper = full, lower = empty),
       direction = "both", k = log(nrow(coffee)))
54 m_h <- stepAIC(object = empty, scope = list(upper = full, lower = empty),
       direction = "both", trace = 0, k = log(nrow(coffee)))
55 summary(m_h)
56
57 # 10 variables is still a fairly small problem:  in this example
58 # all 3 approaches identify the same BIC-based model as the exhaustive
       search.
```

2.11 Model Assumption Check

Recall that, in our MLR model, we have

$$\vec{Y} = X\vec{\beta} + \vec{\varepsilon}, \vec{\varepsilon} \sim \mathbf{MVN}(0, \sigma^2 I_{m\times n})$$

Practically, this means $\varepsilon_i \overset{\text{iid}}{\sim} N(0,\sigma^2)$

1. Independence among all error terms
2. Normally distributed
3. Since $\mathbb{E}(Y_i) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$ is implied by $\mathbb{E}(\varepsilon_i) = 0$ for any $x_{i1}, \cdots, x_{ip}$. This means linear model is appropriate that it can correctly explains response on average.
4. Constant error variance σ^2 (this might be easily violated)

We could asses these assumptions via ε_i's but we cannot observe ε_i directly. Rather, we do have an approximation via the residuals e_i from the fitted model. Recall that

$$\vec{e} \sim \mathbf{MVN}\big(0, (I-H)\sigma^2\big)$$

where H is the hat matrix. So $\vec{e}, \vec{\varepsilon}$ are related:

$$\begin{aligned}\vec{e} &= \vec{Y} - X\hat{\vec{\beta}} = (I-H)\vec{Y} \\ &= (I-H)(X\vec{\beta} + \vec{\varepsilon}) \\ &= (X\vec{\beta} - HX\vec{\beta}) + (I-H)\varepsilon = (I-H)\vec{\varepsilon}\end{aligned}$$

since $HX = X$.

> R We cannot actually solve for $\vec{\varepsilon}$ since $I-H$ is not invertible since it is not full rank. Recall that both H and $I-H$ are idempotent, $\text{Tr}(H) = p+1 = \text{Tr}(X)$. Then, $\text{Tr}(I-H) = n-(p+1) < n$. Similarly, this does not imply that $\vec{Y} = \vec{\varepsilon}$.

So, $e_i = \varepsilon_i - \sum_{j=1}^n h_{ij}\varepsilon_j$ which means e_i is a good approximation to ε_i when the entries of h_{ij} of H are small (which is usually the case, especially when n is large). We note that

$$e_i \sim N(0, \sigma^2(1-h_{ii})) \iff \frac{e_i - 0}{\sigma\sqrt{1-h_{ii}}} \sim N(0,1)$$

if we plug in $\hat{\sigma}$, that defines the **studentized residuals**

$$d_i = \frac{e_i}{\hat{\sigma}\sqrt{1-h_{ii}}}$$

The common practice is to use $\vec{e}$ for following residual plots/diagnostics (using d_i is also possible).

2.11.1 Model Assumption Check Options

Plot $\vec{e}$ vs. $\vec{\mu}$

Elements in $\vec{e}$ and $\vec{\mu}$ are mutually independent since they are **MVN**s with $\mathbf{Cov}(\vec{e}, \vec{\mu}) = 0$.

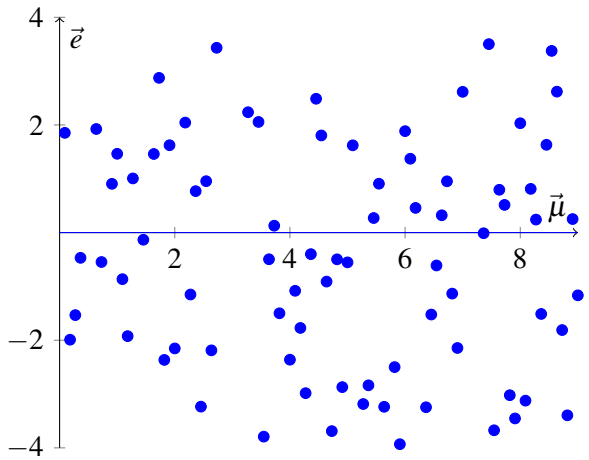


Figure 2.11.1: Scatter plot $\vec{e}$ vs. $\vec{\mu}$ that indicates that our assumption is valid

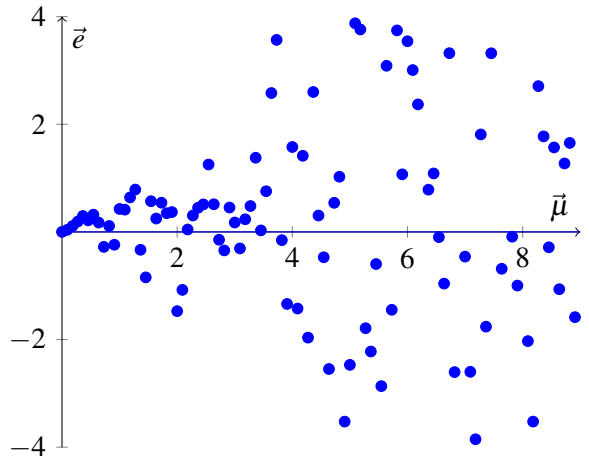


Figure 2.11.2: Scatter plot $\vec{e}$ vs. $\vec{\mu}$ that indicates that our assumption is problematic: variance increases with fitted values

In general, plot of $\vec{e}$ vs. $\vec{\mu}$ can show deviations from independent constant variance if those violated.

Plot $\vec{e}$ vs $\vec{x}_j$ for each predictor

When there are not too many predictors, this can detect non-linearity between $\vec{x}_j$ and $\vec{y}$, not as practical when p is large.

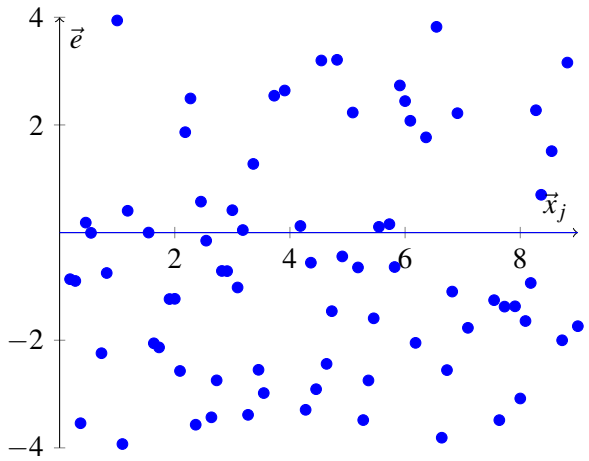


Figure 2.11.3: Scatter plot $\vec{e}$ vs. $\vec{x}_j$ that indicates that our assumption is valid

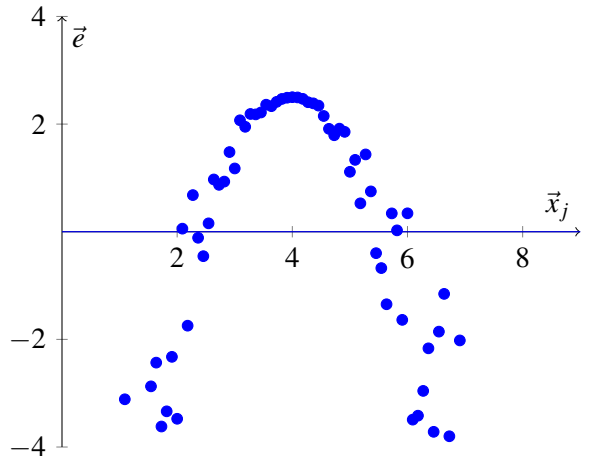


Figure 2.11.4: Scatter plot $\vec{e}$ vs. $\vec{x}_j$ that indicates non-linearity

Also, if $\mathbb{E}(\varepsilon_i) > 0$ for that value of x_j, we might also conclude non-linearity.

If the observation numbers $1, 2, \cdots, n$ were collected in some order (time, space, etc.), also plot e_i vs. indices i to check for any patterns (again, look for random scatter)

Normality Check of the $\vec{e}$

1. **Histogram of $\vec{e}$:** is it bell-shaped and symmetric?

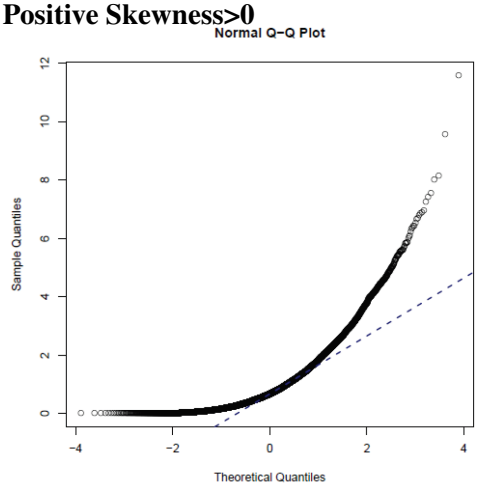
   R This generally works but hard to distinguish normal against distributions with overly fat/thin tails (kurtosis)

2. **QQPlot of $\vec{e}$:** more formally assess normality

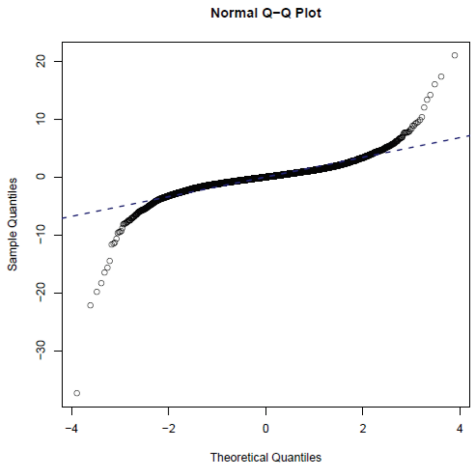
   Definition 2.11.1 — QQPlot. Scatter plot of ordered quantiles from 2 distributions, which are
   (a) empirical quantiles from residuals
   (b) theoretical quantiles from assumed Normal distribution

   If quantiles roughly matches, the points will roughly fall on 45 degree through origin ($y = x$)

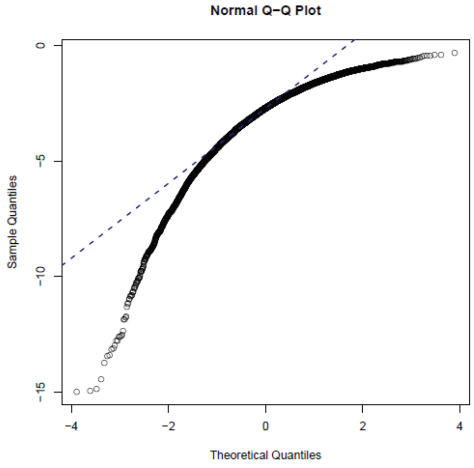
Positive Skewness>0



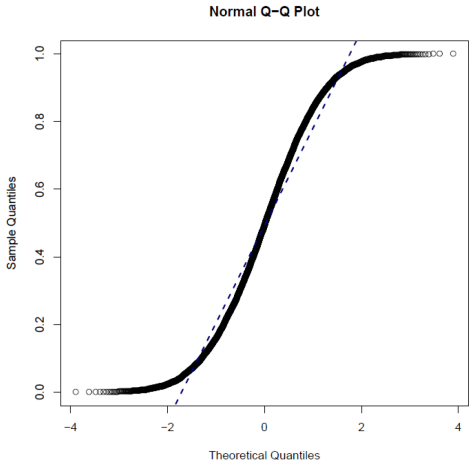
Symmetric; high kurtosis>3



Negative Skewness<0



Symmetric; low kurtosis<3



2.11.2 R Demo - Model Assumption Check

```
%### Residual plots/diagnostics demo

## Florida oranges revisited
dat <- read.csv("florange.csv")
plot(dat$acres,dat$boxes)
lm.1 <- lm(dat$boxes~dat$acres)
summary(lm.1)

# Residual plot: vs fitted values
plot(lm.1$fitted.values, lm.1$residuals, xlab = "Fitted Values", ylab = "
    Residuals")

# Residual plot: vs predictor (just one in this case)
plot(dat$acres, lm.1$residuals, xlab = "Acres", ylab = "Residuals")

# Residual plot: vs i (just to demo plot; no time/space ordering here)
plot(1:nrow(dat), lm.1$residuals, xlab = "Index", ylab = "Residuals")

# Histogram of residuals
hist(lm.1$residuals)

# QQ plot of residuals
qqnorm(lm.1$residuals)
```

```
23 qqline(lm.1$residuals, col="blue", lwd = 2)
```

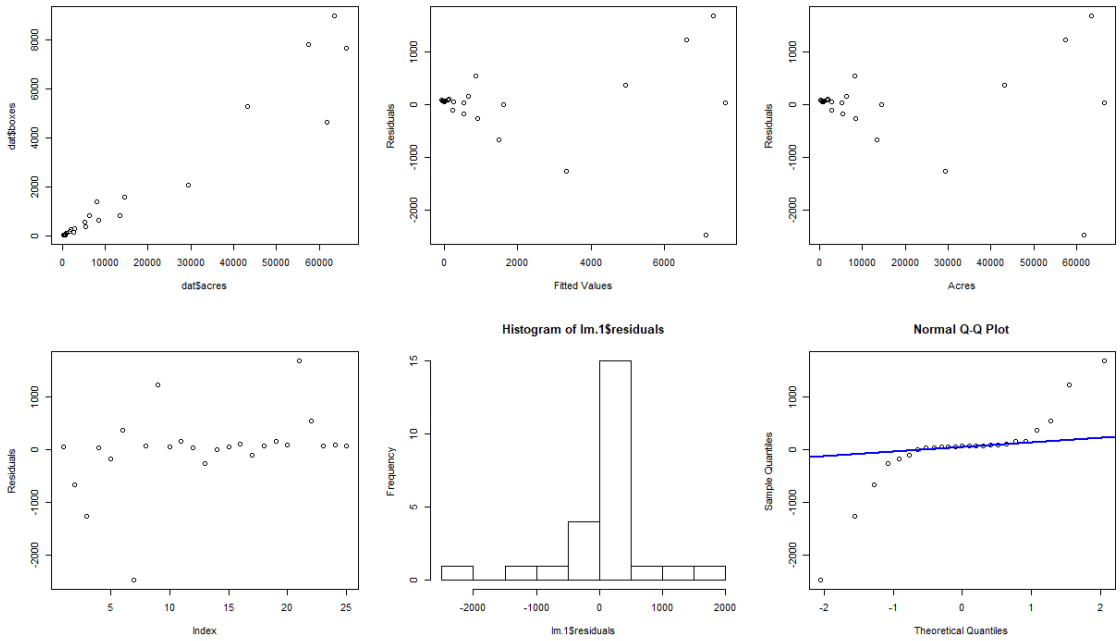


Figure 2.11.5: Orange Data

```
1 %## Rocket data revisited
2 rocket <- read.csv(file="rocket.csv")
3 mr <- lm(thrust ~ nozzle + propratio, data = rocket)
4 summary(mr)
5
6 # Residual plot: vs fitted values
7 plot(mr$fitted.values, mr$residuals, xlab = "Fitted Values", ylab = "
    Residuals")
8
9 # Residual plot: vs predictors
10 plot(rocket$nozzle, mr$residuals, xlab = "Nozzle (1 = large)", ylab = "
    Residuals")
11 plot(rocket$propratio, mr$residuals, xlab = "Propellant to fuel ratio",
    ylab = "Residuals")
12
13 # Histogram of residuals
14 hist(mr$residuals)
15
16 # QQ plot of residuals
17 qqnorm(mr$residuals)
18 qqline(mr$residuals, col="blue", lwd = 2)
```

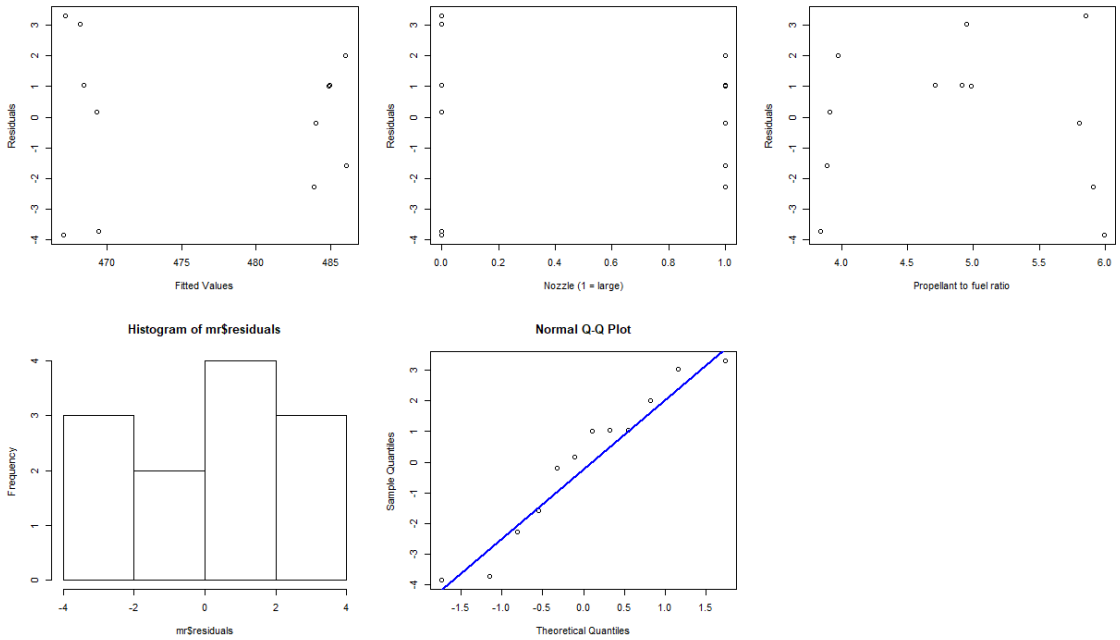


Figure 2.11.6: Rocket Data

```
%## Mystery dataset!
mystery <- read.csv("mystery.csv")
head(mystery)
pairs(mystery)
mm <- lm(Y~X1+X2+X3, data=mystery)
summary(mm)
plot(mm$fitted.values, mm$residuals)
```

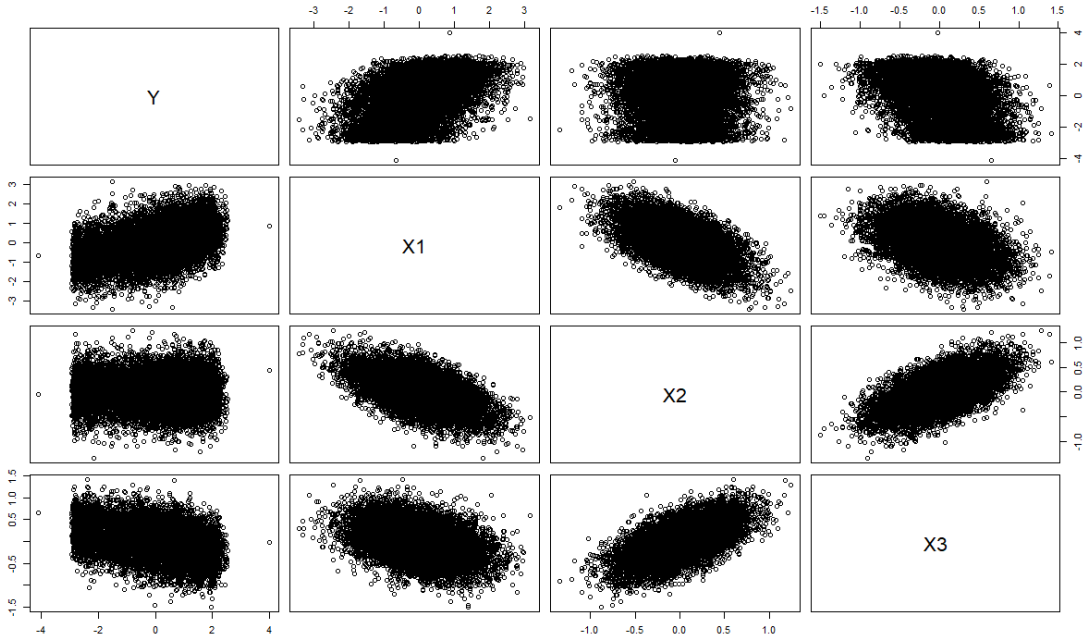


Figure 2.11.7: Mystery Data Pairplot

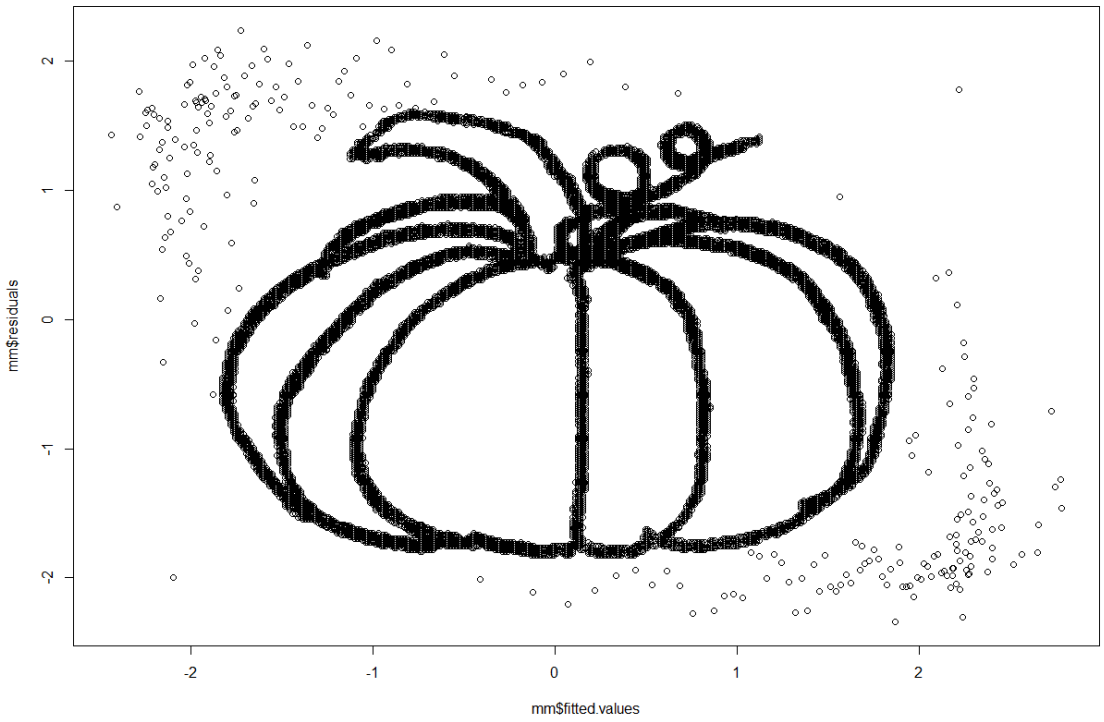


Figure 2.11.8: Quality of education in grad school

2.11.3 Addressing Model Assumption Problems

If residual plots reveal problems with assumptions (although plots don't fully check linearity, independence), we might be able to address via **transformations**, **adding variables to the model**, or **use different error distribution on** ε.

Variance-stabilizing transformations on responses y

This can help address non-constant variance identified in $\vec{e}$ and $\hat{\vec{\mu}}$ plot. The idea is

> Apply function g and fit regression model on the transformed $\{g(y_i)\}_i$
>
> $$g(Y_i) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i$$

Note that this can drastically change **SS(Res)** and $\hat{\sigma}$, so cannot directly use those among different choices of g.

Rationale: the variance of responses might be a function of mean $\mu_i = \mathbb{E}(Y_i)$. For example, larger μ_i have larger variances, like the case with Florida Oranges.

$$\mathbf{Var}(Y_i) = \mathbf{Var}(\varepsilon_i) = h(\mu_i)\sigma^2$$

for some $h > 0$. In which case, we want $\mathbf{Var}(g(Y_i)) \approx \sigma^2$.

By 1st-order Taylor expansion, we have

$$g(Y_i) \approx g(\mu_i) + (Y_i - \mu_i)g'(\mu_i)$$

then,

$$\mathbf{Var}(g(Y_i)) \approx \left[g'(\mu_i)\right]^2 \mathbf{Var}(Y_i)$$

Thus, we need $[g'(\mu_i)]^2 \propto \frac{1}{h(\mu_i)}$

■ **Example 2.9**

1. $h(\mu_i) = \mu_i \Longrightarrow \mathbf{Var}(Y_i) = \sigma^2\mu_i \propto \mu_i$. In this case, variance in response is proportional to the mean response. We need

   $$g'(\mu_i) \propto \frac{1}{\sqrt{h(\mu_i)}} = \frac{1}{\sqrt{\mu_i}} \Longrightarrow g(\mu_i) = \sqrt{\mu_i}$$

   will work. Thus, we can apply $g(y_i) = \sqrt{y_i}$ to obtain approximate constant variance
2. $h(\mu_i) = \mu_i^2 \Longrightarrow \mathbf{Var}(Y_i) = \sigma^2\mu_i^2 \propto \mu_i^2$ or $\mathbf{SD}(Y_i) \propto \mu_i$. We need

   $$g'(\mu_i) \propto \frac{1}{\mu_i} \Longrightarrow g(\mu_i) = \log(\mu_i)$$

   will work. Thus, we can apply $g(y_i) = \log(y_i)$ to obtain approximate constant variance
3. **Power Transformation (Box-Cox):** consider

   $$g(y_i) = \begin{cases} \frac{y_i^\lambda - 1}{\lambda} & \lambda \neq 0 \\ \log(y_i) & \lambda = 0 \end{cases}$$

   note that

   $$g'(\mu_i) = \begin{cases} \mu_i^{\lambda-1} & \lambda \neq 0 \\ \frac{1}{\mu_i} & \lambda = 0 \end{cases} \iff h(\mu_i) \propto \frac{1}{[g'(\mu_i)]^2} = \mu_i^c$$

   where c is some arbitrary power. Thus, Box-Cox transformation can help address non-constant variance of the form $\mu_i^c\sigma^2 = \mathbf{Var}(Y_i)$. For some special cases,
   (a) when $\lambda = \frac{1}{2}$, we have the square-root transfrom

(b) when $\lambda = 0$, we have the log transformation
(c) when $\lambda = 1$, we have the identity transformation
(d) when $\lambda = -1$, we have the reciprocal transformation.

can automatically try a sequence of λ and find the choice that gives best value of likelihood.

■

Note that interpreting $\hat{\beta}_j$ can be less intuitve as a result of transformation, since now increasing x_j be 1 unit corresponds to an estimated change of $\hat{\beta}_j$ in $g(y_i)$.

For the log transformation $g(y_i) = \log(y_i)$, $\hat{\beta}_j$ represents the estimate of expected change of response in the log-scale, which corresponds to $e^{\hat{\beta}_j}$ being the expected multiplicative change applied to the (original) response.

For arbitrary λ, transform might be less interpretable.

Transforming/adding explanatory eariables

This might be applied

1. if y or $g(y)$ has a clear non-linear relation with some x_j. For example, revealed by $\vec{e}$ vs. $\vec{x}_j$, or pairs plot among all predictors and $\vec{y}$. We can consider transforming $\vec{x}_j$ using power transformation.
2. could add polynomial terms, such as $x^2, x^3, \cdots$. For example,

$$Y_1 = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \varepsilon_i$$

suppose we think adding $\vec{x}_i^2$ is appropriate, then we define $x_{i3} = x_{i1}^2$ and fit

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \varepsilon_i$$

which is still linear in β and also note that $\vec{x}_1, \vec{x}_1^2$ are linearly independent.

3. Add interaction terms: if we think the effect of $\vec{x}_i$ on response depends on value of $\vec{x}_j$. For example,

$$Y_1 = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \varepsilon_i$$

suppose we think $\vec{x}_1, \vec{x}_2$ interact, then we might fit

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \varepsilon_i$$

where $x_{i3} = x_{i1} x_{i2}$, so that

$$Y_i = \beta_0 + \underbrace{(\beta_1 + \beta_3 x_{i2})}_{\text{effect depends on } x_{i2}} x_{i1} + \beta_2 x_{i2} + \varepsilon_i$$

we can see the interaction from the expression above.
In general, there are $\binom{p}{3}$ possible intersections (two-way).

R Consider whether interactions are conceptually plausible.

Different $\vec{\varepsilon}$ distribution

Say QQPlot of residuals not Normal even after any approximate transformations. For example, we can apply student-t, Cauchy, or Laplace distribution to $\vec{\varepsilon}$ with heavier tails.

2.11.4 R Demo - Addressing Model Assumption Problems

```
%## Demo for transformations and interactions

## Florida oranges revisited
dat <- read.csv("florange.csv")
lm.1 <- lm(dat$boxes~dat$acres)
summary(lm.1)

# Recall: residuals had non-constant variance
#  (variance increases with fitted values)
plot(lm.1$fitted.values, lm.1$residuals, xlab = "Fitted Values", ylab = "
    Residuals")
qqnorm(lm.1$residuals)
qqline(lm.1$residuals, col="blue", lwd = 2)

# Try log-transforming y
lm.log <- lm( log(dat$boxes)~dat$acres)
summary(lm.log)
plot(lm.log$fitted.values, lm.log$residuals, xlab = "Fitted Values", ylab =
     "Residuals")
plot(dat$acres, lm.log$residuals, xlab = "Fitted Values", ylab = "Residuals
    ")
qqnorm(lm.log$residuals)
qqline(lm.log$residuals, col="blue", lwd = 2)
# Does the plot of residuals vs x suggest a problem
# Let's take a closer look
plot(dat$acres,log(dat$boxes)) # evidently not linear!

# Log-transform x as well
plot(log(dat$acres),log(dat$boxes)) # looks much more linear!
lm.loglog <- lm( log(dat$boxes)~log(dat$acres))
qqnorm(lm.loglog$residuals)
qqline(lm.loglog$residuals, col="blue", lwd = 2)
plot(lm.loglog$fitted.values, lm.loglog$residuals, xlab = "Fitted Values",
    ylab = "Residuals")
plot(log(dat$acres), lm.loglog$residuals, xlab = "Fitted Values", ylab = "
    Residuals")
```

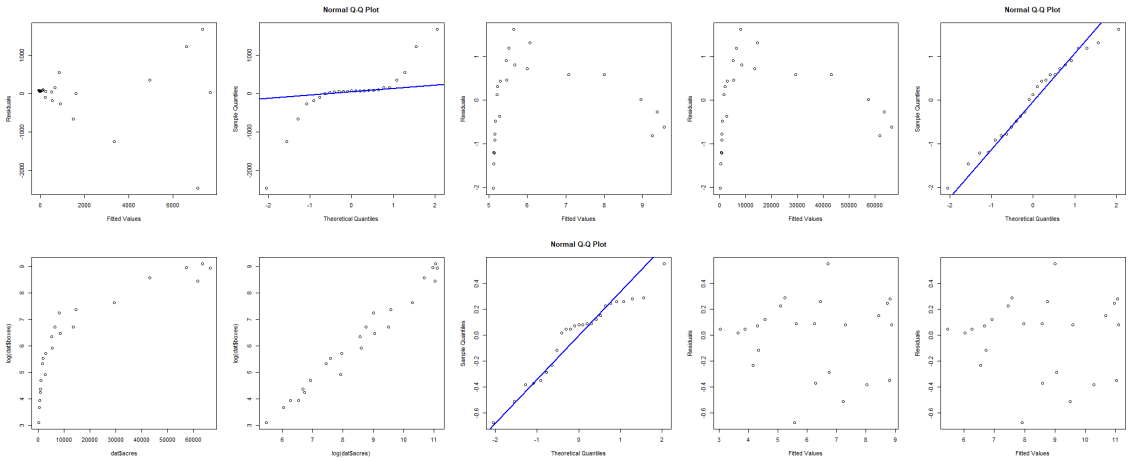


Figure 2.11.9: Florida Orange

```
%## Python data revisited
python <- read.csv("FLpython.csv")
```

```
python$male <- ifelse(python$sex == 'M', 1, 0) # 1 = M, 0 =F
mpf2 <- lm(fat ~ male+mass+svl, data = python)
summary(mpf2)

# Residual plot: vs fitted values
plot(mpf2$fitted.values, mpf2$residuals, xlab = "Fitted Values", ylab = "
    Residuals")
## QQ plot of residuals
qqnorm(mpf2$residuals)
qqline(mpf2$residuals, col="blue", lwd = 2)

# Try a Box-Cox transformation
library(MASS)
bc <- boxcox(mpf2)
lambda <- bc$x[which.max(bc$y)]
mpf3 <- lm( (fat^lambda-1)/lambda ~ male+mass+svl, data = python)
summary(mpf3)
plot(mpf3$fitted.values, mpf3$residuals)
plot(python$mass, mpf3$residuals)
plot(python$svl, mpf3$residuals)
qqnorm(mpf3$residuals)
qqline(mpf3$residuals, col="blue", lwd = 2)
# still some skew, but better!
```

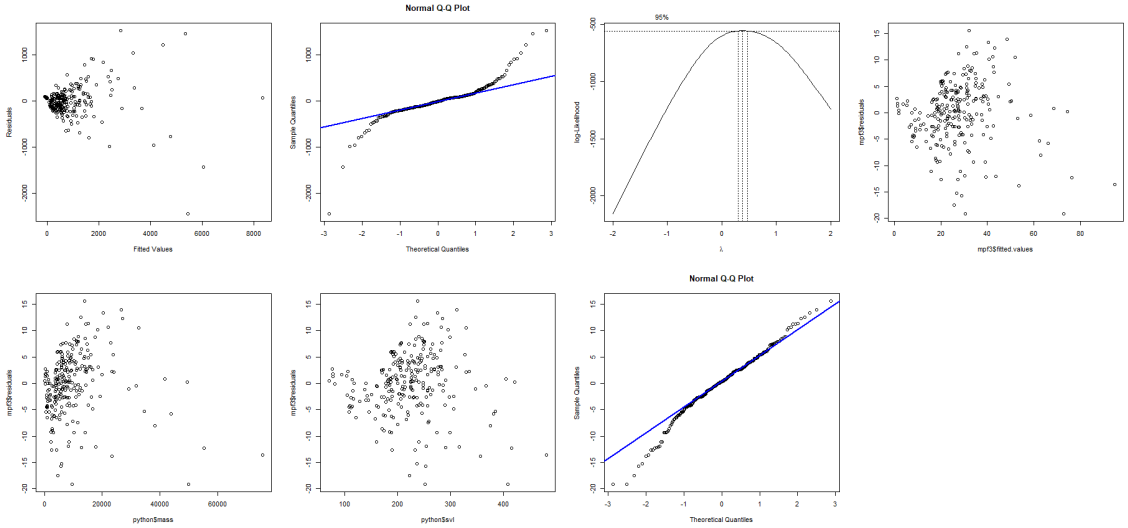


Figure 2.11.10: Florida Python

2.12 Effect of Indivudal Observations

Sometimes an observation (or a few observations) can have an outsized impact on the fitted MLR model.

2.12.1 Outliers

Definition 2.12.1 — Outlier. We say an observation $(y_i, x_{i1}, \cdots, x_{ip})$ is an outlier if it is substantially different from other observations.

This can occur if its response and/or some of its explanatory variables have values that are unusual or extreme compared to others.

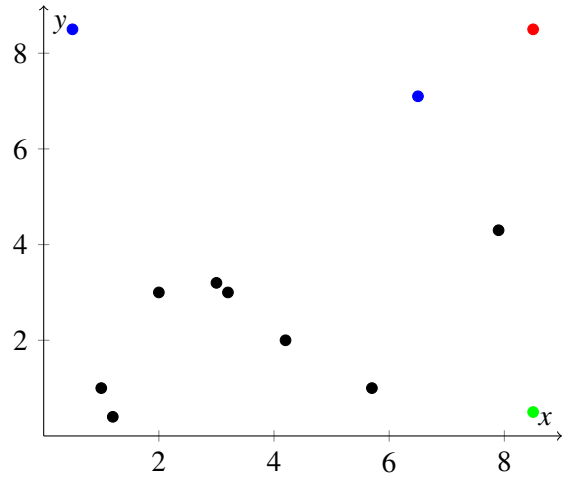


Figure 2.12.1: Simple Linear Regression Outliers

For example, the blue dot in the graph above is an outlier in the response, the green dot is an outlier in the explanatory variable, and the red dot is an outlier in both explanatory and response.
Outliers can occur for different reasons, such as an extraordinary subject and data entry errors. Generally, we don't recommend removing/throwing out outliers unless we have strong reason to believe that observation is an error and does not belong in dataset. But it is useful to investigate what effect it has on our fitted model and also the quality of fit to the rest of data.

2.12.2 Detection/Characterization of Outliers

Studentized Residual

$$d_i = \frac{e_i}{\hat{\sigma}\sqrt{1-h_{ii}}}$$

recall that $e_i \sim N(0, \sigma^2(1-h_{ii}))$. So, if $|d_i|$ is large, the $i-$th observation could be considered an outlier in the sense of having an extreme value of response y_i, for example $|d_i| > 3$. This corresponds to the blue dot in the graph above.

Leverage

Recall that h_{ii} is $i-th$ diagonal element of hat matrix H.

Definition 2.12.2 — Leverage. We call h_{ii} the leverage of the $i-$th observation.

$$\hat{\vec{\mu}} = X\hat{\vec{\beta}} = \underbrace{X(X^\top X)^{-1}X^\top}_{H}\vec{Y} = H\vec{Y}$$

So,

$$\hat{\mu}_i = \begin{bmatrix} h_{i1} & h_{i2} & \cdots & h_{in} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \sum_{j=1}^{n} h_{ij}y_j = h_{ii}y_i + \sum_{j\neq i} h_{ij}y_j$$

So h_{ii}, the leverage, captures contribution of Y_i in determining its corresponding fitted value. If the leverage is large relative to the other $h_{ij}, j \neq i$, then $\hat{\mu}_i$ is mostly determined by Y_i. In A3, you will show that

$$\frac{1}{n} \leq h_{ii} \leq 1$$

If $h_{ii} \approx 1$, then $\mathbf{Var}(e_i) = \sigma^2(1-h_{ii}) \approx 0$, which in turn implies $y_i \approx \hat{\mu}_i$. This means residuals of observation with high leverage tend to be small.

Rule of Thumb: an observation with leverage higher than twice the average is considered to be high

$$h_{ii} > 2\hat{h} = \frac{2}{n}\sum_{i=1}^{n} h_{ii} = \frac{2}{n}\text{Tr}(H) = \frac{2}{n}\text{rank}(X) = \frac{2(p+1)}{n}$$

Recall that $H = X(X^\top X)^{-1}X^\top$ only involves predictors and not response. Thus, leverage is useful to help identify outlier(s) in the sense of having explanatory variables with extreme values, such as the green dot in the graph.

■ **Example 2.10 — Proof by Example - SLR.** In SLR case, $h_{ii} = \frac{1}{n} + \frac{(x_i-\bar{x})^2}{S_{xx}}$ after some algebra. This means if x_i is far from $\bar{x}$, that point will have high leverage. ■

R This generalizes to MLR: an observation with high leverage is an outlier with extreme values in one or more explanatory variables. But leverage does not tell us directly whether that observation is also an outlier in response in fact $y_i \approx \hat{\mu}_i$ for such observation with high leverage.

Influence:

Definition 2.12.3 — Influence. The $i-$th observation is influential if its presence in fitting regression considerably changes estimates compared to when the $i-$th observation is not used to fit model.

Start with fitting $\vec{Y} = X\vec{\beta} + \vec{\varepsilon}$ using all observations and calculate estimates $\hat{\vec{\beta}}$ as usual via LS.
Let $\hat{\vec{\beta}}^{(i)}$ denote LS estimates based on fitting model with $i-$th observation removed.
The idea is that if $\hat{\vec{\beta}}^{(i)}$ is quite different than $\hat{\vec{\beta}}$, then observation i is highly influential. We measure this via Cook's distance.

Definition 2.12.4 — Cook's Distance. For $\hat{\vec{\beta}}$ and $\hat{\vec{\beta}}^{(i)}$, the Cook's Distance is

$$D_i = \frac{(\hat{\vec{\beta}}^{(i)} - \hat{\vec{\beta}})^\top X^\top X(\hat{\vec{\beta}}^{(i)} - \hat{\vec{\beta}})}{\hat{\sigma}^2(p+1)}$$

To see intuition, let $\hat{\vec{\mu}}^{(i)} = X\hat{\vec{\beta}}^{(i)}$ be the fitted values based on removing $i-$th observation and estimating $\vec{\beta}$. Then, we see

$$D_i = \frac{(X\hat{\vec{\beta}}^{(i)} - X\hat{\vec{\beta}})^\top (X\hat{\vec{\beta}}^{(i)} - X\hat{\vec{\beta}})}{\hat{\sigma}^2(p+1)} = \frac{(\hat{\vec{\mu}}^{(i)} - \hat{\vec{\mu}})(\hat{\vec{\mu}}^{(i)} - \hat{\vec{\mu}})}{\hat{\sigma}^2(p+1)} = \frac{\left\|\hat{\vec{\mu}}^{(i)} - \hat{\vec{\mu}}\right\|}{\hat{\sigma}^2(p+1)}$$

Thus, D_i measure the Euclidean distance between fitted values of the two regressions that gives $\hat{\vec{\mu}}^{(i)}$ and $\hat{\vec{\mu}}$, up to a scaling factor. Further, it can be shown that

$$D_i = d_i^2 \frac{h_{ii}}{1-h_{ii}} \cdot \frac{1}{p+1}$$

where we see that both d_i, h_{ii} are key quantities to calculate D_i and observation that are most influential would have large values of $|d_i|$ and h_{ii}.
So, highly influential observation tend to be outliers in the sense of having extreme values in both response and one/more predictors, such as the red dot on the graph (you still know which graph?).

Theorem 12 — Cook's Distance (IInEaR alGEbRa).

$$D_i = d_i^2 \frac{h_{ii}}{1-h_{ii}} \cdot \frac{1}{p+1}$$

Proof. For the $i-$th observation, the values of its predictors are

$$\vec{v}_i = \begin{bmatrix} 1 & x_{i1} & x_{i2} & \cdots & x_{ip} \end{bmatrix}^\top$$

so,

$$X = \begin{bmatrix} - & \vec{v}_1^\top & - \\ & \vdots & \\ - & \vec{v}_n^\top & - \end{bmatrix} \text{ and } X^\top X = \begin{bmatrix} | & & | \\ \vec{v}_1 & \cdots & \vec{v}_n \\ | & & | \end{bmatrix} \begin{bmatrix} - & \vec{v}_1^\top & - \\ & \vdots & \\ - & \vec{v}_n^\top & - \end{bmatrix} = \sum_{j=1}^{n} \vec{v}_j \vec{v}_j^\top$$

Recall $H = X(X^\top X)^{-1}X^\top$, $h_{ii} = \vec{v}_i^\top (X^\top X)^{-1}\vec{v}_i$ are the leverages. Let $X^{(i)}$ be X with $i-$th observation deleted, $\vec{y}^{(i)}$ be $\vec{y}$ with $i-$th response deleted. Then,

$$X^{(i)\top} X^{(i)} = \sum_{j \neq i} \vec{v}_j \vec{v}_j^\top$$

Thus,

$$X^\top X = X^{(i)\top} X^{(i)} + \vec{v}_i \vec{v}_i^\top \qquad (\dagger)$$

Similarly,

$$X^\top \vec{y} = \sum_{j=1}^{n} \vec{v}_j y_j, X^{(i)\top} \vec{y}^{(i)} = \sum_{j \neq i} \vec{v}_j y_j \Longrightarrow X^\top \vec{y} = X^{(i)\top} \vec{y}^{(i)} + \vec{v}_i y_i \qquad (\otimes)$$

Fit LS using $X^{(i)}$ and $\vec{y}^{(i)}$ to obtain $\hat{\vec{\beta}}^{(i)} = (X^{(i)\top} X^{(i)})^{-1} X^{(i)\top} \vec{y}^{(i)}$. By substituting $(\dagger)(\otimes)$, we have

$$\hat{\vec{\beta}}^{(i)} = (X^\top X - \vec{v}_i \vec{v}_i^\top)^{-1} (X^\top \vec{y} - \vec{v}_i y_i)$$

Lemma 2.13 For a square and invertible matrix A, we have

$$(A - \vec{a}\vec{a}^\top)^{-1} = A^{-1} + \frac{A^{-1}\vec{a}\vec{a}^\top A^{-1}}{1 - \vec{a}^\top A^{-1} \vec{a}}$$

Then,

$$\begin{aligned}
\hat{\vec{\beta}}^{(i)} &= \left[(X^\top X)^{-1} + \frac{(X^\top X)^{-1}\vec{v}_i\vec{v}_i^\top(X^\top X)^{-1}}{1-\underbrace{\vec{v}_i^\top(X^\top X)^{-1}\vec{v}_i}_{=h_{ii}}}\right](X^\top\vec{y}-\vec{v}_i y_i) \\
&= \left[(X^\top X)^{-1} + \frac{(X^\top X)^{-1}\vec{v}_i\vec{v}_i^\top(X^\top X)^{-1}}{1-h_{ii}}\right](X^\top\vec{y}-\vec{v}_i y_i) \\
&= \overbrace{(X^\top X)^{-1}X^\top\vec{y}}^{=\hat{\vec{\beta}}} - (X^\top X)^{-1}\vec{v}_i y_i \\
&\quad + \frac{(X^\top X)^{-1}\vec{v}_i\vec{v}_i^\top\overbrace{(X^\top X)^{-1}X^\top\vec{y}}^{=\hat{\vec{\beta}}} - (X^\top X)^{-1}\vec{v}_i\overbrace{\vec{v}_i^\top(X^\top X)^{-1}\vec{v}_i}^{=h_{ii}}y_i}{1-h_{ii}} \\
&= \hat{\vec{\beta}} - (X^\top X)^{-1}\vec{v}_i\left[y_i - \frac{\vec{v}_i^\top\hat{\vec{\beta}} - h_{ii}y_i}{1-h_{ii}}\right] \\
&= \hat{\vec{\beta}} - (X^\top X)^{-1}\vec{v}_i\left[\frac{y_i - h_{ii}y_i - \vec{v}_i^\top\hat{\vec{\beta}} + h_{ii}y_i}{1-h_{ii}}\right] \\
&= \hat{\vec{\beta}} - (X^\top X)^{-1}\vec{v}_i\left[\frac{\overbrace{y_i - \vec{v}_i^\top\hat{\vec{\beta}}}^{=e_i}}{1-h_{ii}}\right]
\end{aligned}$$

Thus, we have

$$\hat{\vec{\beta}}^{(i)} - \hat{\vec{\beta}} = \frac{-e_i}{1-h_{ii}}(X^\top X)^{-1}\vec{v}_i$$

R In fact, we can get $\hat{\vec{\beta}}^{(i)}$ from the regression with all n observations instead of fit a separate model.

Recall the definition of the Cook's Distance

$$D_i = \frac{(\hat{\vec{\beta}}^{(i)} - \hat{\vec{\beta}})^\top X^\top X(\hat{\vec{\beta}}^{(i)} - \hat{\vec{\beta}})}{\hat{\sigma}^2(p+1)}$$

we calculate

$$\begin{aligned}
&\frac{(\hat{\vec{\beta}}^{(i)} - \hat{\vec{\beta}})^\top X^\top X(\hat{\vec{\beta}}^{(i)} - \hat{\vec{\beta}})}{\hat{\sigma}^2(p+1)} \\
=&\frac{\left(\frac{-e_i}{1-h_{ii}}\right)(\vec{v}_i^\top X^\top X)^{-1}X^\top X\left(\frac{-e_i}{1-h_{ii}}\right)(X^\top X)^{-1}\vec{v}_i}{\hat{\sigma}^2(p+1)} \\
=&\frac{e_i^2}{\hat{\sigma}^2(1-h_{ii})^2}\cdot\frac{h_{ii}}{p+1}
\end{aligned}$$

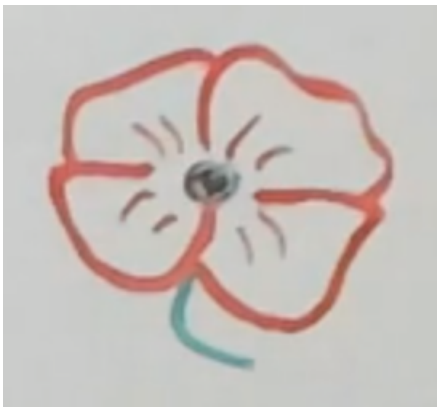
Recall the studentized residual is $d_i = \frac{e_i}{\hat{\sigma}\sqrt{1-h_{ii}}}$. Then,

$$D_i = d_i^2 \frac{h_{ii}}{1-h_{ii}} \cdot \frac{1}{p+1}$$

as claimed. ■

R We can calculate Cook's distance in terms of $|d_i|$ and h_{ii}. So, the most influential observation on estimators of $\vec{\beta}$ are those with high $|d_i|$ and h_{ii}.

Remembrance Day Pause 11.11



Solemn Poppy Drawn by Prof.Wong

In Flanders Fields
by John Mccrae

In Flanders fields the poppies blow
Between the crosses, row on row,
That mark our place; and in the sky
The larks, still bravely singing, fly
Scarce heard amid the guns below.

We are the Dead. Short days ago
We lived, felt dawn, saw sunset glow,
Loved and were loved, and now we lie,
In Flanders fields.

Take up our quarrel with the foe:
To you from failing hands we throw
The torch; be yours to hold it high.
If ye break faith with us who die
We shall not sleep, though poppies grow
In Flanders fields.

2.13.1 R Demo - Effect of Individual Observations

```
%## Effect of individual observations

## Python data revisited
python <- read.csv("FLpython.csv")
python$male <- ifelse(python$sex == 'M', 1, 0) # 1 = M, 0 =F
mpf2 <- lm(fat ~ male+mass+svl, data = python)

# Last time we used a Box-Cox transformation
library(MASS)
bc <- boxcox(mpf2)
lambda <- bc$x[which.max(bc$y)]
mpf3 <- lm( (fat^lambda-1)/lambda ~ male+mass+svl, data = python)
summary(mpf3)
plot(mpf3$fitted.values, mpf3$residuals)
qqnorm(mpf3$residuals)
qqline(mpf3$residuals, col="blue", lwd = 2)

# Quantities for individual observations
```

```
studres(mpf3)  # studentized residuals
hatvalues(mpf3) # leverage
cooks.distance(mpf3) # Cook's distance

par(mfrow = c(2,2))

# Residual plots with studentized residuals
plot(mpf3$fitted.values, studres(mpf3), xlab="Fitted values", ylab="
    Studentized residuals")
abline(h = c(3,-3), col = "red", lty=2)
which(abs(studres(mpf3)) > 3)
qqnorm(studres(mpf3))
qqline(studres(mpf3), col="blue", lwd = 2)

# Leverage
plot(hatvalues(mpf3), ylab="Leverage")
abline(h = 2*mean(hatvalues(mpf3)), col = "red", lty=2)
which(hatvalues(mpf3) > 2*mean(hatvalues(mpf3)))
python[which(hatvalues(mpf3) > 2*mean(hatvalues(mpf3))),]

# Cook's distance
plot(cooks.distance(mpf3),ylab="Cook's distance")
abline(h = 0.5, col = "red", lty=2)
which(cooks.distance(mpf3) > 0.5)

# Let's look at actual changes in beta estimates
mpf3$coefficients # with all the data
# e.g., fit without obs 248
mpf4 <- lm((fat^lambda-1)/lambda ~ male+mass+svl, data = python[-248,])
mpf4$coefficients
# e.g., fit without obs 50
mpf5 <- lm((fat^lambda-1)/lambda ~ male+mass+svl, data = python[-50,])
mpf5$coefficients
```

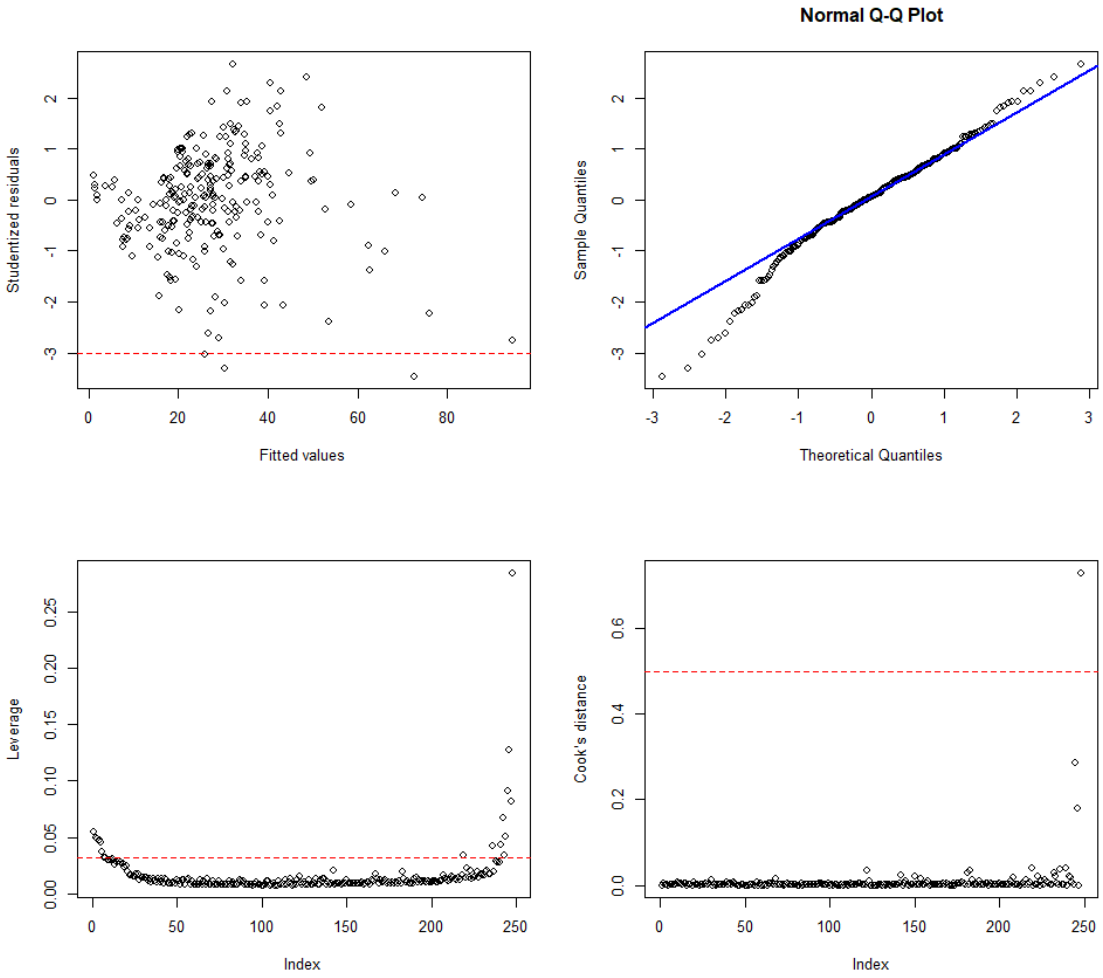


Figure 2.13.1: R Code Graph Output

2.14 Predictive Performance of MLR

Why do we study MLR? For fun? For the purpose of life? For becoming employed? No! For predictions on new data and also its interpretability!

Previously, we saw model selection criteria computed on the fitted models, including R^2_{adj}, **AIC**, **BIC**, which assess the explanatory power of a model on the data **used to fit the model (trained model)**. While these criteria incorporate penalty terms to try to prevent overfitting, they do not directly assess how well a model would perform in predicting the response on new data given the predictors.

We mentioned metrics such as *MSPE* as measures of predictive accuracy. To assess accuracy in prediction, we need metrics for measuring prediction error, such as evaluated over m observations.

Definition 2.14.1 — Mean-Squared Error (MSE).

$$\textbf{MSE} = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$$

where y_i is the actual value and $\hat{y}$ is the predicted value (or fitted value $\hat{\mu}_i$ if calculated on training data). We call **MSE** "MSPE" if applied on new data.

Definition 2.14.2 — Root-Mean-Squared Error (RMSE).

$$\textbf{RMSE} = \sqrt{\textbf{MSE}}$$

R **RMSE** is at the scale of observations.

Definition 2.14.3 — Mean Absolute Error.

$$\textbf{MAE} = \frac{1}{m}\sum_{i=1}^{m}|y_i - \hat{y}_i|$$

Ideally, we have lots of data, conceptualize having 3 parts:

Training Set	**Validation Set**	**Test Set**
n observations	v observations	t observations
$y_1, \cdots, y_n$	$y_{n+1}, \cdots, y_{n+v}$	$y_{n+v+1}, \cdots, y_{n+v+t}$
Fit models (can fit as many as we want)	Estimate prediction error for each fitted model	Used at very end for final assessment of our selected model

Figure 2.14.1: Train-Validation-Test

For example, we can use **MSE** as a metric and compute and compare:

1. **Training MSE**= $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

   Lemma 2.15

   $$\textbf{MSE} = \frac{\textbf{SS(Res)}}{n}$$

   We know that our usual estimate $\hat{\sigma}^2 = \frac{\textbf{SS(Res)}}{n-p-1}$ is a scaled version of this to compensate for the number of predictors.
2. **Validation MSE**= $\frac{1}{b}\sum_{i=n+1}^{n+v}(y_i - \hat{y}_i)^2$ This is considered as an estimate of $MSPE$ on new data
3. **Testing MSE**= $\frac{1}{t}\sum_{i=n+v+1}^{n+v+t}(y_i - \hat{y}_i)^2$ This is the actual test of prediction $MSPE$

R The idea here is that: **MSE** on validation set should approximate **MSE** on testing set since neither set of observation used to fit model.

■ **Example 2.11 — oVeRFItInG?!.** So if we are using MSE/RMSE as metric (as related to $\hat{\sigma}^2, \hat{\sigma}$) and it is significantly larger on validation compared to training set. We probably overfitted and can't expect model to generalize well to new data.

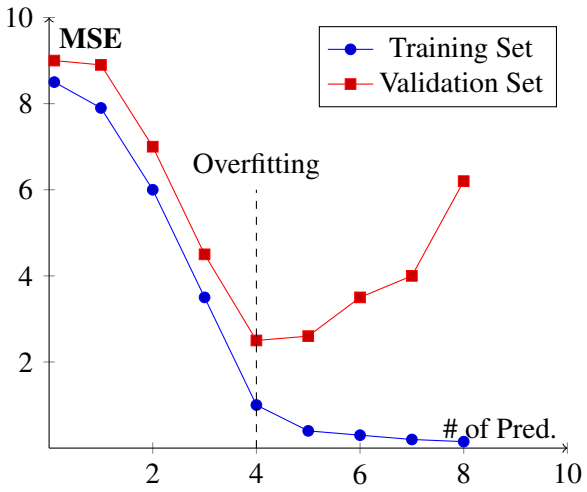


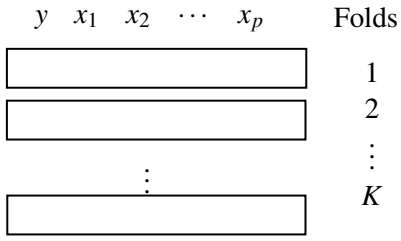
Figure 2.15.1: Error metric vs. # of predictors on the 3-set

■

Practical Applications

1. **Simplest:** randomly divide available data into train/validation, such as 80/20 split for training/validation. It has the following **weaknesses:**
   (a) we do not use all data for training
   (b) we only get estimate of prediction error
2. **Better:** use cross-validation scheme (CV)

How to do CV with K folds?



For example, using **RMSE**, we have $\mathbf{RMSE}_1, \cdots, \mathbf{RMSE}_K$. We can take the average

$$\frac{1}{K}\sum_{k=1}^{K}\mathbf{RMSE}_k$$

as an estimate of $RMSPE$.

1. Divide available data for training and validation into K roughly equal-sized sets (folds) randomly
2. For CV fold k, use data in fold k as validation and train on the rest of data.

Thus, to estimate prediction error for a given model, we fit it K times, each time treating data in fold $1, 2, \cdots, K$ as validation. Thus, we have K estimates of prediction error.
Say we are given some candidate models (e.g. with different variables included). We can do $K-$fold CV using each of them, and choose the one with lowest average predictor error across folds.

2.15.1 R Demo - Cross-Validation

```
%## Cross-validation

## Coffee example  (Coffee Quality Institute, 2018) continued
coffee <- read.csv("coffee_arabica.csv")
coffee$wet <- ifelse(coffee$Processing.Method == 'Washed / Wet', 1, 0) # 1
    = wet, 0 otherwise
coffee$semi <- ifelse(coffee$Processing.Method == 'Semi-washed / Semi-
    pulped', 1, 0) # 1 = semi/dry, 0 otherwise
coffee$Processing.Method <- NULL
```

```

N <- nrow(coffee)

## Train and validation set split
set.seed(12345678)
trainInd <- sample(1:N, round(N*0.8), replace=F)
trainSet <- coffee[trainInd,]
validSet <- coffee[-trainInd,]

# Calculate RMSE on three models each with different variables included
m1 <- lm(Flavor ~ wet + semi + Aroma + Aftertaste + Body, dat=trainSet)
pred1 <- predict(m1, newdata = validSet)
sqrt(mean((validSet$Flavor - pred1)^2)) # RMSE
mean(abs(validSet$Flavor - pred1)) # MAE

m2 <- lm(Flavor~ wet + Aroma + Aftertaste +
              Body + Acidity + Balance + Sweetness + Uniformity + Moisture,
    dat=trainSet)
pred2 <- predict(m2, newdata = validSet)
sqrt(mean((validSet$Flavor - pred2)^2))

m3 <- lm(Flavor ~ Aroma + Aftertaste, dat=trainSet)
pred3 <- predict(m3, newdata = validSet)
sqrt(mean((validSet$Flavor - pred3)^2))

# K fold cross validation
K <- 5
validSetSplits <- sample((1:N) \%\% K + 1)
RMSE1 <- c()
RMSE2 <- c()
RMSE3 <- c()
for (k in 1:K) {
  validSet <- coffee[validSetSplits==k,]
  trainSet <- coffee[validSetSplits!=k,]

  m1 <- lm(Flavor ~ wet + semi + Aroma + Aftertaste + Body, dat=trainSet)
  pred1 <- predict(m1, newdata = validSet)
  RMSE1[k] <- sqrt(mean((validSet$Flavor - pred1)^2))

  m2 <- lm(Flavor~ wet + Aroma + Aftertaste +
             Body + Acidity + Balance + Sweetness + Uniformity + Moisture,
    dat=trainSet)
  pred2 <- predict(m2, newdata = validSet)
  RMSE2[k] <- sqrt(mean((validSet$Flavor - pred2)^2))

  m3 <- lm(Flavor ~ Aroma + Aftertaste, dat=trainSet)
  pred3 <- predict(m3, newdata = validSet)
  RMSE3[k] <- sqrt(mean((validSet$Flavor - pred3)^2))
}

RMSE1
RMSE2
RMSE3
mean(RMSE1)
mean(RMSE2)
mean(RMSE3)
```

2.16 More on K-Fold CV

How to choose K?

Some common choices are:

1. $K = 5$
2. $K = 10$
3. $K = N$ where N is the number of observations for training/validation set where the validation set has only one observation. We also call this **Leave-one-out (LOO) fold**.

As we are increasing the number of folds, the computational time will be higher. This can also lead to better estimation of the prediction error.

What if we are given a list of models to compare?

Or if there are too many predictors to do all possible regression, then we can combine CV with model selection procedure along with certain criteria on the training data and search strategy.
To estimate prediction error a given model selection procedure. We apply it K times, each time treating the observations in fold $1, 2, \cdots, K$ as validation set, e.g, use $\frac{1}{K}\sum_{i=1}^{K} \mathbf{RMSE}_i$ as an estimate, and choose the model selection procedure with the lowest estimated prediction error.

Actual variables selected in each fold might be different. The recommendation is to apply the chosen model selection procedure (with the lowest estimated prediction error) to the entire prediction error) to the entire set of observations available for training and validation to get a final model (for applications to test data).

Beyond AIC and BIC

Recall that

1. $\mathbf{AIC} = 2q - 2\ln L(\hat{\theta})$
2. $\mathbf{BIC} = q\ln(n) - 2\ln L(\hat{\theta})$

We can also do something like

Definition 2.16.1 — L_0-penalized Likelihood.

$$q\lambda - 2\ln L(\hat{\theta})$$

$\lambda > 0$

We can try different values of λ to construct different model selection procedures.

Nice! Maybe it's time to use Bayesian optimization for hyperparameters!

Beyond Stepwise Regression

Recall that stepwise is a deterministic algorithm, which is fast but might not give optimal model selected. We could try stochastic algorithms.

Definition 2.16.2 — Iterative Conditional Minimization (ICM). Start with model with just the intercept. For a given random seed, take predictors $x_1, \cdots, x_p$ and randomly re-order them into $x_{(1)}, \cdots, x_{(p)}$.

```
%x ## set of predictors
S = [] ## S is the set of predictors currently in the model
metric_0 ## initial empty model metric (suppose higher the better for
    now)
x = shuffle(x) ## depends on seed
while (not_converge):
    for j in 1:p:
        if x[j] not in S:
            curr_model = lm(S + x[j])
            if curr_model.metric > metric_0:
                S.append(x[j])
```

```
                metric_0 = curr_model.metric
        else:
            curr_model = lm(S - x[j])
            if curr_model.metric > metric_0:
                S.del(x[j])
                metric_0 = curr_model.metric
```

R Note that different random orderings could give different sets S at end of procedure. Could pick one that has best criterion overall.

Beyond Meat

...

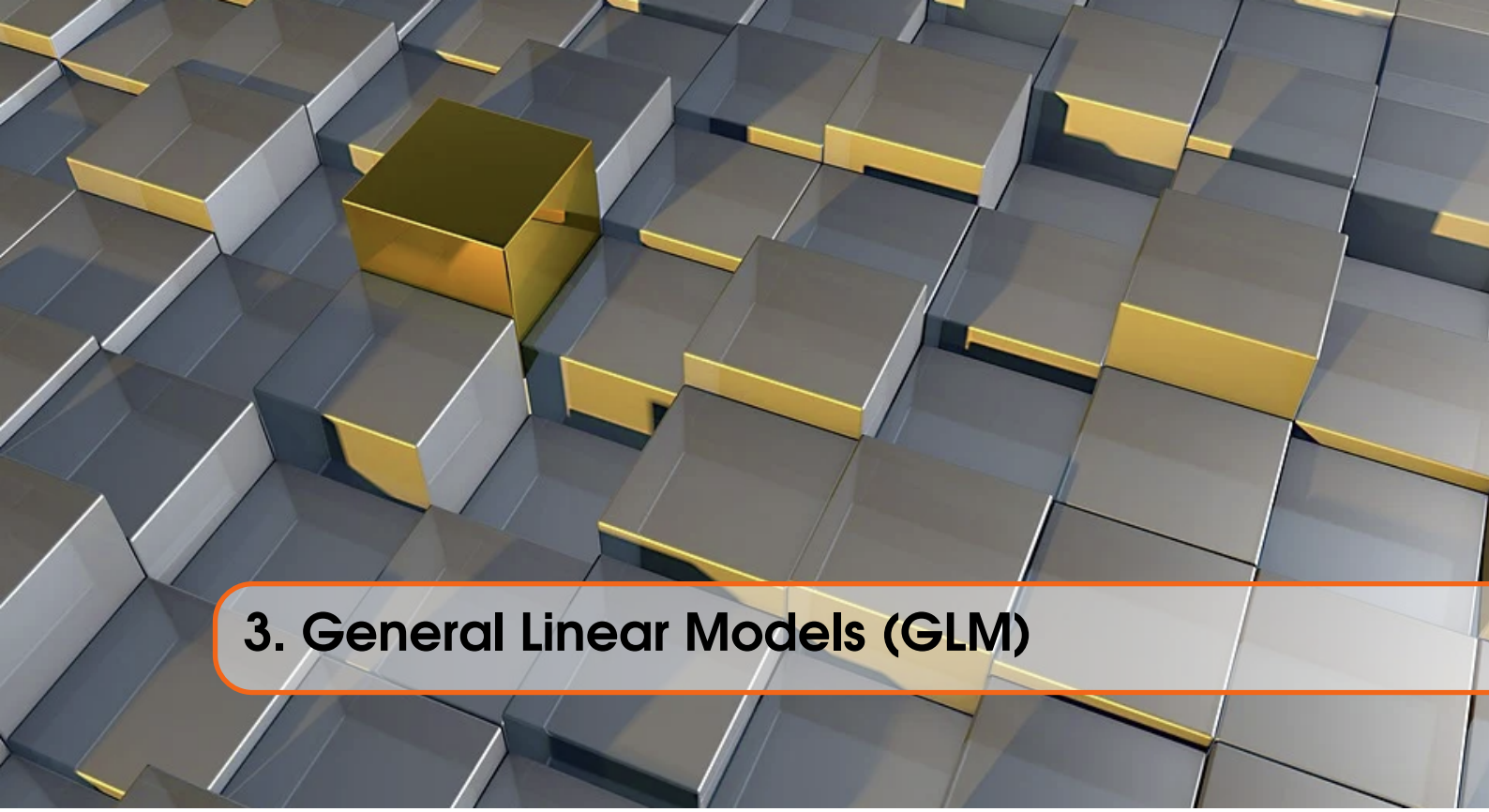
2.16.1 R Demo - Beyond Model Selection

```
%## Cross-validation with model selection

# Dataset from paper "Where does Haydn end and Mozart begin?
# Composer classification of string quartets"
# (Journal of New Music Research, vol 49, 457-476)
HM <- read.csv("haydn-mozart.csv")
HM[,1] <- NULL # first col is just name of quartet, remove it
dim(HM) # 285 observations, 1116 columns

# Let's treat "number of notes in violin part" as the response
# That's variable name "count_pitch_1" and column 683 of data matrix
# So we have 1115 possible predictors

# More model selection, for clarity start with one train/validation split
N <- nrow(HM)
set.seed(12345678)
trainInd <- sample(1:N, round(N*0.8), replace=F)
trainSet <- HM[trainInd,]
validSet <- HM[-trainInd,]

library(MASS)
# Full model and empty model with just intercept
full <- lm(count_pitch_1 ~ ., data = trainSet)
empty <- lm(count_pitch_1 ~ 1, data =trainSet)

# Stepwise forward with BIC
stepAIC(object = empty, scope = list(upper = full, lower = empty),
    direction = "forward", k = log(nrow(trainSet)))
m1 <- lm(formula = count_pitch_1 ~ Prop_m3_num_0_8.1 + count_pitch_3 +
           Prop_m3_num_0_8.3 + Prop_m3_mean_8.1 + count_pitch_4 +
    Dev_count_8_thresh4.393.1 +
           Prop_m3_mean_8.3 + mean_time_3 + Dev_count_t_14_thresh0.216.3 +
           voicepair_int_dist_6_1.2 + Prop_m3_sd_8.3 + Prop_m3_sd_8.1 +
           Prop_m3_num_.6_8.1 + simult_rest_perc + Dev_perc_t_14.1 +
           count_pitch_2 + Prop_m3_num_0_8.2 + Prop_m3_mean_8.2 +
    Dev_count_8_thresh4.024.2 +
           Dev_count_18_thresh3.899.2 + Expo_t_count_14.thresh0.7.1 +
           Prop_m3_num_0_12.1 + Expo_acc_8.1 + Expo_perc_8.2 +
    Prop_m3_num_0_12.2 +
           Dev_count_t_8_thresh0.247.1 + Expo_t_count_8.thresh0.7.1 +
           voicepair_int_dist_1_1.3 + Expo_perc_12.3 +
    Pairwise_voice_int_mean.1.3 +
           Expo_t_count_18.thresh0.7.3 + Prop_m3_q3_16.2 + Dev_perc_t_14.4
    +
```

```
            Prop_m3_q3_14.4 + Dev_count_t_14_thresh0.187.3, data = trainSet)

# we can use the AIC function with our own k for the L0 penalty
AIC(m1, k = log(nrow(trainSet)))
BIC(m1) # in this case matches BIC as we expect (1977.6)

pred1 <- predict(m1, newdata = validSet)
sqrt(mean((validSet$count_pitch_1 - pred1)^2)) # RMSE on validation
sqrt(mean(m1$residuals^2)) # RMSE on train

# Try ICM to search for a model with a potentially better BIC
# than the one found with stepwise
pen <- log(nrow(trainSet)) #
varlist = c()
varnames = names(trainSet)
n = nrow(trainSet)
varorder <- sample(1:ncol(trainSet)) # random order of variables
minCrit = Inf
noChange = F
while (!noChange) {
  noChange = T
  for (i in varorder) {
    if (i == 683)
      next

    if (i %in% varlist & length(varlist) > 1) {
      index = c(683, varlist[varlist != i])
      trainVars = trainSet[, index]

      fit = lm(count_pitch_1 ~ ., data = trainVars)

      if (AIC(fit, k = pen) < minCrit) {
        minCrit = AIC(fit, k = pen)
        varlist = varlist[varlist != i]
        print(paste0("Criterion: ", round(minCrit, 1), ", variables: ",
    paste0(varnames[varlist], collapse = " ")))
        best.model = fit
        noChange = F
      }

    } else if (!i %in% varlist) {
      index = c(683, varlist, i)
      trainVars = trainSet[, index]

      fit = lm(count_pitch_1 ~ ., data = trainVars)

      if (AIC(fit, k = pen) < minCrit) {
        minCrit = AIC(fit, k = pen)
        varlist = c(varlist, i)
        print(paste0("Criterion: ", round(minCrit, 1), ", variables: ",
    paste0(varnames[varlist], collapse = " ")))
        best.model = fit
        noChange = F
      }
    }
  }
}

summary(best.model)
predICM <- predict(best.model, newdata = validSet)
sqrt(mean((validSet$count_pitch_1 - predICM)^2)) # RMSE on validation
```

```
sqrt(mean(best.model$residuals^2)) # RMSE on train


# Try stepwise again, with a larger L0 penalty (e.g., twice the usual BIC penalty)
stepAIC(object = empty, scope = list(upper = full, lower = empty), direction = "forward", k = 2*log(nrow(trainSet)))
m2 <- lm(formula = count_pitch_1 ~ Prop_m3_num_0_8.1 + count_pitch_3 +
           Prop_m3_num_0_8.3 + Prop_m3_mean_8.1 + count_pitch_4 + Dev_count_8_thresh4.393.1 +
           Prop_m3_mean_8.3 + mean_time_3 + Dev_count_t_14_thresh0.216.3,
         data = trainSet)
AIC(m2, k = 2*log(nrow(trainSet)))
# calculate the value of criterion based on this larger L0 penalty
pred2 <- predict(m2, newdata = validSet)
sqrt(mean((validSet$count_pitch_1 - pred2)^2)) # RMSE on validation
sqrt(mean(m2$residuals^2)) # RMSE on train

# Try ICM as well with this penalty
pen <- 2*log(nrow(trainSet))
varlist = c()
varnames = names(trainSet)
n = nrow(trainSet)
varorder <- sample(1:ncol(trainSet))
minCrit = Inf
noChange = F
while (!noChange) {
  noChange = T
  for (i in varorder) {
    if (i == 683)
      next

    if (i %in% varlist & length(varlist) > 1) {
      index = c(683, varlist[varlist != i])
      trainVars = trainSet[, index]

      fit = lm(count_pitch_1 ~ ., data = trainVars)

      if (AIC(fit, k = pen) < minCrit) {
        minCrit = AIC(fit, k = pen)
        varlist = varlist[varlist != i]
        print(paste0("Criterion: ", round(minCrit, 1), ", variables: ", paste0(varnames[varlist], collapse = " ")))
        best.model = fit
        noChange = F
      }

    } else if (!i %in% varlist) {
      index = c(683, varlist, i)
      trainVars = trainSet[, index]

      fit = lm(count_pitch_1 ~ ., data = trainVars)

      if (AIC(fit, k = pen) < minCrit) {
        minCrit = AIC(fit, k = pen)
        varlist = c(varlist, i)
        print(paste0("Criterion: ", round(minCrit, 1), ", variables: ", paste0(varnames[varlist], collapse = " ")))
        best.model = fit
        noChange = F
      }
```

```
154     }
155   }
156 }
157
158 predICM <- predict(best.model, newdata = validSet)
159 sqrt(mean((validSet$count_pitch_1 - predICM)^2)) # RMSE on validation
160 sqrt(mean(best.model$residuals^2)) # RMSE on train
161
162
163 # K fold cross validation to choose model selection method
164 K <- 5
165 validSetSplits <- sample((1:N), K + 1)
166 RMSE1 <- c()
167 RMSE2 <- c()
168 for (k in 1:K) {
169   validSet <- HM[validSetSplits==k,]
170   trainSet <- HM[validSetSplits!=k,]
171
172   full <- lm(count_pitch_1 ~ ., data = trainSet)
173   empty <- lm(count_pitch_1 ~ 1, data = trainSet)
174
175   m1 <- stepAIC(object = empty, scope = list(upper = full, lower = empty),
176                 direction = "forward", k = log(nrow(trainSet)))
177   pred1 <- predict(m1, newdata = validSet)
178   RMSE1[k] <- sqrt(mean((validSet$count_pitch_1 - pred1)^2))
179
180   m2 <- stepAIC(object = empty, scope = list(upper = full, lower = empty),
181                 direction = "forward", k = 2 * log(nrow(trainSet)))
182   pred2 <- predict(m2, newdata = validSet)
183   RMSE2[k] <- sqrt(mean((validSet$count_pitch_1 - pred2)^2))
184
185 }
186
187 RMSE1
188 RMSE2
189 mean(RMSE1)
190 mean(RMSE2)
191
192 # turns out m2 is indeed the better procedure among these two based on CV
      prediction error
193 # if we decide on procedure m2, we can apply procedure m2 to the entire 285
      observations
194 # to get a final model for future prediction
195 # e.g.,
196 full <- lm(count_pitch_1 ~ ., data = HM)
197 empty <- lm(count_pitch_1 ~ 1, data = HM)
198
199 mfinal <- stepAIC(object = empty, scope = list(upper = full, lower = empty)
      ,
200                   direction = "forward", k = 2 * log(nrow(trainSet)))
```



3. General Linear Models (GLM)

3.1 Introduction

Similar to MLR, we still have independent responses $\vec{Y}$ and predictors $\vec{x}_1, \cdots, \vec{x}_p$.

Definition 3.1.1 — General Linear Model. Three ingredient of a GLM:

1. **Random Component:** response Y_i is treated as a random variable with a distribution that is a member of an exponential family. Common examples include Normal, Binomiaol, and Poisson
2. **Systematic Component:** typically a linear predictor based on $x_{i1}, \cdots, x_{ip}$: we denote
$$\eta_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$$
3. **Link Function:** a function $g(\cdot)$ that defines a relationship between $\mathbb{E}(Y_i)$, from the random component, and η_i, from the systematic component. For example, $\eta_i = g(\mathbb{E}(Y_i))$

■ **Example 3.1 — MLR.** Recall that MLR is of the form
$$Y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i, \varepsilon_i \overset{\text{iid}}{\sim} N(0, \sigma^2)$$
Note that $\mathbb{E}(Y_i) = \eta_i$. So, the link function is $g(\mathbb{E}(Y_i)) = \mathbb{E}(Y_i)$, the identity function. ■

■ **Example 3.2 — Logistic Regression.** Logistic regression is commonly used in binary classification. This means
$$Y_i = \begin{cases} 1 & \text{success/on/yes/goose} \\ 0 & \text{failure/off/no/non-goose} \end{cases}$$

1. **Random component:** define $\mathbb{P}(Y_i = 1) = \pi_i$ such that Y_i is binomial with 1 trial success probability of π_i. Correspondingly, $\mathbb{P}(Y_i = 0) = 1 - \pi_i$.
2. **Link function:**
$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} = \eta_i$$
Moreover, note that
$$\mathbb{E}(Y_i) = 1\pi_i + 0(1-\pi_i) = \pi_i$$

so, logistic regression takes $g(\pi_i) = \log\left(\frac{\pi_i}{1-\pi_i}\right)$ **(logit link)**

Say A is an event, then the odds of event A is defined as

$$\frac{\mathbb{P}(A)}{1-\mathbb{P}(A)}$$

then, $\frac{\pi_i}{1-\pi_i}$ is the odds that $Y_i = 1$ and $\log\left(\frac{\pi_i}{1-\pi_i}\right)$ is the log-odds that $Y_i = 1$. Since $\eta_i = \log\left(\frac{\pi_i}{1-\pi_i}\right)$, we can take the inverse to get

$$\pi_i = \frac{e^{\eta_i}}{1+e^{\eta_i}}$$

let's plot this function **(Sigmoid Function)**

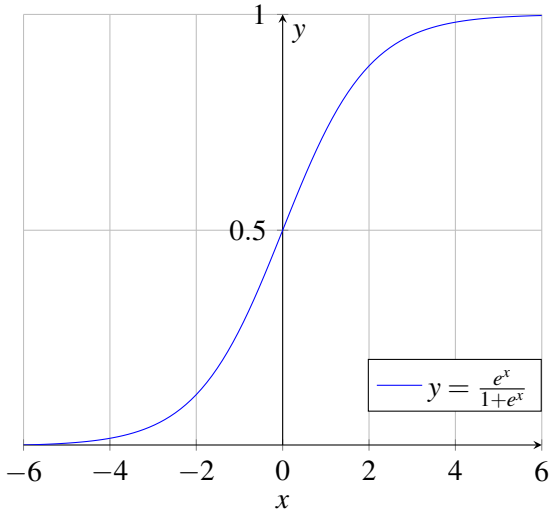


Figure 3.1.1: Sigmoid Function

Notice that π_i is bounded between 0 and 1 while $\log\left(\frac{\pi_i}{1-\pi_i}\right)$ can take any real number. Since

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$$

then β_j's **interpretation** is the expected change in the log-odds of the event that $Y_i = 1$ for a unit increase in the variable x_j while holding other variables constant. And,

$$\frac{\pi_i}{1-\pi_i} = \exp(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip})$$

the e^{β_j} is the expected multiplicative change in odds of $Y_i = 1$ for a unit increase in x_j while holding other variables constant. ∎

How to fit a logistic regression?

We still need MLE to fit these parameters. In `R`, we can use `glm()` function to perform this.

∎ **Example 3.3 — Numerical Example - String Quartet Classification.** Let

$$Y_i = \begin{cases} 1 & \text{Haydn} \\ 0 & \text{Mozart} \end{cases}$$

the model is

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}$$

(just to illustration, the actual model had 7 predictors) From training data, we have the following estimates:

$$\hat{\vec{\beta}} = \begin{bmatrix} -1.12 \\ -15.47 \\ 16.71 \end{bmatrix}$$

where

1. x_1 is the standard deviation of note duration in 1st violin. The range of value is from 0.05 to 0.35.
2. x_2 is the proportion of descending pairwise intervals in 1st violin. The range of value is from 0.2 to 0.5.

Then, higher values of x_1 are more likeli to be Morzart while higher values of x_2 are more likely to be Haydn.

1. Prof.Wong: Okay I digressed a little bit into music literacy
2. LinglingWannabees: MORE!

x_1	0.05	0.2	0.35
x_2	0.2	0.4	0.4
$\hat{\eta}_i$	4.79		
$\hat{\pi}_i$	0.992	0.922	0.537

Table 3.1.1: Estimate Summary

For example, 0.992 is the estimated probability of an observation to be Haydn given $x_1 = 0.05$ and $x_2 = 0.4$. Some more interpretation of $\hat{\beta}_j$:

1. For a unit increase in SD of the note duration in 1st violin, estimated expected change in log-odds of the piece being Haydn is -15.47.

■

3.2 La Dernière Classe

3.2.1 Project Tips/Q&A

Some pro tips:

1. **Explore the dataset:** For example, histograms of some predictors or correlation plots.
2. **No one "right" model:** you can have an amazing MLR model that could do better than some amateur, like myself, fitting a neural network. Try different approaches for sure.
3. **Understand what you are doing:** this should be elaborated in details within the report.

After living in this world for 22 years, I still don't understand what I am doing especially when it comes to doing machine learning type of work.

4. **Present it well:** if you have the following type of code chunk or output in your report, you are probably stabbing yourself from the back [very acrobatic].

```
%    really_hard_r_code <- lm(iam ~ very + hard + to + understand)
    pvalue F anova vif()
    ERROR: STAT331 FAIL
    WARNING: STAT331 Project raw R code

```

The report is intended to be read by well-educated geese that cannot code in R.

5. **Dataset has many variables:** try all possible model search. Do it for the memes. But for sure don't wait till last minute. This many predictors will make your model prone to overfitting. Be cautious about it and do something about it.

3.2.2 Residual (Sur)realism

We want $(\hat{\mu}_i, e_i), i = 1, \cdots, n$ to represent black pixel locations in the image.

R This is like a reverse-engineering problem since most of the time we have x and y.

Recall from fitting a MLR that $\vec{e}$ and $\hat{\vec{\mu}}$ satisfy

$$\begin{cases} \hat{\vec{\mu}} = H\vec{Y} \\ \vec{e} = (I-H)\vec{Y} \end{cases}$$

which is, in fact, a linear system of equations with $2n$ equations. So, we need $\underbrace{\vec{Y}, \vec{x}_1, \cdots, \vec{x}_p}_{n\times(p+1) \text{ free variables}}$.

Due to this generous amount of degrees of freedom, we can fix $\vec{\beta}$, therefore, choosing our regression coefficients.

Recall that $\vec{Y} = X\vec{\beta} + \vec{\varepsilon}$. Thus, the equations can be expressed as

$$\begin{cases} \hat{\vec{\mu}} = H(X\vec{\beta} + \vec{\varepsilon}) = X\vec{\beta} + H\vec{\varepsilon} \\ \vec{e} = (I-H)\vec{\varepsilon} = \vec{\varepsilon} - H\vec{\varepsilon} \end{cases}$$

we need residuals satisfy the condition that $\hat{\vec{\mu}}^\top \vec{e} = 0$ since $\vec{e}$ are orthogonal to $\text{span}(X)$, which might not be satisfied by an arbitrary image. Say we do not have the orthogonality, then we can add some pixels. Guess what? We can add some really influential pixels to the corners! Yeah, you hear that right, we are adding outliers so that we can have the desired orthogonality.

Recall that the LS solution implies

$$\begin{cases} \vec{1}^\top \vec{e} = 0 \\ \tilde{X}^\top \vec{e} = 0 \end{cases}$$

where $X = [\vec{1}, \tilde{X}]$. So, we let

$$\tilde{X} = \left(I_n - \frac{\vec{e}\vec{e}^\top}{\vec{e}^\top \vec{e}} \right) M_{n\times p}$$

so that $\tilde{X}^\top \vec{e} = 0$ is satisfied for any $M_{n\times p}$.

Now, we simulate $\vec{Z}$ iid from $N(0, \tau^2)$ where τ controls the R^2 in MLR and let

$$\vec{\varepsilon} = \vec{e} + H\vec{Z}$$

and substitute in the 1st equation to get

$$\hat{\vec{\mu}} = \underbrace{X\vec{\beta}}_{\sim M_{n\times p}} + \underbrace{H\vec{e}}_{=0} + \underbrace{H\vec{Z}}_{\sim M_{n\times p}}$$

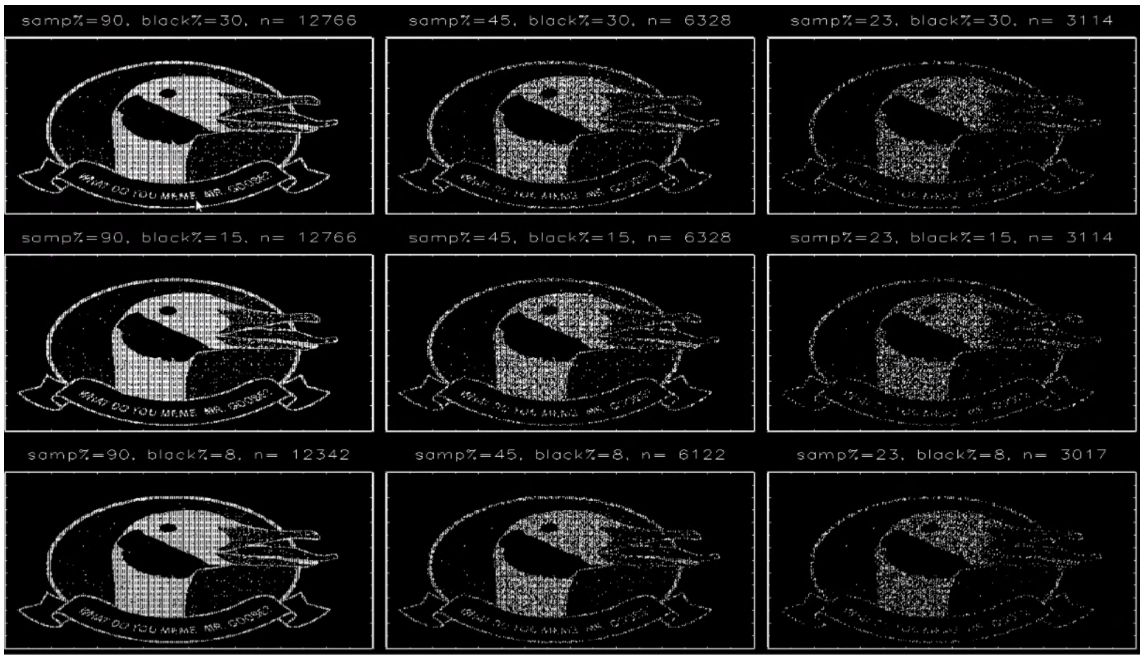
Finally, we use iterative method to update $M_{n\times p}$ until is satisfies the equation.

I think the program released by the author only has the easiest version to use. It's on Windows.

Linux Supremacy, Prof.Wong

3.2.3 R Demo - Residual (Sur)realism

⇓



Autumn (piano 4 hands), E. Grieg
For future reference

END OF STAT331 Course Note

Thank you Mr.Goose and Prof.Wong!
Good luck on the project and wish everyone gets the same RMSPE!
