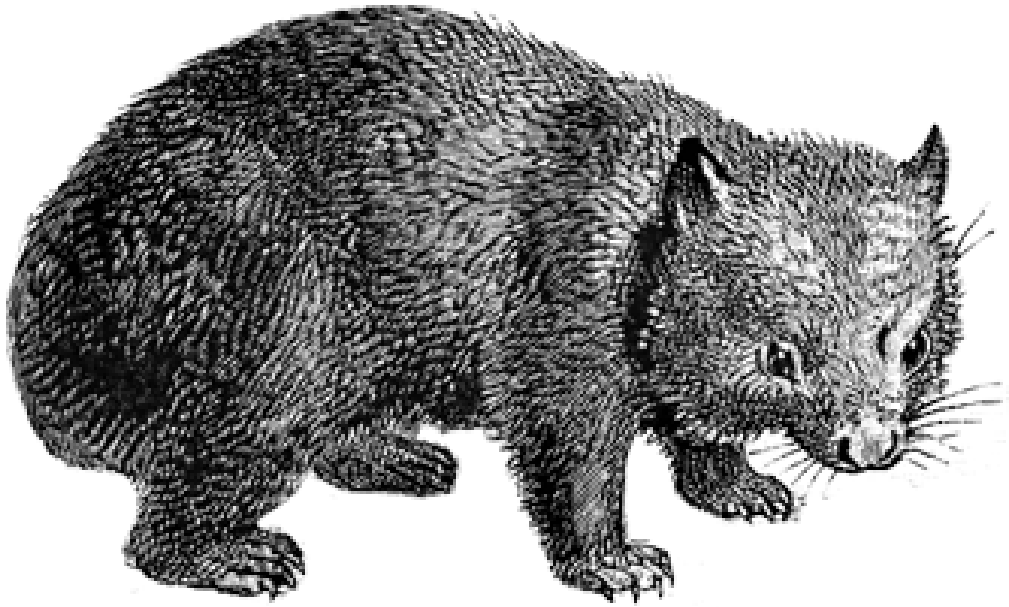
YOUNG-GEUN statistics



Regression Analysis

R Lab

O RLY?

Young-geun Kim

R Lab for Regression Analysis

Young-geun Kim
Department of Statistics, SKKU
dudrms33@g.skku.edu

22 Sep, 2019

Contents

Welcome

This book aims at covering materials of regression analysis. Also, there will be R programming for regression.

```
library(tidyverse)
```

`tidyverse` package will be used in every chapter, so loading step will be hidden.

Linear Regression Analysis

```
data(BioOxyDemand, package = "MPV")
(BioOxyDemand <-
  BioOxyDemand %>%
  tbl_df())
#> # A tibble: 14 x 2
#>        x     y
#>    <int> <int>
#> 1      3     4
#> 2      8     7
#> 3     10     8
#> 4     11     8
#> 5     13    10
#> 6     16    11
#> # ... with 8 more rows
```

Relation

We wonder how `x` affects `y`, especially linearly.

- Functional relation: mathematical equation,

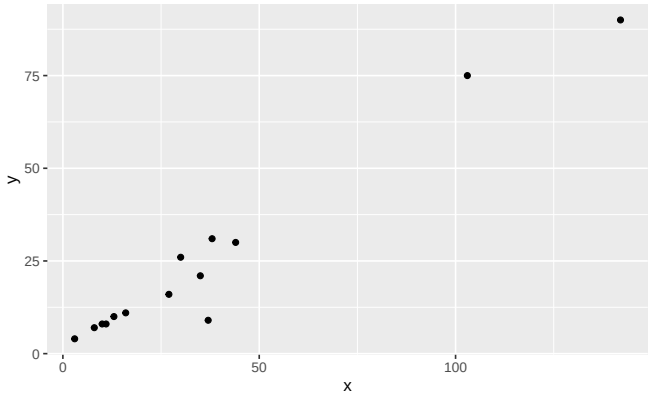
$$y = \beta_0 + \beta_1 x$$

- Statistical relation: embeded with noise

So we try to estimate

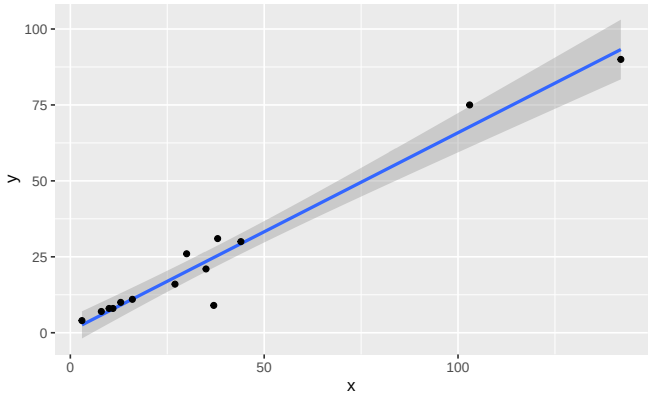
$$y = \beta_0 + \beta_1 x + \epsilon$$

```
BioOxyDemand %>%
  ggplot(aes(x, y)) +
  geom_point()
```



Looking just with the eyes, we can see the linear relationship. Regression analysis estimates the relationship statistically.

```
BioOxyDemand %>%
  ggplot(aes(x, y)) +
  geom_smooth(method = "lm") +
  geom_point()
```



Chapter 1

Simple Linear Regression

1.1 Model

```
delv <- MPV::p2.9 %>% tbl_df()
```

```
delv %>%
  ggplot(aes(x = x, y = y)) +
  geom_point() +
  labs(
    x = "Number of Cases",
    y = "Delivery Time"
  )
```

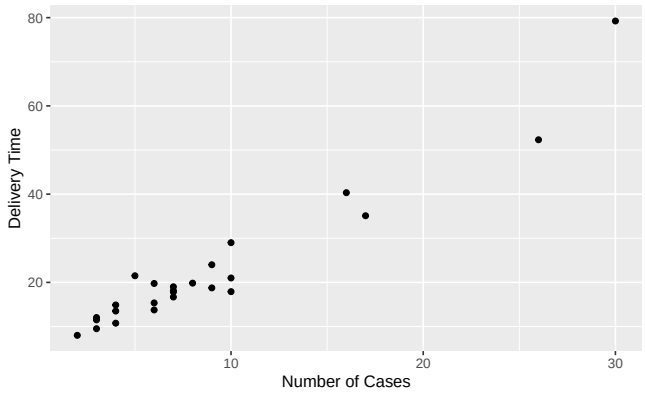


Figure 1.1: The Delivery Time Data

Given data $(x_1, Y_1), \ldots, (x_n, Y_n)$, we try to fit linear model

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

Here ϵ_i is a error term, which is a random variable.

$$\epsilon \overset{iid}{\sim} (0, \sigma^2)$$

It gives the problem of estimating three parameters $(\beta_0, \beta_1, \sigma^2)$. Before estimating these, we set some assumptions.

1. linear relationship
2. ϵ_is are independent
3. ϵ_is are identically destributed, i.e. *constant variance*
4. In some setting, $\epsilon_i \sim N$

1.2 Least Squares Estimation

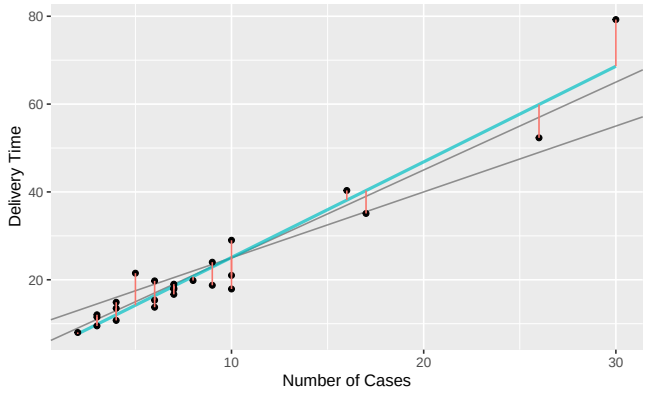


Figure 1.2: Idea of the least square estimation

We try to find β_0 and β_1 that minimize the sum of squares of the vertical distances, i.e.

$$(\beta_0, \beta_1) = \arg\min \sum_{i=1}^{n} (Y_i - \beta_0 - \beta_1 x_i)^2 \tag{1.1}$$

1.2.1 Normal equations

Denote that Equation (1.1) is quadratic. Then we can find its minimum by find the zero point of the first derivative. Set

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

Then we have

$$\frac{\partial Q}{\partial \beta_0} = -2 \sum_{i=1}^{n} (Y_i - \beta_0 - \beta_1 x_i) = 0 \tag{1.2}$$

and

$$\frac{\partial Q}{\partial \beta_1} = -2 \sum_{i=1}^{n} (Y_i - \beta_0 - \beta_1 x_i) x_i = 0 \tag{1.3}$$

From Equation (1.2),

$$\sum_{i=1}^{n} Y_i - n\hat{\beta}_0 - \hat{\beta}_1 \sum_{i=1}^{n} x_i = 0$$

Thus,

$$\hat{\beta}_0 = \overline{Y} - \hat{\beta}_1 \overline{x}$$

Equation (1.3) gives

$$\sum_{i=1}^{n} x_i (Y_i - \overline{Y} + \hat{\beta}_1 \overline{x} - \hat{\beta}_1 x_i) = \sum_{i=1}^{n} x_i (Y_i - \overline{Y}) - \hat{\beta}_1 \sum_{i=1}^{n} x_i (x_i - \overline{x}) = 0$$

Thus,

$$\hat{\beta}_1 = \frac{\sum\limits_{i=1}^{n} x_i (Y_i - \overline{Y})}{\sum\limits_{i=1}^{n} x_i (x_i - \overline{x})}$$

Remark.

$$\hat{\beta}_1 = \frac{S_{XY}}{S_{XX}}$$

where $S_{XX} := \sum\limits_{i=1}^{n} (x_i - \overline{x})^2$ and $S_{XY} := \sum\limits_{i=1}^{n} (x_i - \overline{x})(Y_i - \overline{Y})$

Proof. Note that $\overline{x}^2 = \frac{1}{n^2}\left(\sum_{i=1}^{n} x_i\right)^2$. Then we have

$$\begin{aligned} S_{XX} &= \sum_{i=1}^{n}(x_i - \overline{x})^2 \\ &= \sum_{i=1}^{n} x_i^2 - 2\sum_{i=1}^{n} x_i\overline{x} + \sum_{i=1}^{n}\overline{x}^2 \\ &= \sum_{i=1}^{n} x_i^2 - \frac{2}{n}\left(\sum_{i=1}^{n} x_i\right)^2 + \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2 \\ &= \sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2 \end{aligned} \tag{1.4}$$

It follows that

$$\begin{aligned} \hat{\beta}_1 &= \frac{\sum x_i(Y_i - \overline{Y})}{\sum x_i(x_i - \overline{x})} \\ &= \frac{\sum x_i(Y_i - \overline{Y}) - \overline{x}\sum(Y_i - \overline{Y})}{\sum x_i^2 - \frac{1}{n}(\sum x_i)^2} \qquad \because \sum(Y_i - \overline{Y}) = 0 \\ &= \frac{\sum(x_i - \overline{x})(Y_i - \overline{Y})}{\sum x_i^2 - \frac{1}{n}(\sum x_i)^2} \\ &= \frac{S_{XY}}{S_{XX}} \end{aligned}$$

□

```
lm(y ~ x, data = delv)
#>
#> Call:
#> lm(formula = y ~ x, data = delv)
#>
#> Coefficients:
#> (Intercept)            x
#>        3.32         2.18
```

1.2.2 Prediction and Mean response

"Essentially, all models are wrong, but some are useful."

—George Box

Recall that we have assumed the **linear assumption** between the predictor and the response variables, i.e. the true model. Estimating β_0 and β_1 is same as estimating the *assumed true model*.

Definition 1.1 (Mean response)**.**

$$E(Y \mid X = x) = \beta_0 + \beta_1 x$$

We can estimate this mean resonse by

$$\widehat{E(Y \mid x)} = \hat{\beta}_0 + \hat{\beta}_1 x \tag{1.5}$$

However, in practice, the model might not be true, which is included in ϵ term.

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

Our real problem is predicting individual Y, not the mean. The *prediction* of response can be done by

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i \tag{1.6}$$

Observe that the values of Equations (1.5) and (1.6) are same. However, due to the **error term in the prediction**, it has larger standard error.

1.2.3 Properties of LSE

Parameters β_0 and β_1 have some properties related to the expectation and variance. We can notice that these lse's are **unbiased linear estimator**. In fact, these are the *best unbiased linear estimator*. This will be covered in the Gauss-Markov theorem.

Lemma 1.1.

$$S_{XX} = \sum_{i=1}^{n} x_i^2 - \frac{1}{n}\Big(\sum_{i=1}^{n} x_i\Big)^2 = \sum_{i=1}^{n} x_i(x_i - \overline{x})$$

$$S_{XY} = \sum_{i=1}^{n} x_i Y_i - \frac{1}{n}\Big(\sum_{i=1}^{n} x_i\Big)\Big(\sum_{i=1}^{n} Y_i\Big) = \sum_{i=1}^{n} Y_i(x_i - \overline{x}) = \sum_{i=1}^{n} x_i(Y_i - \overline{Y})$$

Proof. We already proven the first part of S_{XX}. See the Equation (1.4). The second part is tivial. Since $\sum(x_i - \overline{x}) = 0$,

$$S_{XX} = \sum_{i=1}^{n}(x_i - \overline{x})^2 = \sum_{i=1}^{n}(x_i - \overline{x})x_i$$

For the first part of S_{XY},

$$\begin{aligned} S_{XY} &= \sum_{i=1}^{n}(x_i - \overline{x})(Y_i - \overline{Y}) \\ &= \sum_{i=1}^{n} x_i Y_i - \overline{x}\sum_{i=1}^{n} Y_i - \overline{Y}\sum_{i=1}^{n} x_i + n\overline{x}\overline{Y} \\ &= \sum_{i=1}^{n} x_i Y_i - \frac{1}{n}\Big(\sum_{i=1}^{n} x_i\Big)\Big(\sum_{i=1}^{n} Y_i\Big) \end{aligned}$$

Second part of S_{XY} also can be proven from the definition.

$$\begin{aligned} S_{XY} &= \sum_{i=1}^{n}(x_i - \overline{x})(Y_i - \overline{Y}) \\ &= \sum_{i=1}^{n} Y_i(x_i - \overline{x}) - \overline{Y}\sum_{i=1}^{n}(x_i - \overline{x}) \\ &= \sum_{i=1}^{n} Y_i(x_i - \overline{x}) \qquad \because \sum_{i=1}^{n}(x_i - \overline{x}) = 0 \end{aligned}$$

Same for the last part. □

Lemma 1.2 (Linearity)**.** *Each coefficient is a linear estimator.*

$$\hat{\beta}_1 = \sum_{i=1}^{n} \frac{(x_i - \overline{x})}{S_{XX}} Y_i$$

$$\hat{\beta}_0 = \sum_{i=1}^{n} \left(\frac{1}{n} - \frac{(x_i - \overline{x})\overline{x}}{S_{XX}}\right) Y_i$$

Proof. From lemma 1.1,

$$\begin{aligned} \hat{\beta}_1 &= \frac{S_{XY}}{S_{XX}} \\ &= \frac{1}{S_{XX}} \sum_{i=1}^{n}(x_i - \overline{x}) Y_i \end{aligned}$$

It gives that

$$\begin{aligned}\hat{\beta}_0 &= \overline{Y} - \hat{\beta}_1\overline{x} \\ &= \frac{1}{n}\sum_{i=1}^{n} Y_i - \overline{x}\sum_{i=1}^{n}\frac{(x_i-\overline{x})}{S_{XX}}Y_i \\ &= \sum_{i=1}^{n}\left(\frac{1}{n} - \frac{(x_i-\overline{x})\overline{x}}{S_{XX}}\right)Y_i\end{aligned}$$

□

Proposition 1.1 (Unbiasedness)**.** *Both coefficients are unbiased.*

1. $E\hat{\beta}_1 = \beta_1$
2. $E\hat{\beta}_0 = \beta_0$

From the model, $Y_1, \ldots, Y_n \overset{indep}{\sim} (\beta_0 + \beta_1 x_i, \sigma^2)$.

Proof. From lemma 1.1,

$$\begin{aligned}E\hat{\beta}_1 &= \sum_{i=1}^{n}\left[\frac{(x_i-\overline{x})}{S_{XX}}E(Y_i)\right] \\ &= \sum_{i=1}^{n}\frac{(x_i-\overline{x})}{S_{XX}}(\beta_0+\beta_1 x_i) \\ &= \frac{\beta_1\sum(x_i-\overline{x})x_i}{\sum(x_i-\overline{x})x_i} \qquad \because \sum(x_i-\overline{x}) = 0 \\ &= \beta_1\end{aligned}$$

It follows that

$$\begin{aligned}E\hat{\beta}_0 &= E(\overline{Y} - \hat{\beta}_1\overline{x}) \\ &= E(\overline{Y}) - \overline{x}E(\hat{\beta}_1) \\ &= E(\beta_0 + \beta_1\overline{x} + \overline{\epsilon}) - \beta_1\overline{x} \\ &= \beta_0 + \beta_1\overline{x} - \beta_1\overline{x} \\ &= \beta_0\end{aligned}$$

□

Proposition 1.2 (Variances)**.** *Variances and covariance of coefficients*

(a) $Var\hat{\beta}_1 = \frac{\sigma^2}{S_{XX}}$

(b) $Var\hat{\beta}_0 = \left(\frac{1}{n} + \frac{\overline{x}^2}{S_{XX}}\right)\sigma^2$

(c) $Cov(\hat{\beta}_0, \hat{\beta}_1) = -\frac{\overline{x}}{S_{XX}}\sigma^2$

Proof. Proving is just arithmetic.

(a)

$$\begin{aligned} Var\hat{\beta}_1 &= \frac{1}{S_{XX}^2}\sum_{i=1}^{n}\left[(x_i - \overline{x})^2 Var(Y_i)\right] + \frac{1}{S_{XX}^2}\sum_{j\neq k}^{n}\left[(x_j - \overline{x})(x_k - \overline{x})Cov(Y_j, Y_k)\right] \\ &= \frac{\sigma^2}{S_{XX}} \qquad \because Cov(Y_j, Y_k) = 0 \text{ if } j \neq k \end{aligned}$$

(b)

$$\begin{aligned} Var\hat{\beta}_0 &= \sum_{i=1}^{n}\left(\frac{1}{n} - \frac{(x_i - \overline{x})\overline{x}}{S_{XX}}\right)^2 Var(Y_i) + \sum_{j\neq k}\left(\frac{1}{n} - \frac{(x_j - \overline{x})\overline{x}}{S_{XX}}\right)\left(\frac{1}{n} - \frac{(x_k - \overline{x})\overline{x}}{S_{XX}}\right)Cov(Y_j, Y_k) \\ &= \frac{\sigma^2}{n} - 2\sigma^2\frac{\overline{x}}{S_{XX}}\sum_{i=1}^{n}(x_i - \overline{x}) + \frac{\sigma^2\overline{x}^2\sum(x_i - \overline{x})^2}{S_{XX}^2} \\ &= \left(\frac{1}{n} + \frac{\overline{x}^2}{S_{XX}}\right)\sigma^2 \qquad \because \sum(x_i - \overline{x}) = 0 \end{aligned}$$

(c)

$$\begin{aligned} Cov(\hat{\beta}_0, \hat{\beta}_1) &= Cov(\overline{Y} - \hat{\beta}_1\overline{x}, \hat{\beta}_1) \\ &= -\overline{x}Var\hat{\beta}_1 \\ &= -\frac{\overline{x}}{S_{XX}}\sigma^2 \end{aligned}$$

□

1.2.4 Gauss-Markov Theorem

Chapter 1.2.3 shows that the β_0^{LSE} and β_1^{LSE} are the **linear unbiased estimators**. Are these good? Good compared to *what estimators*? Here we consider *linear unbiased estimator*. If variances in the proposition 1.2 are lower than any parameters in this parameter family, β_0^{LSE} and β_1^{LSE} are the **best linear unbiased estimators**.

Theorem 1.1 (Gauss Markov Theorem)**.** *$\hat{\beta}_0$ and $\hat{\beta}_1$ are BLUE, i.e. the best linear unbiased estimator.*

$$Var(\hat{\beta}_0) \leq Var\Big(\sum_{i=1}^{n} a_i Y_i\Big)\ \forall a_i \in \mathbb{R}\ s.t.\ E\Big(\sum_{i=1}^{n} a_i Y_i\Big) = \beta_0$$

$$Var(\hat{\beta}_1) \leq Var\Big(\sum_{i=1}^{n} b_i Y_i\Big)\ \forall b_i \in \mathbb{R}\ s.t.\ E\Big(\sum_{i=1}^{n} b_i Y_i\Big) = \beta_1$$

Bestness of β_1. Consider $\Theta := \left\{ \sum_{i=1}^{n} b_i Y_i \in \mathbb{R} : E\Big(\sum_{i=1}^{n} b_i Y_i\Big) = \beta_1 \right\}$.

Claim: $Var(\sum b_i Y_i) - Var(\hat{\beta}_1) \geq 0$

Let $\sum b_i Y_i \in \Theta$. Then $E(\sum b_i Y_i) = \beta_1$.

Since $E(Y_i) = \beta_0 + \beta_1 x_i$,

$$\beta_0 \sum b_i + \beta_1 \sum b_i x_i = \beta_1$$

It gives

$$\begin{cases} \sum b_i = 0 \\ \sum b_i x_i = 1 \end{cases} \tag{1.7}$$

Then

$$\begin{aligned} 0 \leq Var\Big(\sum b_i Y_i - \hat{\beta}_1\Big) &= Var\Big(\sum b_i Y_i - \sum \frac{(x_i - \bar{x})}{S_{XX}} Y_i\Big) \\ &\overset{indep}{=} \sum \left(b_i - \frac{(x_i - \bar{x})}{S_{XX}}\right)^2 \sigma^2 \\ &= \sum \left(b_i^2 - \frac{2b_i(x_i - \bar{x})}{S_{XX}} + \frac{(x_i - \bar{x})^2}{S_{XX}^2}\right)\sigma^2 \\ &= \sum b_i^2 \sigma^2 - \frac{2\sigma^2}{S_{XX}} \sum b_i x_i + \frac{2\bar{x}\sigma^2}{S_{XX}} \sum b_i + \sigma^2 \frac{\sum (x_i - \bar{x})^2}{S_{XX}^2} \\ &= \sum b_i^2 \sigma^2 - \frac{\sigma^2}{S_{XX}} \qquad \because (1.7) \text{ and } S_{XX} = \sum (x_i - \bar{x})^2 \\ &= Var(\sum b_i Y_i) - Var(\hat{\beta}_1) \end{aligned}$$

Hence,

$$Var(\sum b_i Y_i) \geq Var(\hat{\beta}_1)$$

□

Bestness of β_0. Consider $\Theta := \left\{ \sum_{i=1}^{n} a_i Y_i \in \mathbb{R} : E\Big(\sum_{i=1}^{n} a_i Y_i \Big) = \beta_0 \right\}$.

Claim: $Var(\sum a_i Y_i) - Var(\hat{\beta}_0) \geq 0$

Let $\sum a_i Y_i \in \Theta$. Then $E(\sum a_i Y_i) = \beta_0$.

Since $E(Y_i) = \beta_0 + \beta_1 x_i$,

$$\beta_0 \sum a_i + \beta_1 \sum a_i x_i = \beta_0$$

It gives

$$\begin{cases} \sum a_i = 1 \\ \sum a_i x_i = 0 \end{cases} \tag{1.8}$$

Then

$$\begin{aligned}
0 \leq Var\Big(\sum a_i Y_i - \hat{\beta}_0 \Big) &= Var\Big[\sum a_i Y_i - \sum \Big(\frac{1}{n} - \frac{(x_k - \bar{x})\bar{x}}{S_{XX}} \Big) Y_k \Big] \\
&= \sum \left(a_i - \frac{1}{n} + \frac{(x_i - \bar{x})\bar{x}}{S_{XX}} \right)^2 \sigma^2 \\
&= \sum \left[a_i^2 - 2a_i \Big(\frac{1}{n} - \frac{(x_i - \bar{x})\bar{x}}{S_{XX}} \Big) + \Big(\frac{1}{n} - \frac{(x_i - \bar{x})\bar{x}}{S_{XX}} \Big)^2 \right] \sigma^2 \\
&= \sum a_i^2 \sigma^2 - \frac{2\sigma^2}{n} \sum a_i + \frac{2\bar{x}\sigma^2 \sum a_i x_i}{S_{XX}} - \frac{2\bar{x}^2 \sigma^2 \sum a_i}{S_{XX}} \\
&\quad + \sigma^2 \left(\frac{1}{n} - \frac{2\bar{x}}{n S_{XX}} \sum (x_i - \bar{x}) + \frac{\bar{x}^2 \sum (x_i - \bar{x})^2}{S_{XX}^2} \right) \\
&= \sum a_i^2 \sigma^2 - \frac{2\sigma^2}{n} - \frac{2\bar{x}^2 \sigma^2}{S_{XX}} \qquad \because (1.8) \\
&\quad + \left(\frac{1}{n} + \frac{\bar{x}^2}{S_{XX}} \right) \sigma^2 \qquad \because \sum (x_i - \bar{x}) = 0 \text{ and } S_{XX} := \sum (x_i - \bar{x})^2 \\
&= \sum a_i^2 \sigma^2 - \left(\frac{1}{n} + \frac{\bar{x}^2}{S_{XX}} \right) \sigma^2 \\
&= Var\Big(\sum a_i Y_i \Big) - Var \hat{\beta}_0
\end{aligned}$$

Hence,

$$Var(\sum a_i Y_i) \geq Var(\hat{\beta}_0)$$

□

Example 1.1. Show that $\sum(Y_i - \hat{Y}_i) = 0$, $\sum x_i(Y_i - \hat{Y}_i) = 0$, and $\sum \hat{Y}_i(Y_i - \hat{Y}_i) = 0$.

Solution. Consider the two normal equations (1.2) and (1.3). Note that $\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$.

From the Equation (1.2), we have $\sum(Y_i - \hat{Y}_i) = 0$.

From the Equation (1.3), we have $\sum x_i(Y_i - \hat{Y}_i) = 0$.

It follows that

$$\begin{aligned}\sum \hat{Y}_i(Y_i - \hat{Y}_i) &= \sum(\hat{\beta}_0 + \hat{\beta}_1 x_i)(Y_i - \hat{Y}_i) \\ &= \hat{\beta}_0 \sum(Y_i - \hat{Y}_i) + \hat{\beta}_1 \sum x_i(Y_i - \hat{Y}_i) \\ &= 0\end{aligned}$$

1.2.5 Estimation of σ^2

There is the last parameter, $\sigma^2 = Var(Y_i)$. In the *least squares estimation literary*, we estimate σ^2 by

$$\hat{\sigma}^2 = \frac{1}{n-2}\sum_{i=1}^{n}(Y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2 \tag{1.9}$$

Why $n - 2$? This makes the estimator unbiased.

Proposition 1.3 (Unbiasedness)**.**

$$E(\hat{\sigma}^2) = \sigma^2$$

Proof. Note that

$$(Y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) = (Y_i - \overline{Y}) - \hat{\beta}_1(x_i - \overline{x})$$

Then

$$
\begin{aligned}
E(\hat{\sigma}^2) &= \frac{1}{n-2} E\Big[\sum (Y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2 \Big] \\
&= \frac{1}{n-2} E\Big[\sum (Y_i - \overline{Y})^2 + \hat{\beta}_1^2 \sum (x_i - \overline{x})^2 - 2\hat{\beta}_1 \sum (Y_i - \overline{Y})(x_i - \overline{x}) \Big] \\
&= \frac{1}{n-2} E(S_{YY} + \hat{\beta}_1^2 S_{XX} - 2\hat{\beta}_1 S_{XY}) \\
&= \frac{1}{n-2} E(S_{YY} - \hat{\beta}_1^2 S_{XX}) \qquad \because S_{XY} = \hat{\beta}_1 S_{XX} \\
&= \frac{1}{n-2} \Big(\underbrace{E S_{YY}}_{(a)} - S_{XX} \underbrace{E\hat{\beta}_1^2}_{(b)} \Big)
\end{aligned}
$$

(a)

$$
\begin{aligned}
E S_{YY} &= E\Big[\sum (Y_i - \overline{Y})^2 \Big] \\
&= E\Big[\sum \Big((\beta_0 + \beta_1 x_i + \epsilon_i) - (\beta_0 + \beta_1 \overline{x} + \overline{\epsilon}) \Big)^2 \Big] \\
&= E\Big[\sum \Big(\beta_1 (x_i - \overline{x}) + (\epsilon_i - \overline{\epsilon}) \Big)^2 \Big] \\
&= \beta_1^2 S_{XX} + E\Big(\sum (\epsilon_i - \overline{\epsilon})^2 \Big) + 2\beta_1 \sum (x_i - \overline{x}) E(\epsilon_i - \overline{\epsilon}) \\
&= \beta_1^2 S_{XX} + E\Big(\sum (\epsilon_i - \overline{\epsilon})^2 \Big)
\end{aligned}
$$

Since $E(\overline{\epsilon}) = 0$ and $Var(\overline{\epsilon}) = \frac{\sigma^2}{n}$,

$$
\begin{aligned}
E\Big(\sum (\epsilon_i - \overline{\epsilon})^2 \Big) &= E\Big(\sum (\epsilon_i^2 + \overline{\epsilon}^2 - 2\epsilon_i \overline{\epsilon}) \Big) \\
&= \sum E(\epsilon_i^2) - n E(\overline{\epsilon}^2) \qquad \because \sum \epsilon = n\overline{\epsilon} \\
&= \sum (Var(\epsilon_i) + E(\epsilon_i)^2) - n(Var(\overline{\epsilon}) + E(\overline{\epsilon})^2) \\
&= n\sigma^2 - \sigma^2 \\
&= (n-1)\sigma^2
\end{aligned}
$$

Thus,

$$
E S_{YY} = \beta_1^2 S_{XX} + (n-1)\sigma^2
$$

(b)

$$\begin{aligned} E\hat{\beta}_1^2 &= Var\hat{\beta}_1 + E(\hat{\beta}_1)^2 \\ &= \frac{\sigma^2}{S_{XX}} + \beta_1^2 \end{aligned}$$

It follows that

$$\begin{aligned} E(\hat{\sigma}^2) &= \frac{1}{n-2}\Big(\underbrace{ES_{YY}}_{(a)} - S_{YY}\underbrace{E\hat{\beta}_1^2}_{(b)}\Big) \\ &= \frac{1}{n-2}\bigg(\Big(\beta_1^2 S_{XX} + (n-1)\sigma^2\Big) - S_{XX}\Big(\frac{\sigma^2}{S_{XX}} + \beta_1^2\Big)\bigg) \\ &= \frac{1}{n-2}((n-2)\sigma^2) \\ &= \sigma^2 \end{aligned}$$

□

1.3 Maximum Likelihood Estimation

In this section, we add an assumption to an random errors ϵ_i.

$$\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$$

Example 1.2 (Gaussian Likelihood)**.** Note that $Y_i \overset{indep}{\sim} N(\beta_0+\beta_1 x_i, \sigma^2)$. Then the likelihood function is

$$L(\beta_0, \beta_1, \sigma^2) = \prod_{i=1}^{n}\bigg(\frac{1}{\sqrt{2\pi\sigma^2}}\exp\bigg(-\frac{(Y_i - \beta_0 - \beta_1 x_i)^2}{2\sigma^2}\bigg)\bigg)$$

and so the log-likelihood function can be computed as

$$l(\beta_0, \beta_1, \sigma^2) = -\frac{n}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(Y_i - \beta_0 - \beta_1 x_i)^2$$

1.3.1 Likelihood equations

Definition 1.2 (Maximum Likelihood Estimator)**.**

$$(\hat{\beta}_0^{MLE}, \hat{\beta}_1^{MLE}, \hat{\sigma}^{2MLE}) := \arg\sup L(\beta_0, \beta_1, \sigma^2)$$

Since $l(\cdot) = \ln L(\cdot)$ is monotone,

Remark.

$$(\hat{\beta}_0^{MLE}, \hat{\beta}_1^{MLE}, \hat{\sigma}^{2MLE}) = \arg\sup l(\beta_0, \beta_1, \sigma^2)$$

We can find the maximum of this *quadratic* function by making first derivative.

$$\frac{\partial l}{\partial \beta_0} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(Y_i - \beta_0 - \beta_1 x_i) = 0 \tag{1.10}$$

$$\frac{\partial l}{\partial \beta_1} = \frac{1}{\sigma^2}\sum_{i=1}^{n} x_i(Y_i - \beta_0 - \beta_1 x_i) = 0 \tag{1.11}$$

$$\frac{\partial l}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^{n}(Y_i - \beta_0 - \beta_1 x_i)^2 = 0 \tag{1.12}$$

Denote that Equations (1.10) and (1.11) given $\hat{\sigma}^2$ are equivalent to the normal equations. Thus,

$$\hat{\beta}_0^{MLE} = \hat{\beta}_0^{LSE}, \quad \hat{\beta}_1^{MLE} = \hat{\beta}_1^{LSE}$$

From Equation (1.12),

$$\hat{\sigma}^{2MLE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \beta_0 - \beta_1 x_i)^2 = \frac{n-2}{n}\hat{\sigma}^{2LSE}$$

While $\hat{\sigma}^{2LSE}$ is an unbiased, above *MLE is not an unbiased estimator.* Since $\hat{\sigma}^{2MLE} \approx \hat{\sigma}^{2LSE}$ for large n, howerver, it is *asymptotically unbiased.*

Theorem 1.2 (Rao-Cramer Lower Bound, univariate case)**.** *Let* $X_1, \ldots, X_n \overset{iid}{\sim} f(x;\theta)$*. If* $\hat{\theta}$ *is an unbiased estimator of* θ*,*

$$Var(\hat{\theta}) \geq \frac{1}{I_n(\theta)}$$

where $I_n(\theta) = -E\left(\frac{\partial^2 l(\theta)}{\partial \theta^2}\right)$

To apply this theorem 1.2 in the simple linear regression setting, i.e. (β_0, β_1), we need to look at the *bivariate case.*

Theorem 1.3 (Rao-Cramer Lower Bound, bivariate case)**.** *Let* $X_1, \ldots, X_n \overset{iid}{\sim} f(x; \theta 1, \theta_2)$ *and let* $\boldsymbol{\theta} = (\theta_1, \theta_2)^T$*. If each* $\hat{\theta}_1$*,* $\hat{\theta}_2$ *is an unbiased estimator of* θ_1 *and* θ_2*, then*

$$Var(\boldsymbol{\theta}) := \begin{bmatrix} Var(\hat{\theta}_1) & Cov(\hat{\theta}_1, \hat{\theta}_2) \\ Cov(\hat{\theta}_1, \hat{\theta}_2) & Var(\hat{\theta}_2) \end{bmatrix} \geq I_n^{-1}(\theta_1, \theta_2)$$

where

$$I_n(\theta_1, \theta_2) = -\begin{bmatrix} E\left(\frac{\partial^2 l(\theta_1,\theta_2)}{\partial \theta_1^2}\right) & E\left(\frac{\partial^2 l(\theta_1,\theta_2)}{\partial \theta_1 \partial \theta_2}\right) \\ E\left(\frac{\partial^2 l(\theta_1,\theta_2)}{\partial \theta_1 \partial \theta_2}\right) & E\left(\frac{\partial^2 l(\theta_1,\theta_2)}{\partial \theta_2^2}\right) \end{bmatrix}$$

Assume that σ^2 is **known**. From the Equations (1.10) and (1.11),

$$\begin{cases} \frac{\partial^2 l}{\partial \beta_0^2} = -\frac{n}{\sigma^2} \\ \frac{\partial^2 l}{\partial \beta_1^2} = -\frac{\sum x_i^2}{\sigma^2} \\ \frac{\partial^2 l}{\partial \beta_0 \partial \beta_1} = -\frac{\sum x_i}{\sigma^2} \end{cases}$$

Thus,

$$I_n(\beta_0, \beta_1) = \begin{bmatrix} \frac{n}{\sigma^2} & \frac{\sum x_i}{\sigma^2} \\ \frac{\sum x_i}{\sigma^2} & \frac{\sum x_i^2}{\sigma^2} \end{bmatrix}$$

Applying gaussian elimination,

$$\begin{aligned} \left[\begin{array}{cc|cc} \frac{n}{\sigma^2} & \frac{\sum x_i}{\sigma^2} & 1 & 0 \\ \frac{\sum x_i}{\sigma^2} & \frac{\sum x_i^2}{\sigma^2} & 0 & 1 \end{array}\right] &\leftrightarrow \left[\begin{array}{cc|cc} \frac{n}{\sigma^2} & \frac{\sum x_i}{\sigma^2} & 1 & 0 \\ \frac{\sum x_i}{\sigma^2}\left(\frac{n}{\sum x_i}\right) & \frac{\sum x_i^2}{\sigma^2}\left(\frac{n}{\sum x_i}\right) & 0 & \frac{1}{\overline{x}} \end{array}\right] \\ &\leftrightarrow \left[\begin{array}{cc|cc} \frac{n}{\sigma^2} & \frac{\sum x_i}{\sigma^2} & 1 & 0 \\ 0 & \frac{\sum x_i^2 - \overline{x}\sum x_i}{\sigma^2 \overline{x}} = \frac{S_{XX}}{\sigma^2 \overline{x}} & -1 & \frac{1}{\overline{x}} \end{array}\right] \\ &\leftrightarrow \left[\begin{array}{cc|cc} 1 & \overline{x} & \frac{\sigma^2}{n} & 0 \\ 0 & 1 & -\frac{\overline{x}}{S_{XX}}\sigma^2 & \frac{\sigma^2}{S_{XX}} \end{array}\right] \\ &\leftrightarrow \left[\begin{array}{cc|cc} 1 & 0 & \left(\frac{1}{n} + \frac{\overline{x}^2}{S_{XX}}\right)\sigma^2 & -\frac{\overline{x}}{S_{XX}}\sigma^2 \\ 0 & 1 & -\frac{\overline{x}}{S_{XX}}\sigma^2 & \frac{\sigma^2}{S_{XX}} \end{array}\right] \end{aligned}$$

Hence,

$$I_n^{-1}(\beta_0,\beta_1)=\begin{bmatrix}\left(\frac{1}{n}+\frac{\overline{x}^2}{S_{XX}}\right)\sigma^2 & -\frac{\overline{x}}{S_{XX}}\sigma^2\\ -\frac{\overline{x}}{S_{XX}}\sigma^2 & \frac{\sigma^2}{S_{XX}}\end{bmatrix}=\begin{bmatrix}Var(\hat{\beta}_0) & Cov(\hat{\beta}_0,\hat{\beta}_1)\\ Cov(\hat{\beta}_0,\hat{\beta}_1) & Var(\hat{\beta}_1)\end{bmatrix}$$

Since $Var(\hat{\boldsymbol{\beta}})-I^{-1}=0$ is non-negative definite, each $Var(\hat{\beta}_0)=\left(\frac{1}{n}+\frac{\overline{x}^2}{S_{XX}}\right)\sigma^2$ and $Var(\hat{\beta}_1)=\frac{\sigma^2}{S_{XX}}$ is a theoretical bound.

Remark. This says that $\hat{\beta}_0^{LSE}=\hat{\beta}_0^{MLE}$ and $\hat{\beta}_1^{LSE}=\hat{\beta}_1^{MLE}$ have the smallest variance among all unbiased estimator.

This result is *stronger than Gauss-Markov theorem* 1.1, where the LSE has the smalleset variance among all *linear unbiased* estimators. It can be simply obtained from the *Lehmann-Scheffe Theorem*: If some unbiased estimator is a function of complete sufficient statistic, then this estimator is the unique MVUE (Hogg et al., 2018).

Remark (Lehmann and Scheffe for regression coefficients). $u\Big(\sum Y_i, S_{XY}\Big)$ is CSS in this regression problem, i.e. known σ^2.

Proof. From the example 1.2,

$$\begin{aligned}L(\beta_0,\beta_1)&=(2\pi\sigma^2)^{-\frac{n}{2}}\exp\left[-\frac{1}{2\sigma^2}\sum(Y_i-\beta_0-\beta_1x_i)^2\right]\\&=(2\pi\sigma^2)^{-\frac{n}{2}}\exp\left[-\frac{1}{2\sigma^2}\sum\Big(Y_i^2-(\beta_0+\beta_1x_i)Y_i+(\beta_0+\beta_1x_i)^2\Big)\right]\\&=(2\pi\sigma^2)^{-\frac{n}{2}}\exp\left[-\frac{1}{2\sigma^2}\Big(-\beta_0\sum Y_i-\beta_1\sum x_iY_i\Big)\right]\exp\left[-\frac{1}{2\sigma^2}\Big(\sum Y_i^2+(\beta_0+\beta_1x_i)^2\Big)\right]\end{aligned}$$

By the Factorization theorem, both $\sum Y_i$ and $\sum x_iY_i$ are sufficient statistics. Since S_{XY} is one-to-one function of $\sum x_iY_i$, it is also a sufficient statistic.

Denote that the normal distribution is in exponential family.

Hence, $(\sum Y_i, S_{XY})$ are CSS. □

1.4 Residuals

Definition 1.3 (Residuals)**.**

$$e_i:=Y_i-\hat{Y}_i$$

1.4.1 Prediction error

```
delv %>%
  mutate(yhat = predict(lm(y ~ x))) %>%
  ggplot(aes(x = x, y = y)) +
  geom_smooth(method = "lm", se = FALSE) +
  geom_point() +
  geom_linerange(aes(ymin = y, ymax = yhat), col = I("red"), alpha = .7) +
  labs(
    x = "Number of Cases",
    y = "Delivery Time"
  )
```

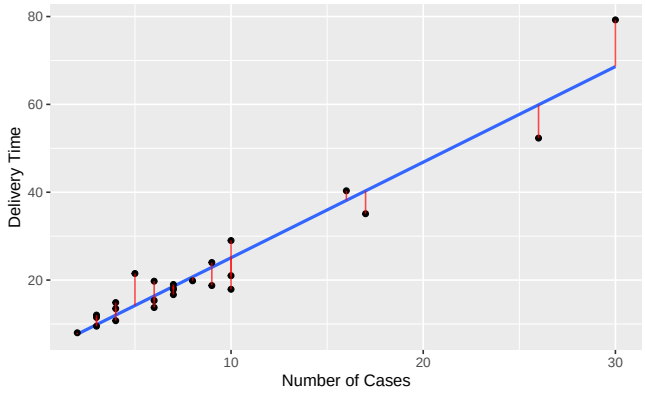


Figure 1.3: Fit and residuals

See Figure 1.3. Each red line is e_i. As we can see, e_i represents the difference between *observed* response and *predicted* response. A large $|e_i|$ indicates a large prediction error. You can call this e_i for each Y_i by `lm()$residuals` or `residuals()`.

```
delv_fit <- lm(y ~ x, data = delv)
delv_fit$residuals
#>      1      2      3      4      5      6      7      8      9     10
#> -1.874  1.651  2.181  2.855 -2.628 -0.444  0.327 -0.724 10.634  7.298
#>     11     12     13     14     15     16     17     18     19     20
#>  2.191 -4.082  1.475  3.372  1.094  3.918 -1.028  0.446 -0.349 -5.216
#>     21     22     23     24     25
#> -7.182 -7.581 -4.156 -0.900 -1.275
```

$\sum e_i^2$, which has been minimized in the procedure of LSE, can be used to see *overall size of prediction errors*.

Definition 1.4 (Residual Sum of Squares)**.**

$$SSE := \sum_{i=1}^{n} e_i^2$$

1.4.2 Residuals and the variance

e_i is a random quantity, which contains the information for ϵ_i. $\sum e_i^2$ can give information about $\sigma^2 = Var(\epsilon_i)$. For this, it is expected that e_i and ϵ_i have similar feature.

Lemma 1.3. *Covriance between Y and each coefficient*

(a) $Cov(\hat{\beta}_0, Y_i) = \left(\frac{1}{n} - \frac{(x_i - \overline{x})\overline{x}}{S_{XX}}\right)\sigma^2$

(b) $Cov(\hat{\beta}_1, Y_i) = \frac{(x_i - \overline{x})}{S_{XX}}\sigma^2$

Proof. (a)

$$\begin{aligned} Cov(\hat{\beta}_0, Y_i) &= Cov(\sum a_i Y_i, Y_i) \\ &= \left(\frac{1}{n} - \frac{(x_i - \overline{x})\overline{x}}{S_{XX}}\right)\sigma^2 \end{aligned}$$

(b)

$$\begin{aligned} Cov(\hat{\beta}_1, Y_i) &= Cov(\sum b_i Y_i, Y_i) \\ &= \frac{(x_i - \overline{x})}{S_{XX}}\sigma^2 \end{aligned}$$

□

Proposition 1.4 (Properties of residuals)**.** *Mean and variance of the residual*

(a) $E(e_i) = 0$

(b) $Var(e_i) \neq \sigma^2$

(c) $\forall i \neq j : Cov(e_i, e_j) \neq 0$

Proof. (a) Recall that this is the assumption of the regression model.

(b) Lemma 1.3 implies that

$$\begin{aligned} Cov(\overline{Y}, \hat{\beta}_1) &= Cov(\frac{1}{n}\sum Y_i, \hat{\beta}_1) \\ &= \frac{1}{n}\sum_{i=1}^{n} \frac{(x_i - \overline{x})}{S_{XX}}\sigma^2 \\ &= 0 \qquad \because \sum (x_i - \overline{x}) = 0 \end{aligned}$$

Then

$$\begin{aligned} Var(\hat{Y}_i) &= Var(\hat{\beta}_0 + \hat{\beta}_1 x_i) \\ &= Var\left[\overline{Y} + (x_i - \overline{x})\hat{\beta}_1\right] \qquad \because \hat{\beta}_0 = \overline{Y} - \hat{\beta}_1\overline{x} \\ &= Var(\overline{Y}) + (x_i - \overline{x})^2 Var(\hat{\beta}_1) + 2(x_i - \overline{x})Cov(\overline{Y}, \hat{\beta}_1) \\ &= \frac{\sigma^2}{n} + (x_i - \overline{x})^2 \frac{\sigma^2}{S_{XX}} + 0 \\ &= \left(\frac{1}{n} + \frac{(x_i - \overline{x})^2}{S_{XX}}\right)\sigma^2 \end{aligned} \tag{1.13}$$

From the same lemma 1.3,

$$\begin{aligned} Cov(Y_i, \hat{Y}_i) &= Cov(Y_i, \overline{Y} + (x_i - \overline{x})\hat{\beta}_1) \\ &= Cov(Y_i, \overline{Y}) + (x_i - \overline{x})Cov(Y_i, \hat{\beta}_1) \\ &= \frac{\sigma^2}{n} + \frac{(x_i - \overline{x})^2}{S_{XX}}\sigma^2 \qquad \because Cov(Y_i, \hat{\beta}_1) = \frac{(x_i - \overline{x})}{S_{XX}}\sigma^2 \\ &= \left(\frac{1}{n} + \frac{(x_i - \overline{x})^2}{S_{XX}}\right)\sigma^2 \end{aligned} \tag{1.14}$$

These Equations (1.13) and (1.14) give that

$$\begin{aligned} Var(e_i) &= Var(Y_i) + Var(\hat{Y}_i) - 2Cov(Y_i, \hat{Y}_i) \\ &= \sigma^2 + \left(\frac{1}{n} + \frac{(x_i - \overline{x})^2}{S_{XX}}\right)\sigma^2 - 2\left(\frac{1}{n} + \frac{(x_i - \overline{x})^2}{S_{XX}}\right)\sigma^2 \\ &= \left(1 - \frac{1}{n} - \frac{(x_i - \overline{x})^2}{S_{XX}}\right)\sigma^2 \\ &\neq \sigma^2 \end{aligned} \tag{1.15}$$

(*c*) Let $i \neq j$. Then

$$
\begin{aligned}
Cov(e_i, e_j) &= Cov\Big(Y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i), Y_j - (\hat{\beta}_0 + \hat{\beta}_1 x_j)\Big) \\
&= Cov(Y_i, Y_j) - Cov\Big(Y_i, (\hat{\beta}_0 + \hat{\beta}_1 x_j)\Big) - Cov((\hat{\beta}_0 + \hat{\beta}_1 x_i), Y_j) + Cov((\hat{\beta}_0 + \hat{\beta}_1 x_i), (\hat{\beta}_0 + \hat{\beta}_1 x_j) \\
&= 0 - \left(\frac{1}{n} - \frac{(x_i - \overline{x})\overline{x}}{S_{XX}}\right)\sigma^2 - \frac{(x_i - \overline{x})x_j}{S_{XX}}\sigma^2 \\
&\quad - \left(\frac{1}{n} - \frac{(x_j - \overline{x})\overline{x}}{S_{XX}}\right)\sigma^2 - \frac{(x_i - \overline{x})x_i}{S_{XX}}\sigma^2 \\
&\quad + \left(\frac{1}{n} + \frac{\overline{x}^2 + x_i x_j - \overline{x}(x_i + x_j)}{S_{XX}}\right)\sigma^2 \\
&= -\left(\frac{1}{n} + \frac{\overline{x}^2 + x_i x_j - \overline{x}(x_i + x_j)}{S_{XX}}\right)\sigma^2 \\
&= -\left(\frac{1}{n} + \frac{(x_i - \overline{x})(x_j - \overline{x})}{S_{XX}}\right)\sigma^2 \\
&\neq 0
\end{aligned}
$$

□

1.5 Decomposition of Total Variability

1.5.1 Total sum of squares

Definition 1.5 (Uncorrected Total Sum of Squares)**.**

$$SST_{uncor} := \sum_{i=1}^{n} Y_i^2$$

Definition 1.6 (Corrected Total Sum of Squares)**.**

$$SST := \sum_{i=1}^{n} (Y_i - \overline{Y})^2$$

What does this total sum of squares mean? To know this, we should know $\overline{Y}$ first.

```
delv %>%
  ggplot(aes(x = x, y = y)) +
  geom_smooth(method = "lm", formula = y ~ 1, se = FALSE) +
  geom_point() +
  labs(
    x = "Number of Cases",
    y = "Delivery Time"
  )
```

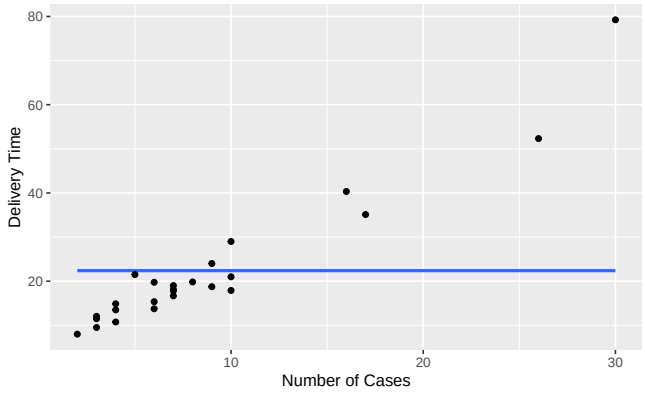


Figure 1.4: Regression without predictor

See Figure 1.4. The line represents the closest line when we use only intercept term for the regression model. In other words, *if we use no information for the response*, i.e. no predictor variables, we will get just average of the response variable. Consider

$$Y_i = \beta_0 + \epsilon_i$$

Then we can get only one normal equation

$$\sum (Y_i - \hat{\beta}_0) = 0$$

Hence,

$$\hat{\beta}_0 = \frac{1}{n} \sum_{i=1}^{n} Y_i \equiv \overline{Y}$$

From this fact, SST implies **total variance**.

1.5.2 Regression sum of squares

Definition 1.7 (Regression Sum of Squares)**.**

$$SSR := \sum_{i=1}^{n} (\hat{Y}_i - \overline{Y})^2$$

This SSR compares $\hat{Y}_i$ versus $\overline{Y}$, computing the sum of squares for difference between predicted values from *regression model* and *model not using predictors.*

1.5.3 Residual sum of squares

Now consider the *residual sum of squares* SSE in the definition 1.4. As mentioned, this is related to the *prediction errors*, which the regression model could not explain the data.

1.5.4 Decomposition of total sum of squares

SST can be decomposed by construction of sum of squares.

Proposition 1.5 (Decomposition of SST)**.**

$$SST = SSR + SSE$$

where $SST = \sum(Y_i - \overline{Y})^2$, $SSR = \sum(\hat{Y}_i - \overline{Y})^2$, *and* $SSE = \sum(Y_i - \hat{Y}_i)$

Proof. From the Example 1.1,

$$\begin{aligned}
\sum_{i=1}^{n}(Y_i - \overline{Y})^2 &= \sum_{i=1}^{n}(Y_i - \hat{Y}_i + \hat{Y}_i - \overline{Y})^2 \\
&= \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 + 2\sum_{i=1}^{n}(Y_i - \hat{Y}_i)(\hat{Y}_i - \overline{Y}) + \sum_{i=1}^{n}(\hat{Y}_i - \overline{Y})^2 \\
&= \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 + \sum_{i=1}^{n}(\hat{Y}_i - \overline{Y})^2 \qquad \because \sum(Y_i - \hat{Y}_i) = 0 \text{ and } \sum(Y_i - \hat{Y}_i)\hat{Y}_i = 0
\end{aligned}$$

□

This represents each SSR and SSE divides total variability as following.

$$\overset{SST}{\text{total variability}} = \overset{SSR}{\text{explained by regression}} + \overset{SSE}{\text{left unexplained by regression}}$$

Denote that the total variability SST is *constant given data set*. If our model is good, SSR grows and SSE flattens. Thus the larger SSR is, the better. The lower SSE is, the better.

1.5.5 Coefficient of determination

We have discussed in the previous section 1.5.4 that SSR and SSE splits the total variability into *explained part and not-explained part by our regression model*. Our first interest is whether the model works well for the data well, so we can think about the *proportion of explained part to the total variance*. The following measure R^2 computes this kind of value.

Definition 1.8 (Coefficient of Determination)**.**

$$R^2 := \frac{SSR}{SST} = 1 - \frac{1 - SSE}{SST}$$

By construction,

$$0 \leq R^2 \leq 1$$

As R^2 goes to 0, the model goes wrong. As R^2 is close to 1, large proportion of variability has been explained. So we prefer large values rather than small.

Proposition 1.6. *R^2 shows the strength of linear relation between two variables x and Y in the simple linear regression.*

$$R^2 = \hat{\rho}_{XY}$$

where $\hat{\rho}_{XY} := \frac{\sum(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum(X_i - \overline{X})^2}\sqrt{\sum(Y_i - \overline{Y})^2}}$ *is the sample correlation coefficients*

Proof. Note that $\hat{Y}_i - \overline{Y} = \hat{\beta}_1(x_i - \overline{x}) = \frac{S_{XY}}{S_{XX}}(x_i - \overline{x})$. Then

$$\begin{aligned}\sum(\hat{Y}_i - \overline{Y})^2 &= \frac{S_{XY}^2}{S_{XX}^2}\sum(x_i - \overline{x})^2 \\ &= \frac{S_{XY}^2}{S_{XX}}\end{aligned}$$

It follows that

$$\begin{aligned}R^2 &= \frac{\sum(\hat{Y}_i - \overline{Y})^2}{\sum(Y_i - \overline{Y})^2} \\ &= \frac{S_{XY}^2}{S_{XX}S_{YY}} \\ &=: \hat{\rho}_{XY}^2\end{aligned}$$

□

In this relation, we can know that R^2 statistic performs as a measure of the linear relationship in the simple linear regression setting.

1.6 Geometric Interpretations

1.6.1 Fundamental subspaces

These linear algebra concepts might be more useful for *multiple linear regression*, but let's briefly recap (Leon, 2014).

Definition 1.9 (Fundamental Subspaces)**.** Let $X \in \mathbb{R}^{n\times(p+1)}$.

Then the Null space is defined by

$$N(X) := \{\mathbf{b} \in \mathbb{R}^n \mid X\mathbf{b} = \mathbf{0}\}$$

The Row space is defined by

$$Row(X) := sp(\{\mathbf{r}_1, \ldots, \mathbf{r}_{p+1}\}) \quad \text{where } X^T = [\mathbf{r}_1^T, \ldots, \mathbf{r}_n^T]$$

The Column space is defined by

$$Col(X) := sp(\{\mathbf{c}_1, \ldots, \mathbf{c}_n\}) \quad \text{where } X = [\mathbf{c}_1, \ldots, \mathbf{c}_{p+1}]$$

The Range of X is defined by

$$R(X) := \{\mathbf{y} \in \mathbb{R}^n \mid \mathbf{y} = X\mathbf{b} \quad \text{for some } \mathbf{b} \in \mathbb{R}^{p+1}\}$$

These spaces have some constructional relationship.

Theorem 1.4 (Fundamental Subspaces Theorem)**.** *Let $X \in \mathbb{R}^{n\times(p+1)}$. Then*

$$N(X) = R(X^T)^\perp = Col(X^T)^\perp = Row(X)^\perp$$

Transposed matrix also satisfy this.

$$N(X^T) = R(X)^\perp = Col(X)^\perp$$

Proof. Let $\mathbf{a} \in N(X)$. Then $X\mathbf{a} = \mathbf{0}$.

Let $\mathbf{y} \in R(X^T)$. Then $X^T\mathbf{b} = \mathbf{y}$ for some $\mathbf{b} \in \mathbb{R}^{p+1}$.

Choose $\mathbf{b} \in \mathbb{R}^{p+1}$ such that $X^T\mathbf{b} = \mathbf{y}$. Then

$$\begin{aligned}\mathbf{0} &= X\mathbf{a} \\ &= \mathbf{b}^T X\mathbf{a} \\ &= \mathbf{y}^T\mathbf{a}\end{aligned}$$

Hence,

$$N(X) \perp R(X^T)$$

Since

$$X^T\mathbf{b} = \mathbf{c}_1\mathbf{b} + \cdots + \mathbf{c}_{p+1}\mathbf{b}$$

it is trivial that $R(X) = Col(X)$ and $R(X^T) = Col(X^T)$.

If $\mathbf{a} \in N(X)$, then

$$X\mathbf{a} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \cdots \\ \mathbf{r}_n \end{bmatrix} \begin{bmatrix} a_1 \\ \cdots \\ a_{p+1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \cdots \\ 0 \end{bmatrix}$$

Thus,

$$\forall i : \mathbf{a}^T\mathbf{r}_i = 0$$

and so

$$N(X) \subseteq Row(X)^\perp$$

Conversely, if $\mathbf{a} \in Row(X)^\perp$, then $\forall i : \mathbf{a}^T\mathbf{r}_i = 0$. This implies that $X\mathbf{a} = \mathbf{0}$. Thus,

$$Row(X)^\perp \subseteq N(X)$$

and so

$$N(X) = Row(X)^\perp$$

□

$N(X^T) = R(X)^\perp$ part in Theorem 1.4 will give the geometric insight to *least squares solution*.

Theorem 1.5. *Let S be a subspace of $\mathbb{R}^n$. Then*

$$dimS + dimS^\perp = n$$

If $\{\mathbf{x}_1, \ldots, \mathbf{x}_r\}$ is a basis for S and $\{\mathbf{x}_{r+1}, \ldots, \mathbf{x}_n\}$ is a basis for $S^\perp$, then $\{\mathbf{x}_1, \ldots, \mathbf{x}_r, \mathbf{x}_{r+1}, \ldots, \mathbf{x}_n\}$ is a basis for $\mathbb{R}^n$.

Theorem 1.6. *Let S be a subspace of $\mathbb{R}^n$. Then*

$$\mathbb{R}^n = S \oplus S^{\perp}$$

1.6.2 Simple linear regression

Theorem 1.7. *Let S be a subspace of $\mathbb{R}^n$. For each $\mathbf{y} \in \mathbf{R}^n$, there exists a unique $\mathbf{p} \in S$ that is closest to $\mathbf{y}$, i.e.*

$$\|\mathbf{y} - \mathbf{p}\| > \|\mathbf{y} - \hat{\mathbf{y}}\|$$

for any $\mathbf{p} \neq \hat{\mathbf{y}}$. Furthermore, a given vector $\mathbf{p} \in S$ will be the closest to a given vector $\mathbf{y} \in \mathbb{R}^n$ if and only if

$$\mathbf{y} - \hat{\mathbf{y}} \in S^{\perp}$$

Least square estimator $(\hat{\beta}_0, \hat{\beta}_1)^T$ minimizes

$$\sum_{i=1}^{n} (Y_i - \beta_0 - \beta_1 x_i)^2 = \|\mathbf{Y} - (\beta_0 \mathbf{1} + \beta_1 \mathbf{x})\|^2 \tag{1.16}$$

with respect to $(\hat{\beta}_0, \hat{\beta}_1)^T \in \mathbb{R}^2$ (where $\mathbf{1} := (1, 1)^T$). Recall that the normal equation gives

$$\sum_{i=1}^{n} (Y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) = \Big(\mathbf{Y} - (\hat{\beta}_0 \mathbf{1} + \hat{\beta}_1 \mathbf{x})\Big)^T \mathbf{1} = 0$$

and

$$\sum_{i=1}^{n} (Y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) x_i = \Big(\mathbf{Y} - (\hat{\beta}_0 \mathbf{1} + \hat{\beta}_1 \mathbf{x})\Big)^T \mathbf{x} = 0$$

These two relation give

$$\mathbf{Y} - (\hat{\beta}_0 \mathbf{1} + \hat{\beta}_1 \mathbf{x}) \perp sp(\{\mathbf{1}, \mathbf{x}\})^{\perp}$$

i.e. $\hat{\mathbf{Y}} = \hat{\beta}_0 \mathbf{1} + \hat{\beta}_1 \mathbf{x}$ is the projection of $\mathbf{Y}$.

Theorem 1.7 can give the same result.

$$\hat{\beta}_0 \mathbf{1} + \hat{\beta}_1 \mathbf{x} \in R([\mathbf{1}, \mathbf{x}])^{\perp} = sp(\{\mathbf{1}, \mathbf{x}\})^{\perp}$$

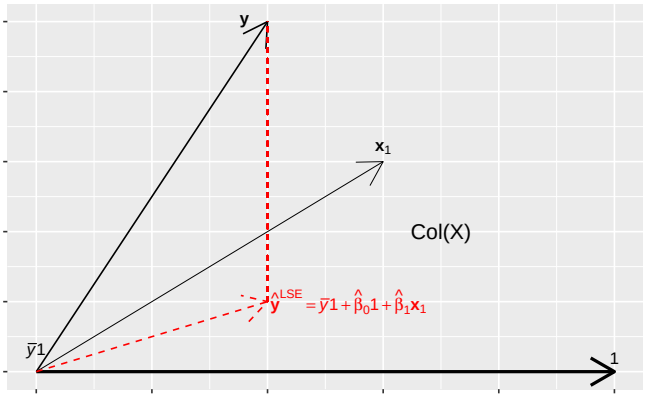


Figure 1.5: Geometric Illustration of Simple Linear Regression

We can see the details from Figure 1.5. In fact, decomposition of SST and R^2 are also in here.

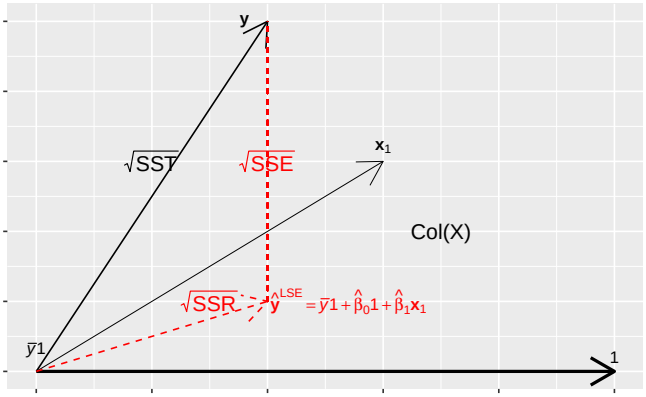


Figure 1.6: Geometric Illustration of Decomposing SST

See Figure 1.6.

$$\begin{cases} SST = \|\mathbf{Y} - \overline{Y}\mathbf{1}\|^2 \\ SSR = \|\hat{\mathbf{Y}} - \overline{Y}\mathbf{1}\|^2 \\ SSE = \|\mathbf{Y} - \hat{\mathbf{Y}}\|^2 \end{cases}$$

Pythagorean law implies that

$$SST = SSR + SSE$$

Also,

$$R^2 = \frac{SSR}{SST} = cos^2\theta = \hat{\rho}_{XY}^2 \tag{1.17}$$

1.6.3 Projection mapping

Look again Figure 1.5. Let $X \equiv [\mathbf{1}, \mathbf{x}] \in \mathbb{R}^{n\times 2}$ and let $\boldsymbol{\beta} \equiv (\beta_0, \beta_1)^T$. By the fundamental subspaces theorem 1.4,

$$\mathbf{Y} - X\hat{\boldsymbol{\beta}} \in Col(X)^\perp = N(X^T)$$

Thus,

$$X^T(\mathbf{Y} - X\hat{\boldsymbol{\beta}}) = \mathbf{0} \tag{1.18}$$

This is the another representation of normal equation. Then we now have

$$\begin{aligned} &X^T\mathbf{Y} - X^TX\hat{\boldsymbol{\beta}} = \mathbf{0} \\ \Leftrightarrow\ &X^T\mathbf{Y} = X^TX\hat{\boldsymbol{\beta}} \end{aligned}$$

If X^TX is nonsingular,

$$\hat{\boldsymbol{\beta}} = (X^TX)^{-1}X^T\mathbf{Y}$$

It follows that

$$\hat{\mathbf{Y}} = X\hat{\boldsymbol{\beta}} = X(X^TX)^{-1}X^T\mathbf{Y}$$

Combining this equation and our figure, we can know that $X(X^TX)^{-1}X^T$ projects $\mathbf{Y}$ from $\mathbb{R}^n$ onto $Col(X) = R(X)$. This is called projection operator/mapping.

Definition 1.10 (Projection matrix)**.** Projection operator or mapping from $\mathbb{R}^n$ to W is written by

$$\Pi(\cdot \mid W) := X(X^TX)^{-1}X^T$$

It can also be called ***Hat matrix*** and written as H.

As mentioned, X^TX should be invertible to get the LSE solution.

Theorem 1.8. *Let* $\mathbf{Y} = X\boldsymbol{\beta}$ *inconsistent and let* $X \in \mathbb{R}^{n\times(p+1)}$ *with* $n > p+1$. *If* $rank(X) = p+1$, *i.e. full rank, then* X^TX *is invertible.*

Proof. Suppose that $(X^TX)\mathbf{b} = \mathbf{0}$. Then

$$X^T(X\mathbf{b}) = \mathbf{0}$$

By the fundamental subspaces theorem 1.4,

$$X\mathbf{b} \in N(X^T) = Col(X)^\perp$$

By construction,

$$X\mathbf{b} \in Col(X) = N(X^T)^\perp$$

Then

$$X\mathbf{b} \in N(X^T) \cap N(X^T)^\perp = \{\mathbf{0}\}$$

It follows that

$$X\mathbf{b} = \mathbf{0}$$

If $rank(X) = \min(n, p+1)$, then the linear equation system has trivial solution $\mathbf{b} = \mathbf{0}$ and so does $X^T(X\mathbf{b}) = \mathbf{0}$. Hence, X^TX is invertible. □

Using projection matrix Π_W, we can re-express each sum of squares. Recall that when we only use y_i for regression fitting, the result becomes its average. It is because $\mathbf{Y}$ vector has been projected onto $sp(\{\mathbf{1}\})$ line.

Remark.

$$\overline{Y}\mathbf{1} = \mathbf{1}(\mathbf{1}^T\mathbf{1})^{-1}\mathbf{1}^T\mathbf{Y} = \Pi_{\mathbf{1}}\mathbf{Y}$$

$$\hat{\mathbf{Y}} = X(X^TX)^{-1}X^T\mathbf{Y} = \Pi_X\mathbf{Y}$$

Intuitively, every projection matrix is idempotent and symmetric. Once projected, the result is same when projecting it again.

Corollary 1.1 (Sum of squares)**.** $\Pi_{\mathbf{1}}$ *and* Π_X *can express each SS as following.*

(i)

$$SST = \mathbf{Y}^T(I - \Pi_{\mathbf{1}})\mathbf{Y}$$

(ii)

$$SSR = \mathbf{Y}^T(\Pi_X - \Pi_{\mathbf{1}})\mathbf{Y}$$

(iii)

$$SSE = \mathbf{Y}^T(I - \Pi_X)\mathbf{Y}$$

1.7 Distributions

1.7.1 Mean response and response

We have already look at predicting each mean response and response from equation (1.5) and (1.6).

Theorem 1.9 (Estimation of the mean response)**.**

$$\hat{\mu}_x \equiv \widehat{E(Y \mid x)} = \hat{\beta}_0 + \hat{\beta}_1 x$$

Theorem 1.10 ((out of sample) Prediction of a response)**.**

$$\hat{Y_x} = \hat{\beta}_0 + \hat{\beta}_1 x$$

Recall that predicting 1.9 targets at

$$\mu_x \equiv E(Y \mid x) = \beta_0 + \beta_1 x$$

which have been assumed to be true model. On the other hand, predicting 1.10 targets at

$$Y = \beta_0 + \beta_1 + \epsilon_x$$

The linearity is not true in reality. So the errors caused by modeling linear model are included in ϵ_x. This error term makes difference between properties of 1.9 and 1.10.

To derive their distribution and see the difference, we additionaly assume Normality, i.e.

$$\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$$

1.7.2 Regression coefficients

Under Normality, we have

$$Y_i \overset{indep}{\sim} N(\beta_0 + \beta_1 x_i, \sigma^2)$$

Then

$$\mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} \sim MVN_n\left(\boldsymbol{\mu} \equiv \begin{bmatrix} \beta_0 + \beta_1 x_1 \\ \beta_0 + \beta_1 x_2 \\ \vdots \\ \beta_0 + \beta_1 x_n \end{bmatrix}, \Sigma \equiv \sigma^2 I = \begin{bmatrix} \sigma^2 & 0 & \cdots & 0 \\ 0 & \sigma^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \sigma^2 \end{bmatrix}\right)$$

Write $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1)^T$. From Lemma 1.2,

$$\hat{\beta}_0 = \mathbf{a}^T \mathbf{Y}$$

where $\mathbf{a} = (a_1, \ldots, a_n)^T \in \mathbb{R}^n$ with $a_i = \left(\frac{1}{n} - \frac{(x_i - \overline{x})\overline{x}}{S_{XX}}\right)$

and

$$\hat{\beta}_1 = \mathbf{b}^T \mathbf{Y}$$

where $\mathbf{b} = (b_1, \ldots, b_n)^T \in \mathbb{R}^n$ with $b_i = \frac{(x_i - \overline{x})}{S_{XX}}$.

Let

$$A = \begin{bmatrix} \mathbf{a}^T \\ \hline \mathbf{b}^T \end{bmatrix} \in \mathbb{R}^{2 \times n}$$

Then

$$\hat{\boldsymbol{\beta}} = A\mathbf{Y}$$

Linearity of the multivariate normal distribution, Proposition 1.1 and 1.2 imply that

$$\hat{\boldsymbol{\beta}} = \begin{bmatrix} \hat{\beta}_0 \\ \hline \hat{\beta}_1 \end{bmatrix} \sim MVN\left(A\boldsymbol{\mu} = \begin{bmatrix} \beta_0 \\ \hline \beta_1 \end{bmatrix}, A\Sigma A^T = \sigma^2 AA^T = \begin{bmatrix} \left(\frac{1}{n} + \frac{\overline{x}^2}{S_{XX}}\right)\sigma^2 & -\frac{\overline{x}}{S_{XX}}\sigma^2 \\ \hline -\frac{\overline{x}}{S_{XX}}\sigma^2 & \frac{\sigma^2}{S_{XX}} \end{bmatrix}\right) \tag{1.19}$$

Since the joint random vector follows multivariate normal distribution, each *partitioned subset follow normal*. For this theorem, see Johnson and Wichern (2013). Hence, we finally get the following result.

Theorem 1.11 (Distributions of regression coefficients)**.** *Each regression coefficient follows Normal distribution.*

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

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

1.7.3 Mean response

In simple linear regression setting, we assume $\mu_x = E(Y \mid x) = \beta_0 + \beta_1 x$ is true.

```
delv %>%
  ggplot(aes(x = x, y = y)) +
  geom_smooth(method = "lm") +
  geom_point(alpha = .7) +
  labs(
    x = "Number of Cases",
    y = "Delivery Time"
  )
```

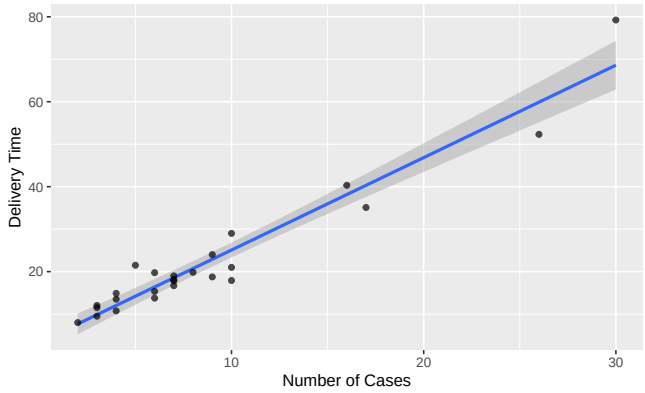


Figure 1.7: Mean response and its standard deviation

For example, in the Figure 1.7, the blue line indicates $E(Y \mid X = x)$ for each point x. Without fitting using `lm()`, `geom_smooth(method = "lm")` let us visualize the fitted line. Since the default method is not the linear regression, the `method` option should be specified.

```
delv %>%
  mutate(eyx = predict(delv_fit, newdata = data.frame(x = x)))
#> # A tibble: 25 x 3
#>       y     x   eyx
#>   <dbl> <dbl> <dbl>
#> 1  16.7     7 18.6
#> 2  11.5     3  9.85
#> 3  12.0     3  9.85
#> 4  14.9     4 12.0
#> 5  13.8     6 16.4
#> 6  18.1     7 18.6
#> # ... with 19 more rows
```

We have already seen in section 1.7.2 that the estimators $\hat{\beta}_0$ and $\hat{\beta}_1$ are random variables. So $\hat{\mu}_x$ is. In fact, the ribbon of the line in Figure 1.7 represents upper and lower confidence limits on mean response. In the later section, we get to know that it is $+t(n-2)\widehat{SE}(\hat{\mu}_x)$ and $-t(n-2)\widehat{SE}(\hat{\mu}_x)$. It can be drawn by default with the option of the `geom_smooth(se = TRUE)`.

Theorem 1.12 (Distribution of mean response estimator)**.** *$\hat{\mu}_x$ is also Normally distributed.*

$$\hat{\mu}_x \sim N\left(\mu_x, \sigma^2\left(\frac{1}{n} + \frac{(x-\overline{x})^2}{S_{XX}}\right)\right)$$

Proof. Since $\hat{\mu}_x = \hat{\beta}_0 + \hat{\beta}_1 x$ is the linear combination of $(\hat{\beta}_0, \hat{\beta}_1)^T$,

$$\hat{\mu}_x \sim N\Big(E(\hat{\mu}_x), Var(\hat{\mu}_x)\Big)$$

From Theorem 1.11,

$$E(\hat{\mu}_x) = E(\hat{\beta}_0) + E(\hat{\beta}_1)x = \beta_0 + \beta_1 x \equiv \mu_x$$

and from Proposition 1.2

$$\begin{aligned} Var(\hat{\mu}_x) &= Var(\hat{\beta}_0 + \hat{\beta}_1 x) \\ &= Var(\hat{\beta}_0) + x^2 Var(\hat{\beta}_1) + 2xCov(\hat{\beta}_0, \hat{\beta}_1) \\ &= \left(\frac{1}{n} + \frac{\overline{x}^2}{S_{XX}}\right)\sigma^2 + \frac{x^2\sigma^2}{S_{XX}} - \frac{2\overline{x}x\sigma^2}{S_{XX}} \\ &= \sigma^2\left(\frac{1}{n} + \frac{(x-\overline{x})^2}{S_{XX}}\right) \end{aligned}$$

□

Corollary 1.2.

$$\hat{\mu}_x - \mu_x \sim N\left(0, \sigma^2\left(\frac{1}{n} + \frac{(x-\overline{x})^2}{S_{XX}}\right)\right)$$

Denote that in both Theorem 1.12 and Corollary 1.2, σ^2 is parameter. So to use $SE(\hat{\mu}_x) = \sqrt{Var(\hat{\mu}_x)}$ in practice we plug in its estimator, usually Equation (1.9).

Corollary 1.3 (Standard error of mean response estimator)**.**

$$\widehat{SE}(\hat{\mu}_x) = \hat{\sigma}\sqrt{\left(\frac{1}{n} + \frac{(x-\overline{x})^2}{S_{XX}}\right)}$$

where $\hat{\sigma}^2 = MSE$

1.7.4 Response

Our goal is to predict each response at each point, i.e. $Y_x = \beta_0 + \beta_1 x + \epsilon_x$. $\epsilon_x \sim N(0, \sigma^2)$ is independent of the given data $(\epsilon_1, \ldots, \epsilon_n)$. In this sense, this prediction is called *out of sample prediction*. This setting makes difference between the *residuals, which are correlated to the data*. See Proposition 1.4 for this. This is occurred because each $\hat{\beta}_0$ and $\hat{\beta}_1$ is linear combination of $Y_1, \ldots, Y_n$, not Y_x.

While $Cov(Y_i, \hat{Y}_i) > 0, i = 1, \ldots, n$ (See Equation (1.14)), in case of out-of-sample Y_x,

$$Cov(Y_x, \hat{Y}_x) = Cov(Y_x, \hat{\beta}_0 + \hat{\beta}_1 x) = 0$$

Hence, arithmetically, this *out of sample prediction becomes to have larger standard error.*

Proposition 1.7 (Joint distribution of coefficients and error term)**.** $(\hat{\beta}_0, \hat{\beta}_1, \epsilon_x)^T$ *is Normally distributed.*

Proof. Want 1: $(\hat{\beta}_0, \hat{\beta}_1)^T \perp\!\!\!\perp \epsilon_x$

We have

$$\begin{aligned} Cov((\hat{\beta}_0, \hat{\beta}_1)^T, \epsilon_x) &= \Big[Cov(\hat{\beta}_i, \epsilon_x)\Big]_{2\times 1} \\ &= \left[Cov\left(\sum_{i=1}^{n} k_i Y_i, \epsilon_x\right)\right]_{2\times 1} \qquad k_i = \text{each linear coefficient for } \hat{\beta}_0, \hat{\beta}_1 \\ &= \mathbf{0} \end{aligned} \tag{1.20}$$

From Equation (1.19),

$$(\hat{\beta}_0, \hat{\beta}_1)^T \sim MVN$$

and from assumption,

$$\epsilon_x \sim N(0, \sigma^2)$$

It follows from Equation (1.20) that (Johnson and Wichern (2013))

$$(\hat{\beta}_0, \hat{\beta}_1)^T \perp\!\!\!\perp \epsilon_x$$

Want 2: $(\hat{\beta}_0, \hat{\beta}_1, \epsilon_x)^T \sim MVN$

From independency, we have (Johnson and Wichern (2013))

$$\left[\begin{array}{c} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hline \epsilon_x \end{array}\right] \sim MVN_{2+1}\left(\left[\begin{array}{c} \beta_0 \\ \beta_1 \\ \hline 0 \end{array}\right], \left[\begin{array}{c|c} Cov(\hat{\boldsymbol{\beta}}) \in \mathbb{R}^{2\times 2} & \mathbf{0} \in \mathbb{R}^2 \\ \hline \mathbf{0}^T \in \mathbb{R}^{2\times 1} & \sigma^2 \end{array}\right] \right)$$

□

This proposition gives clue to distribution of prediction error.

Theorem 1.13 (Distribution of out-of-sample prediction error)**.** *Out of sample prediction error $\hat{Y_x} - Y_x$ is Normally distributed*

$$\hat{Y_x} - Y_x \sim N\left(0, \sigma^2\left(1 + \frac{1}{n} + \frac{(x - \overline{x})^2}{S_{XX}}\right)\right)$$

Proof. Note that

$$\begin{aligned} \hat{Y_x} - Y_x &= (\hat{\beta}_0 + \hat{\beta}_1 x) - (\beta_0 + \beta_1 x + \epsilon_x) \\ &= [1, x, -1](\hat{\beta}_0, \hat{\beta}_1, \epsilon_x)^T - \beta_0 - \beta_1 x \end{aligned}$$

i.e. $\hat{Y_x} - Y_x$ is a linear combination of $(\hat{\beta}_0, \hat{\beta}_1, \epsilon_x)^T$. From prosition 1.7,

$$\begin{aligned}\hat{Y_x} - Y_x &\sim MVN\left([1,x,-1]\begin{bmatrix}\beta_0\\ \beta_1\\ 0\end{bmatrix} - \beta_0 - \beta_1 x, [1,x,-1]\left[\begin{array}{c|c} Cov(\hat{\boldsymbol{\beta}}) \in \mathbb{R}^{2\times 2} & \mathbf{0} \in \mathbb{R}^2 \\ \hline \mathbf{0}^T \in \mathbb{R}^{2\times 1} & \sigma^2 \end{array}\right]\begin{bmatrix}1\\ x\\ -1\end{bmatrix}\right) \\ &\overset{d}{=} MVN\left(0, \sigma^2\left(\frac{1}{n} + \frac{\overline{x}^2}{S_{XX}} - 2\frac{\overline{x}x}{S_{XX}} + \frac{x^2}{S_{XX}}\right) + 1\right) \\ &\overset{d}{=} MVN\left(0, \sigma^2\left(1 + \frac{1}{n} + \frac{(x-\overline{x})^2}{S_{XX}}\right)\right)\end{aligned} \tag{1.21}$$

□

Now we know the standard error of this out-of-sample prediction error.

$$SE(\hat{Y_x} - Y_x) = \sigma\sqrt{\left(1 + \frac{1}{n} + \frac{(x-\overline{x})^2}{S_{XX}}\right)}$$

We can see this standard error is *always larger than of mean response estimator* due to 1 in the bracket, i.e. σ^2. As mentioned, this is due to ϵ term. When we estimate or predict the mean response the model have been assumed to be true. In this out-of-sample prediction setting, however, the model can be wrong. This assumption error is also included in ϵ term and it is called *irreducible error*, which cannot be reduced anymore.

Remark.

$$SE(\hat{\mu}_x - \mu_x) < SE(\hat{Y_x} - Y_x)$$

It might be more clear if we see the inequality in the above remark. We know the fact that $\hat{Y_x}$ and Y_x are uncorrelated in this out-of-sample setting. Y_x is random variable, while μ_x is constant. Then we can re-express the inequality as

$$SE(\hat{\mu}_x) < SE(\hat{Y_x}) + SE(Y_x)$$

Actually, both $\hat{\mu}_x$ and $\hat{Y_x}$ are estimated as $\hat{\beta}_0 + \hat{\beta}_1 x$. Thus, $SE(Y_x) = \sigma^2$ makes out-of-sample more noisy.

To use standard error practically, we use $\hat{\sigma}^2$ as in corollary 1.3.

Corollary 1.4 (Standard error of out-of-sample prediction error)**.**

$$\widehat{SE}(\hat{Y_x} - Y_x) = \hat{\sigma}\sqrt{\left(1 + \frac{1}{n} + \frac{(x-\overline{x})^2}{S_{XX}}\right)}$$

where $\hat{\sigma}^2 = MSE$

1.8 Statistical Inference

Based on each distribution of estimator in section 1.7, we can construct various inferece for each

- β_0
- β_1
- μ_x
- Y_x
- σ^2

We can get the standard error for each coefficient through `summary()` function.

```
summary(delv_fit)
#>
#> Call:
#> lm(formula = y ~ x, data = delv)
#>
#> Residuals:
#>    Min     1Q Median     3Q    Max
#> -7.581 -1.874 -0.349  2.181 10.634
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)    3.321      1.371    2.42    0.024 *
#> x              2.176      0.124   17.55  8.2e-15 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 4.18 on 23 degrees of freedom
#> Multiple R-squared:  0.93,   Adjusted R-squared:  0.927
#> F-statistic:  308 on 1 and 23 DF,  p-value: 8.22e-15
```

Or more state-or-art way, `broom:tidy()` function has a method for each model object to make tidy data: `tibble`.

```
broom::tidy(delv_fit)
#> # A tibble: 2 x 5
#>   term        estimate std.error statistic  p.value
#>   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
#> 1 (Intercept)     3.32     1.37       2.42 2.37e- 2
#> 2 x               2.18     0.124     17.5  8.22e-15
```

1.8.1 Confidence interval

Consider standardization.

$$\frac{\hat{\theta} - \theta}{SE(\hat{\theta})}$$

Each SE includes σ^2 as we have already seen. First think about **known** σ^2 setting. All three estimators follow Normal distribution, and SE is constant by our the setting. Then we can construct each confidence interval as

$$\hat{\theta} \pm z_{\frac{\alpha}{2}} SE(\hat{\theta})$$

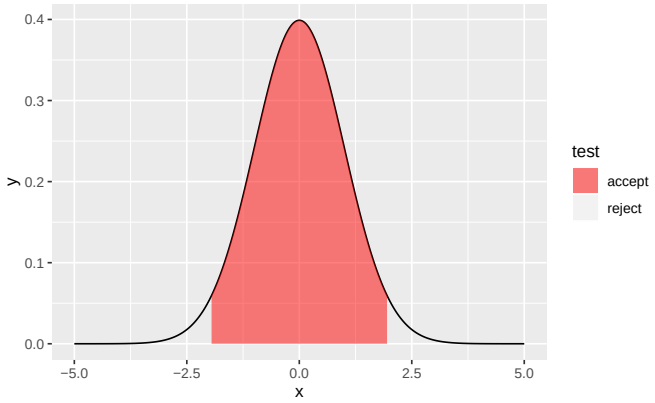


Figure 1.8: Confidence Interval when σ^2 is known

Now just plug in the results of section 1.7. For each regression coefficient,

Proposition 1.8 (Confidence intervals on β)**.** *With known* σ^2*,* $(1-\alpha)100\%$ *confidence intervals on* β_0 *and* β_1 *are given as*

$$\beta_0 : \quad \hat{\beta}_0 \pm z_{\frac{\alpha}{2}} \sqrt{\left(\frac{1}{n} + \frac{\overline{x}^2}{S_{XX}}\right)\sigma^2}$$

$$\beta_1 : \quad \hat{\beta}_1 \pm z_{\frac{\alpha}{2}} \sqrt{\frac{\sigma^2}{S_{XX}}}$$

Proposition 1.9 (Confidence interval on $\hat{\mu}_x$)**.** *With known* σ^2*,* $(1-\alpha)100\%$ *confidence interval on* $\hat{\mu}_x$ *is given as*

$$\mu_x : \quad \hat{\mu}_x \pm z_{\frac{\alpha}{2}} \sqrt{\sigma^2\left(\frac{1}{n} + \frac{(x-\overline{x})^2}{S_{xx}}\right)}$$

In practice, however, we do not know σ^2. In this case, we replace σ^2 with $\hat{\sigma}^2 = \frac{1}{n-2}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2 = MSE$. Then

$$\frac{\hat{\theta}-\theta}{\widehat{SE}} = \frac{\frac{\hat{\theta}-\theta}{\sqrt{Var=\sigma^2(\cdot)}}}{\sqrt{\frac{\frac{SSE}{\sigma^2}}{n-2}\left(\cdot\right)}} = \frac{\frac{\hat{\theta}-\theta}{\sqrt{Var=\sigma^2}} \sim N(0,1)}{\sqrt{\frac{\frac{SSE}{\sigma^2}\sim\chi^2(n-2)}{n-2}}} \sim t(n-2)$$

Thus, we need to replace $z_{\frac{\alpha}{2}}$ with $t_{\frac{\alpha}{2}}(n-2)$.

Proposition 1.10 (Confidence intervals on β when unknown σ^2)**.** *With unknown σ^2, $(1-\alpha)100\%$ confidence intervals on β_0 and β_1 are given as*

$$\beta_0: \quad \hat{\beta}_0 \pm t_{\frac{\alpha}{2}}(n-2)\sqrt{\left(\frac{1}{n}+\frac{\overline{x}^2}{S_{XX}}\right)\hat{\sigma}^2}$$

$$\beta_1: \quad \hat{\beta}_1 \pm t_{\frac{\alpha}{2}}(n-2)\sqrt{\frac{\hat{\sigma}^2}{S_{XX}}}$$

where $\hat{\sigma}^2 = MSE$

Here we can estimate the intervals. Basically, `confint()` function gives this interval.

```
confint(delv_fit, level = .95)
#>             2.5 % 97.5 %
#> (Intercept) 0.484   6.16
#> x           1.920   2.43
```

Proposition 1.11 (Confidence interval on $\hat{\mu}_x$ when unknown σ^2)**.** *With unknown σ^2, $(1-\alpha)100\%$ confidence interval on $\hat{\mu}_x$ is given as*

$$\mu_x: \quad \hat{\mu}_x \pm t_{\frac{\alpha}{2}}(n-2)\sqrt{\hat{\sigma}^2\left(\frac{1}{n}+\frac{(x-\overline{x})^2}{S_{xx}}\right)}$$

where $\hat{\sigma}^2 = MSE$

`predict()` provides options for this confidence interval. Specify `interval = "confidence"`. This argument has three option.

1. `"none"`: just compute fitted value, by default.
2. `"confidence"`: confidence interval of mean response
3. `"prediction"`: prediction interval of out-of-sample prediction

Default `level` is `0.95`.

```
predict(delv_fit, interval = "confidence", level = .95) %>% tbl_df()
#> # A tibble: 25 x 3
#>     fit   lwr   upr
#>   <dbl> <dbl> <dbl>
#> 1 18.6  16.8   20.3
#> 2  9.85  7.57  12.1
#> 3  9.85  7.57  12.1
#> 4 12.0   9.91  14.1
#> 5 16.4  14.5   18.2
#> 6 18.6  16.8   20.3
#> # ... with 19 more rows
```

1.8.2 Prediction interval

One proceeds in a similar way for out-of-sample Y_x.

Proposition 1.12 (Prediction interval on $\hat{Y_x}$)**.** *With known σ^2, $(1-\alpha)100\%$ confidence interval on $\hat{\mu}_x$ is given as*

$$Y_x: \quad \hat{Y_x} \pm z_{\frac{\alpha}{2}} \sqrt{\sigma^2 \left(1 + \frac{1}{n} + \frac{(x-\overline{x})^2}{S_{xx}}\right)}$$

Also, with unknown σ^2,

Proposition 1.13 (Prediction interval on $\hat{Y_x}$ when unknown σ^2)**.** *With unknown σ^2, $(1-\alpha)100\%$ confidence interval on $\hat{\mu}_x$ is given as*

$$Y_x: \quad \hat{Y_x} \pm t_{\frac{\alpha}{2}}(n-2) \sqrt{\hat{\sigma}^2 \left(1 + \frac{1}{n} + \frac{(x-\overline{x})^2}{S_{xx}}\right)}$$

where $\hat{\sigma}^2 = MSE$

Since this is out-of-sample setting, we should also give `newdata` option. Otherwise, we will get warning message. Denote that this argument only receive `data.frame` object with same element names.

```
predict(delv_fit, newdata = data.frame(x = 31:35), interval = "prediction", level = .95
#>    fit  lwr  upr
#> 1 70.8 60.3 81.3
#> 2 73.0 62.3 83.6
#> 3 75.1 64.3 85.9
#> 4 77.3 66.4 88.3
#> 5 79.5 68.4 90.6
```

1.8.3 Hypothesis testing

Look again the output of `summary.lm()` and `broom::tidy.lm()`.

```
summary(delv_fit)
#>
#> Call:
#> lm(formula = y ~ x, data = delv)
#>
#> Residuals:
#>    Min     1Q Median     3Q    Max
#> -7.581 -1.874 -0.349  2.181 10.634
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)    3.321      1.371    2.42    0.024 *
#> x              2.176      0.124   17.55  8.2e-15 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 4.18 on 23 degrees of freedom
#> Multiple R-squared:  0.93,   Adjusted R-squared:  0.927
#> F-statistic:  308 on 1 and 23 DF,  p-value: 8.22e-15
```

We can see `t value` and `Pr(>|t|)`. At the same time, `statistic` and `p.value`. What are these values? These are the results of the following tests.

$$H_0 : \beta_0 = \alpha_0 \qquad \text{vs} \qquad H_1 : \beta_0 \neq \alpha_0$$

$$T = \frac{\hat{\beta}_0 - \alpha_0}{\hat{\sigma}\sqrt{\left(\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}}\right)}} \overset{H_0}{\sim} t(n-2) \tag{1.22}$$

For this test statistic (1.22),

$$\text{reject } H_0 \quad \text{if } |T| > t_{\frac{\alpha}{2}}(n-2)$$

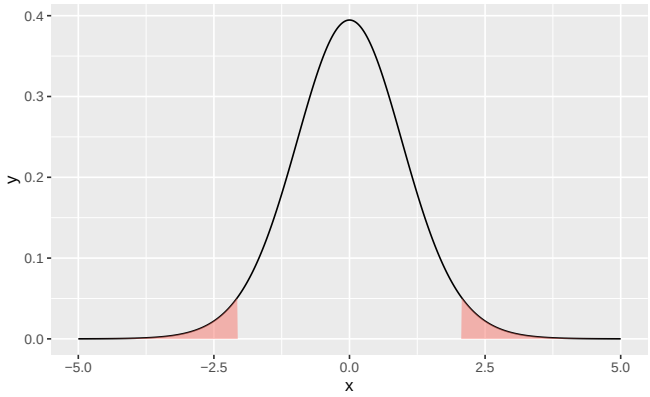


Figure 1.9: Rejection region for β_0

More importantly, we test β_1 which means slope

$$H_0 : \beta_1 = \alpha_1 \qquad \text{vs} \qquad H_1 : \beta_1 \neq \alpha_1$$

$$T = \frac{\hat{\beta}_1 - \alpha_1}{\hat{\sigma}\sqrt{\frac{1}{S_{xx}}}} \overset{H_0}{\sim} t(n-2) \tag{1.23}$$

For this test statistic (1.23),

$$\text{reject } H_0 \quad \text{if } |T| > t_{\frac{\alpha}{2}}(n-2)$$

Looking at these two statistics, we can intuitively know the meaning. As $|\hat{\beta}_1 - \alpha_1|$ becomes larger, the data support H_1.

1.9 Analysis of Variance

1.9.1 Useful distributions

In linear regression setting, we usually assume $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$. There are some useful distributions around Normal.

Proposition 1.14 (χ^2-distribution)**.** *Square of standard normal follows χ^2-distribution.*

If $Z \sim N(0,1)$, then $Z^2 \sim \chi^2(1)$

If $Z_i \overset{indep}{\sim} N(0,1)$, then $Z_1^2 + \cdots + Z_n^2 \sim \chi^2(n)$

Proposition 1.15 (t-distribution)**.** *Let* $Z \sim N(0,1) \perp\!\!\!\perp V \sim \chi^2(m)$*. Then*

$$T = \frac{Z}{\sqrt{V/m}} \sim t(m)$$

Proposition 1.16 (F-distribution)**.** *Let* $V \sim \chi^2(m) \perp\!\!\!\perp W \sim \chi^2(n)$*. Then*

$$F = \frac{V/m}{W/n} \sim F(m,n)$$

Also, there is *non-central analogue* of these three distributions, i.e. starting from $Z \sim N(\mu, 1)$.

Proposition 1.17 (Noncentral χ^2-distribution)**.** *Square of scaled normal follows non-central* χ^2*-distribution.*

If $Z_i \overset{indep}{\sim} N(\mu_i, 1)$*, then* $Z_1^2 + \cdots + Z_n^2 \sim \chi^2(n, \sum_{i=1}^{n} \mu_i^2)$

$\sum_{i=1}^{n} \mu_i^2$ *is called a non-central parameter.*

Proposition 1.18 (Noncentral t-distribution)**.** *Let* $X \sim N(\mu, 1) \perp\!\!\!\perp V \sim \chi^2(m)$*. Then*

$$T = \frac{Z}{\sqrt{V/m}} \sim t(m, \mu)$$

μ *is called a non-central parameter.*

Proposition 1.19 (Noncentral F-distribution)**.** *Let* $V \sim \chi^2(m, \delta) \perp\!\!\!\perp W \sim \chi^2(n)$*. Then*

$$F = \frac{V/m}{W/n} \sim F(m, n, \delta)$$

δ *is called a non-central parameter.*

1.9.2 Quadratic form

Now we can determine the distributions of various quadratic forms. The reason we are taking care of this is ANOVA deals with sum of squares, i.e. quadratic form. See Corollary 1.1 for this.

- $SST = \mathbf{Y}^T(I - \Pi_{\mathbf{1}})\mathbf{Y}$
- $SSR = \mathbf{Y}^T(\Pi_X - \Pi_{\mathbf{1}})\mathbf{Y}$
- $SSE = \mathbf{Y}^T(I - \Pi_X)\mathbf{Y}$

Theorem 1.14 (Idempotent and symmetric)**.** *Let $A \in \mathbb{R}^{k\times k}$ be idempotent and symmetric. Then*

(a) A^n is also idempotent

(b) $I - A$ is also idempotent

(c) Every eigenvalue of A is either 0 or 1 so that $tr(A) = rank(A)$

Proof. (a) and (b) are trivial.

$$(A^n)^2 = (A^2)^n = A^n$$

$$(I - A)^2 = I - 2A + A^2 = I - A$$

(c)

Fix λ an eigenvalue of A. Let $\mathbf{v} \neq \mathbf{0}$ be the corresponding eigenvector.

By definition,

$$A\mathbf{v} = \lambda\mathbf{v}$$

Then

$$A^2\mathbf{v} = \lambda(A\mathbf{v}) = \lambda^2\mathbf{v}$$

and so λ^2 is eigenvalue of A^2.

Since $A^2 = A$,

$$\lambda = \lambda^2$$

Hence,

$$\lambda = 0 \text{ or } 1$$

Note that for every matrix and its eigenvalues λ_j

$$tr(A) = \sum_{j=1}^{p} \lambda_j, \quad rank(A) = \text{the number of non-zero } \lambda_j$$

Since $\lambda = 0, 1$ of A,

$$tr(A) = rank(A)$$

□

Proposition 1.20 (Independence). *Assume* $\mathbf{Y} \sim MVN(\mu, \Sigma)$. *Then*

(i) If A and B are symmetric,

$$Y^T AY \perp\!\!\!\perp Y^T BY \Leftrightarrow A\Sigma B = 0$$

(ii) If A is symmetric,

$$Y^T AY \perp\!\!\!\perp BY \Leftrightarrow B\Sigma A = 0$$

Theorem 1.15 (Distribution of quadratic form). *Assume that* $\mathbf{Y} \sim MVN(\mu, I)$ *and that A is symmetric and idempotent. Then*

$$Y^T AY \sim \chi^2(K, \delta)$$

where $K = rank(A)$ *and* $\delta = \boldsymbol{\mu}^T A\boldsymbol{\mu}$. *Furthermore,*

$$\begin{cases} E(Y^T AY) = K + \delta \\ Var(Y^T AY) = 2(K + 2\delta) \end{cases}$$

Corollary 1.5 (Inner product of standard normal vector). *Let* $\mathbf{Z} = (Z_1, \ldots, Z_n)^T \sim MVN(\mathbf{0}, I_n)$. *Then*

$$\mathbf{Z}^T\mathbf{Z} = \sum_{i=1}^{n} Z_i^2 \sim \chi^2(n)$$

Proof. From Theorem 1.15 point of view,

$$\mathbf{Z}^T\mathbf{Z} = \mathbf{Z}^T I_n \mathbf{Z}$$

Thus,

$$K = rank(I_n) = n$$

$$\delta = \mathbf{0}$$

□

Using the above facts, we can now show distributions of sums of squares. First recall that

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

Proposition 1.21 (Distribution of SSE)**.**

$$\frac{SSE}{\sigma^2} \sim \chi^2(n-2, 0)$$

Proof. From Corollary 1.1, write

$$\frac{SSE}{\sigma^2} = \left(\frac{\mathbf{Y}}{\sigma}\right)^T (I - \Pi_X)\left(\frac{\mathbf{Y}}{\sigma}\right)$$

Note that

$$\frac{\mathbf{Y}}{\sigma} \sim MVN(\frac{1}{\sigma}X\boldsymbol{\beta}, I)$$

Since $I - \Pi_X$ is idempotent and symmetric,

$$K = rank(I - \Pi_X) = tr(I - \Pi_X) = n - rank(\Pi_X) = n - 2$$

$$\begin{aligned} \delta &= \left(\frac{X\boldsymbol{\beta}}{\sigma}\right)^T (I - \Pi_X)\left(\frac{X\boldsymbol{\beta}}{\sigma}\right) \\ &= \frac{\boldsymbol{\beta}^T X^T X\boldsymbol{\beta}}{\sigma^2} - \frac{(\boldsymbol{\beta}^T X^T) X (X^T X)^{-1} X^T (X\boldsymbol{\beta})}{\sigma^2} \\ &= \frac{\boldsymbol{\beta}^T X^T X\boldsymbol{\beta}}{\sigma^2} - \frac{\boldsymbol{\beta}^T X^T X\boldsymbol{\beta}}{\sigma^2} \\ &= 0 \end{aligned} \tag{1.24}$$

Hence,

$$\frac{SSE}{\sigma^2} \sim \chi^2(n-2, 0)$$

□

In case of SSE, it always follows $\chi^2(n-2)$ no matter what H_0 is. However, SSR and SST depend on β_1 that we want to test.

Proposition 1.22 (Distribution of SSR)**.**

$$\frac{SSR}{\sigma^2} \sim \chi^2(1, \delta)$$

where $\delta = \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i - \overline{x})^2\beta_1^2 = \frac{S_{xx}\beta_1^2}{\sigma^2}$

Proof. From Corollary 1.1, write

$$\frac{SSR}{\sigma^2} = \left(\frac{\mathbf{Y}}{\sigma}\right)^T (\Pi_X - \Pi_{\mathbf{1}})\left(\frac{\mathbf{Y}}{\sigma}\right)$$

Note that $\Pi_X - \Pi_{\mathbf{1}}$ is symmetric idempotent. One proceeds in a similar way.

$$K = rank(\Pi_X - \Pi_{\mathbf{1}}) = tr(\Pi_X - \Pi_{\mathbf{1}}) = rank(\Pi_X) - rank(\Pi_{\mathbf{1}}) = 2 - 1 = 1$$

$$\begin{aligned}\delta &= \left(\frac{X\boldsymbol{\beta}}{\sigma}\right)^T (\Pi_X - \Pi_{\mathbf{1}})\left(\frac{X\boldsymbol{\beta}}{\sigma}\right) \qquad \because \frac{\mathbf{Y}}{\sigma} \sim MVN(\frac{1}{\sigma}X\boldsymbol{\beta}, I) \\ &= \frac{\boldsymbol{\beta}^T\Big\{X^T(\Pi_X - \Pi_{\mathbf{1}})X\Big\}\boldsymbol{\beta}}{\sigma^2}\end{aligned}$$

Since $\mathbf{1} \in Col(X)$,

$$\Pi_X\mathbf{1} = \mathbf{1}$$

It gives that

$$\mathbf{1}^T(\Pi_X - \Pi_{\mathbf{1}})\mathbf{1} = 0 \tag{1.25}$$

If $\mathbf{x} \neq \mathbf{1}$, then we have

$$\mathbf{x}^T(\Pi_X - \Pi_{\mathbf{1}})\mathbf{x} = \sum_{i=1}^{n}(x_i - \overline{x})^2 = S_{xx} \tag{1.26}$$

Recall that

$$\overline{x}\mathbf{1} = \mathbf{1}(\mathbf{1}^T\mathbf{1})^{-1}\mathbf{1}^T\mathbf{x} = \Pi_{\mathbf{1}}\mathbf{x}$$

Then we have

$$\mathbf{1}^T(\Pi_X - \Pi_{\mathbf{1}})\mathbf{x} = \sum x_i - n\overline{x} = 0 \tag{1.27}$$

By symmetry,

$$\mathbf{x}^T(\Pi_X - \Pi_{\mathbf{1}})\mathbf{1} = n\overline{x} - \sum x_i = 0 \tag{1.28}$$

Hence by partitioning $X = [\mathbf{1} \mid \mathbf{x}]$,

$$\begin{aligned}
\delta &= \frac{\boldsymbol{\beta}^T \Big\{ [\mathbf{1} \mid \mathbf{x}]^T (\Pi_X - \Pi_{\mathbf{1}}) [\mathbf{1} \mid \mathbf{x}] \Big\} \boldsymbol{\beta}}{\sigma^2} \\
&= \frac{\boldsymbol{\beta}^T \begin{bmatrix} (1.25) & (1.27) \\ (1.28) & (1.26) \end{bmatrix} \boldsymbol{\beta}}{\sigma^2} \\
&= \frac{\boldsymbol{\beta}^T \begin{bmatrix} 0 & 0 \\ 0 & S_{xx} \end{bmatrix} \boldsymbol{\beta}}{\sigma^2} \\
&= \frac{S_{xx}\beta_1^2}{\sigma^2}
\end{aligned} \tag{1.29}$$

□

Proposition 1.23 (Independence)**.** *SSE and SSR are independent, i.e.*

$$SSE \perp\!\!\!\perp SSR$$

Proof. Note that both SSE and SSR are quadratic forms of $\mathbf{Y} \sim MVN(X\boldsymbol{\beta}, \sigma^2 I)$ and that each $I - \Pi_X$ and $\Pi_X - \Pi_{\mathbf{1}}$ is symmetric. Then from Proposition 1.20,

Claim: $(I - \Pi_X)(\sigma^2 I)(\Pi_X - \Pi_{\mathbf{1}}) = 0$, i.e. $(I - \Pi_X)(\Pi_X - \Pi_{\mathbf{1}}) = 0$

It is obvious that

$$\Pi_X \Pi_{\mathbf{1}} = \Pi_{\mathbf{1}}$$

Then

$$\begin{aligned}
(I - \Pi_X)(\Pi_X - \Pi_{\mathbf{1}}) &= \Pi_X - \Pi_{\mathbf{1}} - \Pi_X^2 + \Pi_X \Pi_{\mathbf{1}} \\
&= \Pi_X - \Pi_{\mathbf{1}} - \Pi_X + \Pi_{\mathbf{1}} \qquad \because \text{idempotent} \\
&= 0
\end{aligned}$$

This completes the proof. □

Proposition 1.24 (Independence)**.** *SSE and* $(\hat{\beta}_0, \hat{\beta}_1)$ *are independent, i.e.*

$$SSE \perp\!\!\!\perp (\hat{\beta}_0, \hat{\beta}_1)^T$$

Proof. Note that

$$\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1)^T = (X^T X)^{-1} X^T \mathbf{Y}$$

Since $I - \Pi_X$ of SSE is symmetric, from Proposition 1.20,
Claim: $((X^T X)^{-1} X^T)(\sigma^2 I)(I - \Pi_X) = 0$, i.e. $((X^T X)^{-1} X^T)(I - \Pi_X) = 0$
Since $\Pi_X = X(X^T X)^{-1} X^T$,

$$\begin{aligned}((X^T X)^{-1} X^T)(I - \Pi_X) &= (X^T X)^{-1} X^T - (X^T X)^{-1} X^T X (X^T X)^{-1} X^T \\ &= (X^T X)^{-1} X^T - (X^T X)^{-1} X^T \\ &= 0\end{aligned}$$

This completes the proof. □

Proposition 1.25 (Distribution of SST)**.**

$$\frac{SST}{\sigma^2} \sim \chi^2(n-1, \delta)$$

where $\delta = \sum_{i=1}^{n} (x_i - \overline{x})^2 \beta_1^2 = \frac{S_{xx}\beta_1^2}{\sigma^2}$

Proof. It proceedes in a similary way from Corollary 1.1

$$\frac{SST}{\sigma^2} = \left(\frac{\mathbf{Y}}{\sigma}\right)^T (I - \Pi_{\mathbf{1}}) \left(\frac{\mathbf{Y}}{\sigma}\right)$$

Since $I - \Pi_{\mathbf{1}}$ is symmetric idempotent,

$$K = rank(I - \Pi_{\mathbf{1}}) = tr(I - \Pi_{\mathbf{1}}) = n - rank(\Pi_{\mathbf{1}}) = n - 1$$

Since $\mathbf{1} \in sp(\{\mathbf{1}\})$,

$$\Pi_{\mathbf{1}} \mathbf{1} = \mathbf{1}$$

Hence,

$$\begin{aligned}\delta &= \left(\frac{X\boldsymbol{\beta}}{\sigma}\right)^T (I - \Pi_{\mathbf{1}})\left(\frac{X\boldsymbol{\beta}}{\sigma}\right) \\ &= \frac{S_{xx}\beta_1^2}{\sigma^2} \qquad \because (1.24) \text{ and } (1.29)\end{aligned}$$

□

1.9.3 ANOVA for testing significance of regression

Recall that

$$SST = SSR + SSE$$

- SST: the variation of a response itself
- SSR: the variation of a response *explained by the model*
- SSE: the variation of a response that *cannot be explained by the model*

As mentioned in section 1.5.4, whether the model is useful or not can depend on the proportion of SSR versus SSE in constant SST. When SSR is large compared to SSE, we can say that the model is good. On the other hand, when SSR is not large, the model might be poor. This is what R^2 measures intuitively.

However, this direct comparison somtimes does not work in many times. Both SSR and SSE comes from different distribution, which have different degrees of freedom. So we *compare standardized versions*, i.e. divided by the degrees of freedom.

Definition 1.11 (Degrees of freedom)**.** Degrees of freedom of each sum of squares is

$$df = \text{the number of deviation} - \text{the number of linear constraints}$$

Corollary 1.6 (df of SS)**.** *df of each sum of square is computed as*

(a) $df(SST) = n - 1$

(b) $df(SSR) = 1$

(c) $df(SSE) = n - 2$

Proof. (a)

Since $\sum(Y_i - \overline{Y}) = 0$, we have 1 linear constraints. Thus,

$$df(SST) = n - 1$$

(b)

Note that $\hat{Y}_i - \overline{Y} = \hat{\beta}_1(x_i - \overline{x})$

where $\sum(x_i - \overline{x}) = 0$.

Thus,

$$df(SSR) = n - (n-1) = 1$$

(c)

From Example 1.1, $\sum(Y_i - \hat{Y}_i) = 0$ and $\sum x_i(Y_i - \hat{Y}_i) = 0$.

Thus,

$$df(SSE) = n - 2$$

□

Dividing sum of squares in df, we can standardize it.

Definition 1.12 (Mean square)**.** Mean square is a sum of square SS divided by its degree of freedom df

$$MS := \frac{SS}{df}$$

Using the values of corollary 1.6 we can define each mean square for SSR and SSE.

Definition 1.13 (Regression mean square)**.**

$$MSR := \frac{SSR}{1} = SSR$$

From Proposition 1.22, the following corollary can be drawn.

Corollary 1.7 (Distribution of MSR)**.** *Under* $H_0 : \beta_1 = 0$*,*

$$\frac{SSR}{\sigma^2} \overset{H_0}{\sim} \chi^2(1)$$

Now standardize residual sum of square.

Definition 1.14 (Residual mean square)**.**

$$MSE := \frac{SSE}{n-2}$$

From Proposition 1.22, we can construct same statistic. In fact, $\frac{SSE}{\sigma^2}$ follows $\chi^2(n-2)$ whether or not β_1 is zero. Its $\delta = 0$.

Corollary 1.8 (Distribution of MSE)**.**

$$\frac{SSE}{\sigma^2} \sim \chi^2(n-2)$$

Finally, we can now use Proposition 1.16 so that

$$F \equiv \frac{MSR}{MSE} = \frac{\frac{SSE/\sigma^2 \sim \chi^2(1)}{1}}{\frac{SSR/\sigma^2 \overset{H_0}{\sim} \chi^2(n-2)}{n-2}} \overset{H_0}{\sim} F(1, n-2)$$

By construction, this test statistic is used for

$$H_0 : \beta_1 = 0$$

which means that the predictor does not explain the response anything. In other words, we are testing that

$$H_0 : \text{Model is not useful at all} \qquad \text{vs} \qquad H_1 : \text{Model can explain data} \tag{1.30}$$

Remark (F statistic on testing significance). Null hypothesis (1.30) can be tested with F-statistic.

$$F_0 = \frac{MSR}{MSE} = \frac{SSR/df(SSR)}{SSE/df(SSE)} \overset{H_0}{\sim} F(df(SSR), df(SSE))$$

Here, it is

$$F_0 = \frac{SSR/1}{SSE/(n-2)}$$

Then we reject H_0 if

$$F_0 > F_\alpha\Big(df(SSR), df(SSE)\Big)$$

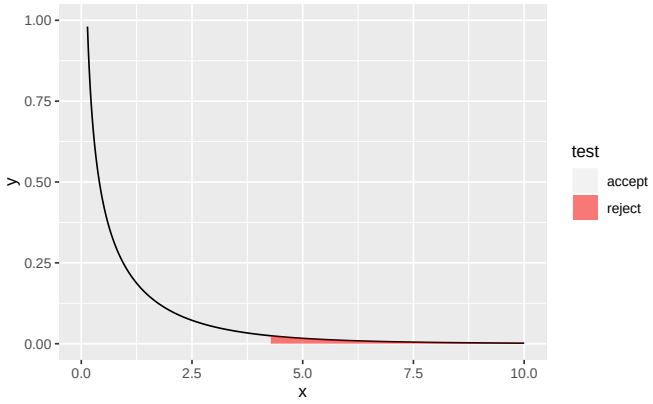


Figure 1.10: Rejection region for significance testing

```
summary(delv_fit)
#>
#> Call:
#> lm(formula = y ~ x, data = delv)
#>
#> Residuals:
#>    Min     1Q Median     3Q    Max
#> -7.581 -1.874 -0.349  2.181 10.634
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)    3.321      1.371    2.42    0.024 *
#> x              2.176      0.124   17.55  8.2e-15 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 4.18 on 23 degrees of freedom
#> Multiple R-squared:  0.93,   Adjusted R-squared:  0.927
#> F-statistic:  308 on 1 and 23 DF,  p-value: 8.22e-15
```

This statistic is `F-statistic` included in `summary.lm()` output. This is saved as `$fstatistic`.

```
summary(delv_fit)$fstatistic
#> value numdf dendf
#>   308     1    23
```

We usually summarize these statistic in table form, so called *ANOVA table*.

Source	SS	df	MS	F	p-value

Source	SS	df	MS	F	p-value
Model	SSR	1	MSR	F_0	p-value
Error	SSE	$n-2$	MSE		
Total	SST	$n-1$			

To get this table, just use `anova()` for `lm` object.

```
anova(delv_fit)
#> Analysis of Variance Table
#>
#> Response: y
#>           Df Sum Sq Mean Sq F value  Pr(>F)
#> x          1   5382    5382     308 8.2e-15 ***
#> Residuals 23    402      17
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Since the last `Total` row is just sum of the model and error, the function does not give it. To use this table as `data.frame` more easily, just implement `broom::tidy` as before.

```
anova(delv_fit) %>%
  broom::tidy()
#> # A tibble: 2 x 6
#>   term         df sumsq meansq statistic   p.value
#>   <chr>     <int> <dbl>  <dbl>     <dbl>     <dbl>
#> 1 x             1 5382. 5382.       308.  8.22e-15
#> 2 Residuals    23  402.   17.5       NA   NA
```

Denote that here *simple linear regression setting* F-statistic and t-statistic of Equation (1.23) perform exactly same thing, $H_0 : \beta_1 = 0$. In fact, we know that

$$F(1, k) \overset{d}{=} T_k^2$$

Remark. In the simple linear regression setting, F-test for significance and t-test for no slope are equivalent, i.e. under $H_0 : \beta_1 = 0$

$$F_0 = \frac{\hat{\beta}_1 S_{xx}}{\hat{\sigma}^2} = \left(\frac{\hat{\beta}_1}{\hat{\sigma} / \sqrt{S_{xx}}} \right) = T_0^2$$

Chapter 2

Multiple Linear Regression

2.1 Model

```
(cem <- MPV::cement %>% tbl_df())
#> # A tibble: 13 x 5
#>        y    x1    x2    x3    x4
#>    <dbl> <dbl> <dbl> <dbl> <dbl>
#> 1  78.5     7    26     6    60
#> 2  74.3     1    29    15    52
#> 3 104.     11    56     8    20
#> 4  87.6    11    31     8    47
#> 5  95.9     7    52     6    33
#> 6 109.     11    55     9    22
#> # ... with 7 more rows
```

Above is a data set about cement and concerning four ingredients from the Montgomery et al. (2015) textbook.

- `y`: heat evolved in calories per gram of cement
- `x1`: tricalcium aluminate
- `x2`: tricalcium silicate
- `x3`: tetracalcium alumino ferrite
- `x4`: dicalcium silicate

Given data $(x_{11}, x_{12}, \ldots, x_{1p}, Y_1), \ldots, (x_{n1}, x_{n2}, \ldots, x_{np}, Y_n)$ $(p = 4)$, we try to fit linear regression model

$$Y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i$$

with

$$\epsilon_i \overset{iid}{\sim} (0, \sigma^2)$$

Compared to simple linear regression problem 1, we have more parameters for coefficients

$$(\beta_0, \beta_1, \ldots, \beta_p, \sigma^2)$$

Each β_j is a change of Y when each predictor variable x_j increases in 1 unit while the others fixed. In this part, we use *matrix notation*. Extending our former matrix work 1.6,

$$\underbrace{\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}}_{\mathbf{Y}} = \underbrace{\begin{bmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{bmatrix}}_{X} \underbrace{\begin{bmatrix} \beta_0 \\ \vdots \\ \beta_p \end{bmatrix}}_{\boldsymbol{\beta}} + \underbrace{\begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}}_{\boldsymbol{\epsilon}}$$

where ϵ_i are i.i.d., and

$$E\boldsymbol{\epsilon} = \mathbf{0}$$

$$Var\boldsymbol{\epsilon} = \sigma^2 I$$

2.2 Least Square Estimation

Write $\boldsymbol{\beta} \equiv (\beta_1, \ldots, \beta_p)^T \in \mathbb{R}^{p+1}$. Extend Equation (1.16).

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\operatorname{argmin}} \sum_{i=1}^{n} (Y_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_p x_{ip})^2 \\ &= \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\operatorname{argmin}} \|\mathbf{Y} - \beta_0 \mathbf{1} - \beta_1 \mathbf{x}_1 - \cdots - \beta_p \mathbf{x}_p\|^2 \\ &= \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\operatorname{argmin}} \|\mathbf{Y} - X\boldsymbol{\beta}\|^2 \end{aligned} \tag{2.1}$$

As discussed, the solution $\hat{\boldsymbol{\beta}}$ is related to the projection. $X\hat{\boldsymbol{\beta}}$ is a projection of $\mathbf{Y}$ onto $Col(X)$.

2.2.1 Normal equation

Now recap the section 1.6.3. Fundamental subspaces theorem 1.4 implies that

$$\mathbf{Y} - X\hat{\boldsymbol{\beta}} \in Col(X)^{\perp} = N(X^T)$$

From the second part of subset, i.e. $N(X^T)$, we now have *Normal equation*

$$X^T(\mathbf{Y} - X\hat{\boldsymbol{\beta}}) = \mathbf{0} \tag{2.2}$$

This is equivalent to

$$X^T\mathbf{Y} = X^TX\hat{\boldsymbol{\beta}}$$

Hence, if X^TX is invertible, the equation gives unique solution

$$\hat{\boldsymbol{\beta}} = (X^TX)^{-1}X^T\mathbf{Y}$$

Our first question is when X^TX is invertible, and Theorem 1.8 have said that it is when the model matrix X is full rank.

Lemma 2.1. *Let $X \in \mathbb{R}^{n\times(p+1)}$ be any model matrix. Then X^TX is always non-negative definite.*

$$\forall \mathbf{v} \in \mathbb{R}^{p+1} : \mathbf{v}^T(X^TX)\mathbf{v} \geq 0$$

Proof. Let $\mathbf{v} \in \mathbb{R}^{p+1}$. Then

$$\mathbf{v}^T(X^TX)\mathbf{v} = (X\mathbf{v})^T(X\mathbf{v}) = \|X\mathbf{v}\|^2 \geq 0$$

□

This lemma can also prove our Theorem 1.8.

Theorem 2.1. *Let $\mathbf{Y} = X\boldsymbol{\beta}$ inconsistent and let $X \in \mathbb{R}^{n\times(p+1)}$ with $n > p+1$. If $rank(X) = p+1$, i.e. full rank, then X^TX is invertible.*

Proof. Let $\mathbf{c} \in \mathbb{R}^{(p+1)}$

Suppose that X^TX is positive definite.

$$\Leftrightarrow \mathbf{c}^T X^T X \mathbf{c} = 0 \quad \text{implies} \quad \mathbf{c} = \mathbf{0}$$
$$\Leftrightarrow X\mathbf{c} = \mathbf{0} \quad \text{implies} \quad \mathbf{c} = \mathbf{0}$$
$$\Leftrightarrow \text{columns of } X \text{ linearly independent}$$
$$\Leftrightarrow rank(X) = p + 1$$

□

2.2.2 Orthogonal decomposition

Theorem 2.2. *Let $Col(X)$ be a subspace of $\mathbb{R}^n$, let $\mathbf{Y} \in \mathbb{R}^n$, and let $\{\mathbf{u}_0, \ldots, \mathbf{u}_p\}$ be an orthonormal basis for $Col(X)$. If*

$$\hat{\mathbf{Y}} = \sum_{j=0}^{p} \hat{\beta}_j \mathbf{u}_j$$

where

$$\hat{\beta}_j = \Pi(\mathbf{Y} \mid R(\mathbf{u}_j)) \quad \textit{for each } i$$

then $\hat{\mathbf{Y}} - \mathbf{Y} \in Col(X)^{\perp}$.

Theorem 2.3. *Under the hypothesis of Theorem 2.2, $\hat{\mathbf{Y}} \in Col(X)$ is the closest to $\mathbf{Y}$ amongst its any element $\mathbf{p}$, i.e.*

$$\|\mathbf{p} - \mathbf{Y}\| > \|\hat{\mathbf{Y}} - \mathbf{Y}\|$$

for any $\mathbf{p} \neq \hat{\mathbf{Y}}$ in $Col(X)$

In other words, projection of $\mathbf{Y}$ onto $Col(X)$, $\hat{\mathbf{Y}}$ can be *represented as sum of projections of $\mathbf{Y}$ onto each (orthogonal) individual variable.* Before looking at individual basis, consider two-block space.

Write

$$X = \left[\begin{array}{c|ccc} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{array}\right] = [\mathbf{1}, \mathbb{X}_A]$$

Consider $R(X)$, $R(\mathbf{1})$, and $R(\mathbb{X}_A)$.

To decompose subspace $R(X)$, we try to orthogonalize $\mathbf{1}$ and $\mathbb{X}_A$. By Theorem 2.2, we have

$$\mathbf{1} \perp \mathbb{X}_A - \Pi_{\mathbf{1}}\mathbb{X}_A$$

In fact, the right one $\mathbb{X}_A - \Pi_{\mathbf{1}}\mathbb{X}_A$ is the *residual after simple linear regression* $\mathbb{X}_A$ onto $\mathbf{1}$. We have seen in Figure 1.6 of section 1.6 that the *residual is orthogonal to predictor vector*. In this procedure, we choose residual as new predictor instead of response in simple linear regression, i.e. $\mathbb{X}_A$. If this is done to individual predictor variables, it is called *successive orthogonalization* and it will be coved next section with QR decomposition.

Theorem 1.6 implies that

$$R(X) = R(\mathbf{1}) \oplus R(\mathbb{X}_A - \Pi_{\mathbf{1}}\mathbb{X}_A)$$

Theorem 2.4 (Orthogonal decomposition)**.** *Let* $X = [\mathbf{1}, \mathbb{X}_A]$. *Then*

(i)

$$R(X) = R(\mathbf{1}) \oplus R(\mathbb{X}_A - \Pi_{\mathbf{1}}\mathbb{X}_A)$$

(ii)

$$\Pi(\cdot \mid R(X)) = \Pi(\cdot \mid R(\mathbf{1})) + \Pi(\cdot \mid R(\mathbb{X}_A - \Pi_{\mathbf{1}}\mathbb{X}_A))$$

Write

$$\mathbb{X}_{A,\perp} := \mathbb{X}_A - \Pi_{\mathbf{1}}\mathbb{X}_A$$

Note that

$$\Pi_{\mathbf{1}} = \mathbf{1}(\mathbf{1}^T\mathbf{1})^{-1}\mathbf{1}^T = \frac{1}{n}\mathbf{1}\mathbf{1}^T$$

Then

$$\begin{aligned} X\hat{\boldsymbol{\beta}} &= \hat{\beta}_0\mathbf{1} + \mathbb{X}_A\hat{\boldsymbol{\beta}}_A \\ &= \hat{\beta}_0\mathbf{1} + (\mathbb{X}_{A,\perp} + \Pi_{\mathbf{1}}\mathbb{X}_A)\hat{\boldsymbol{\beta}}_A \\ &= \mathbf{1}\Big(\hat{\beta}_0 + \frac{1}{n}\mathbf{1}^T\mathbb{X}_A\hat{\boldsymbol{\beta}}_A\Big) + \mathbb{X}_{A,\perp}\hat{\boldsymbol{\beta}}_A \end{aligned} \tag{2.3}$$

From (ii) of Theorem 2.4,

$$\begin{aligned} \Pi(\mathbf{Y} \mid R(X)) &= \Pi(\mathbf{Y} \mid R(\mathbf{1})) + \Pi(\mathbf{Y} \mid R(\mathbb{X}_{A,\perp})) \\ &= \overline{Y}\mathbf{1} + \mathbb{X}_{A,\perp}(\mathbb{X}_{A,\perp}^T\mathbb{X}_{A,\perp})^{-1}\mathbb{X}_{A,\perp}^T\mathbf{Y} \end{aligned} \tag{2.4}$$

Since $\mathbf{1} \perp \mathbb{X}_{A,\perp}$, Equations (2.3) and (2.4) imply that

$$\begin{cases} \hat{\beta}_0 = \overline{Y} - \frac{1}{n}\mathbf{1}^T\mathbb{X}_A\hat{\boldsymbol{\beta}}_A \\ \hat{\boldsymbol{\beta}}_A = (\mathbb{X}_{A,\perp}^T\mathbb{X}_{A,\perp})^{-1}\mathbb{X}_{A,\perp}^T\mathbf{Y} \end{cases} \tag{2.5}$$

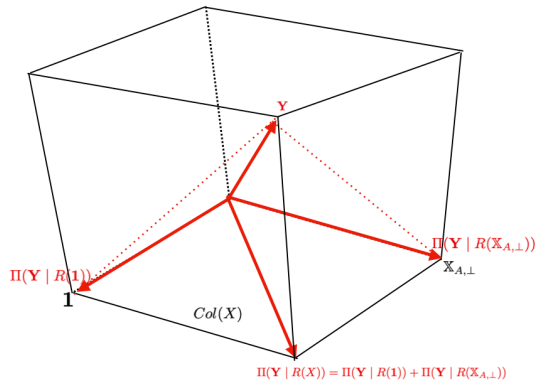


Figure 2.1: Orthogonal decomposition of the column space and LSE

See Figure 2.1. Two are orthogonal, so sum of projections onto them become LSE. In fact, *each projection indicate each regression coefficient*. When we do not have orthogonal basis, however, each projection is nothing.

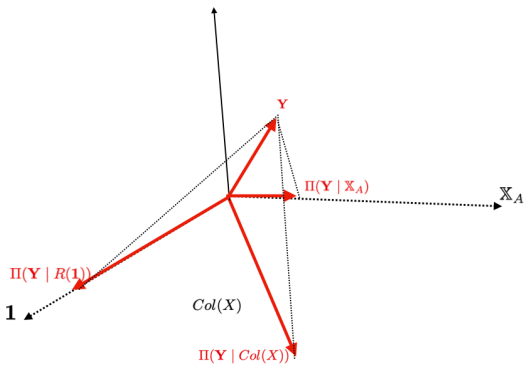


Figure 2.2: Non-orthongality

So what we have done is orthogonalization.

$$\tilde{\mathbb{X}}_A = \Pi_{\mathbf{1}}\mathbb{X}_A + (\mathbb{X}_A - \Pi_{\mathbf{1}}\mathbb{X}_A)$$

2.2.3 Gram-Schmidt QR factorization

Let's briefly look at orthogonalization process. From Theorem 2.2, we can derive following *orthonormalization process.*

Theorem 2.5 (Gram-Schmidt Process)**.** *Let* $\{\mathbf{x}_1, \ldots, \mathbf{x}_{p+1}\}$ *be a basis for the inner product space* V*. Let*

$$\mathbf{u}_1 = \left(\frac{1}{\|\mathbf{x}_1\|}\right)\mathbf{x}_1$$

and define next $\mathbf{u}_2, \ldots, \mathbf{u}_{p+1}$ *recursively by*

$$\mathbf{u}_{k+1} = \frac{1}{\|\mathbf{x}_{k+1} - \mathbf{r}_k^*\|}(\mathbf{x}_{k+1} - \mathbf{r}_k^*)$$

for $k = 1, \ldots, p$*, where*

$$\mathbf{r}_k^* = < \mathbf{x}_{k+1}, \mathbf{u}_1 > \mathbf{u}_1 + < \mathbf{x}_{k+1}, \mathbf{u}_2 > \mathbf{u}_2 + \cdots + < \mathbf{x}_{k+1}, \mathbf{u}_k > \mathbf{u}_k$$

is the projection of $\mathbf{x}_{k+1}$ *onto* $sp(\{\mathbf{u}_1, \ldots, \mathbf{u}_k\})$*.*

Hence, we get $\{\mathbf{u}_1, \ldots, \mathbf{u}_{p+1}\}$ *is an orthonormal basis for* V*.*

Algorithm 1: Gram-schmidt process
input: basis $\{\mathbf{x}_0, \ldots, \mathbf{x}_p\}$
Initialize $\mathbf{v}_0 = \mathbf{x}_0$;
for $k \leftarrow 1$ **to** p **do**
  $\mathbf{u}_{k-1} = \frac{\mathbf{v}_{k-1}}{\|\mathbf{v}_{k-1}\|}$;
  $\mathbf{r}_k^* = < \mathbf{x}_{k+1}, \mathbf{u}_0 > \mathbf{u}_0 + < \mathbf{x}_{k+1}, \mathbf{u}_1 > \mathbf{u}_1 + \cdots + < \mathbf{x}_{k+1}, \mathbf{u}_k > \mathbf{u}_k$;
  $\mathbf{v}_{k+1} = \mathbf{x}_{k+1} - \mathbf{r}_k^*$
end
$\mathbf{u}_p = \frac{\mathbf{v}_p}{\|\mathbf{v}_p\|}$

Our interest is $Col(X)$, and we can factorizae this model matrix so that it represents orthonormalization process 2.5.

Theorem 2.6 (Gram-Schmidt QR factorization)**.** *Let* $X \in \mathbb{R}^{n\times(p+1)}$*. Then* X *can be factored into*

$$X = QR$$

where $Q \in \mathbb{R}^{n\times(p+1)}$ is an orthogonal matrix, i.e. its column vectors are orthonormal and $R \in \mathbb{R}^{(p+1)\times(p+1)}$ is an upper triangular matrix whose diagonal entries are all positive.

Proof. Denote that this is just the representation of Gram-schmidt orthogonalization. Then it gives

$$\mathbf{u}_1 = \frac{\mathbf{x}_1}{\|\mathbf{x}_1\|} \Rightarrow \mathbf{x}_1 = \|\mathbf{x}_1\|\mathbf{u}_1$$

$$\begin{aligned}
&\mathbf{v}_2 = \mathbf{x}_2 - < \mathbf{x}_2, \mathbf{u}_1 > \mathbf{u}_1, \quad \mathbf{u}_2 = \frac{\mathbf{v}_2}{\|\mathbf{v}_2\|} \\
&\Rightarrow \mathbf{x}_2 = < \mathbf{x}_2, \mathbf{u}_1 > \mathbf{u}_1 + \|\mathbf{v}_2\|\mathbf{u}_2 \\
&\Rightarrow \mathbf{x}_2 = \begin{bmatrix} \mathbf{u}_1 \mid \mathbf{u}_2 \end{bmatrix} \begin{bmatrix} < \mathbf{x}_2, \mathbf{u}_1 > \\ \|\mathbf{v}_2\| \end{bmatrix}
\end{aligned}$$

It procees in a simlar way to the others. Hence,

$$\begin{aligned}
X &= \begin{bmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_{p+1} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_{p+1} \end{bmatrix} \begin{bmatrix} \|\mathbf{v}_1\| & < \mathbf{x}_2, \mathbf{u}_1 > & < \mathbf{x}_3, \mathbf{u}_1 > & \cdots & < \mathbf{x}_{p+1}, \mathbf{u}_1 > \\ 0 & \|\mathbf{v}_2\| & < \mathbf{x}_3, \mathbf{u}_2 > & \cdots & < \mathbf{x}_{p+1}, \mathbf{u}_2 > \\ 0 & 0 & \|\mathbf{v}_3\| & \cdots & < \mathbf{x}_{p+1}, \mathbf{u}_3 > \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \|\mathbf{v}_{p+1}\| \end{bmatrix} \\
&\equiv QR
\end{aligned} \tag{2.6}$$

□

Look again the equation in Theorem 2.5. In each process k, the projection is done to the $(k-1)$-dimensional space. In other words, as process goes through, dimension increases. So we try to project each vector only in 1-dimension each step.

Theorem 2.7 (Modified Gram-Schmidt Process)**.** *Let $\{\mathbf{x}_1, \ldots, \mathbf{x}_{p+1}\}$ be a basis for the inner product space V and let $\{\mathbf{q}_1, \ldots, \mathbf{q}_{p+1}\}$ be an orthonormal basis.*

Set $\mathbf{q}_1 = \frac{\mathbf{x}_1}{\|\mathbf{x}_1\|}$. Then consider $sp(\{\mathbf{q}_1\})$.

In the first step, make every $\{\mathbf{x}_2, \ldots, \mathbf{x}_{p+1}\}$ orthogonal to $\mathbf{q}_1$.

$$\mathbf{x}_k^{(1)} = \mathbf{x}_k - (\mathbf{q}_1^T \mathbf{x}_k)\mathbf{q}_1, \quad k = 2, \ldots, p+1$$

So we get orthogonal set $\{\mathbf{q}_1, \mathbf{x}_2^{(1)}, \ldots, \mathbf{x}_{p+1}^{(1)}\}$. *Next, set* $\mathbf{q}_2 = \frac{\mathbf{x}^{(1)}}{\|\mathbf{x}_2^{(2)}\|}$. *Consider* $sp(\{\mathbf{q}_2\})$. *Since we have* $\mathbf{q}_1 \perp \mathbf{q}_2$,

$$\mathbf{x}_k^{(2)} = \mathbf{x}_k^{(1)} - (\mathbf{q}_2^T \mathbf{x}_k^{(1)})\mathbf{q}_2 \perp \mathbf{q}_2, \quad k = 3, \ldots, p+1$$

Thus, get $\{\mathbf{q}_1, \mathbf{q}_2, \mathbf{x}_3^{(2)}, \ldots, \mathbf{x}_{p+1}^{(2)}\}$. $\mathbf{q}_3, \ldots, \mathbf{q}_{p+1}$ *are successively determined in a similary way.*

At the last step, set

$$\mathbf{q}_{p+1} = \frac{\mathbf{x}_{p+1}^{(p)}}{\|\mathbf{x}_{p+1}^{(p)}\|}$$

Since each projection is done in 1-dimension, the algorithm becomes more understandable. Consider

$$Q = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots \mathbf{q}_{p+1} \end{bmatrix} \in \mathbb{R}^{n \times (p+1)} \quad \text{orthogonal}$$

and

$$R = [r_{kj}] = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1,p+1} \\ 0 & r_{22} & \cdots & r_{2,p+1} \\ 0 & 0 & \cdots & r_{3,p+1} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & r_{p+1,p+1} \end{bmatrix} \in \mathbb{R}^{(p+1)\times(p+1)}$$

We can perform QR factorization by following step.

Algorithm 2: QR decomposition for modified G-S process

for $k \leftarrow 1$ **to** $(p+1)$ **do**
  $r_{kk} = \|\mathbf{x}_k\|$;
  $\mathbf{q}_k = \frac{\mathbf{x}_k}{r_{kk}}$;
  for $j \leftarrow 1$ **to** $(p+1)$ **do**
    $r_{kj} = \mathbf{q}_k^T \mathbf{x}_j$;
    $\mathbf{x}_j = \mathbf{x}_j - r_{kj}\mathbf{q}_k$;
  end
end

This *orthonormal basis* gives some useful facts with least squares problem (Leon, 2014).

2.2.4 Successive orthogonalization

In fact, G-S process 1 is equivalent to succesive orthogonalization, i.e. regress(project) $\mathbf{x}_j$ onto the others (Hastie et al., 2013).

Algorithm 3: Successive orthogonalization

Initialize $\mathbf{v}_0 = \mathbf{1}$;
for $k \leftarrow 1$ **to** p **do**
  Regress $\mathbf{x}_k$ on $\mathbf{q}_0, \ldots, \mathbf{q}_{k-1}$;
  $\hat{\beta}_{lk} = \frac{<\mathbf{v}_l, \mathbf{x}_k>}{<\mathbf{v}_l, \mathbf{v}_l>}, l = 0, \ldots, k-1$;
  Residual $\mathbf{v}_k = \mathbf{x}_k - \sum_{l=0}^{k-1} \hat{\beta}_{lk} \mathbf{v}_k$;
end
Regress $\mathbf{Y}$ on $\mathbf{v}_p$

Now we can solve least squares problem using QR decomposition. Recall that

$$X = QR$$

as specified in Theorem 2.6. Then normal equation implies that

$$\begin{aligned} & (X^T X)\hat{\boldsymbol{\beta}} = X^T \mathbf{Y} \\ \Leftrightarrow\ & R^T Q^T Q R \hat{\boldsymbol{\beta}} = R^T Q^T \mathbf{Y} \\ \Leftrightarrow\ & R^T R \hat{\boldsymbol{\beta}} = R^T Q^T \mathbf{Y} \qquad \because Q^T Q = I \\ \Leftrightarrow\ & R \hat{\boldsymbol{\beta}} = Q^T \mathbf{Y} \qquad \text{if } R \text{ is invertible} \end{aligned} \tag{2.7}$$

Hence,

$$\hat{\boldsymbol{\beta}} = R^{-1} Q^T \mathbf{Y} \tag{2.8}$$

It follows that

$$\hat{\mathbf{Y}} = (QR)\hat{\boldsymbol{\beta}} = QQ^T \mathbf{Y} \tag{2.9}$$

Let's compare the result. Base function `qr()` give the QR factorization. Given this object, we can get each Q and R by `qr.Q()` and `qr.R()`.

```
cem_qr <-
  cem %>%
  model.matrix(y ~ ., data = .) %>%
  qr()
```

```
cem_q <- qr.Q(cem_qr)
cem_r <- qr.R(cem_qr)
```

Using Equation (2.8), we get each coefficient as follow.

```
solve(cem_r) %*% t(cem_q) %*% cem$y
#>                 [,1]
#> (Intercept) 62.405
#> x1           1.551
#> x2           0.510
#> x3           0.102
#> x4          -0.144
```

On the other hand, `lm()` gives the following result.

```
lm(y ~ ., data = cem)
#>
#> Call:
#> lm(formula = y ~ ., data = cem)
#>
#> Coefficients:
#> (Intercept)           x1           x2           x3           x4
#>      62.405        1.551        0.510        0.102       -0.144
```

We can check the result is same. In fact, `lm()` fits the model by default `method = "qr"`.

> the method to be used; for fitting, currently only method = "qr" is supported; method = "model.frame" returns the model frame (the same as with model = TRUE, see below).

By default and only way, `lm()` fits the model using QR factorization. What does this orthogonal basis mean? For simplicity, consider simple linear regression problem.

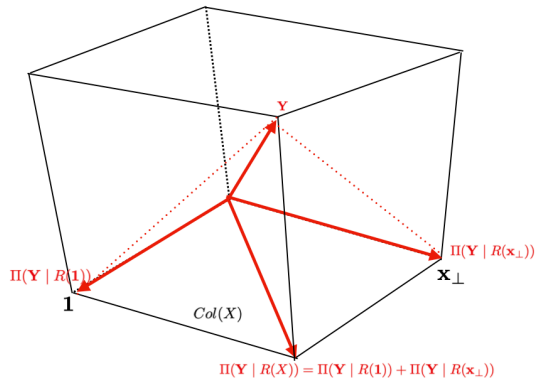


Figure 2.3: Orthogonalized basis

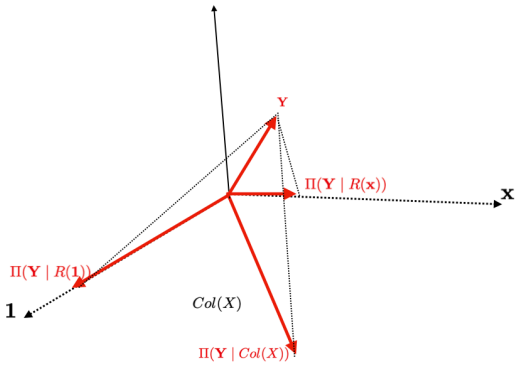


Figure 2.4: Non-orthogonal basis

See Figure 2.3. By construction, projection onto each basis is same as $\hat{\beta}_0$ and $\hat{\beta}_1$. In Figure 2.4, however, each projection is not regression coefficient.

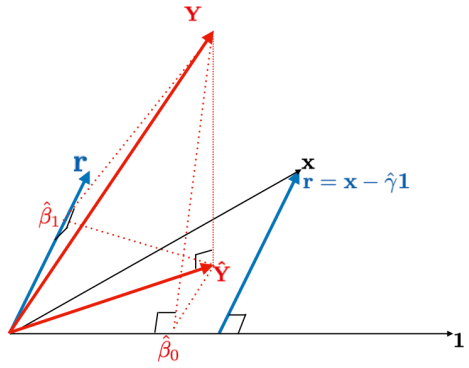


Figure 2.5: QR decomposition for model matrix

Regress **x** onto **1**. Its residual can be a new orthogonalized predictor.

2.2.5 Properties of LSE

We have seen how we extend point estimator $\hat{\boldsymbol{\beta}}$. In turn, we can check this is unbiased, and BLUE.

Proposition 2.1 (Expectation and Variance)**.** $\hat{\boldsymbol{\beta}}$ *is unbiased.*

1. $E\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}$
2. $Var\hat{\boldsymbol{\beta}} = \sigma^2(X^TX)^{-1}$

Proof.

$$\begin{aligned} E\hat{\boldsymbol{\beta}} &= E\Big((X^TX)^{-1}X^T\mathbf{Y}\Big) \\ &= (X^TX)^{-1}X^TE\mathbf{Y} \\ &= (X^TX)^{-1}X^TX\boldsymbol{\beta} \\ &= \boldsymbol{\beta} \end{aligned}$$

Hence, $\hat{\boldsymbol{\beta}}$ is unbiased.

$$\begin{aligned} Var\hat{\boldsymbol{\beta}} &= Var\Big((X^TX)^{-1}X^T\mathbf{Y}\Big) \\ &= (X^TX)^{-1}X^TVar(\mathbf{Y})X(X^TX)^{-1} \\ &= (X^TX)^{-1}X^T(\sigma^2I)X(X^TX)^{-1} \\ &= \sigma^2(X^TX)^{-1} \end{aligned}$$

□

Since the variance of LSE have been revealed, now we want to know if this is the lowest one among estimators. Gauss-Markov theorem states that LSE has the lowest variance among linear unbiased estimators for $\boldsymbol{\beta}$, so called the **best linear unbiased estimator (BLUE)**.

Theorem 2.8 (Gauss-Markov Theorem)**.** $\hat{\boldsymbol{\beta}} = (X^TX)^{-1}X^T\mathbf{Y}$ *is BLUE, i.e.*

For any $\tilde{\boldsymbol{\beta}} \in \Omega \equiv \{\tilde{\boldsymbol{\beta}} : \tilde{\boldsymbol{\beta}} = C\mathbf{Y}, E\tilde{\boldsymbol{\beta}} = \boldsymbol{\beta}\}$,

$$Var(\hat{\boldsymbol{\beta}}) \leq Var(\tilde{\boldsymbol{\beta}})$$

Proof. Let $\tilde{\boldsymbol{\beta}} \in \Omega \equiv \{\tilde{\boldsymbol{\beta}} : \tilde{\boldsymbol{\beta}} = C\mathbf{Y}, E\tilde{\boldsymbol{\beta}} = \boldsymbol{\beta}\}$

Claim: $Var(\tilde{\boldsymbol{\beta}}) - Var(\hat{\boldsymbol{\beta}})$ is non-negative definite.

Note that $\hat{\boldsymbol{\beta}}$ is the one with $C = (X^TX)^{-1}X^T$.

Set $D := C - (X^TX)^{-1}X^T$. From unbiasedness,

$$\begin{aligned} E\tilde{\boldsymbol{\beta}} &= CE\mathbf{Y} \\ &= CX\boldsymbol{\beta} \\ &= \Big((X^TX)^{-1}X^T + D\Big)X\boldsymbol{\beta} \\ &= \boldsymbol{\beta} + DX\boldsymbol{\beta} \\ &= \boldsymbol{\beta} \end{aligned}$$

Since $\forall \boldsymbol{\beta} \in \mathbb{R}^{p+1} : DX\boldsymbol{\beta} = \mathbf{0}$,

$$DX = 0 \tag{2.10}$$

$$
\begin{aligned}
Var\tilde{\boldsymbol{\beta}} &= Var(C\mathbf{Y}) \\
&= \sigma^2 CC^T \\
&= \sigma^2\Big((X^TX)^{-1}X^T + D\Big)\Big((X^TX)^{-1}X^T + D\Big)^T \\
&= \sigma^2\Big((X^TX)^{-1} + DX(X^TX)^{-1} + (X^TX)^{-1}X^TD^T + DD^T\Big) \\
&= \sigma^2\Big((X^TX)^{-1} + DD^T\Big) \qquad \because (2.10) \\
&= Var\hat{\boldsymbol{\beta}} + \sigma^2 DD^T
\end{aligned}
$$

Note that DD^T is non-negative definite. Hence,

$$Var\tilde{\boldsymbol{\beta}} - Var\hat{\boldsymbol{\beta}} = \sigma^2 DD^T$$

is non-negative definite. This completes the proof. □

As in simple linear regression setting, we define *residuals* and explain σ^2.

Definition 2.1 (Residuals)**.** Let $\hat{\mathbf{Y}} = (\hat{Y_1}, \ldots, \hat{Y_n})^T$ with $\hat{Y_i} = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_p x_{ip}$. Then the residual is defined by

$$\mathbf{e} := (\ldots, Y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \cdots - \hat{\beta}_p x_{ip}, \ldots)^T = \mathbf{Y} - \hat{\mathbf{Y}} \in \mathbb{R}^n$$

Extending the simple setting, we estimate σ^2 with inner product of residuals divided by its degrees of freedom, i.e. *MSE*. The degrees of freedom becomes $n -$ the number of coefficients. Thus, $n - (p+1) = n - p - 1$.

Proposition 2.2 (Estimation of σ^2)**.** *Let* $\mathbf{e} = \mathbf{Y} - \hat{\mathbf{Y}}$ *be residuals as in Definition 2.1. Then*

$$\hat{\sigma}^2 = \frac{\|\mathbf{e}\|}{n - p - 1}$$

The reason we divide with degrees of freedom is to make $\hat{\sigma}^2$ unbiased.

Proposition 2.3 (Mean of $\hat{\sigma}^2$)**.** $\hat{\sigma}^2$ *is unbiased, i.e.*

$$E\Big(\|\mathbf{Y} - \hat{\mathbf{Y}}\|^2\Big) = (n - p - 1)\sigma^2$$

Proof. Since $\mathbf{Y} = \Pi_X \mathbf{Y}$,

$$\begin{aligned}\|\mathbf{Y}-\hat{\mathbf{Y}}\|^2 &= \|(I-\Pi_X)\mathbf{Y}\|^2 \\ &= \mathbf{Y}^T(I-\Pi_X)^T(I-\Pi_X)\mathbf{Y} \\ &= \mathbf{Y}^T(I-\Pi_X)\mathbf{Y} \qquad \because (I-\Pi_X) : \text{symmetric idempotent}\end{aligned}$$

Since $\mathbf{Y}^T(I-\Pi_X)\mathbf{Y} \in \mathbb{R}$,

$$\begin{aligned}\mathbf{Y}^T(I-\Pi_X)\mathbf{Y} &= tr\Big(\mathbf{Y}^T(I-\Pi_X)\mathbf{Y}\Big) \\ &= tr\Big((I-\Pi_X)\mathbf{Y}\mathbf{Y}^T\Big)\end{aligned}$$

Then

$$\begin{aligned}E\|\mathbf{Y}-\hat{\mathbf{Y}}\|^2 &= E\Big[\mathbf{Y}^T(I-\Pi_X)\mathbf{Y}\Big] \\ &= E\Big[tr\Big((I-\Pi_X)\mathbf{Y}\mathbf{Y}^T\Big)\Big] \\ &= tr\Big((I-\Pi_X)\underbrace{E\Big[\mathbf{Y}\mathbf{Y}^T\Big]}_{(*)}\Big)\end{aligned}$$

Consider $(*)$.

$$\begin{aligned}E(\mathbf{Y}\mathbf{Y}^T) &= Var\mathbf{Y} + (E\mathbf{Y})(E\mathbf{Y})^T \\ &= \sigma^2 I + X\boldsymbol{\beta}\boldsymbol{\beta}^T X^T\end{aligned}$$

Hence,

$$\begin{aligned}E\|\mathbf{Y}-\hat{\mathbf{Y}}\|^2 &= tr\Big((I-\Pi_X)E\Big[\mathbf{Y}\mathbf{Y}^T\Big]\Big) \\ &= tr\Big((I-\Pi_X)\sigma^2 + (I-\Pi_X)X\boldsymbol{\beta}\boldsymbol{\beta}^T X^T\Big) \\ &= tr\Big((I-\Pi_X)\sigma^2\Big) + tr\Big(\boldsymbol{\beta}^T X^T(I-\Pi_X)X\boldsymbol{\beta}\Big) \qquad \because (I-\Pi_X)X\boldsymbol{\beta} = X\boldsymbol{\beta} - X\boldsymbol{\beta} = 0 \\ &= tr\Big((I-\Pi_X)\sigma^2\Big) \\ &= (n-p-1)\sigma^2 \qquad \because \begin{cases} tr(I) = n \\ tr(\Pi_X) = p+1 \end{cases}\end{aligned}$$

□

In this proposition, we have used model matrix directly. Instead, we can use Equation (2.8) for simplicity.

Proposition 2.4 (Variance using QR decomposition)**.** *Let $X = QR$. Then $\hat{\boldsymbol{\beta}} = R^{-1}Q^T\mathbf{Y}$. It follows that*

$$Var\hat{\boldsymbol{\beta}} = \sigma^2(R^TR)^{-1}$$

Proof. It proceeds in a similar way for $\hat{\boldsymbol{\beta}} = R^{-1}Q^T\mathbf{Y}$.

$$\begin{aligned} Var\hat{\boldsymbol{\beta}} &= Var\left(R^{-1}Q^T\mathbf{Y}\right) \\ &= R^{-1}Q^TVar(\mathbf{Y})Q(R^T)^{-1} \\ &= R^{-1}Q^T(\sigma^2I)Q(R^T)^{-1} \\ &= \sigma^2(R^TR)^{-1} \qquad \because Q^TQ = I \end{aligned}$$

□

Proposition 2.5 (QR representation for residual)**.** *Let $X = QR$. Then*

$$\mathbf{e} = (I - QQ^T)\mathbf{Y}$$

Proof. From Equation (2.9),

$$\Pi_X = QQ^T$$

Hence,

$$\mathbf{e} = (I - \Pi_X)\mathbf{Y} = (I - QQ^T)\mathbf{Y}$$

□

On Figure 2.5, we can see these relations. Operation $Q^T\mathbf{Y}$ is just projection to each orthogonal basis. Q sums these projection so that we get $\hat{\mathbf{Y}}$ which projection of $\mathbf{Y}$ onto the column space of model matrix.

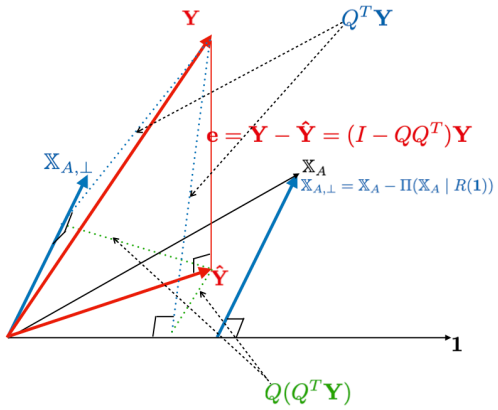


Figure 2.6: Residual vector

2.2.6 Mean response and response

Let $\mathbf{z} = (z_1, \ldots, z_p)^T$.

Theorem 2.9 (Estimation of the mean response)**.**

$$\hat{\mu}_z = \hat{\beta}_0 + \mathbf{z}^T \hat{\boldsymbol{\beta}}_A$$

Theorem 2.10 ((out of sample) Prediction of a response)**.**

$$\hat{Y}_z = \hat{\beta}_0 + \mathbf{z}^T \hat{\boldsymbol{\beta}}_Z$$

Proposition 2.6 (Residual vector)**.** *Let* $\mathbf{e} = (I - \Pi_X)\mathbf{Y}$*. Then*

1. $Var(\mathbf{e}) = \sigma^2(I - \Pi_X)$
2. $\mathbf{e} \perp \hat{\mathbf{Y}}$

$Var(\mathbf{e})$.

$$\begin{aligned} Var(\mathbf{e}) &= Var\Big((I - \Pi_X)\mathbf{Y}\Big) \\ &= (I - \Pi_X)Var(\mathbf{Y})(I - \Pi_X)^T \\ &= \sigma^2(I - \Pi_X) \qquad \because (I - \Pi_X) \text{ symmetric itempotent} \end{aligned}$$

□

$\mathbf{e} \perp \hat{\mathbf{Y}}$. Note that

$$\mathbf{e} \in Col(X)^{\perp}$$

From the properties of projection, we have

$$\begin{cases} \mathbf{e} \perp \mathbf{1} \\ \mathbf{e} \perp \mathbf{x}_j \quad \forall j = 1, 2, \ldots, p \end{cases} \tag{2.11}$$

Since $\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_p x_{ip}$,

$$\mathbf{e} \perp \hat{\mathbf{Y}}$$

□

Equation (2.11) is another form of the *normal equation*.

Remark. The least squares regression line $\{(\mathbf{z}, y) : y = \hat{\beta}_0 + \mathbf{z}^T \hat{\beta}_A\}$ always passes through

$$\left(\frac{1}{n}\mathbb{X}_A^T \mathbf{1}, \overline{Y}\right)$$

In simple linear regression setting,

$$(\overline{x}, \overline{Y})$$

Proof. First consider $p = 1$. Normal equation gives directly that

$$\hat{\beta}_0 = \overline{Y} - \hat{\beta}_1 \overline{x}$$

Thus,

$$\overline{Y} = \hat{\beta}_0 + \hat{\beta}_1 \overline{x}$$

We now give more general proof, i.e. for $p \geq 1$.

Claim: $\overline{Y} = \hat{\beta}_0 + \left(\frac{1}{n}\mathbb{X}_A^T \mathbf{1}\right)^T \hat{\boldsymbol{\beta}}_A$

From Equation (2.5),

$$\hat{\beta}_0 = \overline{Y} - \frac{1}{n}\mathbf{1}^T \mathbb{X}_A \hat{\boldsymbol{\beta}}_A$$

It follows that

$$\overline{Y} = \hat{\beta}_0 + \frac{1}{n}\mathbf{1}^T \mathbb{X}_A \hat{\boldsymbol{\beta}}_A$$

This completes the proof. □

2.3 Analysis of Variance

2.3.1 Decomposition of SST

We have

- $SST = \|\mathbf{Y} - \overline{Y}\mathbf{1}\|^2 = \mathbf{Y}^T(I - \Pi_{\mathbf{1}})\mathbf{Y}$
- $SSR = \|\hat{\mathbf{Y}} - \overline{Y}\mathbf{1}\|^2 = \mathbf{Y}^T(\Pi_X - \Pi_{\mathbf{1}})\mathbf{Y}$
- $SSE = \|\mathbf{Y} - \hat{\mathbf{Y}}\|^2 = \mathbf{Y}^T(I - \Pi_X)\mathbf{Y}$

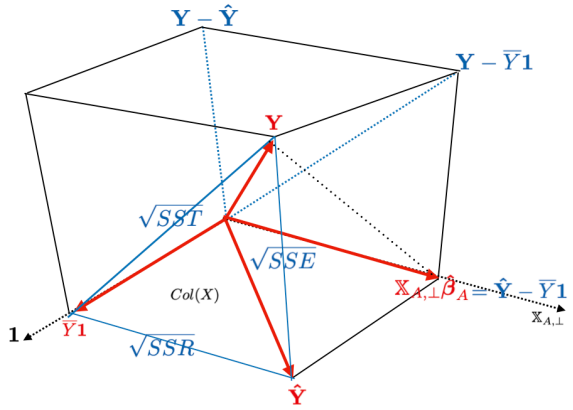


Figure 2.7: Decomposition of SST

In Figure 2.7, Pythagorean law gives

$$SST = SSR + SSE$$

Lemma 2.2. *Let* $X = [\mathbf{1} \mid \mathbb{X}_A]$ *and let* $\mathbb{X}_{A,\perp} = \mathbb{X}_A - \Pi_{\mathbf{1}}\mathbb{X}_A$. *Then*

$$\hat{\mathbf{Y}} - \overline{Y}\mathbf{1} = \mathbb{X}_{A,\perp}\hat{\boldsymbol{\beta}}_A$$

Proof. Note that $\mathbf{1} \perp \mathbb{X}_{A,\perp}$.

Recall that

$$\Pi(\mathbf{Y} \mid R(\mathbf{1})) = \mathbf{1}(\mathbf{1}^T\mathbf{1})^{-1}\mathbf{1}^T\mathbf{Y} = \overline{Y}\mathbf{1}$$

From Theorem 2.4,

$$\begin{aligned}\hat{\mathbf{Y}} &= \Pi(\mathbf{Y} \mid R(\mathbf{1})) + \Pi(\mathbf{Y} \mid R(\mathbb{X}_{A,\perp})) \\ &= \overline{Y}\mathbf{1} + \mathbb{X}_{A,\perp}\hat{\boldsymbol{\beta}}_A\end{aligned}$$

Hence,

$$\hat{\mathbf{Y}} - \overline{Y}\mathbf{1} = \mathbb{X}_{A,\perp}\hat{\boldsymbol{\beta}}_A$$

□

2.3.2 Distributions

Proposition 2.7 (Distribution of SS)**.** *Extending for $p > 1$, we can get each result.*

1. $\frac{SSE}{\sigma^2} \sim \chi^2(n-p-1)$
2. $\frac{SSR}{\sigma^2} \sim \chi^2(p, \delta), \quad \delta = \frac{\boldsymbol{\beta}^T X^T(\Pi_X - \Pi_\mathbf{1})X\boldsymbol{\beta}}{\sigma^2}$
3. $SSR \perp\!\!\!\perp SSE$
4. $SSE \perp\!\!\!\perp \hat{\boldsymbol{\beta}}$
5. $\frac{SST}{\sigma^2} \sim \chi^2(n-1, \delta), \quad \delta = \frac{\boldsymbol{\beta}^T X^T(I - \Pi_\mathbf{1})X\boldsymbol{\beta}}{\sigma^2}$

Distribution of SSE. Note that

$$SSE = \mathbf{Y}^T(I - \Pi_X)\mathbf{Y}$$

From Theorem 1.15,

$$K = rank(I - \Pi_X) = tr(I - \Pi_X) = n - rank(\Pi_X) = n - p - 1$$

δ proof is exactly same as Proposition 1.21.

$$\begin{aligned}\delta &= \left(\frac{X\boldsymbol{\beta}}{\sigma}\right)^T (I - \Pi_X)\left(\frac{X\boldsymbol{\beta}}{\sigma}\right) \\ &= \frac{\boldsymbol{\beta}^T X^T X\boldsymbol{\beta}}{\sigma^2} - \frac{(\boldsymbol{\beta}^T X^T)X(X^T X)^{-1}X^T(X\boldsymbol{\beta})}{\sigma^2} \\ &= \frac{\boldsymbol{\beta}^T X^T X\boldsymbol{\beta}}{\sigma^2} - \frac{\boldsymbol{\beta}^T X^T X\boldsymbol{\beta}}{\sigma^2} \\ &= 0\end{aligned}$$

Hence, $\delta = 0$. □

Distribution of SSR. Note that

$$SSR = \mathbf{Y}^T(\Pi_X - \Pi_{\mathbf{1}})\mathbf{Y}$$

From Theorem 1.15,

$$K = rank(\Pi_X - \Pi_{\mathbf{1}}) = tr(\Pi_X) - tr(\Pi_{\mathbf{1}}) = rank(\Pi_X) - rank(\Pi_{\mathbf{1}}) = p + 1 - 1 = p$$

δ proof is exactly same as Proposition 1.22.

$$\begin{aligned}\delta &= \left(\frac{X\boldsymbol{\beta}}{\sigma}\right)^T (\Pi_X - \Pi_{\mathbf{1}})\left(\frac{X\boldsymbol{\beta}}{\sigma}\right) \qquad \because \frac{\mathbf{Y}}{\sigma} \sim MVN(\frac{1}{\sigma}X\boldsymbol{\beta}, I) \\ &= \frac{\boldsymbol{\beta}^T\Big\{X^T(\Pi_X - \Pi_{\mathbf{1}})X\Big\}\boldsymbol{\beta}}{\sigma^2}\end{aligned} \tag{2.12}$$

This completes the proof. □

In univariate setting, we have seen that δ is expressed in terms of $\hat{\beta}_1$ excluding $\hat{\beta}_0$. How about in multivariate? In Equation (2.12), we can block design matrix X into

$$X = \big[\mathbf{1} \mid \mathbb{X}_A\big]$$

One proceeds in a similar way. Since both $\mathbf{1}, \mathbb{X}_A \in Col(X)$,

$$\Pi_X \mathbf{1} = \mathbf{1}, \qquad \Pi_X \mathbb{X}_A = \mathbb{X}_A$$

Since $\mathbf{1} \in R(\{\mathbf{1}\})$,

$$\Pi_{\mathbf{1}}\mathbf{1} = \mathbf{1}$$

Since $\Pi_{\mathbf{1}} = \frac{1}{n}\mathbf{1}\mathbf{1}^T$,

$$\Pi_{\mathbf{1}}\mathbb{X}_A = \mathbf{1}\overline{\mathbf{x}}^T = \begin{bmatrix} \overline{x}_1 & \overline{x}_2 & \cdots & \overline{x}_p \\ \overline{x}_1 & \overline{x}_2 & \cdots & \overline{x}_p \\ \vdots & \vdots & \vdots & \vdots \\ \overline{x}_1 & \overline{x}_2 & \cdots & \overline{x}_p \end{bmatrix}$$

Using these facts, we have following

$$\mathbf{1}^T(\Pi_X - \Pi_{\mathbf{1}})\mathbf{1} = \mathbf{1}^T(\mathbf{1} - \mathbf{1}) = 0 \tag{2.13}$$

$$\begin{aligned} \mathbf{1}^T(\Pi_X - \Pi_{\mathbf{1}})\mathbb{X}_A &= \mathbf{1}^T(\mathbb{X}_A - \mathbf{1}\overline{\mathbf{x}}^T) \\ &= \mathbf{1}^T \left[x_{ij} - \overline{x}_j\right]_{1\times j} \\ &= \left[\textstyle\sum_i x_{ij} - n\overline{x}_j\right]_{1\times j} \\ &0 \end{aligned} \tag{2.14}$$

$$\mathbb{X}_A^T(\Pi_X - \Pi_{\mathbf{1}})\mathbf{1} = \mathbb{X}_A^T(\mathbf{1} - \mathbf{1}) = 0 \tag{2.15}$$

From Lemma 1.1,

$$\begin{aligned} \mathbb{X}_A^T(\Pi_X - \Pi_{\mathbf{1}})\mathbb{X}_A &= \mathbb{X}_A^T \left[x_{ij} - \overline{x}_j\right]_{1\times j} \\ &= \left[\textstyle\sum_i x_{ij}(x_{jk} - \overline{x}_k)\right]_{j\times k} \\ &= \left[\textstyle\sum_i (x_{ij} - \overline{x}_j)(x_{ik} - \overline{x}_k)\right]_{j\times k} \\ &= (n-1)Var(\mathbb{X}_A) \\ &\equiv S \end{aligned} \tag{2.16}$$

Hence,

$$\begin{aligned}
\delta &= \frac{\boldsymbol{\beta}^T \Big\{ \left[\mathbf{1} \mid \mathbb{X}_A\right]^T (\Pi_X - \Pi_{\mathbf{1}}) \left[\mathbf{1} \mid \mathbb{X}_A\right] \Big\} \boldsymbol{\beta}}{\sigma^2} \\
&= \frac{\boldsymbol{\beta}^T \left[\begin{array}{c|c} (2.13) & (2.14) \\ \hline (2.15) & (2.16) \end{array}\right] \boldsymbol{\beta}}{\sigma^2} \\
&= \frac{\boldsymbol{\beta}^T \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & S \end{array}\right] \boldsymbol{\beta}}{\sigma^2} \\
&= \frac{\boldsymbol{\beta}_A^T S \boldsymbol{\beta}_A}{\sigma^2}
\end{aligned} \tag{2.17}$$

Independence between SSE and SSR. Since SSE and SSR are quadratic form of $\mathbf{Y} \sim MVN(X\boldsymbol{\beta}, \sigma^2 I)$ and each $I - \Pi_X$ and $\Pi_X - \Pi_{\mathbf{1}}$ is symmetric,

Claim: $(I - \Pi_X)(\Pi_X - \Pi_{\mathbf{1}}) = 0$

We have already shown this in Proposition 1.23. Using the fact that $\Pi_X \Pi_{\mathbf{1}} = \Pi_{\mathbf{1}}$,

$$\begin{aligned}
(I - \Pi_X)(\Pi_X - \Pi_{\mathbf{1}}) &= \Pi_X - \Pi_{\mathbf{1}} - \Pi_X^2 + \Pi_X \Pi_{\mathbf{1}} \\
&= \Pi_X - \Pi_{\mathbf{1}} - \Pi_X + \Pi_{\mathbf{1}} \qquad \because \text{idempotent} \\
&= 0
\end{aligned}$$

□

Independence between SSE and regression vector. The proof is same as 1.24, by showing that $((X^TX)^{-1}X^T)(I - \Pi_X) = 0$.

Since $\Pi_X = X(X^TX)^{-1}X^T$,

$$((X^TX)^{-1}X^T)(I - \Pi_X) = (X^TX)^{-1}X^T - (X^TX)^{-1}X^T = 0$$

This completes the proof. □

Distribution of SST. Note that

$$SST = \mathbf{Y}^T (I - \Pi_{\mathbf{1}}) \mathbf{Y}$$

From Theorem 1.15,

$$K = rank(I - \Pi_{\mathbf{1}}) = tr(I - \Pi_{\mathbf{1}}) = n - 1$$

and

$$\begin{aligned}\delta &= \left(\frac{X\boldsymbol{\beta}}{\sigma}\right)^T (I - \Pi_{\mathbf{1}})\left(\frac{X\boldsymbol{\beta}}{\sigma}\right) \\ &= \frac{\boldsymbol{\beta}^T X^T (I - \Pi_{\mathbf{1}}) X\boldsymbol{\beta}}{\sigma^2}\end{aligned}$$

□

2.3.3 ANOVA for testing significance of regression

Under the normality of error term

$$\boldsymbol{\epsilon}_i \sim MVN(\mathbf{0}, \sigma^2 I)$$

a test statistic can follow F-distribution as in univariate setting.

Corollary 2.1 (F-test)**.** *Under normality,*

$$F = \frac{SSR/p}{SSE/(n-p-1)} \sim F(p, n-p-1, \delta)$$

where

$\delta = \frac{\boldsymbol{\beta}^T X^T (\Pi_X - \Pi_1) X\boldsymbol{\beta}}{\sigma^2}$

In the proof of 1.22, we can understand the structure that $\delta = 0$ when all coefficients corresponding to predictors are zero.

$$F \overset{H_0}{\sim} F(p, n-p-1)$$

where

$$H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0$$

As other ordinary test, we just reject this H_0 if observed F_0 is large, i.e.

$$F_0 > F_\alpha(p, n-p-1)$$

See Figure 1.10. **ANOVA table** summarizes these statistics in table form.

Source	SS	df	MS	F	p-value
Model	SSR	p	$MSR = \frac{SSR}{p}$	$F_0 = \frac{MSR}{MSE}$	p-value
Error	SSE	$n-p-1$	$MSE = \frac{SSE}{n-p-1}$		
Total	SST	$n-1$			

Everything is same in R.

```
cem_fit <- lm(y ~ ., data = cem)
summary(cem_fit)
#>
#> Call:
#> lm(formula = y ~ ., data = cem)
#>
#> Residuals:
#>    Min     1Q Median     3Q    Max
#> -3.175 -1.671  0.251  1.378  3.925
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)   62.405     70.071    0.89    0.399
#> x1             1.551      0.745    2.08    0.071 .
#> x2             0.510      0.724    0.70    0.501
#> x3             0.102      0.755    0.14    0.896
#> x4            -0.144      0.709   -0.20    0.844
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 2.45 on 8 degrees of freedom
#> Multiple R-squared:  0.982,  Adjusted R-squared:  0.974
#> F-statistic:  111 on 4 and 8 DF,  p-value: 4.76e-07
```

You can see `F-statistic` with degrees of freedom 4 8.

```
summary(cem_fit)$fstatistic
#> value numdf dendf
#>   111     4     8
```

However, `anova.lm()` gives a bit different format This is related to *extra sum of squares*, which will be covered later.

```
anova(cem_fit)
#> Analysis of Variance Table
#>
#> Response: y
#>           Df Sum Sq Mean Sq F value  Pr(>F)
#> x1         1   1450    1450  242.37 2.9e-07 ***
#> x2         1   1208    1208  201.87 5.9e-07 ***
#> x3         1     10      10    1.64    0.24
#> x4         1      0       0    0.04    0.84
#> Residuals  8     48       6
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2.3.4 Coefficient of determination

In univariate setting, coefficient of determination measures linear relationship based on the SST decomposition.

$$R^2 = \frac{SSR}{SST} = \hat{\rho} = \cos\theta$$

Moreover, it becomes to be same as sample correlation $\hat{\rho}$ and angle between two vectors. In multivariate setting, we also define this kind of measure.

Definition 2.2 (Coefficient of Determination)**.** For $\mathbf{Y}_i = X\boldsymbol{\beta} + \epsilon_i$,

$$R^2 := \max_{\boldsymbol{\beta} \in \mathbb{R}^{p+1}} \hat{\rho}(\mathbf{Y}, X\boldsymbol{\beta})$$

where $\hat{\rho}$ means sample correlation.

We have mentioned about $R^2 = (\cos\theta)^2$ in simple linear regression. See Equation (1.17). Now we try to see detail of this relation. First, in Leon (2014), you might see the relation of $\cos\theta$ and inner product.

Lemma 2.3. *If $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ are two nonzero vectors and θ is the angle between them, then*

$$\mathbf{x}^T\mathbf{y} = \|\mathbf{x}\|\|\mathbf{y}\|\cos\theta$$

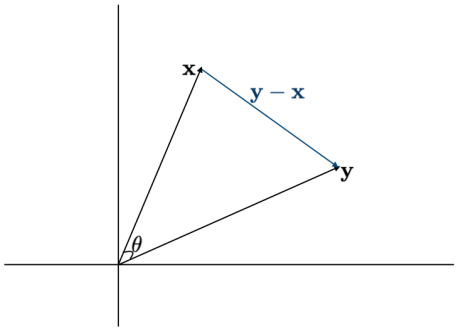


Figure 2.8: Two vectors in $\mathbb{R}^2$

Proof. See Figure 2.8. We have a triangle. Law of cosines gives that

$$\|\mathbf{y}-\mathbf{x}\|^2 = \|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - 2\|\mathbf{x}\|\|\mathbf{y}\|\cos\theta$$

It follows that

$$\begin{aligned}\|\mathbf{x}\|\|\mathbf{y}\|\cos\theta &= \frac{1}{2}(\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - \|\mathbf{y}-\mathbf{x}\|^2) \\ &= \frac{1}{2}(\mathbf{x}^T\mathbf{x} + \mathbf{y}^T\mathbf{y} - (\mathbf{y}-\mathbf{x})^T(\mathbf{y}-\mathbf{x})) \\ &= \mathbf{x}^T\mathbf{y}\end{aligned}$$

□

This implies the *relationship between sample correlation and the angle.*

Theorem 2.11 (Sample correlation and the Angle)**.** *Let* $\mathbf{X} = (X_1, \ldots, X_n)^T$ *and* $\mathbf{Y} = (Y_1, \ldots, Y_n)^T$ *be random variables. Then*

$$\hat{\rho}(\mathbf{X}, \mathbf{Y}) = cos\theta$$

where θ *is the angle between* $\mathbf{X} - \overline{X}\mathbf{1}$ *and* $\mathbf{Y} - \overline{Y}\mathbf{1}$*.*

Proof. Note that

$$\widehat{Cov}(\mathbf{X}, \mathbf{Y}) = \frac{1}{n-1}(\mathbf{X} - \overline{X}\mathbf{1})^T(\mathbf{Y} - \overline{Y}\mathbf{1})$$

$$\hat{\sigma}_{\mathbf{X}} = \sqrt{\frac{1}{n-1}(\mathbf{X} - \overline{X}\mathbf{1})^T(\mathbf{X} - \overline{X}\mathbf{1})}$$

and

$$\hat{\sigma}_{\mathbf{Y}} = \sqrt{\frac{1}{n-1}(\mathbf{Y} - \overline{Y}\mathbf{1})^T(\mathbf{Y} - \overline{Y}\mathbf{1})}$$

and hence it follows that

$$\begin{aligned}\hat{\rho}(\mathbf{X},\mathbf{Y}) &= \frac{\widehat{Cov}(\mathbf{X},\mathbf{Y})}{\hat{\sigma}_{\mathbf{X}}\hat{\sigma}_{\mathbf{Y}}} \\ &= \frac{(\mathbf{X}-\overline{X}\mathbf{1})^T(\mathbf{Y}-\overline{Y}\mathbf{1})}{\sqrt{(\mathbf{X}-\overline{X}\mathbf{1})^T(\mathbf{X}-\overline{X}\mathbf{1})}\sqrt{(\mathbf{Y}-\overline{Y}\mathbf{1})^T(\mathbf{Y}-\overline{Y}\mathbf{1})}} \\ &= \frac{<\mathbf{X}-\overline{X}\mathbf{1},\mathbf{Y}-\overline{Y}\mathbf{1}>}{\|\mathbf{X}-\overline{X}\mathbf{1}\|\|\mathbf{Y}-\overline{Y}\mathbf{1}\|} \\ &= cos\theta\end{aligned}$$

where θ is the angle between $\mathbf{X}-\overline{X}\mathbf{1}$ and $\mathbf{Y}-\overline{Y}\mathbf{1}$. □

Using this fact, we can finally derive that

$$\hat{\rho}(\mathbf{Y}, X\boldsymbol{\beta}) = \cos\theta \tag{2.18}$$

where θ is the angle between $\mathbf{Y}-\overline{Y}\mathbf{1}$ and

$$\beta_1(\mathbf{x}_1-\overline{x}_1\mathbf{1})+\cdots\beta_p(\mathbf{x}_p-\overline{x}_p\mathbf{1}) = \mathbb{X}_{A,\perp}\boldsymbol{\beta}_A \in Col(\mathbb{X}_{A,\perp})$$

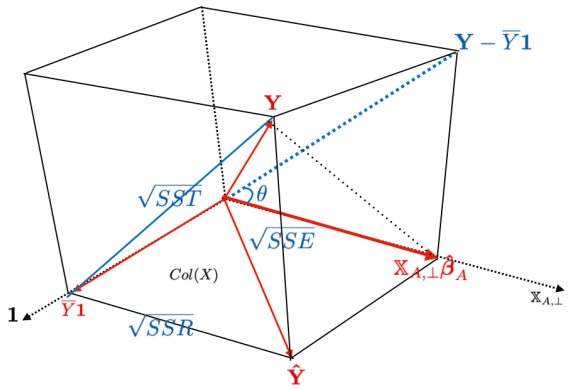


Figure 2.9: R^2 and Projection

See Figure 2.9. θ is marked on Figure 2.7 setting. In this setting, we know that $\theta < \frac{\pi}{2}$ is minimized by projection onto $\mathbb{X}_{A,\perp}$. This means that $\cos\theta$ is maximized. In other words,

$$R^2 = \frac{SSR}{SST} = (\cos\theta)^2$$

is maximized by the projection of $\mathbf{Y} - \overline{Y}\mathbf{1}$ onto $Col(\mathbb{X}_{A,\perp})$. Thus, R^2 can be interpreted as *proportion of variability of Y that is explained by the set of x_js.* It is obvious that $0 \leq R^2 \leq 1$.

R^2 becomes larger if a set of $\mathbb{X}_{A,\perp}$ explains the response well. Is it proper to use this measure as judging *goodness-of-fit*? However, this is not a good measure for model comparison. Model comparison includes different number of predictors. *SSE, however, always decreases when new X_j is added,* while $SST = \sum(Y_i - \overline{Y})$ never changes given Y data. This leads

$$R^2 = 1 - \frac{SSE}{SST}$$

always becomes larger by more predictors. For example,

$$\begin{cases} Y = \beta_0 + \beta_1 X_1 + \epsilon \\ Y = \beta_0 + \beta_1 X_1 + \beta_1 X_2 + \epsilon \end{cases}$$

Whether X_2 additionaly contributes to Y significantly, R^2 increases and we could judge that second model is better than first one. Hence to use this properly, we need some adjustment. As $p+1$ increases, this adjustment should become smaller:

$$\frac{n-1}{n-p-1}$$

Definition 2.3 (Adjusted Rsquared)**.**

$$R_a^2 := 1 - \frac{SSE/(n-p-1)}{SST/(n-1)}$$

Remark (Adjustment).

$$R_a^2 = 1 - \frac{n-1}{n-p-1}(1-R^2)$$

Proof. Note that

$$R^2 = 1 - \frac{SSE}{SST}$$

and hence,

$$\begin{aligned} R_a^2 &:= 1 - \frac{SSE/(n-p-1)}{SST/(n-1)} \\ &= 1 - \frac{n-1}{n-p-1}\left(\frac{SSE}{SST}\right) \\ &= 1 - \frac{n-1}{n-p-1}(1-R^2) \end{aligned}$$

□

So R_a^2 becomes a useful measure for the goodness-of-fit. On the contrary, *it cannot be interpreted as the proportion of total variation in Y that is explained by $X_1, \ldots, X_p$.*

2.4 Distributions

2.4.1 Individual Regression coefficients

Under Normality,

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

Since $\hat{\boldsymbol{\beta}}$ is an unbiased estimator, Proposition 2.1 implies that

$$\hat{\boldsymbol{\beta}} \sim MVN\Big(\boldsymbol{\beta}, \sigma^2 (X^T X)^{-1}\Big) \tag{2.19}$$

Let $C \equiv (X^T X)^{-1}$. Then

$$\begin{bmatrix} \hat{\beta}_0 \\ \hline \hat{\beta}_1 \\ \hline \vdots \\ \hline \hat{\beta}_p \end{bmatrix} \sim MVN\left(\begin{bmatrix} \beta_0 \\ \hline \beta_1 \\ \hline \vdots \\ \hline \beta_p \end{bmatrix}, \sigma^2 \begin{bmatrix} c_{00} & \cdots & \cdots \\ \hline \cdots & c_{11} & \cdots \\ \hline \vdots & \vdots & \vdots \\ \hline \cdots & \cdots & c_{pp} \end{bmatrix} \right)$$

i.e. Individual coefficient follows

$$\hat{\beta}_j \sim N(\beta_j, c_{jj}\sigma^2), \quad j = 0, 1, \ldots, p$$

where c_{jj} is $j+1$th diagonal element of $C = (X^T X)^{-1}$. Then

$$\frac{\hat{\beta}_k - \beta_k}{\sigma\sqrt{c_{kk}}} \sim N(0,1) \tag{2.20}$$

From Proposition 2.7,

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

Since $\hat{\sigma}^2 = \frac{SSE}{n-p-1}$ and $\hat{\beta}_k \perp\!\!\!\perp \hat{\sigma}^2$,

$$\frac{\hat{\beta}_k - \beta_k}{\hat{\sigma}\sqrt{c_{kk}}} = \frac{(\hat{\beta}_k - \beta_k)/(\sigma\sqrt{c_{kk}}) \sim N(0,1)}{\sqrt{\frac{SSE}{\sigma^2}/(n-p-1) \sim \chi^2(n-p-1)}} \sim t(n-p-1), \quad k = 0, 1, \ldots, p \tag{2.21}$$

2.4.2 Mean response

Consider prediction at $\mathbf{z} = (1, z_1, \ldots, z_p)^T$

Mean response targets

$$\hat{\mu}_{\mathbf{z}} = \mathbf{z}^T \hat{\boldsymbol{\beta}}$$

From Equation (2.19),

$$\hat{\mu}_{\mathbf{z}} \sim N(\mathbf{z}^T \boldsymbol{\beta}, \sigma^2 \mathbf{z}^T (X^T X)^{-1} \mathbf{z})$$

Set

$$C_{\mathbf{z}} := \mathbf{z}^T (X^T X)^{-1} \mathbf{z} \tag{2.22}$$

Then by standardization,

$$\frac{\hat{\mu}_{\mathbf{z}} - \mu_{\mathbf{z}}}{\sqrt{C_{\mathbf{z}} \sigma^2}} \sim N(0,1) \tag{2.23}$$

Hence,

$$\frac{\hat{\mu}_{\mathbf{z}} - \mu_{\mathbf{z}}}{\sqrt{C_{\mathbf{z}} \hat{\sigma}^2}} \sim t(n-p-1) \tag{2.24}$$

2.4.3 Response

Now we target $\mathbf{Y}$ at $\mathbf{z} = (1, z_1, \ldots, z_p)^T$ *out-of-sample.* $\boldsymbol{\epsilon}_\mathbf{z}$ at this point is independent of the data set.

Consider

$$\hat{Y}_\mathbf{z} - Y_\mathbf{z} = \mathbf{z}^T(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) + \epsilon_\mathbf{z} \tag{2.25}$$

As in Proposition 1.7, it can be proven that

$$\begin{bmatrix} \hat{\boldsymbol{\beta}} - \boldsymbol{\beta} \\ \epsilon_\mathbf{z} \end{bmatrix} \sim MVN$$

by showing marginal follows Normal and two are independent. First part - marginal follows normal distribution - has been already shown and assumed. Since these are Normal, it is enough to show covariance is zero.

$$\begin{aligned} Cov((\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}), \epsilon_\mathbf{z}) &= Cov\Big((X^TX)^{-1}X^TY, \epsilon_\mathbf{z}\Big) \\ &= (X^TX)^{-1}X^TCov(Y, \epsilon_\mathbf{z}) \\ &= 0 \end{aligned}$$

Hence, the joint distribution

$$\left[\begin{array}{c} \hat{\boldsymbol{\beta}} - \boldsymbol{\beta} \\ \hline \epsilon_\mathbf{z} \end{array}\right] \sim MVN\left(\mathbf{0}, \left[\begin{array}{c|c} \sigma^2(X^TX)^{-1} & 0 \\ \hline 0 & \sigma^2 \end{array}\right]\right)$$

From Equation (2.25),

$$\begin{aligned} \hat{Y}_\mathbf{z} - Y_\mathbf{z} &= \left[\begin{array}{c|c} \mathbf{z}^T & 1 \end{array}\right] \left[\begin{array}{c} \hat{\boldsymbol{\beta}} - \boldsymbol{\beta} \\ \hline \epsilon_\mathbf{z} \end{array}\right] \\ &\sim N\left(\mathbf{0}, \left[\begin{array}{c|c} \mathbf{z}^T & 1 \end{array}\right] \left[\begin{array}{c|c} \sigma^2(X^TX)^{-1} & 0 \\ \hline 0 & \sigma^2 \end{array}\right] \left[\begin{array}{c} \mathbf{z} \\ 1 \end{array}\right]\right) \\ &\stackrel{d}{=} N\Big(\mathbf{0}, \sigma^2\mathbf{z}^T(X^TX)^{-1}\mathbf{z} + \sigma^2\Big) \end{aligned}$$

Using notation from Equation (2.22), we get

$$\hat{\mathbf{Y}}_\mathbf{z} - \mathbf{Y}_\mathbf{z} \sim N\Big(\mathbf{0}, (C_\mathbf{z} + 1)\sigma^2\Big) \tag{2.26}$$

Then standardization gives that

$$\frac{\hat{Y}_{\mathbf{z}} - Y_{\mathbf{z}}}{\sqrt{(C_{\mathbf{z}} + 1)\sigma^2}} \sim N(0, 1) \tag{2.27}$$

and hence,

$$\frac{\hat{Y}_{\mathbf{z}} - Y_{\mathbf{z}}}{\sqrt{(C_{\mathbf{z}} + 1)\hat{\sigma}^2}} \sim t(n - p - 1) \tag{2.28}$$

Compare Statistic (2.28) with Statistic (2.24). We can see that out-of-sample one has larger standard error by σ^2 with same degrees of freedom. This is same as simple regression setting. Error of mean response only comes from $\hat{\boldsymbol{\beta}}$. When we predict out-of-sample individual, however, $Var(\epsilon_{\mathbf{z}})$ is added. Denote that this $\epsilon_{\mathbf{z}}$ should be independent with given data. Otherwise, we cannot get the distribution as ordinary (Johnson and Wichern, 2013).

2.5 Statistical Inference

We have derived basic distributions, so we try to test or build a confidence interval.

2.5.1 Individual Regression coefficients

From Statistic (2.21), we can easily make $(1 - \alpha)100\%$ confidence interval $\hat{\theta} \pm t_{\frac{\alpha}{2}} SE$ and test statistic.

Theorem 2.12 ($(1 - \alpha)100\%$ Confidence interval)**.** $(1 - \alpha)100\%$ *confidence interval for each individual* β_j *is*

$$\left[\hat{\beta}_j \pm t_{\frac{\alpha}{2}}(n - p - 1)\hat{\sigma}\sqrt{c_{kk}}\right]$$

In fact, we have already seen the test statistic form.

Theorem 2.13 (Partial t-test)**.** *Test* $H_0 : \beta_k = \alpha_k \quad vs \quad H_1 : \beta_k \neq \alpha_k$*. For given data, partial t-test computes*

$$T_0 = \frac{\hat{\beta}_k - \alpha_k}{\hat{\sigma}\sqrt{c_{kk}}} \sim t(n - p - 1)$$

where $(X^T X)^{-1} = (c_{ij})_{0 \leq i,j \leq p}$

As usual test, we reject H_0 when

$$|T_0| > t_{\frac{\alpha}{2}}(n-p-1)$$

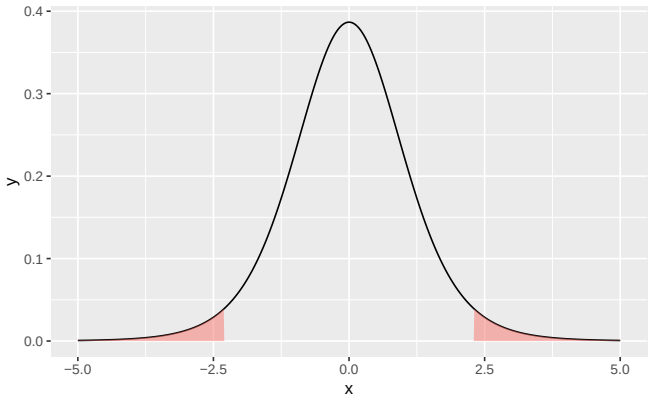


Figure 2.10: Rejection region for β_k

If we use `summary()` to `lm` object, we can get each T statistic(`t value`), standard error(`Std. Error`), and p-value(`Pr(>|t|)`). These are the results of partial t-test.

```
summary(cem_fit)
#>
#> Call:
#> lm(formula = y ~ ., data = cem)
#>
#> Residuals:
#>    Min     1Q Median     3Q    Max
#> -3.175 -1.671  0.251  1.378  3.925
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)   62.405     70.071    0.89    0.399
#> x1             1.551      0.745    2.08    0.071 .
#> x2             0.510      0.724    0.70    0.501
#> x3             0.102      0.755    0.14    0.896
#> x4            -0.144      0.709   -0.20    0.844
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 2.45 on 8 degrees of freedom
```

```
#> Multiple R-squared:  0.982,  Adjusted R-squared:  0.974
#> F-statistic:  111 on 4 and 8 DF,  p-value: 4.76e-07
```

If the test is significant, it means that additional contribution of that variable is significant after all other variables are already in the model. This might be understood well with extra sum of squares concept, later.

2.5.2 Prediction interval

Consider prediction at $\mathbf{z} = (1, z_1, \ldots, z_p)^T$. Write

$$C_{\mathbf{z}} := \mathbf{z}^T (X^T X)^{-1} \mathbf{z}$$

Theorem 2.14 (Prediction or Confidence interval for mean response)**.** $(1 - \alpha)100\%$ *prediction interval for* $\mu_{\mathbf{z}}$ *is*

$$\left[\hat{\mu}_{\mathbf{z}} \pm t_{\frac{\alpha}{2}} (n - p - 1) \sqrt{C_{\mathbf{z}} \hat{\sigma}^2} \right]$$

Theorem 2.15 (Out-of-sample prediction interval)**.** $(1 - \alpha)100\%$ *prediction interval for* $Y_{\mathbf{z}}$ *is*

$$\left[\hat{Y}_{\mathbf{z}} \pm t_{\frac{\alpha}{2}} (n - p - 1) \sqrt{(C_{\mathbf{z}} + 1) \hat{\sigma}^2} \right]$$

See standard error part of Theorem 2.14 and Theorem 2.15. As mentioned, Out of sample prediction interval *always has larger standard error.* This leads to wider interval.

2.5.3 Regression coefficient vector

Now we consider the coefficients simultaneously. For example, $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^T$.

Note that

$$\hat{\boldsymbol{\beta}} - \boldsymbol{\beta} \sim MVN(\mathbf{0}, \sigma^2 (X^T X)^{-1})$$

Then standardization gives

$$\mathbf{Z} \equiv \frac{(X^T X)^{\frac{1}{2}}}{\sigma} (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \sim MVN(\mathbf{0}, I)$$

It follows that

$$\mathbf{Z}^T\mathbf{Z} = \frac{(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T X^T X (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})}{\sigma^2} \sim \chi^2(p+1)$$

Since $\frac{SSE}{\sigma^2} \sim \chi^2(n-p-1)$ and $\hat{\sigma}^2 = MSE$,

$$\begin{aligned} \frac{(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T X^T X (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})}{\hat{\sigma}^2(p+1)} &= \frac{(X\hat{\boldsymbol{\beta}} - X\boldsymbol{\beta})^T (X\hat{\boldsymbol{\beta}} - X\boldsymbol{\beta})}{MSE} \\ &= \frac{SSR/(p+1)}{SSE/(n-p-1)} \\ &\sim F(p+1, n-p-1, \delta) \end{aligned} \tag{2.29}$$

where $\delta = \frac{\boldsymbol{\beta}^T X^T (I - \Pi_1) X \boldsymbol{\beta}}{\sigma^2}$.

Corollary 2.2 (F-test)**.** *Test* $H_0 : \boldsymbol{\beta} = \mathbf{0}$ *vs* $H_1 : \boldsymbol{\beta} \neq \mathbf{0}$*. For given data,* F*-test computes*

$$\frac{(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T X^T X (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})}{\hat{\sigma}^2(p+1)} = \frac{SSR/(p+1)}{MSE} \overset{H_0}{\sim} F(p+1, n-p-1)$$

From the first part, we can get the confidence region for $\boldsymbol{\beta}$.

Theorem 2.16 (Confidence region)**.** $(1-\alpha)100\%$ *confidence interval for* $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^T$ *is*

$$\left\{ \boldsymbol{\beta} : (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T X^T X (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \leq (p+1)\hat{\sigma}^2 F_{1-\alpha}(p+1, n-p-1) \right\}$$

Remark. The confidence region for the vector $\boldsymbol{\beta}$ is the ellipsoid that is centered at $\hat{\boldsymbol{\beta}}$. Eigenvectors and eigenvalues of $X^T X$ determines its orientation and size, respectively. See Johnson and Wichern (2013) for details.

`ellipse::ellipse()` has method for `lm` object. So if you provides the regression object, it will give ellipsoid coordinate as `matrix`. However, this function only supports two variables. By specifying `which` argument, you can select which variable to get coordinates. By default, first two variables `c(1, 2)`.

```
ellipse::ellipse(cem_fit) %>%
  tbl_df() %>% # change to data frame
  ggplot(aes(x = `(Intercept)`, y = x1)) +
  geom_path()
```

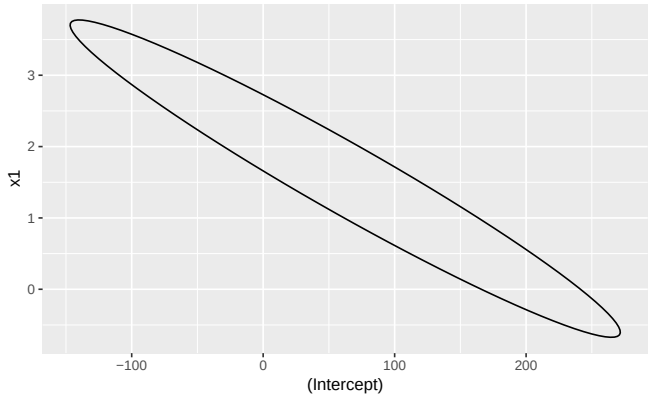


Figure 2.11: Confidence region for (β_0, β_1)

Look again corollary 2.2. Compare this to ANOVA. Something is different. When testing significance in ANOVA, degrees of freedom of SSR was p. Because when testing regression relation, we only need β_1 to β_p, i.e. p parameters. How can we do this here?

2.5.4 Regression coefficient vector $\boldsymbol{\beta}_A$

Consider $\boldsymbol{\beta}_A = (\beta_1, \ldots, \beta_p)^T$. In fact, these tell us significance of variables. We can use $\mathbb{X}_{A,\perp}$ defined before. From Equation (2.5),

$$\hat{\boldsymbol{\beta}}_A = (\mathbb{X}_{A,\perp}^T \mathbb{X}_{A,\perp})^{-1} \mathbb{X}_{A,\perp}^T \mathbf{Y}$$

The reason using $\mathbb{X}_{A,\perp}$ is to decomposing the space orthogonally.

$$\hat{\boldsymbol{\beta}} - \boldsymbol{\beta} = \begin{bmatrix} \hat{\beta}_0 - \beta_0 \\ \hline \hat{\boldsymbol{\beta}}_A - \boldsymbol{\beta}_A \end{bmatrix} \sim MVN\left(\begin{bmatrix} 0 \\ \hline \mathbf{0} \end{bmatrix}, \sigma^2 (X^T X)^{-1} \right)$$

with

$$Var(\hat{\boldsymbol{\beta}}_A - \boldsymbol{\beta}_A) = \sigma^2 (\mathbb{X}_{A,\perp}^T \mathbb{X}_{A,\perp})^{-1}$$

Theorem 2.17 (Confidence region)**.** $(1-\alpha)100\%$ *confidence interval for* $\boldsymbol{\beta}_A = (\beta_1, \ldots, \beta_p)^T$ *is*

$$\left\{ \boldsymbol{\beta}_A : (\hat{\boldsymbol{\beta}}_A - \boldsymbol{\beta}_A)^T \mathbb{X}_{A,\perp}^T \mathbb{X}_{A,\perp} (\hat{\boldsymbol{\beta}}_A - \boldsymbol{\beta}_A) \leq p\hat{\sigma}^2 F_\alpha(p, n-p-1) \right\}$$

where $\hat{\boldsymbol{\beta}}_A = (\mathbb{X}_{A,\perp}^T \mathbb{X}_{A,\perp})^{-1} \mathbb{X}_{A,\perp}^T \mathbf{Y}$

This tests the same hypothesis as ANOVA for significance.

Remark. As for X, the confidence region for the vector $\boldsymbol{\beta}_A$ is the ellipsoid that is centered at $\hat{\boldsymbol{\beta}}_A$. Eigenvectors and eigenvalues of $\mathbb{X}_{A,\perp}^T \mathbb{X}_{A,\perp}$ determines its orientation and size, respectively.

2.6 Nested Models

We have seen t-test and F-test testing $\beta_j = 0$ and $\beta_1 = \cdots = \beta_p = 0$, respectively. Here, we generalize the hypothesis in terms of *nested model*. Consider several types of hypothesis.

Example 2.1 (Examples of three types of hypothesis)**.** We can test if every coefficient is zero or some coefficient is zero, or just test if coefficients are same not specific value.

1. $H_0 : \beta_1 = \cdots = \beta_p = 0$
2. $H_0 : \beta_{p-1} = \beta_p = 0$
3. $H_0 : \beta_{p-1} = \beta_p$

What is the unified approach to these kind of test?

2.6.1 Full model and reduced model

To test some forms of H_0, consider two nested model.

Definition 2.4 (Nested models)**.** We name nested models for the test as follow.

1. ***Full model*** (FM) represents the basic starting model
2. ***Reduced model*** (RM) represents the null model under H_0

In Example 2.1, the basic starting model, i.e. FM is

$$Y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i, \quad \epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$$

See the second case.

$$H_0 : \beta_{p-1} = \beta_p = 0 \qquad \text{vs} \qquad H_1 : \beta_{p-1} \neq 0 \text{ or } \beta_p \neq 0$$

This is equivalent to choosing a model between

$$\begin{cases} FM : Y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i \\ RM : Y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_{p-2} x_{i,p-2} + \epsilon_i \end{cases}$$

Let us define SSR and SSE for each model.

Definition 2.5 (SS of nested models)**.** For each full model and reduced model, compute SSR and SSE.

1. $SSR(FM)$ regression sum of squares after fitting the FM
2. $SSE(FM)$ error sum of squares after fitting the FM
3. $SSR(RM)$ regression sum of squares after fitting the RM
4. $SSR(RM)$ error sum of squares after fitting the RM

By construction, $SST = \sum_{i=1}^{n} (Y_i - \overline{Y})^2$ does not depend on model. This measures variability when no predictor is used. It jus the variation of a response itself. So it becomes consistent with given data.

Remark.

$$SST = SSR(FM) + SSE(FM) = SSR(RM) + SSE(RM)$$

and hence

$$SSE(RM) - SSE(FM) = SSR(FM) - SSR(RM)$$

SSE indicates a variation that cannot be explained by the model. SSR represents a variation explained by the model. We can compare each between the two model. If $SSE(RM) - SSE(FM) = SSR(FM) - SSR(RM)$ is large, then we can think that the full model is explaining given data better than reduced model. Or, SSE says that full model can explain some parts that reduced model have failed to explained. In terms of $H_0 : RM$ vs $H_1 : FM$, it can be said that it is a strong evidene supporting FM against RM. In turn, we conclude that $H_1 : \beta_{p-1} \neq 0$ or $\beta_p \neq 0$ from our first hypothesis set.

Conjecture 2.1.

$$SSR(FM) - SSR(RM) > c \Leftrightarrow \textit{reject}\, H_0$$

We can test any hypothesis using RM-FM pair as in Conjecture 2.1. The question is

1. What is the distribution of $SSR(FM) - SSR(RM)$?
2. Then what is c?

Define two design matrices to build full model and reduced model.

$$Z_1 := \begin{bmatrix} 1 & x_{11} & \cdots & x_{1,p-2} \\ 1 & x_{21} & \cdots & x_{2,p-2} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & \cdots & x_{n,p-2} \end{bmatrix}, \quad \text{and} \quad Z_2 := \begin{bmatrix} x_{1,p-1} & x_{1p} \\ x_{2,p-1} & x_{2p} \\ \vdots & \vdots \\ x_{n,p-1} & x_{np} \end{bmatrix}$$

Z_1 is a design matrix of reduced model, while Z_2 consists of column vector that was not in Z_2, i.e. which were specified in H_0. It is intended to make

$$X^* = \begin{bmatrix} Z_1 \mid Z_2 \end{bmatrix} \in \mathbb{R}^{n\times(p+1)}$$

where X^* can have different column order with original X. It follows that

$$\begin{cases} FM : \mathbf{Y} = X\boldsymbol{\beta}_F + \boldsymbol{\epsilon} \\ RM : \mathbf{Y} = Z_1\boldsymbol{\beta}_R + \boldsymbol{\epsilon} \end{cases}$$

where $\boldsymbol{\beta}_F = (\beta_0, \ldots, \beta_p)^T$ and $\boldsymbol{\beta}_R = (\beta_0, \beta_1, \ldots, \beta_{p-2})^T$. Thus,

$$\begin{cases} SSR(FM) = \mathbf{Y}^T(\Pi_X - \Pi_{\mathbf{1}})\mathbf{Y} \\ SSR(RM) = \mathbf{Y}^T(\Pi_{Z_1} - \Pi_{\mathbf{1}})\mathbf{Y} \end{cases} \tag{2.30}$$

and so

$$\begin{aligned} SSR(FM) - SSR(RM) &= \mathbf{Y}^T(\Pi_X - \Pi_{\mathbf{1}} - \Pi_{Z_1} + \Pi_{\mathbf{1}})\mathbf{Y} \\ &= \mathbf{Y}^T(\Pi_X - \Pi_{Z_1})\mathbf{Y} \end{aligned} \tag{2.31}$$

Lemma 2.4. *$SSR(FM) - SSR(RM)$ satisfies following properties.*

1. *$\Pi_X - \Pi_{Z_1}$ is symmetric and idempotent*
2. *$tr(\Pi_X - \Pi_{Z_1}) = (p+1) -$ number of parameters in reduced model*
3. *$SSR(FM) - SSR(RM) \perp\!\!\!\perp SSE(FM)$*

Proof. Propery of prjection implies that

$$\Pi_X\Pi_{Z_1} = \Pi_{Z_1}$$

Note that

$$SSE(FM) = \mathbf{Y}^T(I - \Pi_X)\mathbf{Y}$$

Then

$$(\Pi_X - \Pi_{Z_1})(I - \Pi_X) = 0$$

Hence, $SSR(FM) - SSR(RM) \perp\!\!\!\perp SSE(FM)$ □

Go back to our Example 2.1. Denote that $\Pi_X - \Pi_{Z_1}$ implies that $R^2_{p-1} > R^2_{p-2}$. It is easy to get

$$tr(\Pi_X - \Pi_{Z_1}) = (p+1) - (p-1) = 2$$

Then

$$\frac{SSR(FM) - SSR(RM)}{\sigma^2} \overset{H_0}{\sim} \chi^2(2) \tag{2.32}$$

From Proposition 2.7

$$\frac{SSE(FM)}{\sigma^2} \overset{H_0}{\sim} \chi^2(n-p-1)$$

Since $SSR(FM) - SSR(RM) \perp\!\!\!\perp SSE(FM)$,

$$\begin{aligned} \frac{(SSR(FM) - SSR(RM))/2}{SSE(FM)/(n-p-1)} &= \frac{\frac{SSR(FM)-SSR(RM)}{\sigma^2}/2}{\frac{SSE(FM)}{\sigma^2}/(n-p-1)} \\ &\overset{H_0}{\sim} F(2, n-p-1) \end{aligned} \tag{2.33}$$

Hence, for our

$$H_0 : \beta_{p-1} = \beta_p = 0 \qquad \text{vs} \qquad H_1 : \beta_{p-1} \neq 0 \text{ or } \beta_p \neq 0$$

reject H_0 if $\frac{(SSR(FM)-SSR(RM))/2}{SSE(FM)/(n-p-1)} > F_\alpha(2, n-p-1)$.

See the numerator part $SSR(FM) - SSR(RM)$ we have been explored. This can be also written as

$$SSR(X_1, \ldots, X_p) - SSR(X_1, \ldots, X_{p-2})$$

and this is called *extra sum of squares*.

2.6.2 Extra sum of squares

Definition 2.6 (Extra sum of squares)**.** Extra sum of squares measures the amount of the additional contribution of variables added after the variables that were already considered. For example,

$$SSR(X_2 \mid X_1) := SSR(X_1, X_2) - SSR(X_1)$$

This measures marginal contribution of X_2 after X_1 is considred. Here, the contribution means marginal increase of SSR. Equivalently, marginal decrease of SSE.

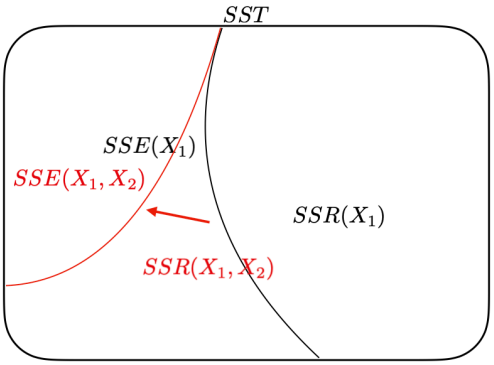


Figure 2.12: Extra sum of squares - SSR increases and SSE decreases

Similarly,

$$SSR(X_3, X_4 \mid X_1, X_2) := SSR(X_1, X_2, X_3, X_4) - SSR(X_1, X_2)$$

measures of X_3 and X_4 after X_1 and X_2 are already in the model.

SST does not change given data. By adding some variables, SSR increases or SSE decreases. *We interpret this change as additional contribution of those variables.* If it is large enough, they are significant. Using this idea, we might test any kind of sets of coefficients. Partial F-test and sequential F-test are typical forms that we can think of and useful.

2.6.3 Partial F test

Focus on marginal contribution given that every other predictor is already in the model.

Definition 2.7 (Partial sum of squares)**.** Partial SS or Type III Sum of squares are following.

- $SSR(X_1 \mid X_2, \ldots, X_p)$
- $SSR(X_2 \mid X_1, X_3, \ldots, X_p)$
- $SSR(X_3 \mid X_1, X_2, X_4, \ldots, X_p)$

- $\vdots$
- $SSR(X_p \mid X_1, \ldots, X_{p-1})$

Remark. Partial SS satisties following properties.

1. $SSR(X_1 \mid X_2, \ldots X_p) + \cdots + SSR(X_p \mid X_1, \ldots, X_{p-1}) \neq SSR(X_1, \ldots, X_p)$
2. Partial SS (Type III SS) represents the additional contribution of each predictor after the other predictor variables are considred.

Partial F-test aims at testing single coefficient.

$$H_0 : \beta_j = 0 \qquad \text{vs} \qquad H_1 : \beta_j \neq 0$$

How to make test statistic is same as previous session: using full model and reduced model. Here, one with and without β_j.

$$F_0 = \frac{SSR(X_j \mid X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_p)/1}{SSE/(n-p-1)} \overset{H_0}{\sim} F(1, n-p-1) \tag{2.34}$$

Hence, reject H_0 if $F_0 > F_\alpha(1, n-p-1)$. Since the first degrees of freedom is 1, we have

$$F_0 = \left(\frac{\hat{\beta}_j}{\widehat{SE}(\hat{\beta}_j)} \right)^2 = T_0^2$$

Remark. Partial F-test is equivalent to partial t-test. Both tests the additional contribution of X_j after the other variables are already considered, not individual significance.

2.6.4 Sequential F test

This sequential F-test adds variable sequentially.

Definition 2.8 (Sequential sum of squares)**.** Sequential SS or Type I Sum of squares are following.

1. $SSR(X_1)$
2. $SSR(X_2 \mid X_1)$
3. $SSR(X_3 \mid X_1, X_2)$
4. $\vdots$
5. $SSR(X_p \mid X_1, \ldots, X_{p-1})$

Remark. Sequential SS satisfies following properties.

1. $SSR(X_1) + SSR(X_2 \mid X_1) + \cdots + SSR(X_p \mid X_1, \ldots, X_{p-1}) = SSR(X_1, \ldots, X_p)$
2. Sequential SS is useful when we consider nested model. That is, when we know that X_1 is the most important, X_2 is the second important, and so on.

Following Definition 2.8, start from intercept-only model.

Algorithm 4: Sequential F Test from beginning

$j \leftarrow 1$;
repeat
  $\begin{cases} FM : Y_i = \beta_0 + \cdots + \beta_j x_{1j} + \epsilon_i \\ RM : Y_i = \beta_0 + \ldots + \beta_{j-1} x_{1,j-1} + \epsilon_i \end{cases}$;
  $H_0 : \beta_j = 0 (RM) \quad \text{vs} \quad H_1 : \beta_j \neq 0 (FM)$;
  $F_0 = \frac{SSR(X_j | X_1, \ldots, X_{j-1})/1}{SSE(X_1, \ldots, X_j)/(n-j-1)}$;
  reject H_0 if $F_0 > F_{1-\alpha}(1, n-j-3)$;
  $j \leftarrow j + 1$;
until *accept* H_0;
output: accepted model, i.e. final RM

Algorithm 4 shows the logic how to add a variable sequentially. It continues until rejecting H_0. In general form, we can test

$$H_0 : \beta_{q+1} = \beta_{q+2} = \cdots = \beta_p = 0 \qquad \text{vs} \qquad H_1 : \text{not } H_0$$

Here, reject H_0 if

$$\frac{\Big(SSR(X_1, \ldots, X_p) - SSR(X_1, \ldots, X_q)\Big)/(p-q)}{SSE(X_1, \ldots, X_p)/(n-p-1)} > F_\alpha(p-q, n-p-1) \quad (2.35)$$

This form of test might be useful when see the significance of categorical predictor. In regression model, quantative predictor is modeled as multiple dummy variables. In this case, we have to see these coefficients all at once and sequential F-test form might be helpful (Hastie et al., 2013).

2.6.5 General linear hypothesis

See Example 2.1 again. The third one, $H_0 : \beta_{p-1} = \beta_p$ can be also tested using F-test by changing model representation.

$$H_0 : \beta_{p-1} = \beta_p = \alpha \quad \forall \alpha$$

Then null model can be set as

$$Y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_{p-2} x_{i,p-2} + \alpha(x_{i,p-1} + x_{ip}) + \epsilon_i$$

However, there is a more general test procedure: linear combination. For instance, $\beta_{p-1} = \beta_p$ can be represented as

$$\beta_{p-1} - \beta_p = 0 \Leftrightarrow (0, \ldots, 0, 1, -1)^T \boldsymbol{\beta} = 0$$

Now we analyze this form of hypothesis.

$$H_0 : C\boldsymbol{\beta} = \mathbf{d} \qquad \text{vs} \qquad H_1 : C\boldsymbol{\beta} \neq \mathbf{d}$$

Since $\hat{\boldsymbol{\beta}} \sim MVN(\boldsymbol{\beta}, \sigma^2 (X^T X)^{-1})$,

$$C\hat{\boldsymbol{\beta}} \sim MVN(C\boldsymbol{\beta}, \sigma^2 C(X^T X)^{-1} C^T)$$

and under $H_0 : C\boldsymbol{\beta} = \mathbf{d}$,

$$C\hat{\boldsymbol{\beta}} \overset{H_0}{\sim} MVN(\mathbf{d}, \sigma^2 C(X^T X)^{-1} C^T)$$

Set

$$\mathbf{Z} \equiv \frac{1}{\sigma}(C(X^T X)^{-1} C^T)^{-\frac{1}{2}}(C\hat{\boldsymbol{\beta}} - \mathbf{d}) \overset{H_0}{\sim} MVN(\mathbf{0}, I)$$

It follows that

$$\mathbf{Z}^T \mathbf{Z} = \frac{1}{\sigma^2}(C\hat{\boldsymbol{\beta}} - \mathbf{d})^T (C(X^T X)^{-1} C^T)^{-1} (C\hat{\boldsymbol{\beta}} - \mathbf{d}) \overset{H_0}{\sim} \chi^2(q = rank(C))$$

Since $SSE \perp\!\!\!\perp \hat{\boldsymbol{\beta}}$ (See Proposition 2.7),

$$\begin{aligned} \frac{\frac{(C\hat{\boldsymbol{\beta}} - \mathbf{d})^T (C(X^T X)^{-1} C^T)^{-1} (C\hat{\boldsymbol{\beta}} - \mathbf{d})}{\sigma^2} / q}{\frac{SSE}{\sigma^2} / (n - p - 1)} &= \frac{(C\hat{\boldsymbol{\beta}} - \mathbf{d})^T (C(X^T X)^{-1} C^T)^{-1} (C\hat{\boldsymbol{\beta}} - \mathbf{d}) / q}{\hat{\sigma}^2} \\ &\overset{H_0}{\sim} F(q, n - p - 1) \end{aligned} \tag{2.36}$$

2.6.6 Extra SS in R

`anova.lm()` gives extra sum of squres by default, which is *sequential sum of squares*, i.e. type I SS.

```
anova(cem_fit)
#> Analysis of Variance Table
#>
#> Response: y
#>           Df Sum Sq Mean Sq F value  Pr(>F)
#> x1         1   1450    1450  242.37 2.9e-07 ***
#> x2         1   1208    1208  201.87 5.9e-07 ***
#> x3         1     10      10    1.64    0.24
#> x4         1      0       0    0.04    0.84
#> Residuals  8     48       6
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

If we change the order,

```
anova(lm(y ~ x2 + x1 + x3 + x4, data = cem))
#> Analysis of Variance Table
#>
#> Response: y
#>           Df Sum Sq Mean Sq F value  Pr(>F)
#> x2         1   1809    1809  302.43 1.2e-07 ***
#> x1         1    848     848  141.81 2.3e-06 ***
#> x3         1     10      10    1.64    0.24
#> x4         1      0       0    0.04    0.84
#> Residuals  8     48       6
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

SS differs. To get Type III SS, we can use `car::Anova()`. This function can compute type II or type III sum of squares. Since it gives type II by default, we should specify `type = "III"` or `type = 3`.

```
car::Anova(cem_fit, type = 3)
#> Anova Table (Type III tests)
#>
#> Response: y
#>             Sum Sq Df F value Pr(>F)
#> (Intercept)    4.7  1    0.79  0.399
#> x1            26.0  1    4.34  0.071 .
#> x2             3.0  1    0.50  0.501
#> x3             0.1  1    0.02  0.896
#> x4             0.2  1    0.04  0.844
#> Residuals     47.9  8
```

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

As base `anova.lm()`, we can also tidy this object.

```
car::Anova(cem_fit, type = 3) %>%
  broom::tidy()
#> # A tibble: 6 x 5
#>   term         sumsq    df statistic p.value
#>   <chr>        <dbl> <dbl>     <dbl>   <dbl>
#> 1 (Intercept)  4.75      1    0.793   0.399
#> 2 x1          26.0       1    4.34    0.0708
#> 3 x2           2.97      1    0.497   0.501
#> 4 x3           0.109     1    0.0182  0.896
#> 5 x4           0.247     1    0.0413  0.844
#> 6 Residuals   47.9       8   NA      NA
```

Look p-value of the type III SS, i.e. partial F-test. Comparing to partial t-test, we can see both are equivalent.

```
car::Anova(cem_fit, type = 3) %>%
  broom::tidy() %>%
  na.omit() %>%
  select(p.value) %>%
  bind_cols(t_test = broom::tidy(cem_fit)$p.value)
#> # A tibble: 5 x 2
#>   p.value t_test
#>     <dbl>  <dbl>
#> 1  0.399  0.399
#> 2  0.0708 0.0708
#> 3  0.501  0.501
#> 4  0.896  0.896
#> 5  0.844  0.844
```

As we can see in the argument of `car::Anova()`, there are also type II and type IV SS. These are not popular ones, though.

`car::linearHypothesis()` function peforms test for linear combination. Just specify C matrix to `hypothesis.matrix`. If you want additional $\mathbf{d}$, specify `rhs`. If this is `NULL` (by default), test will be done with zero. Try $H_0 : \beta_3 = \beta_4$.

```
car::linearHypothesis(cem_fit, hypothesis.matrix = c(0, 0, 0, 1, -1))
#> Linear hypothesis test
#>
#> Hypothesis:
#> x3 - x4 = 0
#>
#> Model 1: restricted model
```

```
#> Model 2: y ~ x1 + x2 + x3 + x4
#>
#>   Res.Df  RSS Df Sum of Sq    F Pr(>F)
#> 1      9 57.2
#> 2      8 47.9  1      9.38 1.57   0.25
```

2.7 Qualitative Variables as Predictors

See the example data set from Chatterjee and Hadi (2015).

```
salary <- haven::read_sav("data/p124.sav")
(salary <-
  salary %>%
  mutate_at(.vars = vars("E", "M"), .funs = factor))
#> # A tibble: 46 x 4
#>       S     X E     M
#>   <dbl> <dbl> <fct> <fct>
#> 1 13876     1 1     1
#> 2 11608     1 3     0
#> 3 18701     1 3     1
#> 4 11283     1 2     0
#> 5 11767     1 3     0
#> 6 20872     2 2     1
#> # ... with 40 more rows
```

- `S`: salary (*response variable*)
- `X`: year of experiece
- `E`: education level
  - `1`: high school (HS)
  - `2`: bachelor degree (BS)
  - `3`: advanced degree (ADV)
- `M`: management status
  - `1`: person with management responsibility (MGT)
  - `0`: otherwise (None)

Unlike with previous setting, we have two qualitative variables `E` and `M`.

```
salary %>%
  ggplot(aes(x = X, y = S)) +
  geom_point(aes(colour = E, shape = M)) +
  labs(
    x = "Experience (year)",
    y = "Salary"
  ) +
  scale_colour_discrete(
    name = "Education",
```

```
    label = c("High School", "Bachelor", "Advanced")
  ) +
  scale_shape_discrete(
    name = "Management",
    label = c("None", "Management")
  )
```

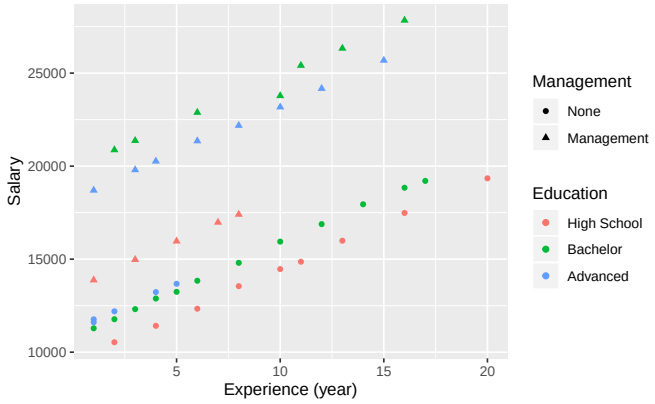


Figure 2.13: Salary Survey Data

See Figure 2.13. There are 6 combinations of `E-M`. For each level, we can see obvious linear relationship between `S` (salary) and `X` (experience). Also, person in management statement has larer salary than not. How can we deal with these variables?

2.7.1 Separate models

The easiest way that we can think of is fit separate regression models for

1. different levels of the qualitative predictor (in case there is only one qualitative predictor)
2. *different combinations of the levels.*

In the above salary survey dataset, we can make 6 models.

levels	HS	BS	ADV
MGT	$Y_i =$ $\beta_{01} + \beta_{11}x_i + \epsilon_{i1}$	$Y_i =$ $\beta_{02} + \beta_{12}x_i + \epsilon_{i2}$	$Y_i =$ $\beta_{03} + \beta_{13}x_i + \epsilon_{i3}$
None	$Y_i =$ $\beta_{04} + \beta_{14}x_i + \epsilon_{i4}$	$Y_i =$ $\beta_{05} + \beta_{15}x_i + \epsilon_{i5}$	$Y_i =$ $\beta_{06} + \beta_{16}x_i + \epsilon_{i6}$

2.7.2 Dummy variables

However, we want to explain linear relationship between response variable and qualitative predictors. Commonly, this is done by defining **dummy variables** or **indicator variables**. For instance, Define E_{i1} and E_{i2} by

$$E_{i1} = \begin{cases} 1 & E = HS \\ 0 & \text{o/w} \end{cases}$$
$$E_{i2} = \begin{cases} 1 & E = BS \\ 0 & \text{o/w} \end{cases}$$

In other words, the last level ADV(3) has value of $E_{i1} = E_{i2} = 0$. This is called *dummy coding* of the last level as baseline. The following function gives how the coding is happened. In R, baseline is set to be the first level, i.e. `base = 1` by default.

```
C(salary$E, contr = contr.treatment, base = 3)
#>  [1] 1 3 3 2 3 2 2 1 3 2 1 2 3 1 3 3 2 2 3 1 1 3 2 2 1 2 1 3 1 1 2 3 2 2 1
#> [36] 2 3 1 2 2 3 2 2 1 2 1
#> attr(,"contrasts")
#>   1 2
#> 1 1 0
#> 2 0 1
#> 3 0 0
#> Levels: 1 2 3
```

For management status, define MGT_i by

$$MGT_i = \begin{cases} 1 & M = MGT \\ 0 & M = None \end{cases}$$

This is the case when `base = 1`, i.e. `0`(None) in our data set.

```
C(salary$M, contr = contr.treatment)
#>  [1] 1 0 1 0 0 1 0 0 0 0 1 1 1 0 1 0 0 0 0 1 0 1 0 1 1 0 1 1 0 0 0 1 1 1 0
#> [36] 0 1 0 1 0 1 1 0 0 0 0
#> attr(,"contrasts")
#>   2
#> 0 0
#> 1 1
#> Levels: 0 1
```

The we now have regression model as

$$Y_i = \beta_0 + \beta_1 x_i + \gamma_1 E_{i1} + \gamma_2 E_{i2} + \delta MGT_i + \epsilon_i \tag{2.37}$$

```
salary_relev <-
  salary %>%
  mutate(
    E = C(E, contr = contr.treatment, base = 3) # different with default
  )
#--------------
salary_relev %>%
  lm(S ~ ., data = .)
#>
#> Call:
#> lm(formula = S ~ ., data = .)
#>
#> Coefficients:
#> (Intercept)            X           E1           E2           M1
#>       11032          546        -2996          148         6884
```

Plug in each value. Then we can make 6 models as previous section.

$$\begin{cases} Y_i = (\beta_0 + \gamma_1) + \beta_1 x_i + \epsilon_i & \text{HS-None} & E_1 = 1, E_2 = 0, MGT = 0 \\ Y_i = (\beta_0 + \gamma_1 + \delta) + \beta_1 x_i + \epsilon_i & \text{HS-MGT} & E_1 = 1, E_2 = 0, MGT = 1 \\ Y_i = (\beta_0 + \gamma_2) + \beta_1 x_i + \epsilon_i & \text{BS-None} & E_1 = 0, E_2 = 1, MGT = 0 \\ Y_i = (\beta_0 + \gamma_2 + \delta) + \beta_1 x_i + \epsilon_i & \text{BS-MGT} & E_1 = 0, E_2 = 1, MGT = 1 \\ Y_i = \beta_0 + \beta_1 x_i + \epsilon_i & \text{ADV-None} & E_1 = 0, E_2 = 0, MGT = 0 \\ Y_i = (\beta_0 + \delta) + \beta_1 x_i + \epsilon_i & \text{ADV-MGT} & E_1 = 0, E_2 = 0, MGT = 1 \end{cases} \tag{2.38}$$

Observe that every line has same slope β_1.

```
salary_relev %>%
  ggplot(aes(x = X, y = S, colour = E, linetype = M)) +
  geom_smooth(method = "lm", se = FALSE, fullrange = TRUE) +
  geom_point(aes(shape = M), alpha = .7, show.legend = FALSE) +
  labs(
    x = "Experience (year)",
    y = "Salary"
  ) +
  scale_colour_discrete(
    name = "Education",
    label = c("High School", "Bachelor", "Advanced")
  ) +
  scale_linetype_discrete(
    name = "Management",
    label = c("None", "Management")
  )
```

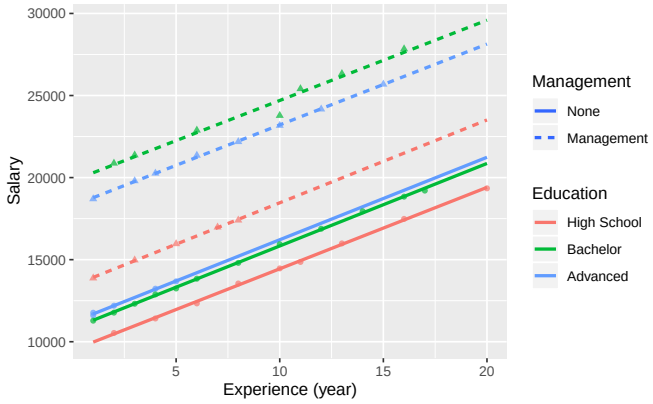


Figure 2.14: Salary Survey Data with Dummy variables

Every line is parallel to each other. In the set of Equations (2.38), we can interpret β_1 as the increment of salary when experience increases in 1 year *when the other predictors are fixed.*

γ_1, the coefficient of E_{i1}, can be interpreted as the increment of salary for `HS` compared to `ADV`(baseline) when the other predictors are fixed. Similarly, we interpret γ_2, the coefficient of E_{i2} as the increment of salary for `BS` compared to the baseline level `ADV` when the other predictors are fixed. The coefficient of MGT δ means the increment of salary for management status for none when the other predictors fixed.

In sum, each coefficient for dummy variables is the increment for corresponding level versus baseline level.

2.7.3 Interaction variables

See Figure 2.13 and focus on education level. Among people in management responsibility, ones with bachelor degrees have the largest salaries and next advanced dgrees. On the other hand, advanced degrees seem to be more important than bachelor among people that are not in management status. The *magnitude of the salary difference between education level also depends management status.* To explain this, we add *interaction term* in the previous model (2.37).

$$Y_i = \beta_0 + \beta_1 x_i + \gamma_1 E_{i1} + \gamma_2 E_{i2} + \delta MGT_i + \alpha_1 (E_{i1} MGT_i) + \alpha_2 (E_{i2} MGT_i) + \epsilon_i \tag{2.39}$$

```
(dum_int1 <-
  salary_relev %>%
  lm(S ~ X + E * M, data = .))
#>
```

```
#> Call:
#> lm(formula = S ~ X + E * M, data = .)
#>
#> Coefficients:
#> (Intercept)            X           E1           E2           M1
#>       11203          497        -1731         -349         7047
#>       E1:M1        E2:M1
#>       -3066         1836
```

$$\begin{cases} Y_i = (\beta_0 + \gamma_1) + \beta_1 x_i + \epsilon_i & \text{HS-None} & E_1 = 1, E_2 = 0, MGT = 0 \\ Y_i = (\beta_0 + \gamma_1 + \delta + \alpha_1) + \beta_1 x_i + \epsilon_i & \text{HS-MGT} & E_1 = 1, E_2 = 0, MGT = 1 \\ Y_i = (\beta_0 + \gamma_2) + \beta_1 x_i + \epsilon_i & \text{BS-None} & E_1 = 0, E_2 = 1, MGT = 0 \\ Y_i = (\beta_0 + \gamma_2 + \delta + \alpha_2) + \beta_1 x_i + \epsilon_i & \text{BS-MGT} & E_1 = 0, E_2 = 1, MGT = 1 \\ Y_i = \beta_0 + \beta_1 x_i + \epsilon_i & \text{ADV-None} & E_1 = 0, E_2 = 0, MGT = 0 \\ Y_i = (\beta_0 + \delta) + \beta_1 x_i + \epsilon_i & \text{ADV-MGT} & E_1 = 0, E_2 = 0, MGT = 1 \end{cases} \tag{2.40}$$

To visualize the model, i.e. get prediction for each grid, we might use `modelr` library.

```
library(modelr)
```

`modelr::data_grid()` expands predictor grids for prediction. `modelr::add_predictions()` add predictions for a given model to a data frame. We can provide multiple models with `modelr::spread_predictions()` and `modelr::gather_predictions()` each as `tidyr::spread()` and `tidyr::gather()`.

```
salary_relev %>%
  data_grid(X, E, M) %>%
  add_predictions(dum_int1) %>%
  ggplot() +
  geom_line(aes(x = X, y = pred, colour = E, linetype = M)) +
  geom_point(
    data = salary_relev,
    aes(x = X, y = S, colour = E, shape = M),
    alpha = .7,
    show.legend = FALSE
  ) +
  labs(
    x = "Experience (year)",
    y = "Salary"
  ) +
  scale_colour_discrete(
    name = "Education",
    label = c("High School", "Bachelor", "Advanced")
```

```
) +
scale_linetype_discrete(
  name = "Management",
  label = c("None", "Management")
)
```

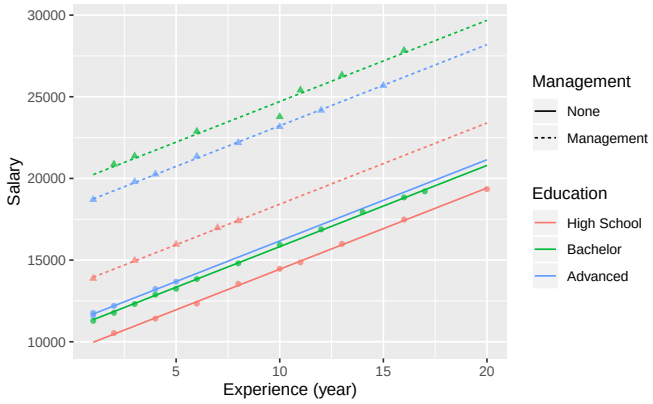


Figure 2.15: Interaction between qualitative variables

Here the increment for experience is also same for each level.

2.7.4 General models with interaction

In the previous section 2.7.2 and 2.7.3, there have been some assumptions in the model. In Model (2.37), *additive assumption* is given. Effects of each predictors do note depend on the others. It means that in the separate model setting 2.7.1 we give conditions

$$\begin{cases} \beta_{11} = \beta_{12} = \cdots = \beta_{16} \equiv \beta_1 \\ \beta_{01} - \beta_{04} = \beta_{02} - \beta_{05} = \beta_{03} - \beta_{06} \equiv \delta \end{cases}$$

Next model (2.39) quite weakens this assumption, i.e. remove the second condition for δ by adding an interaction term between qualitative predictors. Instead, it still assumes that *every combination of levels always has same slope* β_1. Now we want more general one without the first assumption. Add interaction between x_i and other dummy variables.

$$
\begin{aligned}
Y_i = \beta_0 &+ \beta_1 x_i + \gamma_1 E_{i1} + \gamma_2 E_{i2} + \delta MGT_i \\
&+ \alpha_1 (E_{i1} MGT_i) + \alpha_2 (E_{i2} MGT_i) \\
&+ \eta_1 (x_i E_{i1}) + \eta_2 (x_i E_{i2}) + \eta_3 (x_i MGT_i) \\
&+ \eta_4 (x_i MGT_i E_{i1}) + \eta_5 (x_i MGT_i E_{i2}) \\
&+ \epsilon_i
\end{aligned} \tag{2.41}
$$

```
(dum_int2 <-
  salary_relev %>%
  lm(S ~ X * E * M, data = .))
#>
#> Call:
#> lm(formula = S ~ X * E * M, data = .)
#>
#> Coefficients:
#> (Intercept)            X           E1           E2           M1
#>   11189.455      502.364    -1708.064     -381.019     7094.382
#>        X:E1         X:E2         X:M1        E1:M1        E2:M1
#>      -6.247        0.246       -9.850    -3158.650     1904.041
#>     X:E1:M1      X:E2:M1
#>      18.416       -3.645
```

In the same process, we have separate model. Since we have added terms with x_i, the only change from Equation (2.40) is slope part.

$$
\begin{array}{llll}
Y_i = (\beta_0 + \gamma_1) + (\beta_1 + \eta_1) x_i + \epsilon_i & \text{HS-None} & E_1 = 1, E_2 = 0, MGT = 0 \\
Y_i = (\beta_0 + \gamma_1 + \delta + \alpha_1) + (\beta_1 + \eta_1 + \eta_3 + \eta_4) x_i + \epsilon_i & \text{HS-MGT} & E_1 = 1, E_2 = 0, MGT = 1 \\
Y_i = (\beta_0 + \gamma_2) + (\beta_1 + \eta_2) x_i + \epsilon_i & \text{BS-None} & E_1 = 0, E_2 = 1, MGT = 0 \\
Y_i = (\beta_0 + \gamma_2 + \delta + \alpha_2) + (\beta_1 + \eta_2 + \eta_3 + \eta_5) x_i + \epsilon_i & \text{BS-MGT} & E_1 = 0, E_2 = 1, MGT = 1 \\
Y_i = \beta_0 + \beta_1 x_i + \epsilon_i & \text{ADV-None} & E_1 = 0, E_2 = 0, MGT = 0 \\
Y_i = (\beta_0 + \delta) + (\beta_1 + \eta_3) x_i + \epsilon_i & \text{ADV-MGT} & E_1 = 0, E_2 = 0, MGT = 1
\end{array} \tag{2.42}
$$

```
salary_relev %>%
  data_grid(X, E, M) %>%
  add_predictions(dum_int2) %>%
  ggplot() +
  geom_line(aes(x = X, y = pred, colour = E, linetype = M)) +
  geom_point(
    data = salary_relev,
    aes(x = X, y = S, colour = E, shape = M),
    alpha = .7,
    show.legend = FALSE
```

```
  ) +
  labs(
    x = "Experience (year)",
    y = "Salary"
  ) +
  scale_colour_discrete(
    name = "Education",
    label = c("High School", "Bachelor", "Advanced")
  ) +
  scale_linetype_discrete(
    name = "Management",
    label = c("None", "Management")
  )
```

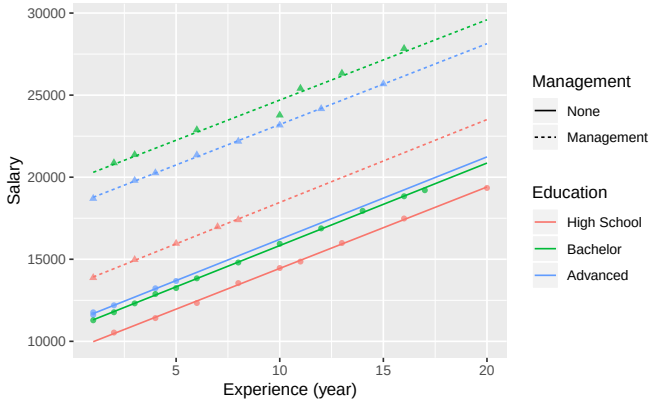


Figure 2.16: Full interaction model

```
summary(dum_int2)
#>
#> Call:
#> lm(formula = S ~ X * E * M, data = .)
#>
#> Residuals:
#>    Min     1Q Median     3Q    Max
#> -918.0  -41.2   14.2   64.8  222.9
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept) 11189.455    155.304   72.05  < 2e-16 ***
#> X             502.364     50.655    9.92  1.4e-11 ***
#> E1          -1708.064    201.534   -8.48  6.7e-10 ***
#> E2           -381.019    183.282   -2.08    0.045 *
```

```
#> M1           7094.382    199.690   35.53  < 2e-16 ***
#> X:E1           -6.247     51.896   -0.12    0.905
#> X:E2            0.246     51.630    0.00    0.996
#> X:M1           -9.850     52.704   -0.19    0.853
#> E1:M1       -3158.650    294.852  -10.71  1.9e-12 ***
#> E2:M1        1904.041    264.434    7.20  2.5e-08 ***
#> X:E1:M1        18.416     62.751    0.29    0.771
#> X:E2:M1        -3.645     55.540   -0.07    0.948
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 184 on 34 degrees of freedom
#> Multiple R-squared:  0.999,  Adjusted R-squared:  0.998
#> F-statistic: 2.68e+03 on 11 and 34 DF,  p-value: <2e-16
```

Denote that the changes of slopes are not that significant.

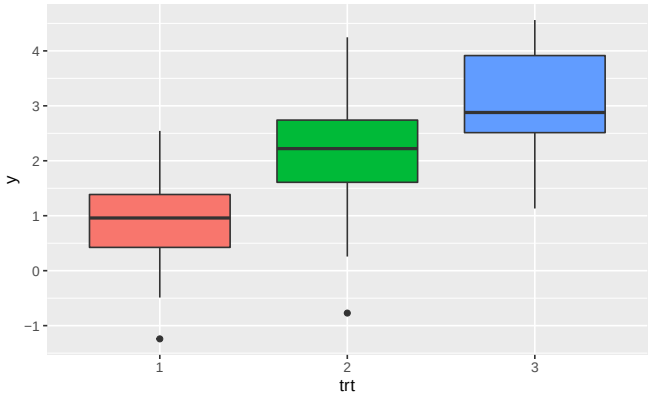
2.7.5 Regression approach to ANOVA

We have generated three grouped data of which group is three.

```
medicine
#> # A tibble: 90 x 2
#>   trt       y
#>   <fct> <dbl>
#> 1 1     1.87
#> 2 1     0.400
#> 3 1     0.283
#> 4 1     1.70
#> 5 1     2.23
#> 6 1     1.27
#> # ... with 84 more rows
```

```
medicine %>%
  group_by(trt) %>%
  summarise(m = mean(y), s = sd(y), N = n())
#> # A tibble: 3 x 4
#>   trt       m     s     N
#>   <fct> <dbl> <dbl> <int>
#> 1 1     0.881 0.818    30
#> 2 2     2.17  1.15     30
#> 3 3     3.02  0.940    30
```

```
medicine %>%
  ggplot(aes(x = trt, y = y, fill = trt)) +
  geom_boxplot(show.legend = FALSE)
```



Each group follows $N(\mu_1 = 1, \sigma^2 = 1)$, $N(\mu_2 = 2, \sigma^2 = 1)$, and $N(\mu_3 = 3, \sigma^2 = 1)$. Our goal is comparison between treatment, i.e. mean comparison.

$$H_0 : \mu_1 = \mu_2 = \mu_3 \quad \text{vs} \quad H_1 : \text{not } H_0$$

```
aov(y ~ trt, data = medicine) %>%
  summary()
#>              Df Sum Sq Mean Sq F value  Pr(>F)
#> trt           2   69.5    34.7    36.3 3.5e-12 ***
#> Residuals    87   83.3     1.0
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Recall that we can build two models, *means model* or *effects model*.

2.7.6 Effects model

In experimental design literature, effects model is more frequently used. Decomposing each group mean with overall mean and *treatment effect*, it gives more intuitive decription (Montgomery, 2012). So we look at this one first.

$$Y_{ij} = \mu + \tau_j + \epsilon_{ij} \tag{2.43}$$

where

$$\begin{cases} 1 \le i \le n_j \\ 1 \le j \le k & j\text{: treatment} \\ Y_{ij} = i\text{-th observation from the } j\text{-th treatment} \\ \mu = \text{overall mean} \\ \tau_j = j\text{-th treatment effect} := \mu_j - \mu \\ \epsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2) \end{cases}$$

Here we test whether there is *no treatment effect*, i.e.

$$H_0 : \tau_1 = \cdots = \tau_k = 0 \quad \text{vs} \quad H_1 : \text{not } H_0$$

which is equivalent to the above $\mu_1 = \cdots = \mu_k$ by construction.

Note that we have $k + 1$ parameters in the model (2.43). Write the parameter vector as

$$\boldsymbol{\theta} \equiv (\mu, \tau_1, \ldots, \tau_k)^T$$

Then we can express the model as

$$\underbrace{\begin{bmatrix} Y_{11} \\ Y_{21} \\ \vdots \\ Y_{n_1,1} \\ \hline Y_{12} \\ \vdots \\ Y_{n_2,2} \\ \hline \vdots \\ \hline Y_{1k} \\ \vdots \\ Y_{n_k,k} \end{bmatrix}}_{\mathbf{Y}} = \underbrace{\begin{bmatrix} 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 0 & \cdots & 0 \\ \hline 1 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 1 & \cdots & 0 \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots \\ \hline 1 & 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & \cdots & 1 \end{bmatrix}}_{X} \underbrace{\begin{bmatrix} \mu \\ \tau_1 \\ \vdots \\ \tau_k \end{bmatrix}}_{\boldsymbol{\theta}} + \underbrace{\begin{bmatrix} \epsilon_{11} \\ \epsilon_{21} \\ \vdots \\ \epsilon_{n_1,1} \\ \hline \epsilon_{12} \\ \vdots \\ \epsilon_{n_2,2} \\ \hline \vdots \\ \hline \epsilon_{1k} \\ \vdots \\ \epsilon_{n_k,k} \end{bmatrix}}_{\boldsymbol{\epsilon}}$$

Look at the model matrix $X = \left[\mathbf{1} \mid \mathbf{x}_1 \mid \cdots \mid \mathbf{x}_k\right] \in \mathbb{R}^{N \times (k+1)}$, where $N = \sum n_j$. From the second column to the last, $\mathbf{x}_j = (x_{1j}, \ldots, x_{n_j,j})$ has value of

$$x_{ij} = \begin{cases} 1 & j\text{-th treatment} \\ 0 & \text{otherwise} \end{cases}$$

However,

$$\mathbf{1} = \mathbf{x}_1 + \cdots + \mathbf{x}_k$$

i.e. column vectors are linearly dependent. This design matrix X is of rank deficient. Hence, the normal equation $(X^T X)^{-1}\hat{\boldsymbol{\beta}} = X^T \mathbf{Y}$ does *not have a unique solution*. This kind of coding is not appropriate. Rawlings et al. (2006) provides four ways *reparameterizing* to remove singularities.

2.7.7 Means model

Next, consider means model.

$$Y_{ij} = \mu_j + \epsilon_{ij} \tag{2.44}$$

where

$$\begin{cases} 1 \le i \le n_j \\ 1 \le j \le k & j\text{: treatment} \\ Y_{ij} = i\text{-th observation from the } j\text{-th treatment} \\ \mu_j = \text{mean of } j\text{-th treatment} \\ \epsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2) \end{cases}$$

With this model (2.44), we test if every μ_j is identical.

$$H_0 : \mu_1 = \cdots = \mu_k \quad \text{vs} \quad H_1 : \text{not } H_0$$

Write a parameter vector by

$$\boldsymbol{\theta}^* \equiv (\mu_1, \ldots, \mu_k)^T \in \mathbb{R}^k$$

One proceeds in a similar way for this $\boldsymbol{\theta}^*$,

$$\underbrace{\begin{bmatrix} Y_{11} \\ Y_{21} \\ \vdots \\ Y_{n_1,1} \\ \hline Y_{12} \\ \vdots \\ Y_{n_2,2} \\ \hline \vdots \\ \hline Y_{1k} \\ \vdots \\ Y_{n_k,k} \end{bmatrix}}_{\mathbf{Y}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & \cdots & 0 \\ \hline 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & \cdots & 0 \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots \\ \hline 0 & 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}}_{X^*} \underbrace{\begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_k \end{bmatrix}}_{\boldsymbol{\theta}^*} + \underbrace{\begin{bmatrix} \epsilon_{11} \\ \epsilon_{21} \\ \vdots \\ \epsilon_{n_1,1} \\ \hline \epsilon_{12} \\ \vdots \\ \epsilon_{n_2,2} \\ \hline \vdots \\ \hline \epsilon_{1k} \\ \vdots \\ \epsilon_{n_k,k} \end{bmatrix}}_{\boldsymbol{\epsilon}}$$

This results in design matrix

$$X^* = \left[\mathbf{x}_1 \mid \mathbf{x}_2 \mid \cdots \mid \mathbf{x}_k \right] \in \mathbb{R}^{N \times k}$$

where $N = \sum n_j$. The regression model *without an intercept* can be built using these indicator variables as

$$Y_{ij} = \beta_1 x_{i1} + \cdots + \beta_k x_{ik} + \epsilon_{ij}$$

The two models have corresponding parameters such that

$$\beta_j = \mu_j$$

So we now estimate $\boldsymbol{\theta}$. Note that $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ is orthogonal. Normalize each column matrix by

$$\mathbf{u}_j = \frac{1}{\sqrt{n_j}} \mathbf{x}_j$$

Let $Q = \begin{bmatrix} \mathbf{u}_1 \mid \cdots \mid \mathbf{u}_k \end{bmatrix}$ be orthogonal matrix and let

$$R = diag(\sqrt{n_1}, \sqrt{n_2}, \ldots, \sqrt{n_k}) \in \mathbb{R}^{k \times k}$$

Then

$$X^* = QR$$

Since

$$X^{*T} X^* = R^T Q^T Q R = R^T T$$

Normal equation gives that

$$\hat{\boldsymbol{\theta}} = R^{-1} Q^T \mathbf{Y}^T = (\overline{Y}_{1.}, \overline{Y}_{2.}, \ldots, \overline{Y}_{k.})^T$$

where $\overline{Y}_{j.} = \sum_{i=1}^{n_j} Y_{ij}$, j-the treatment mean. Hence, each estimate of regression coefficient is a treatment mean, i.e.

$$\hat{\boldsymbol{\beta}} = (\overline{Y}_{1.}, \overline{Y}_{2.}, \ldots, \overline{Y}_{k.})^T$$

2.7.8 Effect coding

Go back to the effects model (2.43). In fact, this model should be constrained by *identifiability condition* such as

$$\forall j : \sum_{i=1}^{n_j} \tau_j = 0 \tag{2.45}$$

Then we have

$$\tau_k = -(\tau_1 + \tau_2 + \cdots + \tau_{k-1})$$

Since τ_k is redundant, parameter vector is given by

$$\boldsymbol{\theta}^* \equiv (\mu, \tau_1, \ldots, \tau_{k-1})^T \in \mathbb{R}^k$$

Assume balanced setting, i.e. $n_1 = n_2 = \cdots = n_k = n$.

$$\underbrace{\begin{bmatrix} Y_{11} \\ Y_{21} \\ \vdots \\ Y_{n1} \\ \hline Y_{12} \\ \vdots \\ Y_{n2} \\ \hline \vdots \\ \hline Y_{1k} \\ \vdots \\ Y_{nk} \end{bmatrix}}_{\mathbf{Y}} = \underbrace{\left[\begin{array}{ccccc} 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 0 & \cdots & 0 \\ \hline 1 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 1 & \cdots & 0 \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots \\ \hline 1 & 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & \cdots & 1 \\ \hline 1 & -1 & -1 & \cdots & -1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & -1 & -1 & \cdots & -1 \end{array}\right]}_{X^*} \underbrace{\begin{bmatrix} \mu \\ \tau_1 \\ \vdots \\ \tau_{k-1} \end{bmatrix}}_{\boldsymbol{\theta}^*} + \underbrace{\begin{bmatrix} \epsilon_{11} \\ \epsilon_{21} \\ \vdots \\ \epsilon_{n1} \\ \hline \epsilon_{12} \\ \vdots \\ \epsilon_{n2} \\ \hline \vdots \\ \hline \epsilon_{1k} \\ \vdots \\ \epsilon_{nk} \end{bmatrix}}_{\boldsymbol{\epsilon}}$$

with $X^* \in \mathbb{R}^{N \times k}$. This is called *effect coding*. From the second column of design matrix, the variable is coded by

$$x_{ij} = \begin{cases} 1 & j\text{-th treatment} \\ -1 & k\text{-th treatment} \\ 0 & \text{otherwise} \end{cases}$$

This results in the regression model of the form

$$Y_{ij} = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_{k-1} x_{i,k-1} + \epsilon_{ij}$$

Here, each parameter corresponds to

$$\beta_0 = \mu, \quad \beta_j = \tau_j$$

Let $X^* = \left[\mathbf{1} \mid \mathbb{X}_A\right]$.

Note that

$$\mathbf{1} \perp \mathbb{X}_A$$

Then

$$\begin{aligned} X^* \hat{\boldsymbol{\theta}}^* &= \hat{\mu}\mathbf{1} + \mathbb{X}_A \boldsymbol{\tau} \quad \text{where } \boldsymbol{\tau} = (\tau_1, \ldots, \tau_{k-1})^T \\ &= \mathbf{1}\Pi(\mathbf{Y} \mid R(\mathbf{1})) + \mathbb{X}_A \Pi(\mathbf{Y} \mid R(\mathbb{X}_A)) \\ &= \overline{Y}_{\cdot\cdot}\mathbf{1} + \mathbb{X}_A (\mathbb{X}_A^T \mathbb{X}_A)^{-1} \mathbb{X}_A^T \mathbf{Y} \end{aligned}$$

Hence,

$$\hat{\mu} = \overline{Y}_{\cdot\cdot}$$

and

$$\boldsymbol{\tau} = (\mathbb{X}_A^T \mathbb{X}_A)^{-1} \mathbb{X}_A^T \mathbf{Y}$$

```
C(medicine$trt, contr = contr.sum)
#>  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2
#> [36] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3
#> [71] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
#> attr(,"contrasts")
#>   [,1] [,2]
#> 1    1    0
#> 2    0    1
#> 3   -1   -1
#> Levels: 1 2 3
```

We can specify the coding by transformation or `contrasts` argument in `lm()` function directly. `contrasts = list(variablename = "contr")` makes it possible.

```
(med_fit1 <- lm(y ~ trt, data = medicine, contrasts = list(trt = "contr.sum")))
#>
#> Call:
#> lm(formula = y ~ trt, data = medicine, contrasts = list(trt = "contr.sum"))
#>
#> Coefficients:
#> (Intercept)         trt1         trt2
#>       2.024       -1.142        0.147
```

ANOVA for this regression model produces the same result.

```
anova(med_fit1)
#> Analysis of Variance Table
#>
#> Response: y
#>           Df Sum Sq Mean Sq F value  Pr(>F)
#> trt        2   69.5    34.7    36.3 3.5e-12 ***
#> Residuals 87   83.3     1.0
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Denote that `trt` is for `trt1 = trt2 = 0`.

2.7.9 Dummy coding

In effects model, we can set baseline to be zero. For instance, the last level. This is what we have done in section 2.7.2.

$$\tau_k = 0$$

Then for the baseline level,

$$Y_{ik} = \mu + \epsilon_{ik}$$

Since the baseline τ_k is redundant, we have reduced parameter vector

$$\boldsymbol{\theta}^* \equiv (\mu, \tau_1, \ldots, \tau_{k-1})^T \in \mathbb{R}^k$$

This implies that

$$\underbrace{\begin{bmatrix} Y_{11} \\ Y_{21} \\ \vdots \\ Y_{n_1,1} \\ \hline Y_{12} \\ \vdots \\ Y_{n_2,2} \\ \hline \vdots \\ \hline Y_{1,k-1} \\ \vdots \\ Y_{n_{k-1},k-1} \\ \hline Y_{1k} \\ \vdots \\ Y_{n_k,k} \end{bmatrix}}_{\mathbf{Y}} = \underbrace{\left[\begin{array}{ccccc} 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 0 & \cdots & 0 \\ \hline 1 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 1 & \cdots & 0 \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots \\ \hline 1 & 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & \cdots & 1 \\ \hline 1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & \cdots & 0 \end{array}\right]}_{X^*} \underbrace{\begin{bmatrix} \mu \\ \tau_1 \\ \vdots \\ \tau_{k-1} \end{bmatrix}}_{\boldsymbol{\theta}^*} + \underbrace{\begin{bmatrix} \epsilon_{11} \\ \epsilon_{21} \\ \vdots \\ \epsilon_{n_1,1} \\ \hline \epsilon_{12} \\ \vdots \\ \epsilon_{n_2,2} \\ \hline \vdots \\ \hline \epsilon_{1,k-1} \\ \vdots \\ \epsilon_{n_{k-1},k-1} \\ \hline \epsilon_{1k} \\ \vdots \\ \epsilon_{n_k,k} \end{bmatrix}}_{\boldsymbol{\epsilon}}$$

where $X^* = \left[\mathbf{1} \mid \mathbf{x}_1 \mid \cdots \mid \mathbf{x}_{k-1}\right] \in \mathbb{R}^{N \times k}$. This makes $k-1$ indicator variables such that

$$x_{ij} = \begin{cases} 1 & j\text{-th treatment} \\ 0 & \text{otherwise} \end{cases}$$

Then the regression model becomes

$$Y_{ij} = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_{k-1} x_{i,k-1} + \epsilon_{ij}$$

with

$$\beta_0 = \mu, \quad \beta_j = \tau_j$$

Denote that

$$X^{*T} X^* = \begin{bmatrix} \sum n_j & n_1 & n_2 & \cdots & n_{k-1} \\ n_1 & n_1 & 0 & \cdots & 0 \\ n_2 & 0 & n_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ n_{k-1} & 0 & 0 & \cdots & n_{k-1} \end{bmatrix}$$

If follows that

$$
\begin{aligned}
&X^{*T}X^*\boldsymbol{\beta} = X^{*T}\mathbf{Y} \\
\Leftrightarrow &\begin{bmatrix} \sum n_j & n_1 & n_2 & \cdots & n_{k-1} \\ n_1 & n_1 & 0 & \cdots & 0 \\ n_2 & 0 & n_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ n_{k-1} & 0 & 0 & \cdots & n_{k-1} \end{bmatrix} \begin{bmatrix} \mu \\ \tau_1 \\ \vdots \\ \tau_{k-1} \end{bmatrix} = \begin{bmatrix} \sum_{i,j} Y_{ij} \\ \sum_i Y_{i1} = n_1 \overline{Y}_1 \\ \vdots \\ \sum_i Y_{i,k-1} = n_{k-1}\overline{Y}_{k-1} \end{bmatrix}
\end{aligned}
$$

In the above linear system, subtract every other line to the first one, i.e.

$$
\Big(\sum n_j - n_1 - \cdots n_{k-1}\Big)\mu + (n_1 - n_1)\tau_1 + \cdots + (n_{k-1} - n_{k-1})\tau_{k-1} = \sum_{i,j} Y_{ij} - \Big(\sum_i Y_{i1} + \cdots + \sum_i Y_{i,k-1}\Big)
$$

Then we can get

$$
n_k \mu = \sum_i Y_{ik}
$$

Therefore,

$$
\hat{\mu} = \frac{1}{n_k} \sum_i Y_{ik} = \overline{Y}_k \tag{2.46}
$$

Plug-in for each line of the system gives

$$
\begin{cases} \hat{\tau}_1 = \overline{Y}_1 - \overline{Y}_k \\ \hat{\tau}_2 = \overline{Y}_2 - \overline{Y}_k \\ \vdots \\ \hat{\tau}_{k-1} = \overline{Y}_{k-1} - \overline{Y}_k \end{cases} \tag{2.47}
$$

```
C(medicine$trt, contr = contr.treatment, base = nlevels(medicine$trt))
#>  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2
#> [36] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3
#> [71] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
#> attr(,"contrasts")
#>   1 2
#> 1 1 0
#> 2 0 1
#> 3 0 0
#> Levels: 1 2 3
```

```
(med_fit2 <-
  medicine %>%
  mutate(trt = C(medicine$trt, contr = contr.treatment, base = nlevels(medicine$trt)))
  lm(y ~ trt, data = .))
#>
#> Call:
#> lm(formula = y ~ trt, data = .)
#>
#> Coefficients:
#> (Intercept)         trt1         trt2
#>       3.018       -2.137       -0.847
```

```
anova(med_fit2)
#> Analysis of Variance Table
#>
#> Response: y
#>           Df Sum Sq Mean Sq F value  Pr(>F)
#> trt        2   69.5    34.7    36.3 3.5e-12 ***
#> Residuals 87   83.3     1.0
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2.7.10 Testing treatment effects

Recall that the main interest is to test

$$\begin{aligned} & H_0 : \mu_1 = \mu_2 = \cdots = \mu_k \\ \Leftrightarrow\ & H_0 : \tau_1 = \tau_2 = \cdots = \tau_k = 0 \end{aligned}$$

Compute

$$\begin{aligned} SSR &= \mathbf{Y}^T(\Pi_X - \Pi_{\mathbf{1}})\mathbf{Y} \\ &= \sum_{j=1}^{k} n_j(\overline{Y}_{j.})^2 - N(\overline{Y}_{..})^2, \quad N \equiv \sum n_j \\ &= \sum_{j=1}^{k} n_j(\overline{Y}_{j.} - \overline{Y}_{..})^2 \\ &= \text{between sum of squares} \end{aligned} \tag{2.48}$$

and

$$SSE = \sum_{j=1}^{k} \sum_{i=1}^{n_j} (Y_{ij} - \overline{Y}_{j.})^2 = \text{within sum of squares} \tag{2.49}$$

Since variance of each group is same,

$$\frac{SSR}{\sigma^2} \overset{H_0}{\sim} \chi^2(k-1) \perp\!\!\!\perp \frac{SSE}{\sigma^2} \sim \chi^2(N-(k-1)-1)$$

and so

$$F_0 = \frac{SSR/(k-1)}{SSE/(N-k)} \overset{H_0}{\sim} F(k-1, N-k)$$

```
anova(med_fit2)
#> Analysis of Variance Table
#>
#> Response: y
#>            Df Sum Sq Mean Sq F value  Pr(>F)
#> trt         2   69.5    34.7    36.3 3.5e-12 ***
#> Residuals 87   83.3     1.0
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2.8 Maximum Likelihood Estimation

2.8.1 Maximum likelihood estimator

For MLE, distributional assumption is needed.

$$\boldsymbol{\epsilon} \sim MVN(\mathbf{0}, \sigma^2 I) \tag{2.50}$$

Then we now have

$$Y_i \overset{indep}{\sim} N(\beta_0 + \beta_1 x_{i1} + \cdots \beta_p x_{ip}, \sigma^2) \tag{2.51}$$

It follows that

$$\begin{aligned} L(\beta_0, \ldots, \beta_p, \sigma^2) &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Big(-\frac{1}{2\sigma^2}\Big(Y_i - (\beta_0 + \cdots + \beta_p x_{ip})\Big)^2\Big) \\ &= \Big(\frac{1}{\sqrt{2\pi\sigma^2}}\Big)^n \exp\Big(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\Big(Y_i - (\beta_0 + \cdots + \beta_p x_{ip})\Big)^2\Big) \\ &= (2\pi\sigma^2)^{-\frac{n}{2}} \exp\Big(-\frac{1}{2\sigma^2}\|\mathbf{Y} - X\boldsymbol{\beta}\|^2\Big) \\ &\equiv L(\boldsymbol{\beta}, \sigma^2) \end{aligned} \tag{2.52}$$

Then log-likelihood is given by

$$\begin{aligned} l(\boldsymbol{\beta}, \sigma^2) &= \ln L(\boldsymbol{\beta}, \sigma^2) \\ &= -\frac{n}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\|\mathbf{Y} - X\boldsymbol{\beta}\|^2 \end{aligned} \tag{2.53}$$

Now we can find MLE by finding the maximum of this l.

Remark (Likelihood Equation). Since $l = -\frac{n}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}(\mathbf{Y} - X\boldsymbol{\beta})^T(\mathbf{Y} - X\boldsymbol{\beta})$,

1. $\frac{\partial l}{\partial \boldsymbol{\beta}} = \frac{1}{\sigma^2}X^T(Y - X\boldsymbol{\beta}) = 0$
2. $\frac{\partial l}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}\|\mathbf{Y} - X\boldsymbol{\beta}\|^2 = 0$

Hence,

$$\begin{cases} \hat{\boldsymbol{\beta}}^{MLE} = (X^TX)^{-1}X^T\mathbf{Y} = \hat{\boldsymbol{\beta}}^{LSE} \\ \hat{\sigma}^{2MLE} = \frac{\|\mathbf{Y} - X\hat{\boldsymbol{\beta}}\|^2}{n} \neq \hat{\sigma}^{2LSE} \end{cases} \tag{2.54}$$

2.8.2 Rao-cramer lower bound

Since $\hat{\boldsymbol{\beta}}^{MLE} = \hat{\boldsymbol{\beta}}^{LSE}$, $\hat{\boldsymbol{\beta}}^{MLE}$ is also an *ubiased estimator*, i.e.

$$E\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}$$

Extending Theorem 1.3, we can see if this unbiased estimator has the *minimum variance*.

Theorem 2.18 (Rao-Cramer Lower Bound, multivariate case)**.** *Let* $X_1, \ldots, X_n \overset{iid}{\sim} f(x; \theta 1, \ldots, \theta_{p+1})$ *and let* $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{p+1})^T$. *If* $\hat{\boldsymbol{\theta}}$ *is an unbiased estimator of* $\boldsymbol{\theta}$*, then*

$$Var(\boldsymbol{\theta}) \geq I_n^{-1}(\boldsymbol{\theta})$$

where

$$I_n(\boldsymbol{\theta}) = -E\left[\frac{\partial^2 l}{\partial \boldsymbol{\theta}^2}\right]$$

As in simple linear regression setting, assume that σ^2 is **known**. From likelihood equation, we have

$$\frac{\partial l}{\partial \boldsymbol{\beta}} = \frac{1}{\sigma^2} X^T (Y - X\boldsymbol{\beta}) \quad \text{and} \quad \frac{\partial^2 l}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} = -\frac{1}{\sigma^2} X^T X \tag{2.55}$$

It gives that

$$I_n(\boldsymbol{\beta}) = -E\left[\frac{\partial^2 l}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T}\right] = \frac{1}{\sigma^2} X^T X \tag{2.56}$$

and so

$$I_n^{-1}(\boldsymbol{\beta}) = \sigma^2 (X^T X)^{-1} = Var(\hat{\boldsymbol{\beta}}^{LSE}) \tag{2.57}$$

Hence, $\hat{\boldsymbol{\beta}}^{MLE} = \hat{\boldsymbol{\beta}}^{LSE}$ is the *minimum variance unbiased estimator*. In fact, it is same when σ^2 is unknown. Compute derivatives for σ^2 additionaly.

$$\frac{\partial l}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} \|\mathbf{Y} - X\boldsymbol{\beta}\|^2 \quad \text{and} \quad \frac{\partial^2 l}{\partial^2 \sigma^2} = \frac{n}{2\sigma^4} - \frac{1}{\sigma^6} \|\mathbf{Y} - X\boldsymbol{\beta}\|^2 \tag{2.58}$$

Then

$$I_n(\boldsymbol{\beta}, \sigma^2) = \left[\begin{array}{c|c} \frac{\partial^2 l}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} & \frac{\partial^2 l}{\partial \boldsymbol{\beta} \partial \sigma^2} \\ \hline \frac{\partial^2 l}{\partial \boldsymbol{\beta}^T \partial \sigma^2} & \frac{\partial^2 l}{\partial^2 \sigma^2} \end{array}\right] \tag{2.59}$$

By construction,

$$I_n^{-1}(\boldsymbol{\beta}, \sigma^2) = \left[\begin{array}{c|c} I_n^{-1}(\boldsymbol{\beta}) & \vdots \\ \hline \cdots & I_n^{-1}(\sigma^2) \end{array}\right] \tag{2.60}$$

In other words, it is still valid that $I_n^{-1}(\boldsymbol{\beta})$ implies minimum variance of $\hat{\boldsymbol{\beta}}$.

Chapter 3

Model Adequacy and Regression Diagnostics

$$Y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i, \quad \epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$$

From this regression model, we conduct analysis such as

- estimate $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}^2$
- inference
- predict $\hat{Y}$
- ANOVA

However, all these results make sense only when the model satisfies its assumption. Chatterjee and Hadi (2015) categorizes the assumption into four: form of the model, error term, predictors, and observations.

3.1 The Standard Regression Assumptions

3.1.1 Linearity assumption

First of all, the relation between response Y and predictors $X_1, \ldots, X_p$ is assumed to be linear.

$$E(Y \mid \mathbf{X} = \mathbf{x}) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$$

```
delv %>%
  add_predictions(delv_fit) %>%
  ggplot(aes(x = x, y = y)) +
  geom_linerange(aes(ymin = y, ymax = pred), col = gg_hcl(1)) +
  geom_smooth(method = "lm") +
```

```
  geom_point() +
  labs(
    x = "Number of Cases",
    y = "Delivery Time"
  )
```

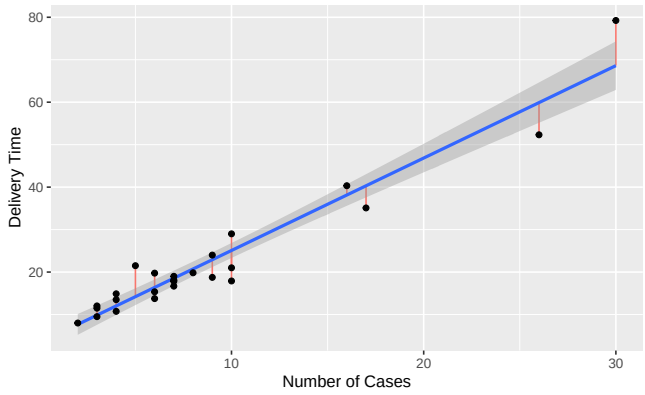


Figure 3.1: Linearity assumption

3.1.2 Errors

Error term $\boldsymbol{\epsilon}$ is assumed as

$$\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$$

This involves the following assumptions.

1. **Mean-zero assumption**: ϵ_i has zero mean.
2. **Homoskedasticity** (or homogeneity): ϵ_i has constant variance. If variance varies according to observation, we call it as *heteroskedasticity* (or heterogeneity).
3. **Independence assumption**: ϵ_i is mutually independent.
4. **Normality assumption**: ϵ_i follows Normal distribution.

As mentioned many times, ϵ_j cannot be observed. Instead, we gain information from **residuals**.

```
delv %>%
  add_predictions(delv_fit) %>%
  add_residuals(delv_fit) %>%
  ggplot(aes(x = pred, y = resid)) +
  geom_ref_line(h = 0) +
  geom_linerange(aes(ymin = resid, ymax = 0), col = gg_hcl(1), linetype = "dotted") +
  geom_point()
```

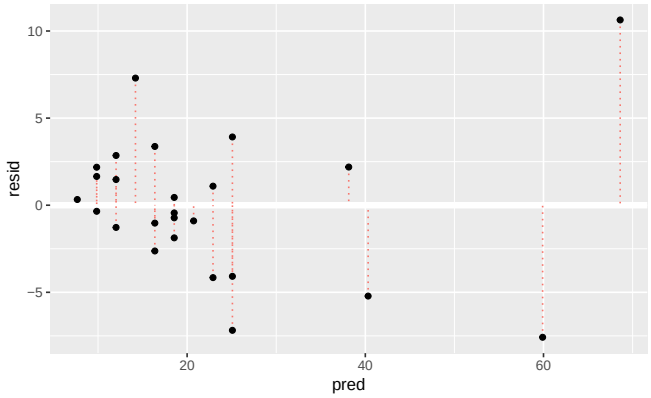


Figure 3.2: Residual plot preview - Does this model satisfy the assumptions?

3.1.3 Predictors

- **Non-random**: While Y is random variable, $X_1, \ldots, X_p$ are not. They are assumed fixed or conditioned, i.e. $E(Y \mid \mathbf{X} = \mathbf{x})$.
- **measured without error**: We assume here that $x_{i1}, \ldots, x_{ip}$ are measured without error, i.e. there is *no measurement error*. If there is, the measurement error will affect every residual variance, regression coefficient, et cetera. Let w_i be the measurement error of i-th observation. Then what we observe is $Z_i = x_i + w_i$, not x_i. In turn, we estimate regression coefficient and the others using (Z_i, Y_i), i.e. relationship between Z_i and Y_i. *Residual variance will increase by construction*. Additionaly, correlation coefficients will be reduced.
  - We have assumed that there is no measurement error, but we already know that *it is hardly not true*.
  - So correction for the errors can be considered. But it requires the ratio between $Var(w_i)$ and $Var(\epsilon_i)$, which we seldomly know.
  - As a result, we just sacrifice some accuracy for impossible correcting task.
- **Linear independence**: A set of predictor variables $\{X_1, \ldots, X_p\}$ is linearly independent. Recall Theorem 2.1. This makes the design matrix full rank and guarantee unique least squares solution. This is violated by so-called *multicollinearity* problem.

First two assumptions cannot be validated, but should be always remembered in the interpretation of the analysis results.

3.1.4 Observations

Every observation is equally reliable and plays an equal role in determining the results. When we estimate mean, for instance, each $X_1, \ldots, X_n$ has equal role so that

$$\hat{\mu} = \frac{1}{n}X_1 + \cdots + \frac{1}{n}X_n = \overline{X}$$

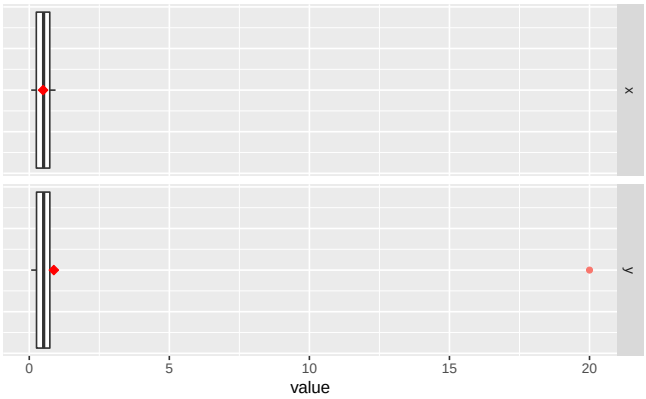


Figure 3.3: Existence of Outlier

However, if there exists *outlier*, it can be changed critically. See Figure 3.3. In lower box plot, just one outlier is added: `20`. Each red dot is average. Thus, we should be careful about *outliers*. In regression literature, we would see the following observations.

- Leverage point
- Influential point

3.2 Residuals

3.2.1 Raw residual

Looking at ϵ_i, we might be able to check if the model violates the assumptions directly in section 3.1.1 and 3.1.2. The problem is ϵ_i is non-observable. So as a surrogate, we use residuals

$$e_i := Y_i - \hat{Y}_i = Y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_p x_{ip}), \quad i = 1, \ldots, n \tag{3.1}$$

We already know that `residuals(model)` or `model$residuals` gives residuals. `modelr::add_residuals(model)`, additionaly, mutates a colum named `resid` by default. We can change its name with `var` argument. See Figure 3.2 and its code.

Let

$$\mathbf{e} := (e_1, \ldots, e_p)^T$$

From Proposition 2.6,

$$\mathbf{e} \sim N(\mathbf{0}, \sigma^2(I - H_X)), \quad H_X = \text{projection onto } R(X)$$

This implies that

$$Var(e_i) = (1 - h_{ii})\sigma^2$$

where h_{ii} is the i-th diagonal element of H_X. Recall that

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

As a backup of σ^2, e_i should reflect the assumption of ϵ_i. *Scaling* might be needed.

3.2.2 Standardized residual

Applying usual standardization procedure, one may use

$$e_i^* = \frac{e_i}{\sigma\sqrt{1 - h_{ii}}} \tag{3.2}$$

σ is unkown. So replace σ with $\hat{\sigma}$.

Definition 3.1 (Standardized residual)**.** Standardized residual can be obtained by replacing σ with $\hat{\sigma}$.

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

where $\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} e_i^2}{n-p-1}$.

We can get values of d_i with `rstandard()`.

```
rstandard(delv_fit)
#>       1       2       3       4       5       6       7       8       9
#> -0.4581  0.4092  0.5406  0.7041 -0.6437 -0.1085  0.0815 -0.1770  3.3893
#>      10      11      12      13      14      15      16      17      18
#>  1.7931  0.5480 -0.9972  0.3637  0.8260  0.2670  0.9569 -0.2517  0.1090
#>      19      20      21      22      23      24      25
#> -0.0866 -1.3146 -1.7544 -2.1694 -1.0145 -0.2198 -0.3146
```

In fact, we can get both residuals using `broom::augment()`.

- `.fitted`: fitted values

- `.resid`: raw residuals
- `.std.resid`: standardized residuals

```
delv_fit %>%
  broom::augment() %>%
  gather(.resid, .std.resid, key = "residual", value = "value") %>%
  ggplot(aes(x = .fitted, y = value, colour = residual)) +
  geom_ref_line(h = 0) +
  geom_linerange(aes(ymin = value, ymax = 0), linetype = "dotted") +
  geom_point() +
  labs(
    y = "Residuals",
    x = "Prediction"
  ) +
  scale_colour_discrete(
    name = "Residuals",
    label = c("Raw", "Std")
  )
```

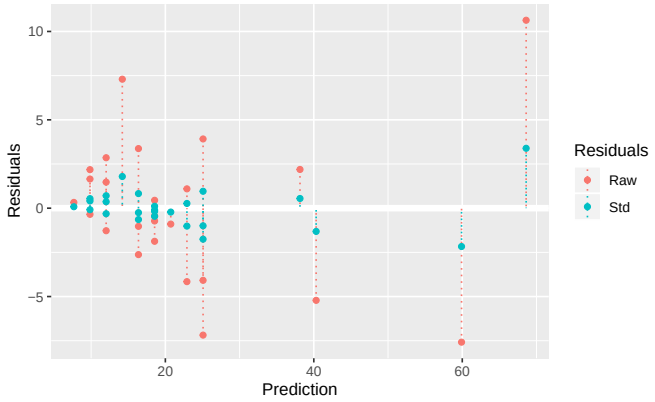


Figure 3.4: Standardized residuals

Does this d_i follow $t(n-p-1)$? $Z := \frac{e_i}{\sigma\sqrt{1-h_{ii}}} \sim N(0,1)$ and $X := \frac{\hat{\sigma}}{\sigma} \sim \chi^2(n-p-1)$. In the other context, this typically leads to t-distribution. Here, however, Z and X are not independent. Denote that $\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} e_i^2}{n-p-1}$ includes e_i, so it cannot be independent with e_i. Thus, d_i is not t distributed.

3.2.3 Studentized residual

To force d_i to be t random variable, we just prevent from each e_i encountering each e_i in $\hat{\sigma}^2$. This can be done by computing MSE without i-th observation, i.e. using data set

$$D_{(-i)} := \begin{bmatrix} (\mathbf{x}_1, Y_1) \\ (\mathbf{x}_2, Y_2) \\ \vdots \\ \cancel{(\mathbf{x}_i, Y_i)} \\ \vdots \\ (\mathbf{x}_n, Y_n) \end{bmatrix}$$

compute MSE, $\hat{\sigma}^2_{(-i)}$.

Definition 3.2 (Studentized residual)**.** The studentized residual is defined by

$$r_i := \frac{e_i}{\hat{\sigma}_{(-i)}\sqrt{1-h_{ii}}}$$

where $\hat{\sigma}^2_{(-i)}$ is MSE from data without i-th observation, $D_{(-i)}$.

In principle, we need to fit n different models to get this residuals. In fact, this is not necessary because $\hat{\sigma}^2_{(-i)}$ has some relation to $\hat{\sigma}^2$.

Lemma 3.1. *$\hat{\sigma}^2_{(-i)}$ can be computed from the original $\hat{\sigma}^2$.*

$$\hat{\sigma}^2_{(-i)} = \frac{SSE_{(-i)}}{n-p-2} = \frac{(n-p-1)\hat{\sigma}^2 - \frac{e_i^2}{1-h_{ii}}}{n-p-2}$$

where $SSE_{(-i)}$ is the SSE when fitting the model without i-th observation.

This lemma removes repeating procedure.

Corollary 3.1. *Studentized residuals and standardized residuals are related by*

$$r_i = d_i\sqrt{\frac{n-p-2}{n-p-1-d_i^2}}$$

where d_i are standardized residuals and r_i are studentized residuals.

`rstudent()` gives this r_i in `R`.

```
rstudent(delv_fit)
#>       1       2       3       4       5       6       7       8       9
#> -0.4500  0.4017  0.5321  0.6962 -0.6353 -0.1062  0.0797 -0.1732  4.6851
#>      10      11      12      13      14      15      16      17      18
#>  1.8908  0.5395 -0.9970  0.3567  0.8201  0.2615  0.9551 -0.2466  0.1067
#>      19      20      21      22      23      24      25
#> -0.0847 -1.3369 -1.8436 -2.3790 -1.0152 -0.2152 -0.3084
```

Check Corollary 3.1 in our example.

```
# n = 25
# p = 1
rstandard(delv_fit) * sqrt((25 - 3) / (25 - 2 - rstandard(delv_fit)^2))
#>       1       2       3       4       5       6       7       8       9
#> -0.4500  0.4017  0.5321  0.6962 -0.6353 -0.1062  0.0797 -0.1732  4.6851
#>      10      11      12      13      14      15      16      17      18
#>  1.8908  0.5395 -0.9970  0.3567  0.8201  0.2615  0.9551 -0.2466  0.1067
#>      19      20      21      22      23      24      25
#> -0.0847 -1.3369 -1.8436 -2.3790 -1.0152 -0.2152 -0.3084
```

`broom::(augment)` does not provide `rstudent()` yet, so here `mutate` them all in hand.

```
delv %>%
  add_predictions(delv_fit) %>%
  mutate(
    resid = residuals(delv_fit),
    std = rstandard(delv_fit),
    stud = rstudent(delv_fit)
  ) %>%
  gather(resid, std, stud, key = "residual", value = "value") %>%
  ggplot(aes(x = pred, y = value, colour = residual)) +
  geom_ref_line(h = 0) +
  geom_linerange(aes(ymin = value, ymax = 0), linetype = "dotted") +
  geom_point() +
  labs(
    y = "Residuals",
    x = "Prediction"
  ) +
  scale_colour_discrete(
    name = "Residuals",
    label = c("Raw", "Std", "Student")
  )
```

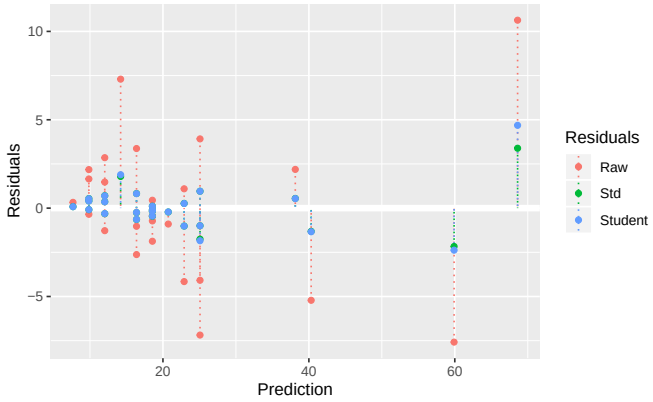


Figure 3.5: Studentized residuals

Chatterjee and Hadi (2015) names the above three (3.2) e_i^*, 3.1 d_i, and 3.2 r_i as follows.

Remark. The form of residual in Equation (3.2), Definition 3.1, and 3.2 can be called as

1. $e_i^* := \frac{e_i}{\sigma\sqrt{1-h_{ii}}}$ *Standardized residual*
2. $d_i := \frac{e_i}{\hat{\sigma}\sqrt{1-h_{ii}}}$ *Internally studentized residual*
3. $r_i := \frac{e_i}{\hat{\sigma}_{(-i)}\sqrt{1-h_{ii}}}$ *Externally studentized residual*

The word *internal* and *external* is due to direct involvement of $\hat{\sigma}$ in e_i. In externally studentized residual, $\hat{\sigma}_{(-i)}$ is not involved in e_i, i.e. external to e_i.

3.3 Residual Plots

Graphical methods are able to become some kind of caveat. It is effective way to investigate model adequacy.

```
gg_fit <- function(data, mapping) {
  pt <- gg_scatter(data, mapping = mapping, alpha = 1)
  pt$layers <- c(geom_smooth(method = "lm"), pt$layers)
  pt # point layer on line layer
}
#-----------------------------------
draw_dot <- function(data, mapping, ...) {
  data %>%
    ggplot(mapping = mapping) +
    geom_dotplot(...)
}
```

Since `gg_scatter()` is pre-defined function drawing a scatter plot, we can change the order of layers by indexing its `layers`. This is because every `geom_*` and `stat_*` returns `layer()`. For example,

```
geom_smooth
#> function (mapping = NULL, data = NULL, stat = "smooth", position = "identity",
#>     ..., method = "auto", formula = y ~ x, se = TRUE, na.rm = FALSE,
#>     show.legend = NA, inherit.aes = TRUE)
#> {
#>     params <- list(na.rm = na.rm, se = se, ...)
#>     if (identical(stat, "smooth")) {
#>         params$method <- method
#>         params$formula <- formula
#>     }
#>     layer(data = data, mapping = mapping, stat = stat, geom = GeomSmooth,
#>         position = position, show.legend = show.legend, inherit.aes = inherit.aes,
#>         params = params)
#> }
#> <bytecode: 0x7f8679e503e0>
#> <environment: namespace:ggplot2>
```

`GGally::ggpairs()` easily draws a matrix of plots. We can specify each `lower`, `upper`, and `diag`.

```
cem %>%
  GGally::ggpairs(
    lower = list(continuous = gg_fit), # regression
    diag = list(continuous = draw_dot) # dot plot
  )
```

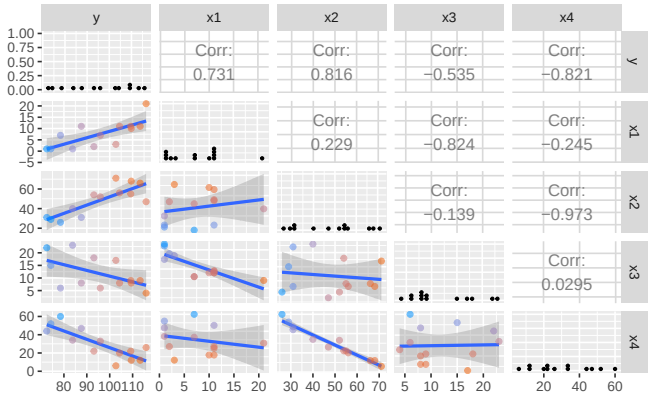


Figure 3.6: Plot matrix for cement data set

Plots that have scatter plot in the matrix form and pairwise correlation such as

Figure 3.6 is called *plot matrix* or *scatter matrix*. First column let us know the relationship between `y` and each `x`. Correlation is important in that we have assumed linear independence. This would be covered later.

3.3.1 Residual plot

Direct illustration of Y and $X_1, \ldots, X_p$ is helpful, but has its limit to check the assumptions. Residuals might be more informative. Among the three we have defined, we often draw *externally studentized residual* 3.2, i.e. `rstudent()`.

$$r_i := \frac{e_i}{\hat{\sigma}_{(-i)}\sqrt{1-h_{ii}}}$$

Definition 3.3 (Residual plot)**.** Residual plot is a plot of r_i versus corresponding fitted values $\hat{Y}_i$

In `modelr` syntax,

```
add_rstudent <- function(data, model, var = ".stud.resid") {
  data[[var]] <- rstudent(model, x = data)
  data
}
#---------------------------------------
residplot <- function(data, model, ...) {
  data %>%
    add_predictions(model) %>%
    add_rstudent(model) %>%
    ggplot(data, mapping = aes(x = pred, y = .stud.resid)) +
    geom_ref_line(h = 0) +
    geom_point(...)
}
```

```
residplot(cem, cem_fit) +
  labs(
    x = "Fitted values",
    y = "Residuals"
  )
```

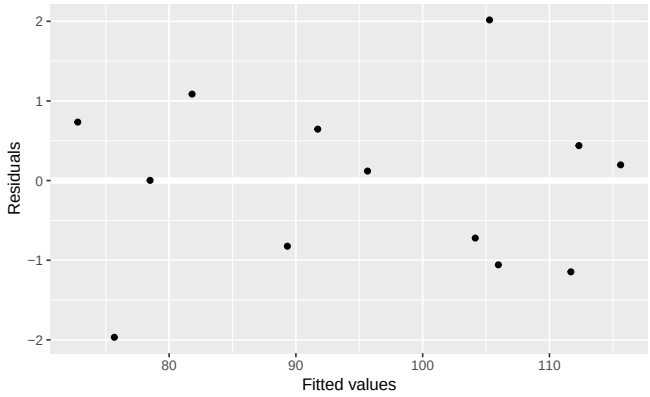


Figure 3.7: Residual plot for regression with cement data set

Denote that

$$r_i \overset{iid}{\approx} N(0,1)$$

So we can deduct some ideal behavior of r_i.

Remark. Ideally, residual plot should be

- have *no systematic pattern*
- equal variance, i.e. *variability of r_i shows constancy*, independent of $\hat{Y}_i$
- most r_is fall between -2 and 2, approximately 95%

Now consider various scenarios and see how to interpret this plot.

```
residplot(hetero, hetero_fit)
```

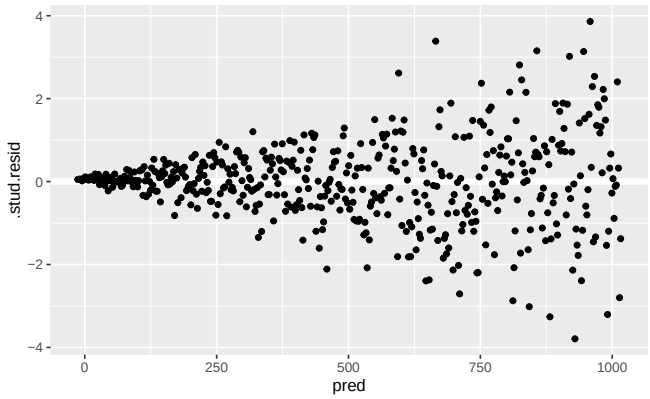


Figure 3.8: Heteroskedasticity

As i grows, variance becomes larger. Constant variance assumption is violated.

```
residplot(cubic, cubic_fit)
```

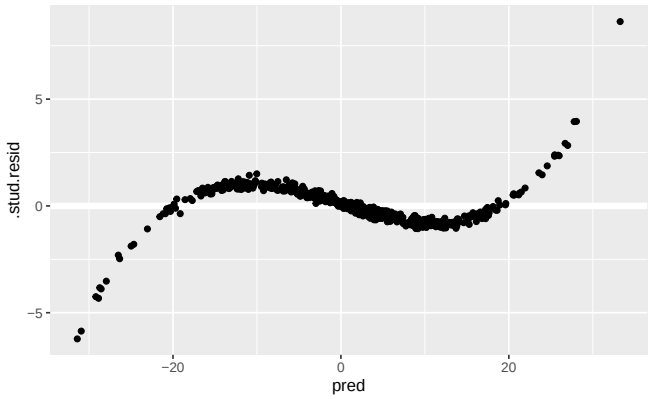


Figure 3.9: Non-linear relationship and autocorrelation

By its definition, residuals are values after removing linear effects. If the data embeds non-linear relationships, it would not be removed and it would be still remained as pattern. Figure 3.9 indicates that Y and x have non-linear relationship.

```
residplot(outlier, outlier_fit) +
  geom_point(aes(colour = isout), na.rm = TRUE, shape = 1, size = 3, show.legend = FALSE) +
  scale_colour_manual(values = c("TRUE" = "red", "FALSE" = NA))
```

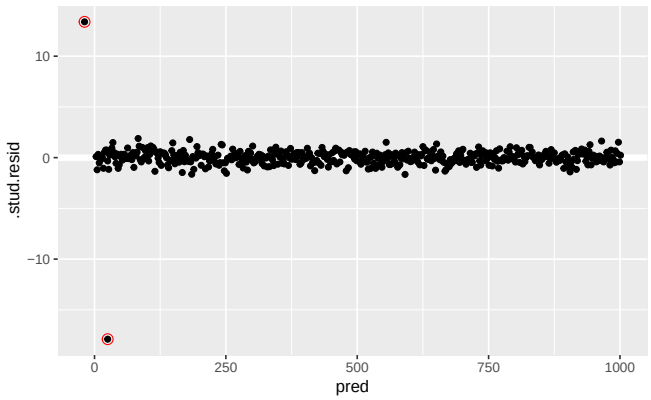


Figure 3.10: Existence of outlier

Large values of residuals implies they are outliers.

3.3.2 Normal quantile-quantile plot

We expect that

$$r_i \overset{iid}{\sim} N(0,1)$$

Normal quantile-quantile plot (q-q plot) or normal probability plot can tell us whether observations of r_i is closed to *Normal distribution*.

Definition 3.4 (Normal q-q plot)**.** Normal Q-Q plot is the plot of *ordered studentized residuals* $r_{(1)} < r_{(2)} < \cdots < r_{(n)}$ versus *theoretical normal quantiles* $\Phi^+\left(\frac{i-\frac{3}{8}}{n+\frac{1}{4}}\right)$.

If Q-Q plot is close to a straight line, this supports the Normality of residuals. Otherwise, we can say that the assumption is violated.

`ggplot2` provides `stat_qq_line()` and `stat_qq()`. One draws the guide line and the other draws a points of q-q plot. There is an argument `distribution`. We can compare observations with any distribution function. The default, of course, is `stats::qnorm`, i.e. normal distribution.

```
cem %>%
  add_rstudent(cem_fit) %>%
  ggplot(aes(sample = .stud.resid)) +
  stat_qq_line(
    distribution = stats::qnorm,
    col = I("white"),
    size = 2
  ) +
  stat_qq(distribution = stats::qnorm)
```

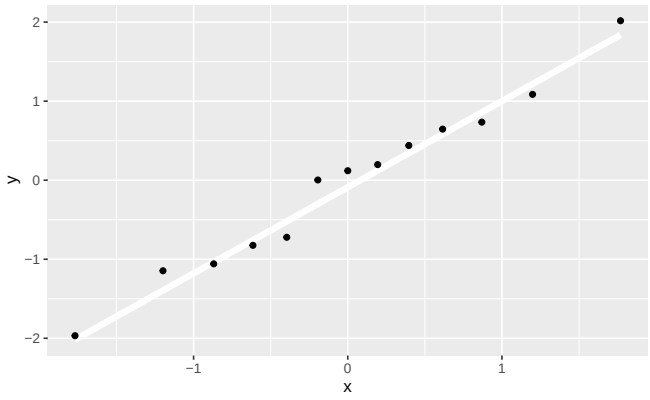


Figure 3.11: Q-Q plot for cement residuals

In general, we filter *skewness* here.

```
draw_qq <- function(data, model, distribution = stats::qnorm, ...) {
  data %>%
    add_rstudent(model) %>%
    ggplot(aes(sample = .stud.resid)) +
    stat_qq_line(distribution = distribution, ...) +
    stat_qq(distribution = distribution) +
    xlab("Theoretical Quantiles")
}
```

```
draw_qq(ideal, ideal_fit, col = I("white"), size = 2)
```

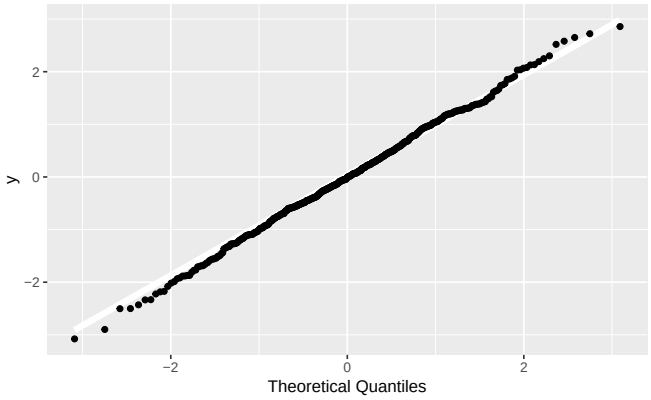


Figure 3.12: Q-Q plot - Ideal case

Figure 3.12 shows the ideal case. Observations resemble the straight line.

```
draw_qq(fat, fat_fit, col = I("white"), size = 2)
```

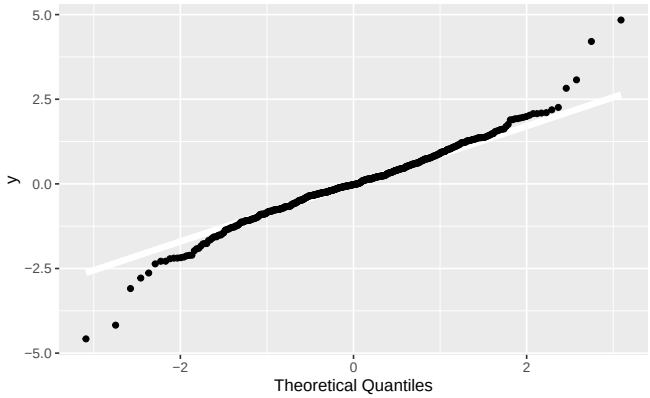


Figure 3.13: Q-Q plot - Heavy-tailed distribution

Recall that the x-axis is *theoretical* normal quantile and y-axis is the *observed* ordered studentized residuals. In case of small values of r_i, i.e. *left tail*, observed values indicating empirical quantiles are less than theoretical quantiles. *Smaller quantiles at left tail means heavier left tail.* At *right tail*, on the other hand, observed values indicating empirical quantiles are larger than theoretical quantiles. *Larger quantiles at right tail means heavier right tail.* In sum, Figure 3.13 form is of heavy-tailed distribution.

```
draw_qq(thin, thin_fit, col = I("white"), size = 2)
```

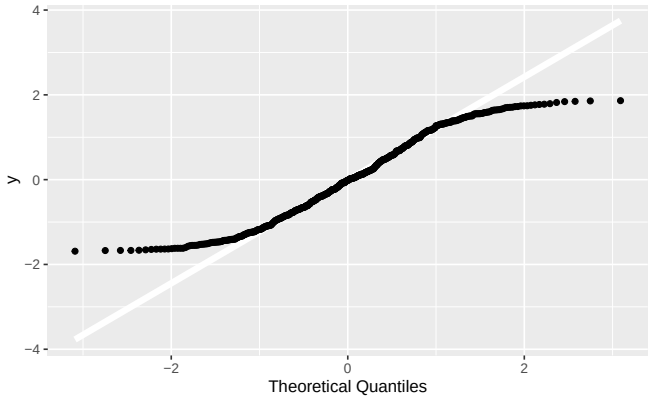


Figure 3.14: Q-Q plot - Light-tailed distribution

This is exactly opposite result with Figure 3.13. Larger empirical quantiles at left tail implies light left tail. Smaller empirical quantiles at right tail implies light right tail. This is the form of light-tailed distribution.

```
draw_qq(right, right_fit, col = I("white"), size = 2)
```

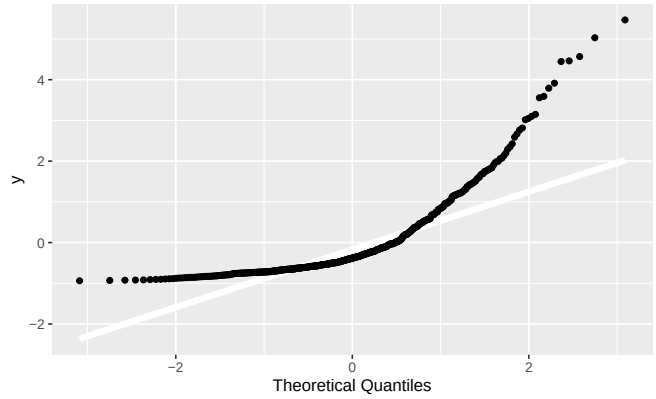


Figure 3.15: Q-Q plot - Positive skew

In this case, r_i is observed as heavier right tail than left. This is called *positive skew* or *right skew*, i.e. its mass leans to the left.

```
draw_qq(left, left_fit, col = I("white"), size = 2)
```

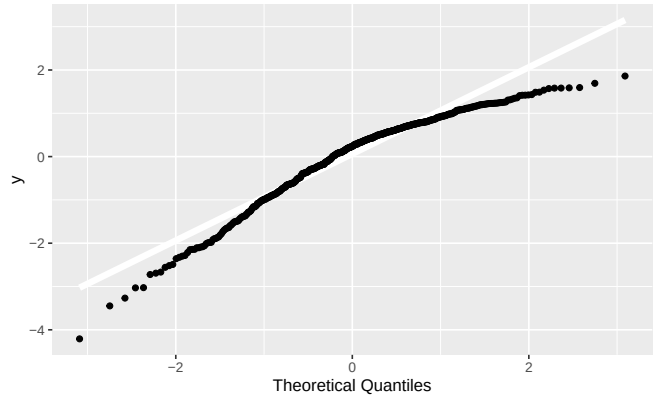


Figure 3.16: Q-Q plot - Negative skew

Left tail is observed heavier than right. This is called *negative skew* or *left skew*. Its mass leans to the right.

3.3.3 Partial residual plots

3.4 Outliers

An outlier is an extreme observation. Outliers appear in two direction: X and Y. There exist appropriate measures to find outliers in corresponding direction.

3.4.1 Leverage

Consider the design matrix and its row vectors

$$X = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix}$$

Note that each $\mathbf{x}_i^T = (1, x_{i1}, x_{i2}, \ldots, x_{ip})^T$ represents i-th observation.

Definition 3.5 (Leverage)**.** Let H be hat matrix, i.e. projection onto $Col(X)$. Then leverage is defined by its i-th diagonal element.

$$h_{ii} := \mathbf{x}_i^T (X^T X)^{-1} \mathbf{x}_i$$

Let us consider the simple linear regression model, $p = 1$. Then we have

$$(X^T X)^{-1} = \frac{1}{\sigma^2} Var(\hat{\boldsymbol{\beta}}) = \begin{bmatrix} \left(\frac{1}{n} + \frac{\overline{x}^2}{S_{xx}}\right) & -\frac{\overline{x}}{S_{xx}} \\ -\frac{\overline{x}}{S_{xx}} & \frac{1}{S_{xx}} \end{bmatrix}$$

It follows that

$$\begin{aligned} h_{ii} &= \begin{bmatrix} 1 & x_i \end{bmatrix} \begin{bmatrix} \left(\frac{1}{n} + \frac{\overline{x}^2}{S_{xx}}\right) & -\frac{\overline{x}}{S_{xx}} \\ -\frac{\overline{x}}{S_{xx}} & \frac{1}{S_{xx}} \end{bmatrix} \begin{bmatrix} 1 \\ x_i \end{bmatrix} \\ &= \frac{1}{n} + \frac{(x_i - \overline{x})^2}{S_{xx}} \end{aligned}$$

$(x_i - \overline{x})^2$ implies that h_{ii} becomes larger as x_i is far from $\overline{x}$. In other words, h_{ii} represents how far x_i is away from the center. Extending to p, h_{ii} represents *how far* $\mathbf{x}_i$ *is away from the center of observations.*

Proposition 3.1 (Properties of h_{ii})**.** *Leverage values* h_{ii} *possess several properties*

1. $\frac{1}{n} \leq h_{ii} \leq 1$
2. $\sum_{i=1}^{n} h_{ii} = p + 1 \Rightarrow \overline{h} = \frac{1}{n}\sum h_{ii} = \frac{p+1}{n}$
3. $Var(\hat{Y}_i) = h_{ii}\sigma^2$

Conjecture 3.1 (High leverage point)**.** *If $h_{ii} > 2\overline{h} = \frac{2(p+1)}{n}$ (twice the average value), then we regard i-th observation as* ***high leverage point****.*

If i-th observation is a high leverage point, we can consider that this observation is ***unusual*** *in X-space.*

In `R`, h_{ii} can be get in various ways. `influence()` returns `list` including `hat` which is leverage values. `broom::augment()` has a column `.hat` and this is the leverage values.

```
broom::augment(cem_fit) %>%
  select(.hat)
#> # A tibble: 13 x 1
#>    .hat
#>   <dbl>
#> 1 0.550
#> 2 0.333
#> 3 0.577
#> 4 0.295
#> 5 0.358
#> 6 0.124
#> # ... with 7 more rows
```

High leverage point is potentially dangerous for estimation of regression coefficients.

$$\hat{Y}_i = h_{i1}Y_1 + h_{i2}Y_2 + \cdots + h_{ii}Y_i + \cdots + h_{in}Y_n$$

h_{ii} is a function of $\mathbf{x}_i$. By construction, it measures a role of each $\mathbf{x}_i$ in weight of observation Y_i in determining $\hat{Y}_i$. So a small change of Y_i corresponding to a high leverage can dramatically change the estimators. However, *high leverage points are not always influential points.*

3.4.2 Influence measure

Influence points are the points that can change the values of estimates by their existence. To see the influence of each data point, we can focus on this meaning. *How much would the regression results change if the i-th observation were deleted?*

Consdier $D_{(-i)}$. Let $X_{(-i)} \in (n-1) \times (p+1)$ be the design matrix from this data set, i.e. design matrix without i-th observation.

Lemma 3.2. *Let X be any design matrix. Then*

$$X = \begin{bmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_i^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix}$$

where $\mathbf{x}_i = (1, \mathbf{x}_{i1}, \ldots, \mathbf{x}_{ip})^T$. By construction,

$$X^T X = \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T$$

Let $Y = (Y_1, \ldots, Y_n)^T$ be any observation vector. Then

$$X^T \mathbf{Y} = \sum_{i=1}^{n} \mathbf{x}_i Y_i$$

Proof. It is just arithmetic.

$$X^T X = \begin{bmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_i & \cdots & \mathbf{x}_n \end{bmatrix} \begin{bmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_i^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix} = \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T$$

and

$$X^T \mathbf{Y} = \begin{bmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_i & \cdots & \mathbf{x}_n \end{bmatrix} \begin{bmatrix} Y_1 \\ \vdots \\ Y_i \\ \vdots \\ Y_n \end{bmatrix} = \sum_{i=1}^{n} \mathbf{x}_i Y_i$$

□

Lemma 3.3. *It can be shown that*

$$(A + BCB)^{-1} = A^{-1} - A^{-1}B(C^{-1} + B^T A^{-1} B)^{-1} B^T A^{-1}$$

From Lemma 3.2,

$$\begin{aligned}\hat{\boldsymbol{\beta}}_{(-i)} &= (X_{(-i)}^T X_{(-i)})^{-1} X_{(-i)}^T \mathbf{Y}_{(-i)} \\ &= \left(\sum_{j\neq i} \mathbf{x}_j \mathbf{x}_j^T\right)^{-1} \left(\sum_{j\neq i} \mathbf{x}_j Y_j\right) \\ &= (X^T X - \mathbf{x}_i \mathbf{x}_i^T)^{-1} (X^T \mathbf{Y} - \mathbf{x}_i Y_i)\end{aligned} \tag{3.3}$$

In Lemma 3.3, take $A = X^T X$, $B = \mathbf{x}_i$, and $C = -1$. It gives that

$$\begin{aligned}(X^T X - \mathbf{x}_i \mathbf{x}_i^T)^{-1} &= (X^T X)^{-1} - (X^T X)^{-1} \mathbf{x}_i (-1 + \mathbf{x}_i^T (X^T X)^{-1} \mathbf{x}_i)^{-1} \mathbf{x}_i^T (X^T X)^{-1} \\ &= (X^T X)^{-1} + \frac{1}{1 - h_{ii}} (X^T X)^{-1} \mathbf{x}_i \mathbf{x}_i^T (X^T X)^{-1}\end{aligned}$$

Thus,

$$\begin{aligned}\hat{\boldsymbol{\beta}}_{(-i)} &= (X^T X - \mathbf{x}_i \mathbf{x}_i^T)^{-1} (X^T \mathbf{Y} - \mathbf{x}_i Y_i) \\ &= \left[(X^T X)^{-1} + \frac{1}{1 - h_{ii}} (X^T X)^{-1} \mathbf{x}_i \mathbf{x}_i^T (X^T X)^{-1}\right] (X^T \mathbf{Y} - \mathbf{x}_i Y_i) \\ &= (X^T X)^{-1} X^T \mathbf{Y} - (X^T X)^{-1} \mathbf{x}_i Y_i \\ &\quad + \frac{1}{1 - h_{ii}} (X^T X)^{-1} \mathbf{x}_i \mathbf{x}_i^T (X^T X)^{-1} X^T \mathbf{Y} - \frac{1}{1 - h_{ii}} (X^T X)^{-1} \mathbf{x}_i \mathbf{x}_i^T (X^T X)^{-1} \mathbf{x}_i Y_i \\ &= \hat{\boldsymbol{\beta}} - (X^T X)^{-1} \mathbf{x}_i Y_i + \frac{1}{1 - h_{ii}} (X^T X)^{-1} \mathbf{x}_i \mathbf{x}_i^T \hat{\boldsymbol{\beta}} - \frac{h_{ii}}{1 - h_{ii}} (X^T X)^{-1} \mathbf{x}_i Y_i \\ &= \hat{\boldsymbol{\beta}} - \left[1 + \frac{h_{ii}}{1 - h_{ii}}\right] (X^T X)^{-1} \mathbf{x}_i Y_i + \frac{1}{1 - h_{ii}} (X^T X)^{-1} \mathbf{x}_i \hat{Y}_i \quad \because \mathbf{x}_i^T \hat{\boldsymbol{\beta}} = \hat{Y}_i \\ &= \hat{\boldsymbol{\beta}} - \frac{1}{1 - h_{ii}} (X^T X)^{-1} \mathbf{x}_i (Y_i - \hat{Y}_i) \quad \because 1 + \frac{h_{ii}}{1 - h_{ii}} = \frac{1}{1 - h_{ii}} \\ &= \hat{\boldsymbol{\beta}} - \frac{1}{1 - h_{ii}} (X^T X)^{-1} \mathbf{x}_i e_i\end{aligned}$$

Theorem 3.1. *Regression coefficient without i-th observation can come from the original desing matrix. Also, it is affected by i-th residual.*

$$\hat{\boldsymbol{\beta}}_{(-i)} = \hat{\boldsymbol{\beta}} - \frac{1}{1 - h_{ii}} (X^T X)^{-1} \mathbf{x}_i e_i$$

3.4.3 PRESS residual

See Theorem 3.1. Previous residuals might be able to detect outliers. However, other kind of residuals can be considered here.

Definition 3.6 (PRESS residuals)**.** *PRESS* residuals are defined by

$$e_{i,-i} = Y_i - \hat{Y}_{i,-i}$$

where $\hat{Y}_{i,-i} = \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_{(-i)}$ is the fitted value of the i-th response without i-th observation.

Note that

$$\hat{Y}_{i,-i} = \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_{(-i)}$$

From Theorem 3.1, we can get a useful identity about PRESS residual.

Theorem 3.2. *PRESS residuals are related to Raw residuals e_i and leverage values h_{ii}.*

$$e_{i,-i} = \frac{e_i}{1 - h_{ii}}$$

Proof. Theorem 3.1 implies that

$$\begin{aligned}
e_{i,-i} &= Y_i - \hat{Y}_{i,-i} \\
&= Y_i - \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_{(-i)} \\
&= Y_i - \mathbf{x}_i^T \left[\hat{\boldsymbol{\beta}} - \frac{1}{1 - h_{ii}} (X^T X)^{-1} \mathbf{x}_i e_i \right] \\
&= Y_i - \mathbf{x}_i^T \hat{\boldsymbol{\beta}} + \frac{h_{ii}}{1 - h_{ii}} e_i \\
&= \frac{e_i}{1 - h_{ii}} \qquad \because Y_i - \mathbf{x}_i^T \hat{\boldsymbol{\beta}} = e_i
\end{aligned}$$

□

Althogh the definition of PRESS requires refitting different models, this Thoerem 3.2 let us easily compute $e_{i,-i}$ without refitting data.

Remark (Standardized PRESS residual). We standardize PRESS residuals $e_{i,-i}$ by

$$\frac{e_{i,-i}}{\sqrt{Var(e_{i,-i})}} = \frac{e_i / 1 - h_{ii}}{\sqrt{\sigma^2/(1 - h_{ii})}} = \frac{e_i}{\sigma\sqrt{1 - h_{ii}}}$$

It is same as *standardized residuals* if replacing σ^2 with $\hat{\sigma}^2$.

3.4.4 DFFITS

Definition 3.7 (DFFITS)**.**

$$(DFFITS)_i := \frac{\hat{Y}_i - \hat{Y}_{i,-i}}{\hat{\sigma}_{(-i)}\sqrt{h_{ii}}}$$

where $\hat{Y}_i = \mathbf{x}_i^T \hat{\boldsymbol{\beta}}$ and $\hat{Y}_{i,-i} = \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_{(-i)}$

Corollary 3.2.

$$(DFFITS)_i = \frac{e_i}{\hat{\sigma}_{(-i)}\sqrt{h_{ii}}} \times \sqrt{\frac{h_{ii}}{1-h_{ii}}} = \textit{studentized residual} \times \textit{leverage measure}$$

Proof.

$$\begin{aligned}
\hat{Y}_i - \hat{Y}_{i,-i} &= -\Big[(Y_i - \hat{Y}_i) - (Y_i - \hat{Y}_{i,-i})\Big] \\
&= -e_i + e_{i,-i} \\
&= \frac{h_{ii}}{1-h_{ii}} e_i
\end{aligned}$$

□

The second part represents leverage. As leverage values become large, i.e. $h_{ii} \to 1$ (Proposition 3.1), this becomes ∞. This form of function expressed by h_{ii} is called *potential function*

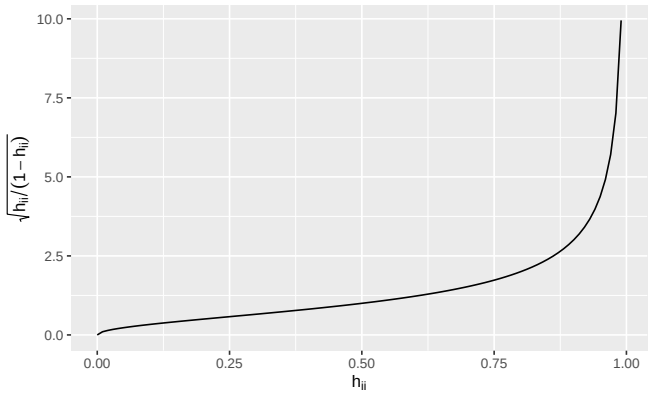


Figure 3.17: Shape of potential function

Conjecture 3.2 (Rule of thumb). *If* $|(DFFITS)_i| > 2\sqrt{\frac{p+1}{n-p-1}}$, *then* i*-th observation is considered to be influential.*

There is `dffits()` so that we can easilty get only $(DFFITS)_i$ for every i

```
tibble(DFFITS = dffits(cem_fit)) %>%
  mutate(i = 1:n()) %>%
  ggplot(aes(x = i, y = DFFITS)) +
  geom_ref_line(h = 2 * sqrt( (4 + 1) / (nrow(cem) - 4 - 1) )) +
  geom_point() +
  geom_linerange(aes(ymin = DFFITS, ymax = 0), col = gg_hcl(1), linetype = "dotted")
```

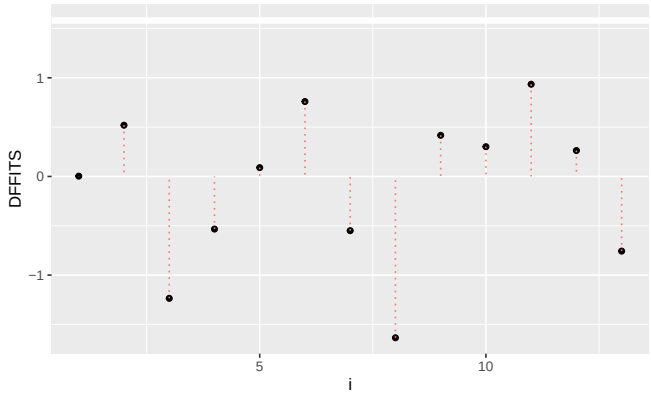


Figure 3.18: DFFITS for fitted cement dataset

3.4.5 Cook's Distance

Definition 3.8 (Cookś distance).

$$C_i := \frac{\sum_{j=1}^{n} (\hat{Y}_j - \hat{Y}_{j,-i})^2}{\hat{\sigma}(p+1)}$$

where $\hat{Y}_{j,-i} = \mathbf{x}_j^T \hat{\boldsymbol{\beta}}_{(-i)}$.

Cook's distnace is related to the distance between $\hat{\mathbf{Y}}$ and $\hat{\mathbf{Y}}_{(-i)}$.

Corollary 3.3. *Cooks's distance is related to the distance between* $\hat{\boldsymbol{\beta}}$ *and* $\hat{\boldsymbol{\beta}}_{(-i)}$.

$$C_i = \frac{(\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{(-i)})^T X^T X (\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{(-i)})}{\hat{\sigma}(p+1)}$$

Proof. It is just arithmetic if plug-in $\hat{Y}_j = \mathbf{x}_j^T \hat{\boldsymbol{\beta}}$ $\hat{Y}_{j,-i} = \mathbf{x}_j^T \hat{\boldsymbol{\beta}}_{(-i)}$.

$$\begin{aligned} C_i &= \frac{(\mathbf{Y}-\mathbf{Y}_{-i})^T(\mathbf{Y}-\mathbf{Y}_{-i})}{\hat{\sigma}^2(p+1)} \\ &= \frac{(X\hat{\boldsymbol{\beta}}-X\hat{\boldsymbol{\beta}}_{(-i)})^T(X\hat{\boldsymbol{\beta}}-X\hat{\boldsymbol{\beta}}_{(-i)})}{\hat{\sigma}^2(p+1)} \\ &= \frac{(\hat{\boldsymbol{\beta}}-\hat{\boldsymbol{\beta}}_{(-i)})^T X^T X(\hat{\boldsymbol{\beta}}-\hat{\boldsymbol{\beta}}_{(-i)})}{\hat{\sigma}^2(p+1)} \end{aligned}$$

□

Remark. C_i can be represented by internally studentized residual and potential function

$$C_i = \left(\frac{e_i}{\hat{\sigma}\sqrt{1-h_{ii}}}\right)^2 \times \frac{1}{p+1} \times \frac{h_{ii}}{1-h_{ii}}$$

Proof. From Theorem 3.1 ($\hat{\boldsymbol{\beta}}_{(-i)} = \hat{\boldsymbol{\beta}} - \frac{1}{1-h_{ii}}(X^TX)^{-1}\mathbf{x}_i e_i$) and Corollary 3.3,

$$\begin{aligned} C_i &= \frac{(\hat{\boldsymbol{\beta}}-\hat{\boldsymbol{\beta}}_{(-i)})^T X^T X(\hat{\boldsymbol{\beta}}-\hat{\boldsymbol{\beta}}_{(-i)})}{\hat{\sigma}^2(p+1)} \\ &= \frac{\left[\frac{1}{1-h_{ii}}(X^TX)^{-1}\mathbf{x}_i e_i\right]^T X^T X\left[\frac{1}{1-h_{ii}}(X^TX)^{-1}\mathbf{x}_i e_i\right]}{\hat{\sigma}^2(p+1)} \\ &= \frac{\frac{e_i^2}{(1-h_{ii})^2}\mathbf{x}_i^T(X^TX)^{-1}\mathbf{x}_i}{\hat{\sigma}^2(p+1)} \\ &= \left(\frac{e_i}{\hat{\sigma}\sqrt{1-h_{ii}}}\right)^2 \times \frac{1}{p+1} \times \frac{h_{ii}}{1-h_{ii}} \quad \because \mathbf{x}_i^T(X^TX)^{-1}\mathbf{x}_i = h_{ii} \\ &= (\text{internally studentized residual})^2 \times \frac{1}{p+1} \times \text{leverage measure} \end{aligned}$$

□

Conjecture 3.3. *In practice, if $C_i > 1$, i-th observation is considered to be influential.*

Corollary 3.3 is form of F-statistic. More specifically, $F(p, n-p)$. 1 is actually come from $F_{0.5}(p, n-p)$. If the case is out of 50th percentile of F-distribution, then we will say that it is influential point.

In `R`, `cooks.distance()` gives C_i alone.

```
tibble(cook = cooks.distance(cem_fit)) %>%
  mutate(i = 1:n()) %>%
  ggplot(aes(x = i, y = cook)) +
  geom_ref_line(h = 1) +
  geom_point() +
  geom_linerange(aes(ymin = cook, ymax = 0), col = gg_hcl(1), linetype = "dotted")
```

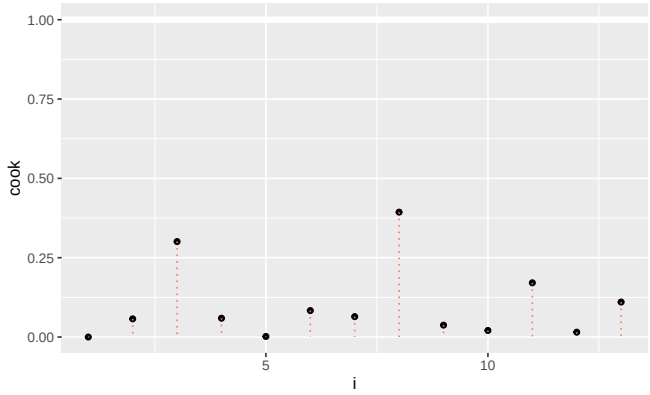


Figure 3.19: Cookś Distance for fitted cement dataset

In fact, we can get every influential statistic at once. `influence()` gives as `list` object.

```
influence(cem_fit)
#> $hat
#>     1     2     3     4     5     6     7     8     9    10    11    12
#> 0.550 0.333 0.577 0.295 0.358 0.124 0.367 0.409 0.294 0.700 0.426 0.263
#>    13
#> 0.304
#>
#> $coefficients
#>    (Intercept)        x1        x2        x3        x4
#> 1      -0.0797  0.000615  0.000865  0.000563  0.000923
#> 2      14.3977 -0.188059 -0.144479 -0.166288 -0.130195
#> 3     -76.1808  0.763715  0.777935  0.811840  0.772107
#> 4     -16.9320  0.135355  0.183354  0.170637  0.163203
#> 5      -1.4097  0.002222  0.018383  0.002002  0.017145
#> 6      11.3466 -0.078436 -0.114107 -0.113965 -0.119354
#> 7       3.8884 -0.003697 -0.053510 -0.042756 -0.031897
#> 8      25.1599 -0.328637 -0.223616 -0.416775 -0.248256
#> 9      18.0763 -0.202827 -0.182419 -0.176212 -0.185153
#> 10     -3.9568  0.092798  0.031933  0.063502  0.035052
#> 11    -22.6795  0.278931  0.213431  0.345600  0.217330
```

```
#> 12    -10.0419  0.106271  0.107888  0.101899  0.098533
#> 13     24.5346 -0.219201 -0.274452 -0.214449 -0.247089
#>
#> $sigma
#>    1    2    3    4    5    6    7    8    9   10   11   12   13
#> 2.61 2.52 2.43 2.50 2.61 2.08 2.52 2.10 2.54 2.61 2.42 2.58 2.40
#>
#> $wt.res
#>        1        2        3        4        5        6        7        8
#>  0.00476  1.51120 -1.67094 -1.72710  0.25076  3.92544 -1.44867 -3.17499
#>        9       10       11       12       13
#>  1.37835  0.28155  1.99098  0.97299 -2.29433
```

On the other hand, `influence.measures()` gives `matrix`. Its class is `infl` and we can extract `matrix` by `$infmat`.

```
influence.measures(cem_fit)
#> Influence measures of
#>   lm(formula = y ~ ., data = cem) :
#>
#>      dfb.1_    dfb.x1   dfb.x2    dfb.x3   dfb.x4   dffit cov.r   cook.d
#> 1  -0.00106  0.000773  0.00112  0.000698  0.00122  0.0030 4.335 2.06e-06
#> 2   0.19947 -0.245131 -0.19378 -0.213899 -0.17826  0.5193 2.017 5.72e-02
#> 3  -1.09529  1.033072  1.08281  1.083708  1.09704 -1.2356 2.195 3.01e-01
#> 4  -0.23674  0.178058  0.24819  0.221515  0.22551 -0.5333 1.741 5.93e-02
#> 5  -0.01884  0.002793  0.02378  0.002484  0.02264  0.0894 3.004 1.82e-03
#> 6   0.19047 -0.123878 -0.18544 -0.177619 -0.19800  0.7594 0.225 8.34e-02
#> 7   0.05380 -0.004812 -0.07168 -0.054930 -0.04362 -0.5497 2.151 6.43e-02
#> 8   0.41856 -0.514380 -0.36015 -0.643741 -0.40814 -1.6352 0.365 3.94e-01
#> 9   0.24840 -0.262227 -0.24268 -0.224818 -0.25144  0.4171 2.068 3.75e-02
#> 10 -0.05297  0.116875  0.04138  0.078926  0.04637  0.3016 6.330 2.07e-02
#> 11 -0.32727  0.378689  0.29816  0.463023  0.30992  0.9345 1.558 1.71e-01
#> 12 -0.13589  0.135302  0.14134  0.128027  0.13177  0.2625 2.309 1.53e-02
#> 13  0.35692 -0.300024 -0.38654 -0.289655 -0.35523 -0.7568 1.185 1.10e-01
#>      hat inf
#> 1  0.550   *
#> 2  0.333
#> 3  0.577   *
#> 4  0.295
#> 5  0.358   *
#> 6  0.124
#> 7  0.367
#> 8  0.409
#> 9  0.294
#> 10 0.700   *
#> 11 0.426
```

```
#> 12 0.263
#> 13 0.304
```

In `broom` package, `broom::augment()` includes influene measures. To use `tibble`, we can use this function.

```
broom::augment(cem_fit)
#> # A tibble: 13 x 12
#>       y    x1    x2    x3    x4 .fitted .se.fit   .resid  .hat .sigma
#>   <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>   <dbl>    <dbl> <dbl>  <dbl>
#> 1  78.5     7    26     6    60    78.5   1.81   0.00476 0.550   2.61
#> 2  74.3     1    29    15    52    72.8   1.41   1.51    0.333   2.52
#> 3 104.     11    56     8    20   106.    1.86  -1.67    0.577   2.43
#> 4  87.6    11    31     8    47    89.3   1.33  -1.73    0.295   2.50
#> 5  95.9     7    52     6    33    95.6   1.46   0.251   0.358   2.61
#> 6 109.     11    55     9    22   105.    0.862  3.93    0.124   2.08
#> # ... with 7 more rows, and 2 more variables: .cooksd <dbl>,
#> #   .std.resid <dbl>
```

3.5 Remedial Measures

In many times, given data set violates our assumptions. We have seen some of those scenarios. In that case, typical procedure will not be welcomed.

3.5.1 Transformations

As quick remedies for violations, transformations of variables can be conducted.

First, See Figure 3.8. When **variance becomes larger** as $\hat{Y}_i$ increases, transform Y_i into

$$\ln Y_i \quad \text{or} \quad \sqrt{Y_i}$$

i.e. the model becomes

$$\begin{cases} \ln Y_i = \beta_0 + \beta_1 x_{i1} + \cdots \beta_p x_{ip} + \epsilon_i \\ \sqrt{Y_i} = \beta_0 + \beta_1 x_{i1} + \cdots \beta_p x_{ip} + \epsilon_i \end{cases}$$

Since $\ln(.)$ and $\sqrt{.}$ is increasing with decreasing derivative, it might be able to suppress increasing variance which is depending on $\hat{Y}_i$.

```
hetero_scale <-
  hetero %>%
  mutate(
    logy = log(y - min(y) + .1),
```

```
    sqy = sqrt(y - min(y) + .1)
  )
#--------------
hetero_log <- lm(logy ~ x, data = hetero_scale)
hetero_sqrt <- lm(sqy ~ x, data = hetero_scale)
```

```
residplot(hetero_scale, hetero_log)
```

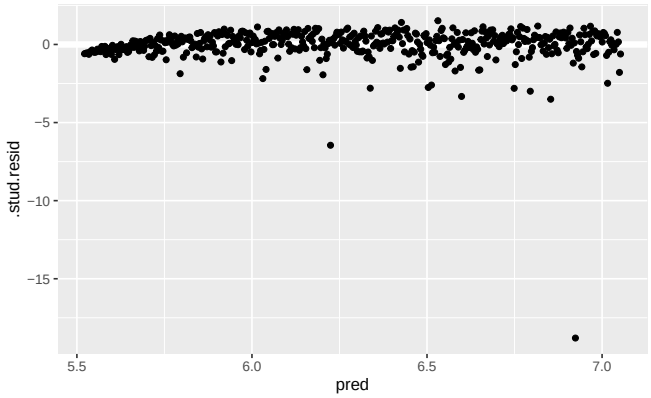


Figure 3.20: Log transformation against increasing variance

```
residplot(hetero_scale, hetero_sqrt)
```

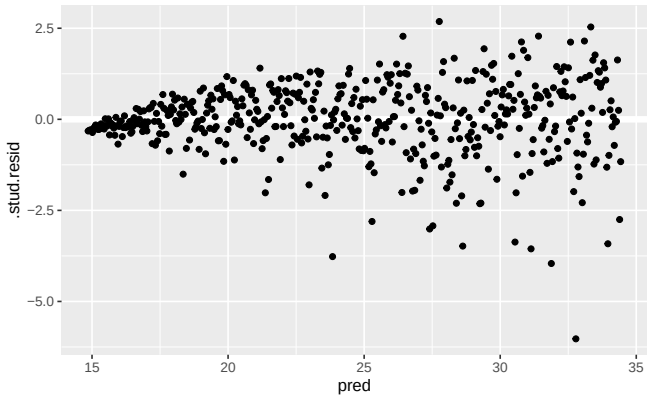


Figure 3.21: Square root transformation against increasing variance

See Figures 3.20 and 3.21. In this example, log-transformation was more affective. This might be because log function is smaller than square root function in large values.

Second, see Figure 3.9. Sometimes we observe a **certain pattern** in residual plot. Here, non-linear - cubic. Then transform x_i into corresponding non-linear function of x_i.

```
cubic_poly <- lm(y ~ poly(x, 3), data = cubic) # cubic polynomial
residplot(cubic, cubic_poly)
```

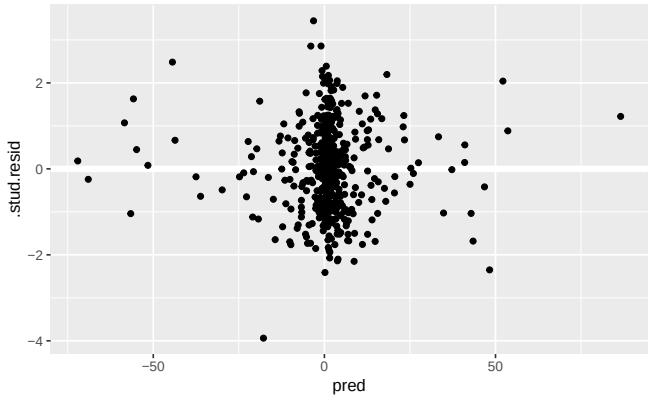


Figure 3.22: Cubic transformation against cubic pattern

Third, consider the case when Q-Q plot shows **non-normality**. Of course it depends on a situation, but

$$\ln Y_i$$

helpful sometimes.

3.5.2 Variance stabilizing transformation

Suppose that the variance is **not constant** and it is the **function of mean**.

$$Y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i, \quad \epsilon_i \sim N(0, f^2(\mu_i))$$

where $\mu_i = E(Y_i)$ and f is a smooth function for *standard deviation*. Before looking at how to deal with this heteroskedasticity, review a theorem from mathematical statistics.

Theorem 3.3 (Delta Method of moments)**.** *Let $h(Y_i)$ be any change of variable.*

$$E[h(Y_i)] \approx h(Y_i)$$

and

$$Var[h(Y_i)] \approx \Big(h'(\mu_i)\Big)^2 Var(Y_i)$$

Proof. Taylor formula implies that

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

Then by taking mean and variance,

$$Eh(Y_i) \approx h(Y_i)$$

$$Varh(Y_i) \approx \Big(h'(\mu_i)\Big)^2 Var(Y_i)$$

□

We try to find some *variance stabilizing transformation* $h(Y_i)$ such that

$$Var[h(Y_i)] = \sigma^2$$

From Theorem 3.3,

$$\sigma^2 \approx \Big(h'(\mu_i)\Big)^2 f^2(\mu_i) \tag{3.4}$$

It follows that

$$\begin{aligned} h'(\mu_i) &\approx \frac{\sigma}{f(\mu_i)} \\ \Rightarrow h'(\mu_i) &\propto \frac{1}{f(\mu_i)} \\ \Rightarrow h(\mu_i) &\propto \int \frac{1}{f(\mu_i)} d\mu_i \end{aligned}$$

Theorem 3.4 (Variance stabilizing transformation)**.** *h can be calculated by*

$$h(\mu_i) \propto \int \frac{1}{f(\mu_i)} d\mu_i$$

Theorem 3.4 shows how we can stabilize variance using the direct pattern of variance.

Example 3.1 (Poisson error term)**.**

$$Y_i \mid \mathbf{x}_i \overset{indep}{\sim} Poisson(\lambda(\mathbf{x}_i))$$

```
residplot(vst_pois, pois_fit)
```

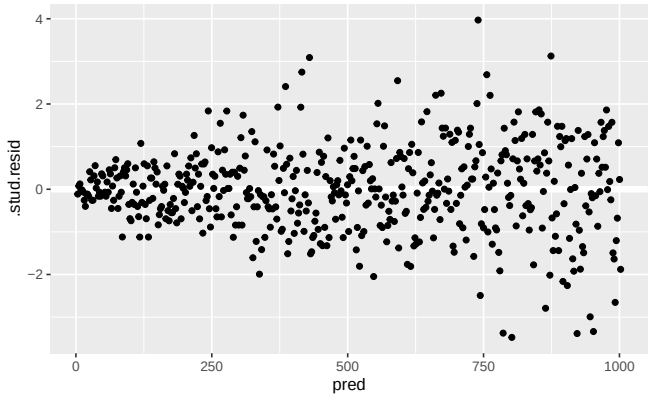


Figure 3.23: Residual plot after fitting Poisson error model

Solution. Poisson error term results in non-constant variance. It is natural because the variance is the same as mean

$$Var(Y_i \mid \mathbf{x}_i) = E(Y_i \mid \mathbf{x}_i) = \lambda(\mathbf{x}_i)$$

Let $\lambda_i \equiv \lambda(\mathbf{x}_i)$. Applying VST,

$$h(\lambda_i) \propto \int \frac{1}{\sqrt{\lambda_i}} d\lambda_i \propto 2\sqrt{\lambda_i}$$

Hence, we set VST by

$$h(Y_i) = \sqrt{Y_i}$$

```
vst_pois <-
  vst_pois %>%
  mutate(ysq = sqrt(y))
vstpois_fit <- lm(ysq ~ x, data = vst_pois)
#---------------
residplot(vst_pois, vstpois_fit)
```

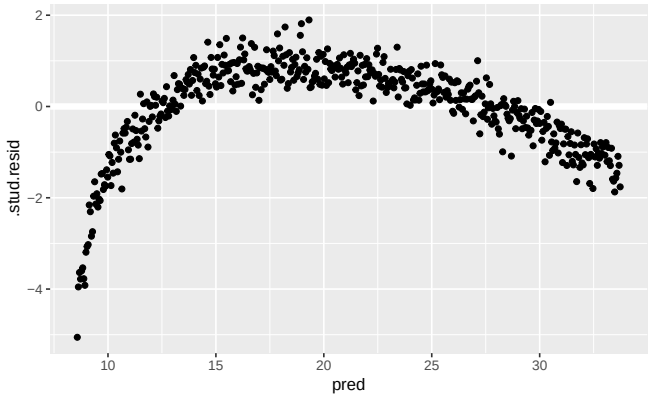


Figure 3.24: Poisson error term model residuals after VST

In fact, a *better way is to fit Poisson regression model, as a special case of generalized linera models*. This can explain Poisson random error with random component.

In general, we do not know f. In this case, we just

1. trial and error
2. previous results
3. If replications (multiple observations of Y_i at same $\mathbf{x}_i$ values) exist
    1. compute sd S_i and average $\overline{Y}_i$
    2. simple linear regression $S_i = \alpha + \lambda \overline{Y}_i$
    3. $f(\mu_i) = \hat{\alpha}\mu_i^{\lambda}$
4. If no replication, group

3.5.3 Box-Cox Transformation

Box-Cox transformation is usually considered when Normality assumption is violated.

$$Y' = \begin{cases} \frac{Y^{\lambda}-1}{\lambda} & \text{if } \lambda \neq 0 \\ \ln Y & \text{if } \lambda = 0 \end{cases}$$

How can we decide λ? λ can be estimated by ML method. Note that Y' become gaussian. Assume that

$$Y' \sim N(\mathbf{x}_i\boldsymbol{\beta}, \sigma^2)$$

Then

$$L(\lambda) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(- \frac{(Y_i'(\lambda) - \mathbf{x}_i^T\boldsymbol{\beta})^2}{\sigma^2} \right)$$

3.6 Generalized and Weighted Least Squares

We can meet diverse values of σ_i^2. As advanced approaches, we can fit method of **generalized least squares** (GLS). One of special cases of GLS, **weighted least squares** is also widely used.

3.6.1 Generalized least squares

When the errors do not have equal variance or they are not independent, we change our previous least squares procedure, so-called *ordinary least squres* (OLS) method. Our assumption for variance is $\sigma^2 I$, i.e. constant variance. Combined with normality, this also implies independence. Suppose that these two assumptions are violated. Let

$$Var(\boldsymbol{\epsilon}) = \sigma^2 V$$

where $V \neq I$ known. Assume that V is *positive definite*. Consider the following transformation

$$\underset{\mathbf{Y}'}{V^{-\frac{1}{2}}\mathbf{Y}} = \underset{X'}{V^{-\frac{1}{2}}X}\underset{\boldsymbol{\beta}}{\boldsymbol{\beta}} + \underset{\boldsymbol{\epsilon}'}{V^{-\frac{1}{2}}\boldsymbol{\epsilon}} \tag{3.5}$$

Then

$$\begin{aligned} Var(\boldsymbol{\epsilon}') &= Var(V^{-\frac{1}{2}}\boldsymbol{\epsilon}) \\ &= V^{-\frac{1}{2}} Var(\boldsymbol{\epsilon}) V^{-\frac{1}{2}} \quad \because V \text{ symmetric} \\ &= \sigma^2 I \end{aligned} \tag{3.6}$$

i.e. the errors become to have constant variance and independent. We now decide to find solution for our estimator. As in OLS,

$$\hat{\boldsymbol{\beta}}_G := \underset{\boldsymbol{\beta}\in\mathbb{R}^{p+1}}{\operatorname{argmin}} (\mathbf{Y}' - X'\boldsymbol{\beta})^T (\mathbf{Y}' - X'\boldsymbol{\beta})$$

From Equation (3.5),

$$(\mathbf{Y}' - X'\boldsymbol{\beta})^T (\mathbf{Y}' - X'\boldsymbol{\beta}) = (\mathbf{Y} - X\boldsymbol{\beta})^T V^{-1} (\mathbf{Y} - X\boldsymbol{\beta})$$

Corollary 3.4 (GLS criterion)**.** *GLS estimator finds*

$$\hat{\boldsymbol{\beta}}_G = \underset{\boldsymbol{\beta}\in\mathbb{R}^{p+1}}{\operatorname{argmin}} (\mathbf{Y} - X\boldsymbol{\beta})^T V^{-1} (\mathbf{Y} - X\boldsymbol{\beta})$$

From Corollary 3.4, GLS estimator is

$$\hat{\boldsymbol{\beta}}_G = (X^T V^{-1} X)^{-1} X^T V^{-1} \mathbf{Y} \tag{3.7}$$

Proposition 3.2 (Properties of $\hat{\boldsymbol{\beta}}_G$)**.** $\hat{\boldsymbol{\beta}}_G$ *satisfies following properties.*

i $E\hat{\boldsymbol{\beta}}_G = \boldsymbol{\beta}$ *and* $Var\hat{\boldsymbol{\beta}}_G = \sigma^2 (X^T V^{-1} X)^{-1}$

ii $X\hat{\boldsymbol{\beta}}_G$ *is the projection of* $\mathbf{Y}$ *onto* $Col(X)$ *when we define a normed space* $(\mathbb{R}^n, \|\cdot\|_{V^{-1}})$ *with* $\|\mathbf{u}\|_{V^{-1}}^2 := \mathbf{u}^T V^{-1} \mathbf{u}$

Proof. (i)-1

$$\begin{aligned} E\hat{\boldsymbol{\beta}}_G &= E\Big[(X^T V^{-1} X)^{-1} X^T V^{-1} \mathbf{Y}\Big] \\ &= (X^T V^{-1} X)^{-1} X^T V^{-1} E\mathbf{Y} \\ &= (X^T V^{-1} X)^{-1} X^T V^{-1} X\boldsymbol{\beta} \\ &= \boldsymbol{\beta} \end{aligned}$$

(i)-2

Note that

$$Var\mathbf{Y} = \sigma^2 V$$

Then

$$\begin{aligned} Var\hat{\boldsymbol{\beta}}_G &= Var\Big[(X^T V^{-1} X)^{-1} X^T V^{-1} \mathbf{Y}\Big] \\ &= (X^T V^{-1} X)^{-1} X^T V^{-1} Var(\mathbf{Y}) V^{-1} X (X^T V^{-1} X)^{-1} \\ &= \sigma^2 (X^T V^{-1} X)^{-1} (X^T V^{-1} X)(X^T V^{-1} X)^{-1} \\ &= \sigma^2 (X^T V^{-1} X)^{-1} \end{aligned}$$

□

Remember the definition of normed space.

Definition 3.9 (Normed space)**.** A *normed space* is a vector space X with a ***norm*** $\|\cdot\| : X \to \mathbb{R}$ satisfying for all $x, y \in X$ and $\alpha \in K$

(N1) $\|x\| \geq 0$

(N2) $\|x\| = 0 \Leftrightarrow x = 0$

(N3) $\|\alpha x\| = |\alpha|\|x\|$

(N4) $\|x + y\| \leq \|x\| + \|y\|$

Remark. Consider Definition 3.9.

1. $(N3)$ and $(N4)$ imply $(N1)$. Hence, it is enough for any X to satisfy $(N2)$, $(N3)$ and $(N4)$ to show that it is normed space.
2. $x = 0$ implies $\|x\| = 0$, i.e. proving *only if* part is enough for $(N2)$.

Proof. 1

Let $x \in X$. Then

$$\begin{aligned} 0 = \|0\| &= \|x + (-x)\| \\ &\leq \|x\| + \|-x\| \quad \because (N4) \\ &= 2\|x\| \quad \because (N3) \end{aligned}$$

Thus,

$$\|x\| \geq 0$$

2

$$\|0\| = \|0x\| \overset{(N3)}{=} |0|\|x\| = 0$$

□

Denote that in our regression setting, $K = \mathbb{R}$. Now continue the proof of Proposition 3.2.

Proof. (ii) normed space

Let $\mathbf{u} \in \mathbb{R}^n$ and let $\|\mathbf{u}\|_{V^{-1}}^2 := \mathbf{u}^T V^{-1} \mathbf{u}$.

Want 1: $\|\mathbf{u}\| = 0 \Rightarrow \mathbf{u} = \mathbf{0}$

Suppose that $\|\mathbf{u}\| = 0$. Then by definition,

$$\mathbf{u}^T V^{-1} \mathbf{u} = 0$$

Since V is positive defintie, V^{-1} is also positive definite. Thus,

$$\mathbf{u} = 0$$

and $(N1)$ holds.

Want 2: $(N3)$

Let $c \in \mathbb{R}$. Then

$$\begin{aligned} \|c\mathbf{u}\|_{V^{-1}}^2 &= (c\mathbf{u})^T V^{-1}(c\mathbf{u}) \\ &= c^2 \mathbf{u}^T V^{-1}\mathbf{u} \end{aligned}$$

Thus,

$$\|c\mathbf{u}\|_{V^{-1}} = |c|\|\mathbf{u}\|_{V^{-1}}$$

Want 3: $(N4)$

Let $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$. Then we have

$$\begin{aligned} \|\mathbf{u}+\mathbf{v}\|_{V^{-1}}^2 &= (\mathbf{u}+\mathbf{v})^T V^{-1}(\mathbf{u}+\mathbf{v}) \\ &= \mathbf{u}^T V^{-1}\mathbf{u} + \mathbf{v}^T V^{-1}\mathbf{v} + \mathbf{u}^T V^{-1}\mathbf{v} + \mathbf{v}^T V^{-1}\mathbf{u} \\ &= \mathbf{u}^T V^{-1}\mathbf{u} + \mathbf{v}^T V^{-1}\mathbf{v} + (V^{-\frac{1}{2}}\mathbf{u})^T V^{-\frac{1}{2}}\mathbf{v} + (V^{-\frac{1}{2}}\mathbf{v})^T V^{-\frac{1}{2}}\mathbf{u} \\ &\leq \|\mathbf{u}\|_{V^{-1}} + \|\mathbf{v}\|_{V^{-1}} + 2\|V^{-\frac{1}{2}}\mathbf{u}\| \cdot \|V^{-\frac{1}{2}}\mathbf{v}\| \quad \leftarrow \text{Cauchy-Schwarz inequality in } (\mathbb{R}^n, <\cdot,\cdot>) \\ &= \|\mathbf{u}\|_{V^{-1}} + \|\mathbf{v}\|_{V^{-1}} + 2\left(\mathbf{u}^T V^{-1}\mathbf{u} \cdot \mathbf{v}^T V^{-1}\mathbf{v}\right) \\ &= \|\mathbf{u}\|_{V^{-1}} + \|\mathbf{v}\|_{V^{-1}} + 2\|\mathbf{u}\|_{V^{-1}}\|\mathbf{v}\|_{V^{-1}} \\ &= (\|\mathbf{u}\|_{V^{-1}} + \|\mathbf{v}\|_{V^{-1}})^2 \end{aligned}$$

This completes the proof. □

Refer to the GLS model (3.5). In fact, this is a transformed model, so $\boldsymbol{\beta}$ here is the different coefficient with the original model. By Equation (3.7) and Proposition 3.2, $\hat{\boldsymbol{\beta}}_G$ is a linear unbiased estimator for $\boldsymbol{\beta}$. Then is this the BLUE, i.e. is

$$Var\hat{\boldsymbol{\beta}}_G = \sigma^2(X^T V^{-1} X)^{-1}$$

the smallest variance among of the linear unbiased estimators?

Theorem 3.5. *GLS estimator $\hat{\boldsymbol{\beta}}_G$ is the best linear unbiased estimator of $\boldsymbol{\beta}$, i.e.* ***BLUE****.*

Proof. One proceeds in a similar way to the proof of Theorem 2.8.

Let $\tilde{\boldsymbol{\beta}} \in \Omega \equiv \{\tilde{\boldsymbol{\beta}} : \tilde{\boldsymbol{\beta}} = C\mathbf{Y}, E\tilde{\boldsymbol{\beta}} = \boldsymbol{\beta}\}$.

Claim: $Var\tilde{\boldsymbol{\beta}} - Var\hat{\boldsymbol{\beta}}_G$ is non-negative definite.

Set $D := C - (X^TV^{-1}X)^{-1}X^TV^{-\frac{1}{2}}$. From unbiasedness,

$$\begin{aligned} E\tilde{\boldsymbol{\beta}} &= CE\mathbf{Y} \\ &= CV^{-\frac{1}{2}}X\boldsymbol{\beta} \\ &= \Big((X^TV^{-1}X)^{-1}X^TV^{-\frac{1}{2}} + D\Big)V^{-\frac{1}{2}}X\boldsymbol{\beta} \\ &= \boldsymbol{\beta} + DV^{-\frac{1}{2}}X\boldsymbol{\beta} \\ &= \boldsymbol{\beta} \end{aligned}$$

It implies that

$$DV^{-\frac{1}{2}}X = 0$$

$$\begin{aligned} Var\tilde{\boldsymbol{\beta}} &= Var(C\mathbf{Y}) \\ &= \sigma^2CC^T \\ &= \sigma^2\Big((X^TV^{-1}X)^{-1}X^TV^{-\frac{1}{2}} + D\Big)\Big((X^TV^{-1}X)^{-1}X^TV^{-\frac{1}{2}} + D\Big)^T \\ &= \sigma^2\Big((X^TV^{-1}X)^{-1} + DV^{-\frac{1}{2}}X(X^TV^{-1}X)^{-1} + (X^TV^{-1}X)^{-1}X^TV^{-\frac{1}{2}}D^T + DD^T\Big) \\ &= \sigma^2\Big((X^TV^{-1}X)^{-1} + DD^T\Big) \\ &= Var\hat{\boldsymbol{\beta}}_G + \sigma^2DD^T \end{aligned}$$

Note that σ^2DD^T is a non-negative definite matrix. Hence,

$$Var\tilde{\boldsymbol{\beta}} - Var\hat{\boldsymbol{\beta}}_G = \sigma^2DD^T$$

is non-negative definite. This completes the proof. □

In practice, however, we do not know what V is.

3.6.2 Weighted least squares

As mentioned, WLS is a special case of GLS. Set $W := V^{-1}$ in GLS by

$$W = diag\Big(\frac{1}{w_1}, \frac{1}{w_2}, \ldots, \frac{1}{w_n}\Big)$$

with $w_i > 0$. This method is appropriate to the *uncorrelated error term of unequal variances with*

$$Var(\epsilon_i) = \frac{\sigma^2}{w_i}$$

As in GLS, weighted least squares deals with the model of the form

$$W^{\frac{1}{2}}\mathbf{Y} = W^{\frac{1}{2}}X\boldsymbol{\beta} + W^{\frac{1}{2}}\boldsymbol{\epsilon} \tag{3.8}$$

From Corollary 3.4, it minimizes

$$(\mathbf{Y} - X\boldsymbol{\beta})^T W(\mathbf{Y} - X\boldsymbol{\beta})$$

This is equivalent to the next corollary.

Corollary 3.5 (WLS criterion)**.** *WLS minimizes the weighted sum of squred errors*

$$\sum_{i=1}^{n} w_i(Y_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_p x_{ip})^2$$

Corollary 3.5 means that each data point is *weighted by inversely proportional to the variance of the corresponding response.* The estimator would be computed as

$$\hat{\boldsymbol{\beta}}^{WLS} = (X^T W X)X^T W\mathbf{Y} \tag{3.9}$$

Since we do not know each σ_i in typical analysis, we should estimate each w_i.

This weights of WLS is specified in `weights` argument of `lm()`.

Chapter 4

Multicollinearity

4.1 Multicollinearity

See a dataset about equal opportunity in public education.

```
(eeo <- haven::read_sav("data/p228.sav"))
#> # A tibble: 70 x 4
#>     ACHV    FAM    PEER SCHOOL
#>    <dbl>  <dbl>   <dbl>  <dbl>
#> 1 -0.431  0.608  0.0351  0.166
#> 2  0.800  0.794  0.479   0.534
#> 3 -0.925 -0.826 -0.620  -0.786
#> 4 -2.19  -1.25  -1.22   -1.04
#> 5 -2.85   0.174 -0.185   0.142
#> 6 -0.662  0.202  0.128   0.273
#> # ... with 64 more rows
```

In 1965, 70 schools were selected at random. 4 measurements were taken for each school.

- `ACHV`: student achievements, *response variable*
- `FAM`: faculty credentials
- `PEER`: influence of their peer group in the school
- `SCHOOL`: school facilities

(Chatterjee and Hadi, 2015).

```
eeo_fit <- lm(ACHV ~ ., data = eeo)
```

Definition 4.1 (Multicollinearity)**.** A set of predictors $\{X_1, X_2, \ldots, X_p\}$ is said to have ***multicollinearity*** iff there exist linear or near-linear dependencies among predictors.

4.1.1 Effects of multicollinearity

Consider design matrix

$$X = \begin{bmatrix} \mathbf{1} & \mathbf{x}_1 & \cdots & \mathbf{x}_p \end{bmatrix} \in \mathbb{R}^{n \times (p+1)}$$

When there exists any linear dependency among the predictors, column vectors of X are *lineary dependent*. Or equivalently, centered columns $\{\mathbf{x}_1 - \overline{x}_1\mathbf{1}, \ldots, \mathbf{x}_p - \overline{x}_p\mathbf{1}\}$ are lineary dependent. Then

$$rank(X) < p + 1 \leq n$$

i.e. rank deficient. From either Theorem 1.8 or 2.1, X^TX can be *singular*. In this case, we cannot get the LSE solution. It is not only the problem of computing but also the variance. Consider *total variance*, sum of every coefficient variance. From Proposition 2.1,

$$\begin{aligned} \sum_{j=0}^{p} Var(\hat{\beta}_j) &= trace\Big(Var(\hat{\boldsymbol{\beta}})\Big) \\ &= trace\Big(\sigma^2(X^TX)^{-1}\Big) \\ &= \sigma^2 trace\Big((X^TX)^{-1}\Big) \\ &= \sigma^2 \sum_{j=0}^{p} \frac{1}{\kappa_j} \quad \kappa_j := \text{eigenvalues of } X^TX = \text{singular values}^2 \end{aligned}$$

If X is not of full rank, X^TX is not of full rank. In other words,

$$\exists j : \kappa_j = 0$$

By construction, even one $\kappa_j = 0$ results in

$$\sum_{j=0}^{p} Var(\hat{\beta}_j) = \infty$$

It is found that *linear dependency leads to increasing variance of the estimates*. This variance problem occurs when nearly-linear dependeny situation, of course. So we should detect and remedy this.

4.2 Multicollinearity diagnostics

4.2.1 Correlation matrix

Multicollinearity leads to unstable regression coefficients. From Equation (2.18), we know that $\hat{\boldsymbol{\beta}}$ is related to sample correlation between response and predictors. Pairwise correlation gives information about linear relationship between X_j.

```
cor(eeo)
#>          ACHV   FAM  PEER SCHOOL
#> ACHV   1.000 0.419 0.440  0.418
#> FAM    0.419 1.000 0.960  0.986
#> PEER   0.440 0.960 1.000  0.982
#> SCHOOL 0.418 0.986 0.982  1.000
```

4.2.2 Variance inflation factor

Lemma 4.1. *Consider regression model j-th predictor X_j on the remaining X_k, $k \neq j$, i.e.*

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

Let $s_{jj} = \sum_{i=1}^{n} (x_{ij} - \overline{x}_j)^2$ be corrected sum of squares and let R_j^2 be the coefficient of determination. Then

$$Var(\hat{\beta}_j) = \frac{1}{1 - R_j^2} \frac{\sigma^2}{s_{jj}}, \quad 1 \leq j \leq p$$

Proof. Assume $j = 1$ without loss of generality. Write

$$X = \begin{bmatrix} \mathbf{x}_1 & X_B \end{bmatrix}$$

Then we have the original regression model for $\mathbf{Y}$ as

$$X\boldsymbol{\beta} = \mathbf{x}_1 \boldsymbol{\beta}_A + X_B \boldsymbol{\beta}_B \tag{4.1}$$

Orthogonalize $\mathbf{x}_1$ by projecting onto X_B as before.

$$\mathbf{x}_{1,\perp} := \mathbf{x}_1 - \Pi_B \mathbf{x}_1$$

where $\Pi_B = X_B (X_B^T X_B)^{-1} X_B^T$. It follows that

$$\begin{aligned} X\hat{\boldsymbol{\beta}} &= \mathbf{x}_1\hat{\boldsymbol{\beta}}_1 + X_B\hat{\boldsymbol{\beta}}_B \\ &= \mathbf{x}_{1,\perp}\hat{\boldsymbol{\beta}}_1 + X_B\Big(\hat{\boldsymbol{\beta}}_B + (X_B^TX_B)^{-1}X_B^T\mathbf{x}_1\hat{\boldsymbol{\beta}}_1\Big) \\ &= \Pi(\mathbf{Y} \mid R(\mathbf{x}_{1,\perp})) + \Pi(\mathbf{Y} \mid R(X_B)) \end{aligned}$$

Thus,

$$\begin{cases} \hat{\boldsymbol{\beta}}_{1,\perp} = (\mathbf{x}_{1,\perp}^T\mathbf{x}_{1,\perp})^{-1}\mathbf{x}_{1,\perp}^T\mathbf{Y} \\ \hat{\boldsymbol{\beta}}_B = (X_B^TX_B)^{-1}X_B^T(\mathbf{Y} - \mathbf{x}_1\hat{\boldsymbol{\beta}}_1) \end{cases}$$

Note that $\mathbf{x}_{1,\perp}$ is the *residual vector in our previous resgression model* in the statement. By construction,

$$\mathbf{x}_{1,\perp}^T\mathbf{x}_{1,\perp} = SSE$$

computed from the model X_1 on the other predictors. Since $SST = s_{11}$,

$$R_1^2 = \frac{SST - SSE}{SST} = \frac{s_{11} - \mathbf{x}_{1,\perp}^T\mathbf{x}_{1,\perp}}{s_{11}}$$

Therefore,

$$\mathbf{x}_{1,\perp}^T\mathbf{x}_{1,\perp} = s_{11}(1 - R_1^2) \tag{4.2}$$

From Equation (4.2),

$$Var(\hat{\beta}_1) = \sigma^2(\mathbf{x}_{1,\perp}^T\mathbf{x}_{1,\perp})^{-1} = \frac{\sigma^2}{s_{11}(1 - R_1^2)}$$

One proceeds in a similary way for the other j.

$$\mathbf{x}_{j,\perp}^T\mathbf{x}_{j,\perp} = s_{jj}(1 - R_j^2) \tag{4.3}$$

□

What does Lemma 4.1 mean? Recall that in simple linear regression setting,

$$Var(\hat{\beta}_j) = \frac{\sigma^2}{s_{jj}}$$

When multicollinearity occurs, one term is multiplied so that

$$Var(\hat{\beta}_j) = \frac{1}{1 - R_j^2} \frac{\sigma^2}{s_{jj}} \tag{4.4}$$

Naturally, we can think of $\frac{1}{1-R_j^2}$ detecting multicollinearity.

Definition 4.2 (Variance inflation factor)**.** Let R_j^2 be the coefficient of determination that results when X_j is regressed against remaining X_k, $k \neq j$. Then the variance inflation factor for X_j is defined by

$$VIF_j := \frac{1}{1 - R_j^2},\ j = 1, \ldots, p$$

From Equation (4.4),

$$Var(\hat{\beta}_j) \propto (VIF_j)\sigma^2 \tag{4.5}$$

Variance of j-th regression coefficient is proportion to VIF_j. The term *variance inflation factor* originated from this fact.

Corollary 4.1. *VIF_j is simple the inflation rate of $Var(\hat{\beta}_j)$ in comparison with the case where X_j is not correlated with other predictors, i.e.*

$$s_{jk} = \sum_{i=1}^{n} (x_{ij} - \overline{x}_j)(x_{ik} - \overline{x}_k) = 0 \quad \forall k \neq j$$

Proof. By definition,

$$\begin{aligned}
&VIF_j^2 = 1 \\
&\Leftrightarrow R_j^2 = 0 \\
&\Leftrightarrow s_{jj} - \mathbf{x}_{j,\perp}^T \mathbf{x}_{j,\perp} = \sum_{i=1}^{n} (x_{ij} - \overline{x}_j)^2 - \sum_{i'=1}^{n} (x_{i'j} - \overline{x}_j)(x_{i'k} - \overline{x}_k) = 0 \\
&\Leftrightarrow s_{jk} = \sum_{i'=1}^{n} (x_{ij} - \overline{x}_j)(x_{ik} - \overline{x}_k) = 0 \quad \forall k \neq j
\end{aligned}$$

□

For example, suppose that

$$x_1 \approx c_0 + c_2 x_2 + \cdots + c_p x_p$$

Then

$$R_1^2 \approx 1$$

and so

$$VIF_1 \to \infty$$

Remark. Large VIF_j for one or multiple js indicate multicollinearity.

Rawlings et al. (2006) and Chatterjee and Hadi (2015) suggest some thresholds with references.

Conjecture 4.1. *If*

$$VIF_j > 10$$

, then β_j would be poorly estimated

Note that the *precision of OLS* is measured by its variance. Using the proportionality (4.5), let D^2 be the *expected squred distance of OLS* estimators (Chatterjee and Hadi, 2015).

$$D^2 = \sigma^2 \sum_{j=1}^{p} VIF_j$$

The smaller, the more accurate OLS is.

Remark. If predictor variables are orthogonal, then

$$\forall j : VIF_j = 1$$

and so

$$D^2 = p\sigma^2$$

Consider the ratio of D^2 to orthogonal D^2.

$$\frac{\sigma^2 \sum VIF_j}{p\sigma^2} = \frac{1}{p} \sum VIF_j \equiv \overline{VIF}$$

Definition 4.3. Write average of every VIF_j by

$$\overline{VIF} := \frac{1}{p} \sum_{j=1}^{p} VIF_j$$

$\overline{VIF}$ is not just average. The remark implies that $\overline{VIF}$ estimates the ratio of the true multicollinearity to a model when predictors are uncorrelated. Hence, this can also be used as an criterion of multicollinearity.

Conjecture 4.2. *If*

$$\overline{VIF} >> 1$$

, then serious multicollinearity might occur.

`car` library has a function `vif()`.

```
car::vif(eeo_fit)
#>    FAM   PEER SCHOOL
#>   37.6   30.2   83.2
```

4.2.3 Condition number

Theorem 4.1. *Let $\Sigma \in \mathbb{R}^{p\times p}$. Then the eigenvalues of Σ are all real.*

Proof. Let

$$\Sigma\mathbf{x} = \lambda\mathbf{x}$$

Then

$$\mathbf{x}^T\Sigma\mathbf{x} = \lambda\mathbf{x}^T\mathbf{x} = \lambda\|\mathbf{x}\|^2$$

Write $\alpha := \mathbf{x}^T\Sigma\mathbf{x}$.

Claim: $\alpha \in \mathbb{R}$

Since $\mathbf{x} \in \mathbb{R}^p$ and $\Sigma \in \mathbb{R}^{p\times p}$, i.e. real matrix, it is obvious that α is real.

Since $\|\mathbf{x}\|^2 \neq 0$,

$$\lambda = \frac{\alpha}{\|\mathbf{x}\|^2}$$

is also a real number. □

Theorem 4.2. *Let $X \in \mathbb{R}^{n\times p}$. Then the eigenvalues of X^TX are all non-negative real number.*

Proof. Since $X \in \mathbb{R}^{n\times p}$, $X^TX \in \mathbb{R}^{n\times n}$ and is symmetric. Let

$$(X^TX)\boldsymbol{\beta} = \lambda\boldsymbol{\beta}, \quad \boldsymbol{\beta} \neq \mathbf{0}$$

Since X^TX is symmetric, every $\lambda \in \mathbb{R}$.

$$\begin{aligned}\|X\boldsymbol{\beta}\|^2 &= (X\boldsymbol{\beta})^T(X\boldsymbol{\beta}) \\ &= \boldsymbol{\beta}^T X^T X\boldsymbol{\beta} \\ &= \lambda\boldsymbol{\beta}^T\boldsymbol{\beta} \qquad X^T X\boldsymbol{\beta} = \lambda\boldsymbol{\beta} \\ &= \lambda\|\boldsymbol{\beta}\|^2\end{aligned}$$

Hence,

$$\lambda = \frac{\|X\boldsymbol{\beta}\|^2}{\|\boldsymbol{\beta}\|^2} \geq 0 \tag{4.6}$$

□

From Theorem 4.2, let

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{p+1} \geq 0$$

be eigenvalues of $X^T X$. Then we have

$$\lambda_{p+1} > 0$$

guarantees existence of $(X^T X)^{-1}$. Reversely,

$$\lambda_{p+1} \approx 0$$

results in nearly-non-invertibility.

Definition 4.4 (Condition number)**.** Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{p+1} \geq 0$ be eigenvalues of $X^T X$. Then ***Condition number*** of $X^T X$ is defined by

$$\kappa := \frac{\lambda_1}{\lambda_{p+1}} = \frac{(\text{maximal eigenvalue of } X^T X)}{(\text{minimal eigenvalue of } X^T X)}$$

Some textbooks such as Chatterjee and Hadi (2015) define this measure additional square root, i.e. singular values of X. Since how large is important, it does not matter much. κ measures *how small minimal eigenvalue compared to maximal eigenvalue*. This means spread of the eigenvalue spectrum of $X^T X$.

Conjecture 4.3. *The larger κ, the more serious multicollinearity is. ($\lambda_{p+1} \approx 0 \Rightarrow \kappa \to \infty$)*

1. *Weak dependence* $\kappa \approx 100$
2. *Moderate to strong* $100 \leq \kappa \leq 1000$

3. *Severe* $\kappa \geq 1000$

There is a base function called `kappa()` calculating the condition number of a `matrix`. Since this function has a `S3` method for `lm`, we can just provide `lm` object to get κ of our model.

```
kappa(eeo_fit)
#> [1] 11.6
```

4.3 Principal Component Analysis

Multicollinearity violates the assumption of OLS model. This makes the estimator unstable - large variance. Sometimes we cannot even get the solution. In a linear regression frame, there exist some alternative to OLS dealing with large variance. Here we present two methods.

1. **Principal component regression**: By construction, principal components are uncorrelated.
2. Shrinkage methods: **Ridge regression** enables least squres get the solution and shrinks its variance.

Before looking at principal component analysis (PCA), we see some preliminary matrix algebra theorems: spectral decomposition and singular value decomposition.

4.3.1 Spectral decomposition

Theorem 4.3 (Spectral Decomposition)**.** *If* $A \in \mathbb{R}^{p\times p}$ *is a real symmetric matrix, then* A *is diagonalizable as*

$$A = P\Lambda P^T$$

where

$$P = \begin{bmatrix}\mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_p\end{bmatrix} = \begin{bmatrix} v_{11} & \cdots & v_{1p} \\ \vdots & \ldots & \vdots \\ v_{p1} & \cdots & v_{pp}\end{bmatrix} \text{ orthogonal}$$

and

$$\Lambda = diag(\lambda_1, \ldots, \lambda_p)$$

with eigenvalue and corresponding orthonormal eigenvector of A $(\lambda_j, \mathbf{v}_j)$, $j = 1, \ldots, p$.

The theorem satisfies when a matrix is *symmetric*. For example, covariance matrix or correlation matrix.

```
(spec_exm <- cov(eeo) %>% eigen())
#> eigen() decomposition
#> $values
#> [1] 6.09834 2.09533 0.03927 0.00808
#>
#> $vectors
#>       [,1]   [,2]     [,3]     [,4]
#> [1,] 0.876  0.483  0.00834  0.00648
#> [2,] 0.296 -0.544  0.67029 -0.40931
#> [3,] 0.258 -0.450 -0.74023 -0.42715
#> [4,] 0.280 -0.518 -0.05197  0.80621
```

`$values` are eigenvalues. Λ is a diagonal matrix with these elements.

```
diag(spec_exm$values)
#>      [,1] [,2]   [,3]    [,4]
#> [1,]  6.1  0.0 0.0000 0.00000
#> [2,]  0.0  2.1 0.0000 0.00000
#> [3,]  0.0  0.0 0.0393 0.00000
#> [4,]  0.0  0.0 0.0000 0.00808
```

`$vectors` is P whose column vectors are corresponding orthonormal eigenvectors to `$values`. We can check the theory in the eye.

```
near(spec_exm$vectors %*% diag(spec_exm$values) %*% t(spec_exm$vectors), cov(eeo))
#>        ACHV  FAM PEER SCHOOL
#> ACHV   TRUE TRUE TRUE   TRUE
#> FAM    TRUE TRUE TRUE   TRUE
#> PEER   TRUE TRUE TRUE   TRUE
#> SCHOOL TRUE TRUE TRUE   TRUE
```

The matrix decomposition can also be expressed as following corollary.

Corollary 4.2. *If $A \in \mathbb{R}^{p\times p}$ is a real symmetric matrix, then*

$$A = \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T + \cdots + \lambda_p \mathbf{v}_p \mathbf{v}_p^T$$

where each $(\lambda_j, \mathbf{v}_j)$ is defined as in Theorem 4.3.

Spectral decomposition $A = P\Lambda P^T$ gives some useful facts to A.

Proposition 4.1 (Properties of spectral decomposition)**.** *Let $A \in \mathbb{R}^{p\times p}$ be a real symmetric matrix. Then $A = P\Lambda P^T$.*

- $PP^T = P^T P = I$ *so that* $P^{-1} = P^T$
- $A^{-1} = P\Lambda^{-1}P^T$ *with* $\Lambda = diag\left(\frac{1}{\lambda_1}, \ldots, \frac{1}{\lambda_p}\right)$

- $A^k = P\Lambda^k P^T$ *with* $\Lambda = diag(\lambda_1^k, \ldots, \lambda_p^k)$

Proof. Since eigenvectors are orthonormalized, P is orthogonal.

$$A^{-1} = (P^T)^{-1}\Lambda^{-1}P^{-1} = P\Lambda^{-1}P^T$$

$$A^k = (P\Lambda\underbrace{P^T)(P}_{=I}\Lambda P^T)\cdots(P\Lambda\underbrace{P^T)(P}_{=I}\Lambda P^T) = P\Lambda^k P^T$$

Recall that Λ is diagonal. □

Collaborating with quadrating form, spectral decomposition can give a geometric meaning.

Theorem 4.4 (Principal Axes Theorem)**.** *If $A \in \mathbb{R}^{p\times p}$ is a real symmetric matrix, then*

$$\exists \textit{ change of variables } \mathbf{u} = P^T\mathbf{X} \quad \textit{such that} \quad \mathbf{X}^T A\mathbf{X} = \mathbf{u}^T\Lambda\mathbf{u}$$

where Λ is a diagonal matrix.

Proof. Spectral decomposition 4.3 implies that

$$P^T AP = \Lambda$$

Then

$$\mathbf{X}^T A\mathbf{X} = \mathbf{u}^T P^T AP\mathbf{u} = \mathbf{u}^T\Lambda\mathbf{u}$$

□

Corollary 4.3. *If $A \in \mathbb{R}^{p\times p}$ is a real symmetric matrix, then there exists a change of variables $\mathbf{u} = P^T\mathbf{X}$ s.t.*

$$\mathbf{X}^T A\mathbf{X} = \lambda_1(\mathbf{v}_1^T\mathbf{X})^2 + \cdots + \lambda_p(\mathbf{v}_p^T\mathbf{X})^2$$

Above Theorem 4.4 and Corollary 4.3 is linearly transforming its coordinates of conic section. The directions of coordinate are $\mathbf{v}_j$, orthonormal eigenvectors. Axes are determined by corresponding egenvalues.

4.3.2 Singular value decomposition

Let $X \in \mathbb{R}^{n\times(p+1)}$ be any real matrix. Note that by Theorem 4.2, every eigenvalue of X^TX is non-negative real number. Then we can compute its square root.

Definition 4.5 (Singular values of a real matrix)**.** Let $X \in \mathbb{R}^{n\times(p+1)}$ be any real matrix and let λ_j be eigenvalues of X^TX, $j = 1, \ldots, p+1$. Then signular values of X is

$$\sigma_j := \sqrt{\lambda_j}, \quad j = 1, \ldots, p+1$$

Singular value decomposition decomposes any matrix X into

$$X = UDV^T$$

D is a diagonal matrix which consists of singular values. There are several types of SVD according to shape of D. One is *full SVD* and the other is *reduced SVD*. In general, the latter is frequently used in regression literature. First, in full SVD, D has same dimension as X (Leon, 2014).

Theorem 4.5 (Full SVD)**.** *If $X \in \mathbb{R}^{n\times(p+1)}$ be any real matrix, then X has a full SVD*

$$X = UDV^T$$

with

$$D = \left[\begin{array}{ccc|ccc} \sigma_1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \sigma_{p+1} & 0 & \cdots & 0 \\ \hline 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{array}\right] \in \mathbb{R}^{n\times(p+1)}$$

where

$$\sigma_1 \geq \sigma \geq \cdots \geq \sigma_{p+1} \geq 0$$

are singular values of X, $U \in \mathbb{R}^{n\times n}$ orthogonal, and $V \in \mathbb{R}^{(p+1)\times(p+1)}$ also orthogonal.

Other U and V is not unique.

Theorem 4.6. *Let $X \in \mathbb{R}^{n\times(p+1)}$ be any real matrix and let $\sigma_1 \geq \sigma \geq \cdots \geq \sigma_{p+1} \geq 0$ be its singular values. Suppose that*

$$\sigma_1 \geq \sigma \geq \cdots \geq \sigma_r > 0$$

with $r \leq p+1$. If

$$U = \left[\begin{array}{ccc|ccc} \mathbf{u}_1 & \cdots & \mathbf{u}_r & \mathbf{u}_{r+1} & \cdots & \mathbf{u}_n \end{array}\right]$$

where

- *$\{\mathbf{u}_1, \ldots, \mathbf{u}_r\}$ is an orthonormal basis for $R(X)$*
- *$\{\mathbf{u}_{r+1}, \ldots, \mathbf{u}_n\}$ is an orthonormal basis for $N(X^T) = R(X)^\perp$*
- *Thus, $U = [\cdots \mid \cdots]$ is orthogonal*

$$V = \left[\begin{array}{ccc|ccc} \mathbf{v}_1 & \cdots & \mathbf{v}_r & \mathbf{v}_{r+1} & \cdots & \mathbf{v}_{p+1} \end{array}\right]$$

where

- *$\{\mathbf{v}_1, \ldots, \mathbf{v}_r\}$ is an orthonormal eigenvectors of X^TX, which in fact is orthonormal basis for $Row(X)$*
- *$\{\mathbf{v}_{r+1}, \ldots, \mathbf{v}_{p+1}\}$ is an orthonormal eigenvectors of X^TX belonging to $\lambda = 0$, which is orthonormal basis for $N(X) = Row(X)^\perp$*
- *Thus, $V = [\cdots \mid \cdots]$ is orthogonal*

then

$$X = UDV^T$$

From the equation, it is easy to get

$$XV = UD \tag{4.7}$$

From this equation, we call each $\mathbf{v}_j$ and $\mathbf{u}_j$ *right singular vector* and *left singular vector*. Next, we see reduced SVD. This uses left upper block of D, i.e. it has dimension of $(p+1)\times(p+1)$.

Theorem 4.7 (Reduced SVD)**.** *If $X \in \mathbb{R}^{n\times(p+1)}$ be any real matrix, then X has a reduced SVD*

$$X = UDV^T$$

with

$$D = diag(\sigma_1, \sigma_2, \ldots, \sigma_{p+1}) \in \mathbb{R}^{(p+1)\times(p+1)}$$

where

$$\sigma_1 \geq \sigma \geq \cdots \geq \sigma_{p+1} \geq 0$$

are singular values of X, $U \in \mathbb{R}^{n\times(p+1)}$ *orthogonal, and* $V \in \mathbb{R}^{(p+1)\times(p+1)}$ *also orthogonal.*

Since D is diagonal matrix, U has different dimension with previous SVD.

Theorem 4.8. *Let* $X \in \mathbb{R}^{n\times(p+1)}$ *be any real matrix and let* $\sigma_1 \geq \sigma \geq \cdots \geq \sigma_{p+1} \geq 0$ *be its singular values. Suppose that*

$$\sigma_1 \geq \sigma \geq \cdots \geq \sigma_r > 0$$

with $r \leq p+1$. *If*

$$U = \left[\begin{array}{ccc|ccc} \mathbf{u}_1 & \cdots & \mathbf{u}_r & \mathbf{u}_{r+1} & \cdots & \mathbf{u}_n \end{array}\right]$$

where

- $\{\mathbf{u}_1, \ldots, \mathbf{u}_r\}$ *is an orthonormal basis for* $R(X)$
- $\{\mathbf{u}_{r+1}, \ldots, \mathbf{u}_{p+1}\}$ *is an orthonormal basis for* $N(X^T) = R(X)^\perp$
- *Thus,* $U = [\cdots \mid \cdots]$ *is orthogonal*

$$V = \left[\begin{array}{ccc|ccc} \mathbf{v}_1 & \cdots & \mathbf{v}_r & \mathbf{v}_{r+1} & \cdots & \mathbf{v}_{p+1} \end{array}\right]$$

where

- $\{\mathbf{v}_1, \ldots, \mathbf{v}_r\}$ *is an orthonormal eigenvectors of* X^TX, *which in fact is orthonormal basis for* $Row(X)$
- $\{\mathbf{v}_{r+1}, \ldots, \mathbf{v}_{p+1}\}$ *is an orthonormal eigenvectors of* X^TX *belonging to* $\lambda = 0$, *which is orthonormal basis for* $N(X) = Row(X)^\perp$
- *Thus,* $V = [\cdots \mid \cdots]$ *is orthogonal*

then

$$X = UDV^T$$

In `R`, `svd()` produces this form of decomposition by default.

```
svd(eeo)
#> $d
#> [1] 20.518 12.033  1.646  0.756
#>
#> $u
#>            [,1]     [,2]     [,3]     [,4]
#>  [1,] -0.00690  0.05326 -0.22433  0.17599
#>  [2,]  0.05894  0.04462 -0.09485  0.12443
#>  [3,] -0.06997 -0.05721  0.03781  0.04957
#>  [4,] -0.14117 -0.05892 -0.05872 -0.23628
#>  [5,] -0.11939  0.12146 -0.13529 -0.13771
#>  [6,] -0.01998  0.05228 -0.01315 -0.10405
#>  [7,]  0.11553 -0.09623 -0.15080  0.00498
#>  [8,]  0.11905 -0.03743 -0.13980 -0.12862
#>  [9,] -0.06268 -0.03666  0.01520  0.07240
#> [10,]  0.06035  0.08449 -0.09853 -0.11151
#> [11,]  0.09800  0.09452 -0.01231 -0.03544
#> [12,]  0.10288 -0.00447  0.16010 -0.01966
#> [13,]  0.02701  0.10078  0.01617  0.10398
#> [14,] -0.14969  0.03293  0.01437 -0.14010
#> [15,]  0.15737 -0.18913  0.06356 -0.08148
#> [16,]  0.23906 -0.00966 -0.03734 -0.09601
#> [17,]  0.12660  0.11946 -0.05121  0.10874
#> [18,]  0.16035 -0.18531  0.06480 -0.01929
#> [19,]  0.09712  0.11181  0.05970  0.09287
#> [20,] -0.08336 -0.15010  0.00513 -0.08016
#> [21,] -0.03822 -0.00900  0.12366  0.14732
#> [22,] -0.10196  0.03368 -0.11999  0.24609
#> [23,] -0.02128  0.18663 -0.11713  0.18227
#> [24,]  0.15232 -0.07111 -0.07595 -0.07378
#> [25,] -0.14961 -0.08608  0.15842  0.16188
#> [26,]  0.11459  0.02395  0.09186  0.08478
#> [27,] -0.15323 -0.00663  0.06278 -0.00687
#> [28,] -0.34718  0.06231 -0.02012  0.16642
#> [29,]  0.04636  0.03081  0.10077  0.22808
#> [30,] -0.01085 -0.01021  0.13241  0.09381
#> [31,] -0.11897  0.05215 -0.03983  0.01440
#> [32,] -0.14383  0.11954  0.03977 -0.22294
#> [33,] -0.18174 -0.13472  0.00262 -0.01183
#> [34,]  0.10970  0.01414  0.10006  0.04811
#> [35,]  0.20978 -0.22341  0.12869  0.14858
#> [36,]  0.06108 -0.14862 -0.14610 -0.00321
#> [37,]  0.00784 -0.02109  0.00440  0.13452
#> [38,] -0.19574 -0.09842  0.20952 -0.00292
#> [39,] -0.10492 -0.08540 -0.13392  0.07673
```

```
#> [40,]  0.00476  0.05062 -0.07881  0.03740
#> [41,] -0.01485  0.13959 -0.06979  0.13048
#> [42,] -0.11053 -0.05512 -0.07241  0.06784
#> [43,]  0.00940 -0.07098 -0.00892  0.12019
#> [44,]  0.06344 -0.04091  0.15783  0.16184
#> [45,] -0.12411  0.10278  0.05589 -0.14154
#> [46,]  0.19127 -0.07341 -0.14263  0.10051
#> [47,] -0.19534 -0.14913 -0.03448  0.18142
#> [48,]  0.00663  0.01195 -0.24117 -0.20325
#> [49,] -0.02342 -0.16000  0.19426 -0.06717
#> [50,]  0.02202 -0.04244 -0.07099 -0.04297
#> [51,]  0.09884 -0.12388  0.02242  0.27041
#> [52,]  0.05379  0.05924  0.16078 -0.03502
#> [53,] -0.02698 -0.18979  0.10155  0.08442
#> [54,] -0.09002 -0.10432 -0.22834  0.09920
#> [55,]  0.00628  0.08315  0.29747 -0.02728
#> [56,] -0.12627  0.04514 -0.10000 -0.17134
#> [57,]  0.06228  0.18790  0.18247 -0.08109
#> [58,]  0.14704  0.02233  0.00744  0.01677
#> [59,] -0.20980  0.07850  0.14361  0.11547
#> [60,]  0.12021  0.03307 -0.12879 -0.01103
#> [61,] -0.02504 -0.29292 -0.14482 -0.13992
#> [62,]  0.10318  0.34753  0.04735 -0.04666
#> [63,]  0.05664  0.23366 -0.00399 -0.03063
#> [64,] -0.13139 -0.02086  0.12511 -0.12123
#> [65,]  0.20635  0.01377 -0.05582  0.03845
#> [66,]  0.05582  0.00585 -0.14121  0.01831
#> [67,]  0.15660 -0.06033  0.24952 -0.18930
#> [68,] -0.02031  0.26505  0.11598  0.00286
#> [69,]  0.10601  0.21614  0.03125  0.09715
#> [70,] -0.03703  0.18843 -0.17869  0.11882
#>
#> $v
#>       [,1]   [,2]     [,3]    [,4]
#> [1,] 0.875 -0.483 -0.00833 -0.0064
#> [2,] 0.297  0.543 -0.66982  0.4099
#> [3,] 0.259  0.450  0.74072  0.4260
#> [4,] 0.281  0.518  0.05092 -0.8065
```

Let $X = UDV^T$. Then for every $\mathbf{b} \in \mathbb{R}^{p+1}$,

$$X\mathbf{b} = UDV^T\mathbf{b}$$

Each V, D, and U linearly transform $\mathbf{b}$. How? For reproducibility, think about a circle.

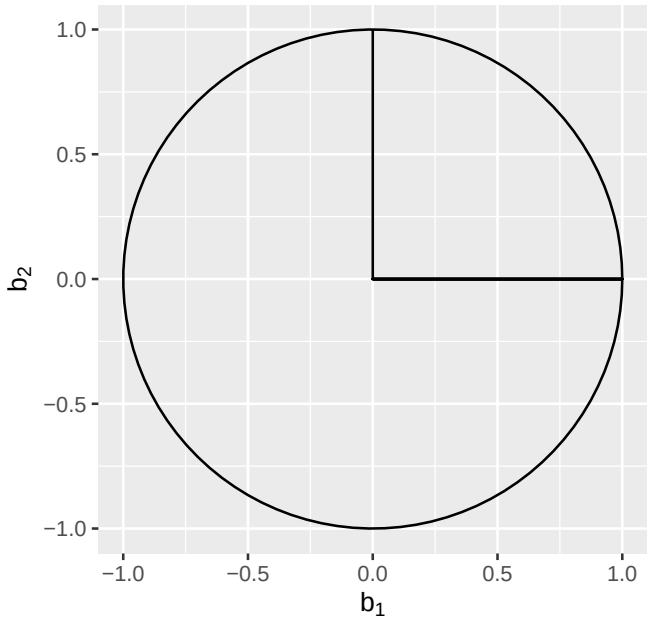


Figure 4.1: A circle with radius 1

Now consider linear transformation

$$V^T\mathbf{b}$$

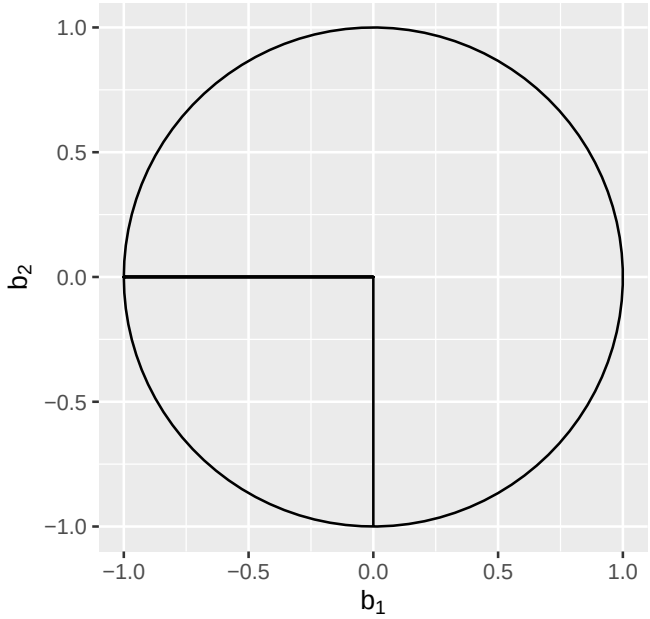


Figure 4.2: $V^T\mathbf{b}$

In Figure 4.2, we can see that V maintains length and angle, but just rotate axes. Next,

$$D(V^T\mathbf{b})$$

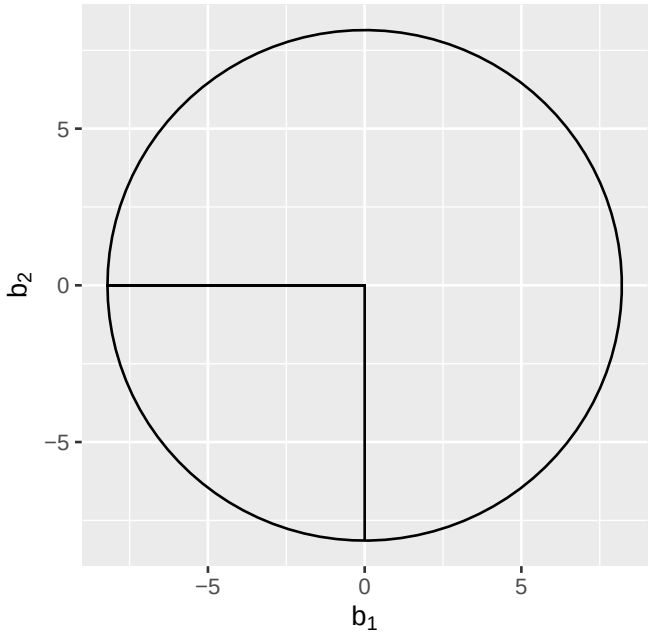


Figure 4.3: $D(V^T\mathbf{b})$

In Figure 4.3, length has changed. In fact, angle has also changed a little bit while rotating. Finally,

$$U(DV^T\mathbf{b})$$

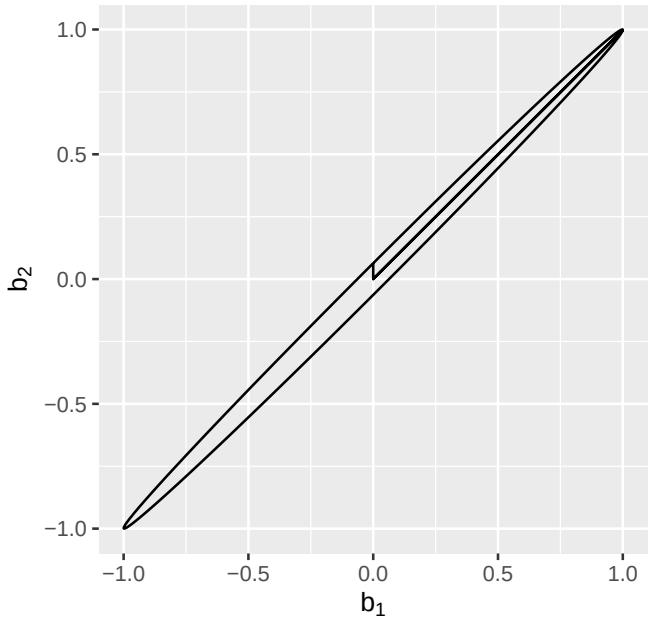


Figure 4.4: $U(DV^T\mathbf{b})$

Figure 4.4 is the final result of SVD. Coordinate is rotated each.

4.3.3 OLS via SVD

Using reduced SVD 4.8, it is possible to solve OLS problem (Hastie et al., 2013). Before applying it, consider centered input. Write

$$X = [\mathbf{1} \mid \mathbb{X}_A]$$

as before.

Lemma 4.2 (Centered input)**.** *Let*

$$\widetilde{\mathbb{X}_A} = \left[x_{ij} - \overline{x}_j\right]$$

be centered design matrix and let

$$\mathbb{X}_{A,\perp} = \mathbb{X}_A - \Pi_{\mathbf{1}}\mathbb{X}_A$$

be an projection of predictor variable observations into $R(\mathbf{1})$. Two are equivalent, i.e.

$$\widetilde{\mathbb{X}_A} = \mathbb{X}_{A,\perp}$$

Proof. Note that

$$\Pi_{\mathbf{1}} = \frac{1}{n}\mathbf{1}\mathbf{1}^T$$

Then

$$\Pi_1 \mathbb{X}_A = \frac{1}{n}\mathbf{1}\mathbf{1}^T X = \begin{bmatrix} \overline{x}_1 & \overline{x}_2 & \cdots & \overline{x}_p \\ \overline{x}_1 & \overline{x}_2 & \cdots & \overline{x}_p \\ \vdots & \vdots & \cdots & \vdots \\ \overline{x}_1 & \overline{x}_2 & \cdots & \overline{x}_p \end{bmatrix}$$

where $\overline{x}_j$ is the sample average for observations of j-th variable. It follows that

$$\mathbb{X}_A - \Pi_1 \mathbb{X}_A = \begin{bmatrix} x_{11} - \overline{x}_1 & x_{12} - \overline{x}_2 & \cdots & x_{1p} - \overline{x}_p \\ x_{21} - \overline{x}_1 & x_{22} - \overline{x}_2 & \cdots & x_{2p} - \overline{x}_p \\ \vdots & \vdots & \cdots & \vdots \\ x_{n1} - \overline{x}_1 & x_{n2} - \overline{x}_2 & \cdots & x_{np} - \overline{x}_p \end{bmatrix} = \widetilde{\mathbb{X}_A}$$

□

Note that

$$\mathbf{1} \perp \mathbb{X}_{A,\perp} = \widetilde{\mathbb{X}_A}$$

Then

$$\begin{aligned} X\hat{\boldsymbol{\beta}} &= \mathbf{1}\hat{\beta}_0^* + \widetilde{\mathbb{X}_A}\hat{\boldsymbol{\beta}}_A \\ &= \Pi(\mathbf{Y} \mid R(\mathbf{1})) + \Pi(\mathbf{Y} \mid R(\widetilde{\mathbb{X}_A})) \end{aligned}$$

Hence,

$$\begin{cases} \hat{\beta}_0^* = \overline{Y} \\ \hat{\boldsymbol{\beta}}_A = (\widetilde{\mathbb{X}_A}^T \widetilde{\mathbb{X}_A})^{-1} \widetilde{\mathbb{X}_A}^T \mathbf{Y} \end{cases}$$

Since the intercept is estimated as response average, we only need to care about input part from now on. Apply SVD to $\widetilde{\mathbb{X}_A}$.

$$\widetilde{\mathbb{X}_A} = UDV^T$$

Then the LSE becomes

$$\begin{aligned}
\hat{\boldsymbol{\beta}}_A &= (\widetilde{\mathbb{X}_A}^T \widetilde{\mathbb{X}_A})^{-1} \widetilde{\mathbb{X}_A}^T \mathbf{Y} \\
&= (VD^2V^T)^{-1} VDU^T \mathbf{Y} \\
&= VD^{-1}U^T \mathbf{Y} \\
&= V diag\Big(\frac{1}{\sigma_j}\Big) U^T \mathbf{Y}
\end{aligned} \tag{4.8}$$

This implies that rank deficiency, i.e. existence of $\sigma_j = 0$, makes OLS not properly work. In turn, fitted values can be computed in a compact form. Suppose that $\sigma_j \neq 0$.

$$\begin{aligned}
\hat{\mathbf{Y}} &= \mathbf{1}\overline{Y} + \widetilde{\mathbb{X}_A}\hat{\boldsymbol{\beta}}_A \\
&= \mathbf{1}\overline{Y} + UDV^T VD^{-1}U^T \mathbf{Y} \\
&= \mathbf{1}\overline{Y} + UDD^{-1}U^T \mathbf{Y} \\
&= \mathbf{1}\overline{Y} + UU^T \mathbf{Y}
\end{aligned} \tag{4.9}$$

Here, $U^T\mathbf{Y}$ are the coordinates of $\mathbf{Y}$ with respect to the orthonormal basis U, i.e. of $Col(X)$. For more details, rewrite the Equation (4.9). If $\sigma_j \neq 0$, then

$$\hat{\mathbf{Y}} = UU^T\mathbf{Y} = \sum_{j=1}^{p} \mathbf{u}_j \mathbf{u}^T \mathbf{Y}$$

Each $\mathbf{u_j}$ indicating basis of $Col(X)$ is a coordinate. In other words, we can think of SVD as another orthogonalization process with QR decomposition in estimating regression model. The fitted value, which is projection onto $Col(X)$ and $\hat{\theta}_j = \mathbf{u}^T\mathbf{Y}$ are corresponding coordinates of $\mathbf{Y}$. Summing every $\hat{\theta}_j$, we can get the fitted value.

Compare this with QR decomposition

$$\hat{\mathbf{Y}} = QQ^T\mathbf{Y}$$

whose orthonormal basis is Q. Observe that Q and U are generally different orthonormal bases for $Col(X)$ (Hastie et al., 2013).

4.3.4 Principal component analysis

Principal component analysis (PCA) finds a set of p orthogonal variables made by linear combination of original variables. Furthermore, we try to reduce the dimension $M < p$. It is called *dimension reduction* method. For instance, scatterplot needs at most two or three dimension. PCA enables us to visualize more easily. Since we remove some elements, we should preserve information of the data as much as possible. What are the information of data? How to preserve them?

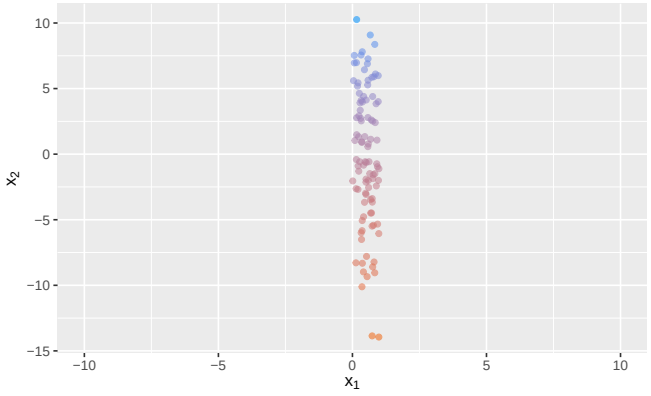


Figure 4.5: Summarizing the features

See Figure 4.5. With $X_1 = x_{i1}$, it is hard to distinguish the observations.

Conjecture 4.4. *Small variance of a variable implies small information about the data.*

Thus, we try to find the direction having the largest variance. For consistency, here we also use centered predictors. Let $\mathbf{X} = (X_1, \ldots, X_p)^T$ be a random vector with a mean $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_p)^T$ and a variance Σ. Then $\widetilde{\mathbf{X}} = (X_1 - \mu_1, \ldots, X_p - \mu_p)^T$ has a mean $\mathbf{0}$ and a variance Σ.

Definition 4.6 (Population principal components)**.** Consider $\widetilde{\mathbf{X}}$. $Z_1, \ldots, Z_p$ are said to be ***principal components*** of $\widetilde{\mathbf{X}}$ iff

1. First PC $Z_1 = \underset{\mathbf{a}_1^T\widetilde{\mathbf{X}}}{\operatorname{argmax}} \mathbf{a}_1^T \Sigma \mathbf{a}_1$ subject to $\mathbf{a}_1^T \mathbf{a}_1 = 1$
2. Second PC $Z_2 = \underset{\mathbf{a}_2^T\widetilde{\mathbf{X}}}{\operatorname{argmax}} \mathbf{a}_2^T \Sigma \mathbf{a}_2$ subject to $\mathbf{a}_2^T \mathbf{a}_2 = 1$ and $\mathbf{a}_2^T \mathbf{a}_1 = 0$
3. $\vdots$
4. Last PC $Z_p = \underset{\mathbf{a}_p^T\widetilde{\mathbf{X}}}{\operatorname{argmax}} \mathbf{a}_p^T \Sigma \mathbf{a}_1$ subject to $\mathbf{a}_p^T \mathbf{a}_p = 1$ and $\forall k < p : \mathbf{a}_p^T \mathbf{a}_k = 0$

By construction, PCA aims to finding uncorrelated set of linear combinations. How to find the solution $\mathbf{a}_j = (a_{1j}, \ldots, a_{pj})^T$? It can be shown that *Spectral decomposition* of Σ gives it (Johnson and Wichern, 2013).

Theorem 4.9 (Population PC with covariance)**.** *Continuing the PCA of* $\widetilde{\mathbf{X}}$*, let*

$$\lambda_1 \geq \cdots \geq \lambda_p \geq 0$$

be the eigenvalues of Σ*, and let* $\mathbf{v}_1, \ldots, \mathbf{v}_p$ *be the corresponding orthonormal eigenvectors. Then each* j*-th principal components is computed by*

$$Z_j = \mathbf{v}_j \widetilde{\mathbf{X}}$$

with

$$\begin{cases} Var(Z_j) = \lambda_j \\ \forall j \neq k : Cov(Z_j, Z_k) = 0 \end{cases}$$

Proof. Note that covariance matrix Σ is symmetric. Then by spectral decomposition,

$$\Sigma = P \Lambda P^T$$

where $P = \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_p \end{bmatrix} \in \mathbb{R}^{p \times p}$ and $\Lambda = diag(\lambda_1, \ldots, \lambda_p) \in \mathbb{R}^{p \times p}$

Step 1: First PC

Let $\mathbf{a} \in \mathbb{R}^p$. Then

$$\begin{aligned} Var(\mathbf{a}^T \widetilde{\mathbf{X}}) &= \mathbf{a}^T \Sigma \mathbf{a} \\ &= \mathbf{a}^T P \Lambda P^T \mathbf{a} \\ &= \mathbf{b}^T \Lambda \mathbf{b} \quad \leftarrow \text{Set } \mathbf{b} := \mathbf{a}^T P \\ &= \lambda_1 b_1^2 + \cdots + \lambda_p b_p^2 \\ &\leq \lambda_1 (b_1^2 + \cdots + b_p^2) \quad \because \lambda_1 \geq \lambda_2, \ldots, \lambda_p \\ &= \lambda_1 \mathbf{b}^T \mathbf{b} \\ &= \lambda_1 \mathbf{a}^T P P^T \mathbf{a} \\ &= \lambda_1 \mathbf{a}^T \mathbf{a} \\ &= \lambda_1 \quad \because \text{construction} \end{aligned}$$

Thus, λ_1 is the upper bound of $Var(\mathbf{a}^T \widetilde{\mathbf{X}})$ subject to $\mathbf{a}^T \mathbf{a} = 1$.

Claim: $\mathbf{a} = \mathbf{v}_1$

This is quite trivial. Set $\mathbf{a} = \mathbf{v}_1$. Then

$$\mathbf{v}_1^T \Sigma \mathbf{v}_1 = \lambda_1 \mathbf{v}_1^T \mathbf{v}_1 = \lambda_1$$

Hence,

$$Z_1 = \mathbf{v}_1 \widetilde{\mathbf{X}}$$

with $Var(Z_1) = \lambda_1$.

Step 2: Find $\mathbf{a}_2 = \underset{\mathbf{a} \in \mathbb{R}^p}{\operatorname{argmax}} \, \mathbf{a}_2^T \Sigma \mathbf{a}_2$ subject to $\mathbf{a}_2^T \mathbf{a}_2 = 1$ and $\mathbf{a}_2^T \Sigma \mathbf{v}_1 = 0$

Assume that $\mathbf{a}_2^T \mathbf{a}_2 = 1$ and that $\mathbf{a}_2 \perp \mathbf{v}_2$. Since $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_p\}$ is an orthonormal basis of $\mathbb{R}^p$,

$$\mathbf{a}_2 \in \{\mathbf{v}_2, \ldots, \mathbf{v}_p\}$$

Then we now have

$$\begin{aligned} \mathbf{a}_2^T \Sigma \mathbf{a}_2 &= \lambda_j \quad \text{for some } j \in \{2, \ldots, p\} \\ &\leq \lambda_2 \quad \because \lambda_2 \geq \lambda_3, \ldots, \lambda_p \end{aligned}$$

By symmetry, we can get the upper bound λ_2 at $\mathbf{a}_2 = \mathbf{v}_2$. Therefore,

$$Z_2 = \mathbf{v}_2^T \widetilde{\mathbf{X}}$$

with $Var(Z_2) = \lambda_2$. In addition,

$$Cov(Z_2, Z_1) = \mathbf{v}_2^T \Sigma \mathbf{v}_1 = \lambda_1 \mathbf{v}_2^T \mathbf{v}_1 = 0$$

One proceeds in a similar way for the next $j = 3, \ldots, p$. Since

$$\mathbf{a}^T \Sigma \mathbf{a} \leq \lambda_j$$

in each step,

$$Z_j = \mathbf{v}_j^T \widetilde{\mathbf{X}}$$

, $Var(Z_j) = \lambda_j$, and $Cov(Z_j, Z_k)$ for every $k < j$. □

Using Principal axes theorem 4.4, we can understand this covariance PCA better. Recall that a gaussian random variable $\mathbf{X} \sim MVN(\boldsymbol{\mu}, \Sigma)$ can be represented by ellipse $\widetilde{\mathbf{X}}^T \Sigma^{-1} \widetilde{\mathbf{X}} = c^2$.

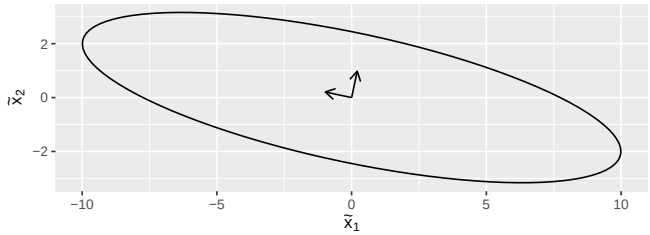


Figure 4.6: Ellipse for some bivariate normal distribution

From Corollary 4.3,

$$\begin{aligned} c^2 &= \widetilde{\mathbf{X}}^T \Sigma^{-1} \widetilde{\mathbf{X}} \\ &= \frac{1}{\lambda_1} (\mathbf{v}_1 \widetilde{\mathbf{X}})^2 + \cdots + \frac{1}{\lambda_p} (\mathbf{v}_p \widetilde{\mathbf{X}})^2 \\ &= \frac{1}{\lambda_1} Z_1^2 + \cdots + \frac{1}{\lambda_p} Z_p^2 \quad \leftarrow Z_j = j\text{-th } PC \end{aligned} \tag{4.10}$$

Hence, PCA is a linear transformation of coordinate

$$V^T \widetilde{\mathbf{X}}$$

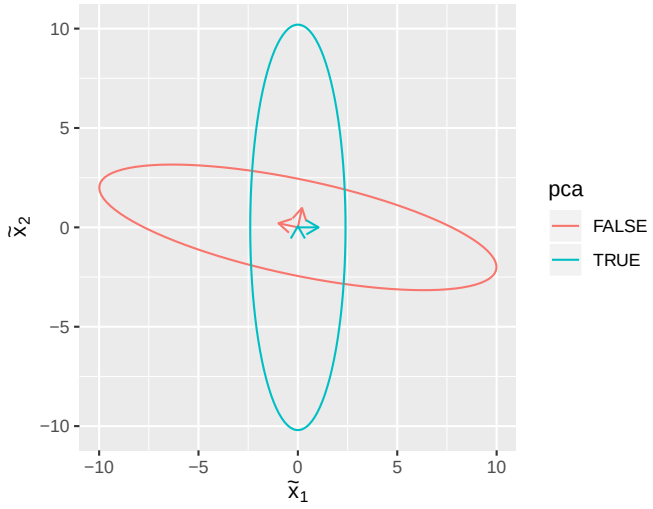


Figure 4.7: Ellipse with respect to principal components

4.3.5 Sample principal components

Our aim is to find principal components given observed sample. In this case,

$$\widetilde{X} = [x_{ij} - \overline{x}_j] \in \mathbb{R}^{n \times p}$$

Remark (Sample covariance matrix). For centered input $\widetilde{X}$, empirical covariance matrix is given by

$$S = \frac{1}{n-1}\widetilde{X}^T\widetilde{X} = \left[\frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)(x_{ik} - \overline{x}_k)\right]_{j\times k} \in \mathbb{R}^{p\times p}$$

Conduct the spectral decomposition for S, i.e.

$$S = P\Lambda P^T$$

where

$$\Lambda = diag(\lambda_1, \ldots, \lambda_p) \qquad \text{and} \qquad P = \begin{bmatrix}\mathbf{v}_1 & \cdots & \mathbf{v}_p\end{bmatrix}$$

with eigenvalue-orthonormal eigenvector pair $(\lambda_j, \mathbf{v}_j)$, $\lambda_1 \geq \cdots \geq \lambda_p \geq 0$. We can get an observed vector for j-th PC using Theorem 4.9

$$\mathbf{z}_j = \widetilde{X}\mathbf{v}_j = \begin{bmatrix} z_{1j} \\ z_{2j} \\ \vdots \\ z_{nj} \end{bmatrix} \in \mathbb{R}^n \tag{4.11}$$

This is called the j-th *principal component scores*.

Theorem 4.10 (Sample principal components)**.** *Given observed sample $\widetilde{X}$, entire principal component scrore matrix can be computed by*

$$\begin{aligned} Z &= \widetilde{X}P \in \mathbb{R}^{n \times p} \\ &= \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1p} \\ z_{21} & z_{22} & \cdots & z_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ z_{n1} & z_{n2} & \cdots & z_{np} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{z}_1 & \mathbf{z}_2 & \cdots & \mathbf{z}_p \end{bmatrix} \\ &= \textit{linear combination for observations} \end{aligned}$$

To reduce dimension, we just use the first $q < p$ columns of P.

$$Z = \widetilde{X}P_q$$

where P_q consists of the first q column of P. Or we just discard the last columns after calculating Z.

```
eeo_cent <- scale(eeo, scale = FALSE)
#-----------------------------------
eeo_p <- eigen(var(eeo_cent))$vectors
colnames(eeo_p) <- colnames(eeo)
eeo_p
#>        ACHV    FAM     PEER   SCHOOL
#> [1,] 0.876  0.483  0.00834  0.00648
#> [2,] 0.296 -0.544  0.67029 -0.40931
#> [3,] 0.258 -0.450 -0.74023 -0.42715
#> [4,] 0.280 -0.518 -0.05197  0.80621
```

```
eeo_cent %*% eeo_p %>% # XP
  as_tibble()
#> # A tibble: 70 x 4
#>     ACHV    FAM    PEER  SCHOOL
#>    <dbl>  <dbl>   <dbl>   <dbl>
#> 1 -0.194 -0.586  0.372  -0.119
#> 2  1.16  -0.483  0.159  -0.0801
```

```
#> 3 -1.49   0.744 -0.0594 -0.0230
#> 4 -2.95   0.766  0.0992  0.194
#> 5 -2.50  -1.40   0.225   0.118
#> 6 -0.463 -0.574  0.0242  0.0926
#> # ... with 64 more rows
```

In R, `princomp()` conduct PCA through spectral decomposition.

```
princomp(~ ., data = eeo)$scores %>%
  as_tibble()
#> # A tibble: 70 x 4
#>   Comp.1 Comp.2  Comp.3  Comp.4
#>    <dbl>  <dbl>   <dbl>   <dbl>
#> 1 -0.194 -0.586  0.372  -0.119
#> 2  1.16  -0.483  0.159  -0.0801
#> 3 -1.49   0.744 -0.0594 -0.0230
#> 4 -2.95   0.766  0.0992  0.194
#> 5 -2.50  -1.40   0.225   0.118
#> 6 -0.463 -0.574  0.0242  0.0926
#> # ... with 64 more rows
```

This function implements `eigen()`. Of course, the result is the same as we calculated directly.

4.3.6 PC as linear manifolds

Principal components finds linear combination having maximal variances in order. We can find another components. In fact, these are equivalent. Find the closest hyperplane to the points (Hastie et al., 2013).

$$\mathbf{f}(\mathbf{z}) = \boldsymbol{\mu} + V_q \mathbf{z}, \quad V_q \in \mathbb{R}^{p \times q} \text{ orthogonal}$$

with $V_q = \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_q \end{bmatrix}$.

This find least squres solution minimizing the errors between the hyperplane. In OLS, loss was calculated with respect to the original coordinate but here it is calculated vertical to the hyperplane.

Definition 4.7 (Reconstruction error)**.** PCA deals with reconstruction error.

$$RE := \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{f}(\mathbf{z}_i)\|^2$$

PCA corresponds to the problem of finding least squares for

$$(\hat{\boldsymbol{\mu}}, \{\mathbf{z}_i\}, \hat{P}_q) = \underset{\boldsymbol{\mu}, \{\mathbf{z}_i\}, V_q}{\operatorname{argmin}} \sum_{i=1}^{n} \|\mathbf{x}_i - \boldsymbol{\mu} - V_q \mathbf{z}_i\|^2 \tag{4.12}$$

First partially optimize for $\boldsymbol{\mu}$ and $\{\mathbf{z}_i\}$.

Since

$$RE = \sum (\mathbf{x}_i - \boldsymbol{\mu} - V_q \mathbf{z}_i)^T (\mathbf{x}_i - \boldsymbol{\mu} - V_q \mathbf{z}_i)$$

$$\frac{\partial RE}{\partial \boldsymbol{\mu}} = -2 \sum (\mathbf{x}_i - \boldsymbol{\mu} - V_q \mathbf{z}_i)$$

Setting it to be $\mathbf{0}$,

$$\hat{\boldsymbol{\mu}} = \overline{\mathbf{x}} - V_q \overline{\mathbf{z}} \tag{4.13}$$

Since

$$RE = \sum \left\{ \|\mathbf{x}_i - \boldsymbol{\mu}\|^2 - 2(\mathbf{x}_i - \boldsymbol{\mu})^T V_q \mathbf{z}_i + \mathbf{z}_i^T \mathbf{z}_i \right\}$$

$$\frac{\partial RE}{\partial \mathbf{z}_i} = -2 V_q^T (\mathbf{x}_i - \boldsymbol{\mu}) + 2\mathbf{z}_i$$

Setting this to be $\mathbf{0}$,

$$\hat{\mathbf{z}}_i = V_q^T (\mathbf{x}_i - \boldsymbol{\mu}) \tag{4.14}$$

From Equations (4.13) and (4.14),

$$\begin{aligned} &\hat{\boldsymbol{\mu}} = \overline{\mathbf{x}} - V_q V_q^T (\overline{\mathbf{x}} - \hat{\boldsymbol{\mu}}) \\ &\Leftrightarrow V_q V_q^T (\overline{\mathbf{x}} - \hat{\boldsymbol{\mu}}) = \overline{\mathbf{x}} - \hat{\boldsymbol{\mu}} \\ &= (I - V_q V_q^T)(\overline{\mathbf{x}} - \hat{\boldsymbol{\mu}}) = \mathbf{0} \end{aligned}$$

Since V_q is orthogonal, $V_q V_q^T = V_q (V_q^T V_q)^{-1} V_q^T$ is a projection onto $R(V_q)$. Theorem 1.4 implies that $I - V_q V_q^T$ is a projection onto $R(V_q)^\perp$. Thus,

$$\hat{\boldsymbol{\mu}} = \overline{\mathbf{x}} + \mathbf{h}, \quad \mathbf{h} \in R(V_q)^\perp$$

Setting $\mathbf{h} = \mathbf{0}$,

$$\begin{cases} \hat{\boldsymbol{\mu}} = \overline{\mathbf{x}} \\ \hat{\mathbf{z}}_i = V_q^T(\mathbf{x}_i - \overline{\mathbf{x}}) \end{cases} \tag{4.15}$$

We are using centered input, so

$$\boldsymbol{\mu} = \mathbf{0}, \quad \overline{\mathbf{x}} = \mathbf{0}$$

Denote that we do not know V_q yet. Partially optimized,

$$RE = \sum_{i=1}^{n} \|\widetilde{\mathbf{x}}_i - V_q V_q^T \widetilde{\mathbf{x}}_i\|^2$$

To get V_q, solve

$$\hat{P}_q = \underset{V_q}{\operatorname{argmin}} \sum_{i=1}^{n} \|\widetilde{\mathbf{x}}_i - V_q V_q^T \widetilde{\mathbf{x}}_i\|^2 \tag{4.16}$$

Again, $V_q V_q^T$ is a projection onto $Col(V_q)$. It is easy to realize that we might use Equation (4.9).

Theorem 4.11 (Principal components by SVD)**.** *Given centered input $\widetilde{\mathbb{X}}_A$, SVD is conducted. Then*

$$\begin{aligned} \widetilde{\mathbb{X}}_A V = UD &= Z \\ &= \begin{bmatrix} \mathbf{z}_1 & \cdots & \mathbf{z}_p \end{bmatrix} \end{aligned}$$

with dimension reduction

$$\hat{V}_q = \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_q \end{bmatrix}$$

While `princomp()` does it with `eigen()`, `prcomp()` conduct PCA using singular value decomposition. `R` is recommending this function more that the former.

> The calculation is done by a singular value decomposition of the (centered and possibly scaled) data matrix, not by using eigen on the covariance matrix. This is generally the preferred method for numerical accuracy. The print method for these objects prints the results in a nice format and the plot method produces a scree plot.

The following is the result of UD

```
eeo_svd <- svd(eeo_cent)
eeo_svd$u %*% diag(eeo_svd$d) %>%
  head()
#>        [,1]   [,2]    [,3]    [,4]
#> [1,] -0.194  0.586 -0.3721  0.1187
#> [2,]  1.156  0.483 -0.1589  0.0801
#> [3,] -1.487 -0.744  0.0594  0.0230
#> [4,] -2.948 -0.766 -0.0992 -0.1935
#> [5,] -2.503  1.405 -0.2252 -0.1183
#> [6,] -0.463  0.574 -0.0242 -0.0926
```

Of course, $\widetilde{\mathbb{X}}_A V$ gives the same result.

```
head(eeo_cent %*% eeo_svd$v)
#>        [,1]   [,2]    [,3]    [,4]
#> [1,] -0.194  0.586 -0.3721  0.1187
#> [2,]  1.156  0.483 -0.1589  0.0801
#> [3,] -1.487 -0.744  0.0594  0.0230
#> [4,] -2.948 -0.766 -0.0992 -0.1935
#> [5,] -2.503  1.405 -0.2252 -0.1183
#> [6,] -0.463  0.574 -0.0242 -0.0926
```

Try `prcomp()`. `$x` has PC score.

```
prcomp(eeo)$x %>%
  as_tibble()
#> # A tibble: 70 x 4
#>      PC1    PC2     PC3     PC4
#>    <dbl>  <dbl>   <dbl>   <dbl>
#> 1 -0.194  0.586 -0.372   0.119
#> 2  1.16   0.483 -0.159   0.0801
#> 3 -1.49  -0.744  0.0594  0.0230
#> 4 -2.95  -0.766 -0.0992 -0.194
#> 5 -2.50   1.40  -0.225  -0.118
#> 6 -0.463  0.574 -0.0242 -0.0926
#> # ... with 64 more rows
```

Recall that these results are same as previous spectral decomposition. In fact, linear approximation approach is equivalent to variance maximization.

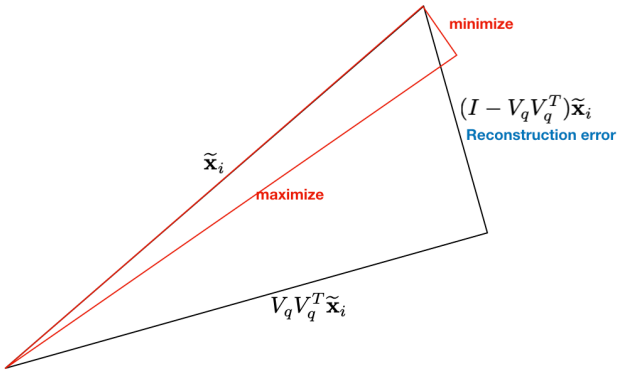


Figure 4.8: Equivalence of two approaches

See Figure 4.8. By the Pythagorean law,

$$\begin{aligned}\|\widetilde{\mathbf{x}}_i\|^2 &= \|V_q V_q^T \widetilde{\mathbf{x}}_i\|^2 + \|(I - V_q V_q^T)\widetilde{\mathbf{x}}_i\|^2 \\ &= \|V_q^T \widetilde{\mathbf{x}}_i\|^2 + \|(I - V_q V_q^T)\widetilde{\mathbf{x}}_i\|^2\end{aligned}$$

It follows that

$$\begin{aligned}Var(\widetilde{\mathbf{x}}_i) &= Var(V_q^T \widetilde{\mathbf{x}}_i) + \sum_{i=1}^{n} \|(I - V_q V_q^T)\widetilde{\mathbf{x}}_i\|^2 \\ \sum_{j=1}^{p} \mathbf{v}_j^T S \mathbf{v}_j &= \sum_{j=1}^{q} \mathbf{v}_j^T S \mathbf{v}_j + \sum_{i=1}^{n} \|(I - V_q V_q^T)\widetilde{\mathbf{x}}_i\|^2 \\ \sum_{j=1}^{p} \mathbf{v}_j^T S \mathbf{v}_j &= \text{maximal variance} + \text{minimal reconstruction error} \\ \sum_{j=1}^{p} \mathbf{v}_j^T S \mathbf{v}_j &= \text{Spectral decomposition} + \text{reduced SVD}\end{aligned}$$

In fact, spectral decomposition is just a special case of SVD. Recall that

$$S = \frac{1}{n-1} \widetilde{\mathbb{X}}_A^T \widetilde{\mathbb{X}}_A$$

The only difference in eigenvalues of S and $\widetilde{\mathbb{X}}_A^T\widetilde{\mathbb{X}}_A$ is proportion $\frac{1}{n-1}$. Arithmetically, the two calculations are equivalent.

4.3.7 Variance

Theorem 4.12 (Sample variance of PC)**.** *Let $\mathbf{z}_j$ be j-th PC score. Then*

- *Total variance*

$$\sum_{j=1}^{p} Var(\mathbf{Z}_j) = \sum_{j=1}^{p} Var(\widetilde{\mathbf{X}}_j) = \frac{1}{n}\sum \sigma_j^2$$

- *Sample variance explained by $\mathbf{z}_j$*

$$Var(\mathbf{z}_j) = \frac{\sigma_j^2}{n}$$

- *Proportion of variance explained using q PCs*

$$PVE_q = \frac{\sigma_1^2 + \cdots + \sigma_q^2}{\sum\limits_{j=1}^{p} \sigma_j^2}$$

Proof. Consider principal component scores

$$Z = \widetilde{\mathbb{X}}_A P = UD = \widetilde{\mathbb{X}}_A V$$

Then

$$\mathbf{z}_j = \sigma_j \mathbf{u}_j$$

and so

$$\begin{aligned} Var(\mathbf{z}_j) &= \frac{1}{n} < \mathbf{z}_j, \mathbf{z}_j > \\ &= \frac{1}{n} < \sigma_j \mathbf{u}_j, \sigma_j \mathbf{u}_j > \\ &= \frac{\sigma^2}{n} \end{aligned}$$

In turn,

$$\begin{aligned}
\sum Var(\widetilde{\mathbf{X}}_j) &= \frac{1}{n}\sum_{i,j}\widetilde{x}_{ij}^2 \\
&= \frac{1}{n}tr(\widetilde{\mathbb{X}}_A^T\widetilde{\mathbb{X}}_A) \\
&= \frac{1}{n}tr(VD^2V^T) \\
&= \frac{1}{n}tr(V^TVD^2) \\
&= \frac{1}{n}tr(D^2) = \sum_j \frac{\sigma_j}{n} \\
&= \sum_{j=1}^{p} Var(\mathbf{Z}_j) \\
&> \sum_{j=1}^{q} Var(\mathbf{Z}_j), \quad q < p
\end{aligned}$$

Hence,

$$PVE_q = \frac{Var(\mathbf{Z}_1) + \cdots + Var(\mathbf{Z}_q)}{\sum Var(\mathbf{Z})} = \frac{\sigma_1^2 + \cdots + \sigma_q^2}{\sum_{j=1}^{p}\sigma_j^2}$$

□

4.3.8 Principal components regression

Using principal components of centered inputs

$$Z = \widetilde{\mathbb{X}}_A P = UD = \widetilde{\mathbb{X}}_A V$$

one may rewrite the regression equation.

$$\begin{aligned}
X\boldsymbol{\beta} &= \mathbf{1}\beta_0^* + \mathbb{X}_{A,\perp}\boldsymbol{\beta}_A \\
&= \mathbf{1}\beta_0^* + \mathbb{X}_{A,\perp}VV^T\boldsymbol{\beta}_A \\
&= \mathbf{1}\beta_0^* + Z(V^T\boldsymbol{\beta}_A) \\
&= \mathbf{1}\beta_0^* + Z\boldsymbol{\alpha}, \quad \alpha \equiv V^T\boldsymbol{\beta}_A
\end{aligned} \tag{4.17}$$

It is trivial that

$$\hat{\beta}_0^* = \overline{Y}$$

In case of $\boldsymbol{\alpha}$,

$$\begin{aligned} \hat{\boldsymbol{\alpha}} &= (Z^T Z)^{-1} Z^T \mathbf{Y} \\ &= (D^T U^T U D)^{-1} D^T U^T \mathbf{Y} \\ &= D^{-1} U^T \mathbf{Y} \end{aligned} \tag{4.18}$$

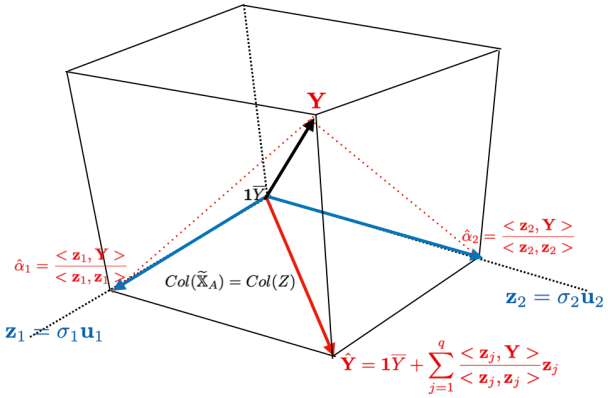


Figure 4.9: Geometry of PCR

Figure 4.9 presents what $\hat{\boldsymbol{\alpha}}$ indicates. It also holds when dimension is reduced by $q \leq p$. Unlike G-S process, PCA rotates the original axes and make orthogonal axes. Since orthogonal estimating process is same. But to use this properly, $\hat{\boldsymbol{\beta}}_A$ should be same as LSE. Is it?

Remark. Let $\hat{\boldsymbol{\beta}}_A^{PCR}$ be $\hat{\boldsymbol{\beta}}_A$ in Equation (4.17). If $q = p$, then

$$\hat{\boldsymbol{\beta}}_A^{PCR} = \hat{\boldsymbol{\beta}}_A^{OLS}$$

Proof. From Equation (4.18) and Equation (4.8),

$$\hat{\boldsymbol{\beta}}_A^{PCR} = V\hat{\boldsymbol{\alpha}} = VD^{-1}U^T\mathbf{Y} = \hat{\boldsymbol{\beta}}_A^{OLS} \tag{4.19}$$

□

Hence, when we use every component in PCR, it is guaranteed that the regression coefficient is same as LSE. Practically, we sometimes discard some components. Or in the presence of multicollinearity, $\sigma_j \approx 0$ for some j. We do not have to use these j. Choose the first q PCs $Z_1, \ldots, Z_q$ with $q < p$.

$$\begin{aligned}
\widetilde{\mathbb{X}}_A \left[\begin{array}{cccc|ccc} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_q & \cdots & \mathbf{v}_p \end{array} \right] &= Udiag(\sigma_1, \sigma_2, \ldots, \sigma_q \Big| \ldots, \sigma_p) \\
&= \widetilde{\mathbb{X}}_A \left[\begin{array}{cccc|ccc} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_q & \cdots & \mathbf{v}_p \end{array} \right] \quad \text{from } P\Lambda P^T \\
&= \left[\begin{array}{cccc|ccc} \mathbf{z}_1 & \mathbf{z}_2 & \cdots & \mathbf{z}_q & \cdots & \mathbf{z}_p \end{array} \right] \\
&= \Big[Z_q \quad \Big| \quad \text{discard } q-p \text{ smallest } \sigma_j \Big]
\end{aligned} \tag{4.20}$$

We just remove entire columns with the smallest σ_j, or equivalently λ_j. How does this removal work? One can worry that removal negatively affect the other variable. Recall that

$$\forall j \neq k : \quad < \mathbf{z}_j, \mathbf{z}_k >= 0$$

i.e. principal components are uncorrelated. Thus, we can freely remove any component.

Look at the Equation (4.17) and Figure 4.9.

$$\hat{\mathbf{Y}}^{PCR} = \mathbf{1}\overline{Y} + \sum_{j=1}^{p} \frac{< \mathbf{z}_j, \mathbf{Y} >}{< \mathbf{z}_j, \mathbf{z}_j >} \mathbf{z}_j \tag{4.21}$$

Here, when $q = p$,

$$\hat{\alpha}_j = \frac{< \mathbf{z}_j, \mathbf{Y} >}{< \mathbf{z}_j, \mathbf{z}_j >}$$

and

$$\hat{\boldsymbol{\beta}}^{PCR} = \sum_{j=1}^{p} \hat{\alpha}_j \mathbf{v}_j$$

is same as regression coefficient estimate. Furthermore, each element is of simple LSE, $\hat{\beta}_j^{LSE}$. Since uncorrelated, try to remove the last $p - q$.

$$\hat{\mathbf{Y}}^{PCR}(q) = \mathbf{1}\overline{Y} + \begin{bmatrix} \hat{\alpha}_1 \\ \vdots \\ \hat{\alpha}_q \end{bmatrix} \begin{bmatrix} \mathbf{z}_1 & \cdots & \mathbf{z}_q \end{bmatrix}$$

In this case, each $\hat{\alpha}_j$ still has same value. However,

$$\hat{\boldsymbol{\beta}}^{PCR} = \sum_{j=1}^{q} \hat{\alpha}_j \mathbf{v}_j \neq \hat{\boldsymbol{\beta}}^{LSE}$$

Refer to Equation (4.9). LSE can be interpreted by principal components.

$$\hat{\mathbf{Y}}^{LSE} = \mathbf{1}\overline{Y} + UDD^{-1}U^T\mathbf{Y} = \mathbf{1}\overline{Y} + \sum_{j=1}^{p} \mathbf{z}_j \frac{\mathbf{u}^T\mathbf{Y}}{d_j}$$

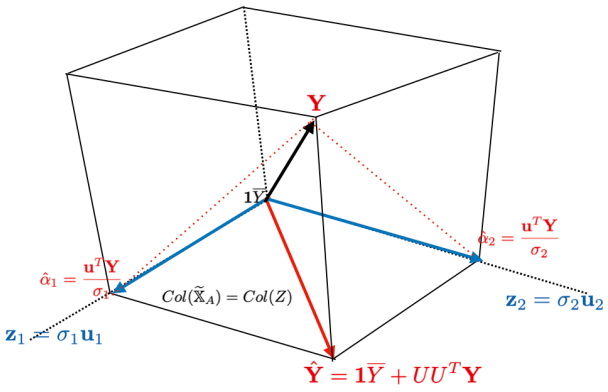


Figure 4.10: LSE and PC

See Figure 4.10. LSE projects the response vector onto principal components hyperplane.

Principal components are chosen in direction of maximal variance of predictors. In other words, it is determined only by predictors, not response.

Conjecture 4.5 (Key idea of PCR)**.** *The directions in which $X_1, \ldots, X_p$ show the most variation are probably going to be associated with Y.*

Conjecture 4.5 is just hunch. It is not guaranteed the best performance of the model (James et al., 2013). Against this kind of disadvantages, alternative methods such as *partial least squares* are sometimes used.

4.4 Ridge Regression

Consider centered input $\widetilde{\mathbb{X}}_A = \mathbb{X}_{A,\perp}$ facing with multicollinearity. Multicollinearity makes estimation unstable.

```
summary(eeo_fit)
#>
#> Call:
#> lm(formula = ACHV ~ ., data = eeo)
#>
#> Residuals:
#>    Min     1Q Median     3Q    Max
#> -5.210 -1.393 -0.295  1.142  4.588
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)   -0.070      0.251   -0.28     0.78
#> FAM            1.101      1.411    0.78     0.44
#> PEER           2.322      1.481    1.57     0.12
#> SCHOOL        -2.281      2.220   -1.03     0.31
#>
#> Residual standard error: 2.07 on 66 degrees of freedom
#> Multiple R-squared:  0.206,  Adjusted R-squared:  0.17
#> F-statistic: 5.72 on 3 and 66 DF,  p-value: 0.00153
```

Look at `Std. Error`. These values of standard error can be thought to be large compared to `Estimate`. **Ridge regression** shrinks unstable variance of coefficient $\frac{1}{\epsilon}$ to $\frac{1}{\epsilon+\kappa}$.

4.4.1 Original motivation of ridge regression

Originally, ridge regression is developed to deal with singularity of X^TX due to linear dependency. When multicollinearity occurs, the matrix becomes rank deficient. Then $(X^TX)^{-1}$ does not exist. In this situation, LSE does not have unique solution.

To make it possible to get the inverse matrix, *add positive values to the diagonal element.*

Remark (Ridge estimator). Ridge estimator always has unique solution even when $(\widetilde{\mathbb{X}}_A^T\widetilde{\mathbb{X}}_A)^{-1}$ does not exist.

$$\hat{\boldsymbol{\beta}}_{A,R} = (\widetilde{\mathbb{X}}_A^T\widetilde{\mathbb{X}}_A + \kappa I)^{-1}\widetilde{\mathbb{X}}_A^T\mathbf{Y} \tag{4.22}$$

with $\kappa > 0$.

This is called *biased regression* in that the model has sacrificed biasedness condition for smaller variance.

Theorem 4.13 (Expectation of ridge estimator)**.** *The ridge estimator* $\hat{\boldsymbol{\beta}}_{A,R}$ *is biased when* $\kappa > 0$*.*

$$E\hat{\boldsymbol{\beta}}_{A,R} = (\widetilde{\mathbb{X}}_A^T\widetilde{\mathbb{X}}_A + \kappa I)^{-1}\widetilde{\mathbb{X}}_A^T\widetilde{\mathbb{X}}_A\boldsymbol{\beta}_A \neq \boldsymbol{\beta}_A$$

Proof. Note that ridge estimator is linear function of $\mathbf{Y}$. Since $E\mathbf{Y} = \widetilde{\mathbb{X}}_A\boldsymbol{\beta}_A$,

$$E\hat{\boldsymbol{\beta}}_{A,R} = (\widetilde{\mathbb{X}}_A^T\widetilde{\mathbb{X}}_A + \kappa I)^{-1}\widetilde{\mathbb{X}}_A^T\widetilde{\mathbb{X}}_A\boldsymbol{\beta}_A$$

□

4.4.2 Ridge penalty

Nowadays, ridge estimator is defined to be the minimizer of penalized least square estimation problem.

Remark (Regularized estimates). Let κ be a tuning parameter for penalty, which controls complexity of the model.

- $$LSE^R = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \Big[RSS + \kappa\text{penalty}\Big]$$
- $$MLE^R = \underset{\boldsymbol{\beta}}{\operatorname{argmax}} \Big[L(\boldsymbol{\beta}) - \kappa\text{penalty}\Big]$$

In ridge regression, l_2 penalty is used.

Definition 4.8 (Ridge penalty)**.** Let $\beta_1, \ldots, \beta_p$ be the regression coefficients. Then ***ridge penalty*** is squared l_2 norm compted by

$$\|\boldsymbol{\beta}_A\|^2 = \sum_{j=1}^{p} \beta_j^2$$

Thus, ridge estimator becomes

$$\hat{\boldsymbol{\beta}}_{A,R} = \underset{\boldsymbol{\beta}_A}{\operatorname{argmin}} \left\{ (\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A\boldsymbol{\beta}_A)^T(\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A\boldsymbol{\beta}_A) + \kappa\boldsymbol{\beta}_A^T\boldsymbol{\beta}_A \right\} \quad (4.23)$$

Theorem 4.14 (Ridge solution)**.** *Solution to the penalized least squares estimation problem* (4.23) *gives exactly same solution as the original one* (4.22)*.*

$$\hat{\boldsymbol{\beta}}_{A,R} = (\widetilde{\mathbb{X}}_A^T\widetilde{\mathbb{X}}_A + \kappa I)^{-1}\widetilde{\mathbb{X}}_A^T\mathbf{Y}$$

Proof. Write

$$Q(\kappa) \equiv (\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A\boldsymbol{\beta}_A)^T(\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A\boldsymbol{\beta}_A) + \kappa\boldsymbol{\beta}_A^T\boldsymbol{\beta}_A$$

Recall that $\overline{Y}$ is an estimate for β_0, not just a constant. So it would be removed when differentiating w.r.t. $\boldsymbol{\beta}_A$. Then we now have

$$\frac{\partial}{\partial\boldsymbol{\beta}_A}Q(\kappa) = -2\widetilde{\mathbb{X}}_A\mathbf{Y} + 2\widetilde{\mathbb{X}}_A^T\widetilde{\mathbb{X}}_A\boldsymbol{\beta}_A + 2\kappa\boldsymbol{\beta}_A$$

Setting it to $\mathbf{0}$, we obtain normal equation

$$(\widetilde{\mathbb{X}}_A^T\widetilde{\mathbb{X}}_A + \kappa I)\hat{\boldsymbol{\beta}}_{A,R} = \widetilde{\mathbb{X}}_A^T\mathbf{Y} \tag{4.24}$$

Hence,

$$\hat{\boldsymbol{\beta}}_{A,R} = (\widetilde{\mathbb{X}}_A^T\widetilde{\mathbb{X}}_A + \kappa I)^{-1}\widetilde{\mathbb{X}}_A^T\mathbf{Y}$$

□

As in section 4.3.3, we might implement SVD for ridge solution formula. For $\widetilde{\mathbb{X}}_A$, we have

$$\widetilde{\mathbb{X}}_A = UDV^T$$

Then we have

$$\widetilde{\mathbb{X}}_A^T\widetilde{\mathbb{X}}_A = VD^2V^T$$

It follows that

$$\begin{aligned}
\hat{\boldsymbol{\beta}}_{A,R} &= (\widetilde{\mathbb{X}}_A^T\widetilde{\mathbb{X}}_A + \kappa I)^{-1}\widetilde{\mathbb{X}}_A^T\mathbf{Y} \\
&= (VD^2V + \kappa I)^{-1}VDU^T\mathbf{Y} \\
&= \Big((VD)^{-1}VD^2V^T + \kappa(VD)^{-1}\Big)^{-1}U^T\mathbf{Y} \\
&= (DV^T + \kappa D^{-1}V^T)^{-1}U^T\mathbf{Y} \\
&= V(D + \kappa D^{-1})^{-1}D^{-1}DU^T\mathbf{Y} \\
&= V(D^2 + \kappa I)^{-1}DU^T\mathbf{Y} \\
&= Vdiag\Big(\frac{\sigma_j}{\sigma_j^2 + \kappa}\Big)U^T\mathbf{Y}
\end{aligned} \tag{4.25}$$

Compare this to OLS (4.8).

Remark (Regression coefficents via SVD). Ridge estimator shrinks OLS estimator.

- $\hat{\boldsymbol{\beta}}_A = V diag\Big(\frac{1}{\sigma_j}\Big) U^T \mathbf{Y}$
- $\hat{\boldsymbol{\beta}}_{A,R} = V diag\Big(\frac{\sigma_j}{\sigma_j^2+\kappa}\Big) U^T \mathbf{Y}$

The difference between the two is κ in denominator. Look at OLS. As mentioned, if $\sigma_j = 0$, we cannot get a unique solution. On the other hand, ridge solution adds $\kappa > 0$, so $\frac{1}{\sigma_j^2+\kappa} < \infty$ is certified. Also, if

$$\kappa \to \infty$$

$\hat{\boldsymbol{\beta}}_{A,B}$ goes close to zero, i.e. *As we give larger penalty, the coefficient shrinks.*

One proceeds in a similar way for **fitted values**.

$$\begin{aligned}
\hat{\mathbf{Y}}_R &= \mathbf{1}\overline{Y} + \widetilde{\mathbb{X}}_A \hat{\boldsymbol{\beta}}_{A,R} \\
&= \mathbf{1}\overline{Y} + UDV^T V diag\Big(\frac{\sigma_j}{\sigma_j^2+\kappa}\Big) U^T \mathbf{Y} \\
&= \mathbf{1}\overline{Y} + UD diag\Big(\frac{\sigma_j}{\sigma_j^2+\kappa}\Big) U^T \mathbf{Y} \\
&= \mathbf{1}\overline{Y} + U diag\Big(\frac{\sigma_j^2}{\sigma_j^2+\kappa}\Big) U^T \mathbf{Y}
\end{aligned} \tag{4.26}$$

In OLS (4.9), UU^T means projection. However, this is derived only when $DD^{-1} = I$, i.e. when $\sigma_j \neq 0$. However, ridge regression always project. In fact, ridge regression projects the response vector *into PC hyperplane*. Re-express Equation (4.26).

$$\begin{aligned}
\hat{\mathbf{Y}}_R &= \mathbf{1}\overline{Y} + U diag\Big(\frac{\sigma_j^2}{\sigma_j+\kappa}\Big) U^T \mathbf{Y} \\
&= \mathbf{1}\overline{Y} + UD diag\Big(\frac{1}{\sigma_j+\kappa}\Big) D U^T \mathbf{Y} \\
&= \mathbf{1}\overline{Y} + \sum_{j=1}^{p} \mathbf{z}_j \left(\frac{1}{\sigma_j^2+\kappa}\right) \mathbf{z}_j^T \mathbf{Y} \quad \mathbf{z}_j = \sigma_j \mathbf{u}_j = j\text{-th } PC \text{ score}
\end{aligned} \tag{4.27}$$

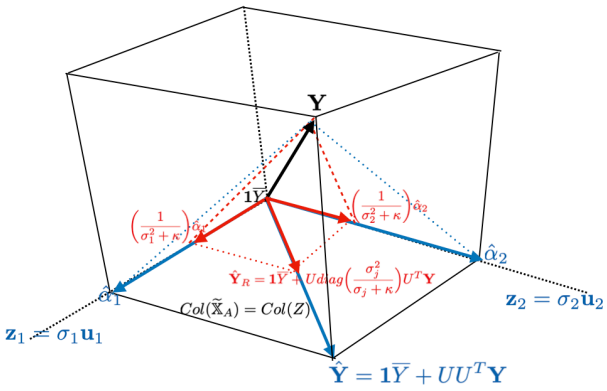


Figure 4.11: Ridge regression and PC

Note that

$$< \mathbf{z}_j, \mathbf{z}_j >= 1$$

Then in Equation (4.27), we have

$$\hat{\alpha}_j = \mathbf{z}_j^T\mathbf{Y}$$

Observe that $\left(\frac{1}{\sigma_j^2+\kappa}\right)$ shrinks each PC coefficient $\hat{\alpha}_j$. For more detail, look at Figure 4.11.

4.4.3 Variance of ridge estimator

Theorem 4.13 shows that the ridge estimator is biased. However, it has much smaller variance.

Theorem 4.15 (Variance of ridge estimator)**.** *Suppose that* $Var\mathbf{Y} = \sigma^2 I$*. Then*

- $$Var\hat{\boldsymbol{\beta}}_A = \sigma^2 V diag\left(\frac{1}{\sigma_j^2}\right)V^T$$
- $$Var\hat{\boldsymbol{\beta}}_{A,R} = \sigma^2 V diag\left(\frac{\sigma_j^2}{(\sigma_j^2+\kappa)^2}\right)V^T$$

Proof. Consider SVD expression (4.8) and (4.25).

$$\begin{aligned} Var\hat{\boldsymbol{\beta}}_A &= \sigma^2(\widetilde{\mathbb{X}}_A^T\widetilde{\mathbb{X}}_A)^{-1} \\ &= \sigma^2(VD^2V^T)^{-1} \\ &= \sigma^2 V diag\left(\frac{1}{\sigma_j^2}\right)V^T \end{aligned}$$

$$\begin{aligned} Var\hat{\boldsymbol{\beta}}_{A,R} &= \left(V diag\Big(\frac{\sigma_j}{\sigma_j^2+\kappa}\Big)U^T\right) Var\mathbf{Y}\left(V diag\Big(\frac{\sigma_j}{\sigma_j^2+\kappa}\Big)U^T\right)^T \\ &= \sigma^2 V diag\Big(\frac{\sigma_j}{\sigma_j^2+\kappa}\Big)U^T U diag\Big(\frac{\sigma_j}{\sigma_j^2+\kappa}\Big)V^T \\ &= \sigma^2 V diag\Big(\frac{\sigma_j^2}{(\sigma_j^2+\kappa)^2}\Big)V^T \end{aligned}$$

□

When we face multicollinearity problem, there exist j such that

$$\sigma_j \approx 0$$

It makes

$$Var\hat{\boldsymbol{\beta}}_A \to \infty$$

i.e. very unstable. However, $Var\hat{\boldsymbol{\beta}}_{A,R}$ has $\kappa > 0$ in denominator. Denote that

$$\frac{\sigma_j^2}{(\sigma_j^2+\kappa)^2} \to 0 \quad \text{as} \quad \sigma_j \to 0$$

Hence, multicollinearity does not make variance large. It has small variance in the cost of bias.

Theorem 4.16 (Total variance)**.** *Let $\kappa > 0$. Then total variance of ridge regression is always smaller than of OLS.*

$$\sum_{j=1}^{p} Var(\hat{\beta}_{j,R}) = \sigma^2 \sum_{j=1}^{p} \frac{\sigma_j^2}{(\sigma_j^2+\kappa)^2} < \sum_{j=1}^{p} Var(\hat{\beta}_j) = \sigma^2 \sum_{j=1}^{p} \frac{1}{\sigma_j^2}$$

Proof. Note that

$$\sum_j Var(\hat{\beta}_j) = tr\Big(Var\hat{\boldsymbol{\beta}}_A\Big)$$

Then

$$\begin{aligned}\sum_j Var(\hat{\beta}_j) &= \sigma^2 tr\bigg(Vdiag\bigg(\frac{1}{\sigma_j^2}\bigg)V^T\bigg)\\ &= \sigma^2 tr\bigg(V^T Vdiag\bigg(\frac{1}{\sigma_j^2}\bigg)\bigg)\\ &= \sigma^2 tr\bigg(diag\bigg(\frac{1}{\sigma_j^2}\bigg)\bigg)\\ &= \sigma^2 \sum_j \frac{1}{\sigma_j^2}\end{aligned} \tag{4.28}$$

One proceeds in a similar way for ridge estimator that

$$\begin{aligned}\sum_j Var(\hat{\beta}_{j,R}) &= tr\Big(Var\hat{\boldsymbol{\beta}}_{A,R}\Big)\\ &= \sigma^2 tr\bigg(Vdiag\bigg(\frac{\sigma_j^2}{(\sigma_j^2+\kappa)^2}\bigg)V^T\bigg)\\ &= \sigma^2 tr\bigg(diag\bigg(\frac{\sigma_j^2}{(\sigma_j^2+\kappa)^2}\bigg)\bigg)\\ &= \sigma^2 \sum_j \frac{\sigma_j^2}{(\sigma_j^2+\kappa)^2}\end{aligned} \tag{4.29}$$

Since $\kappa > 0$,

$$\frac{1}{\sigma_j^2} > \frac{\sigma_j^2}{(\sigma_j^2+\kappa)^2}$$

for all κ.

Hence,

$$\sum_j Var(\hat{\beta}_j) > \sum_j Var(\hat{\beta}_{j,R})$$

□

4.4.4 Ivanov regularization

Refer to L_2 penalization (4.23).

$$\hat{\boldsymbol{\beta}}_{A,R} = \underset{\boldsymbol{\beta}_A}{\operatorname{argmin}} \left\{ \|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A \boldsymbol{\beta}_A\|^2 + \kappa \|\boldsymbol{\beta}_A\|^2 \right\} \tag{4.30}$$

This form of regularization is called *Tikhonov regularization*. Recall that this constraint has shrinked the coefficients. There is another regularization that directly gives constraint on the size of the coefficients.

Remark (Ivavnov regularization). An equivalent way to write the ridge problem is

$$\hat{\boldsymbol{\beta}}_{A,R} = \underset{\boldsymbol{\beta}_A}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A \boldsymbol{\beta}_A\|^2 \quad \text{subject to } \|\boldsymbol{\beta}_A\|^2 \leq d \tag{4.31}$$

which makes *explicit the size constraint* on the parameters.

Following the definitions of Wade (2017), we first presents the *lagrangian multipliers* to show two regularizations are equivalent.

Lemma 4.3 (Lagrange Multipliers)**.** *Let $n > p$, V be open in $\mathbb{R}^n$, and $f, g_j : V \to \mathbb{R}$ be $\mathcal{C}^1$ on V for $j = 1, \ldots, p$. Suppose that*

$$\exists\, \mathbf{a} \in V : \quad \frac{\partial(g_1, \ldots, g_p)}{\partial(x_1, \ldots, x_p)}(\mathbf{a}) \neq \mathbf{0} \tag{4.32}$$

If $f(\mathbf{a})$ is a local extremum of f subject to the constraints

$$g_k(\mathbf{a}) = 0, \quad k = 1, \ldots, p$$

then

$$\exists \text{ scalars } \lambda_1, \ldots, \lambda_p : \quad \nabla f(\mathbf{a}) + \sum_{j=1}^{p} \lambda_j \nabla g_j(\mathbf{a}) = \mathbf{0}$$

Definition 4.9 (Nested parameter space)**.** Ivanov regularization is constraint by nested parameter space defined by

$$\begin{aligned} \Omega_d &:= \overline{B_d(\mathbf{0})} \\ &= \{\boldsymbol{\beta}_A \in \mathbb{R}^p : \|\boldsymbol{\beta}_A\|^2 \leq d\} \end{aligned}$$

Remark. If $d_1 < d_2 < \cdots < \infty$, then

$$\Omega_{d_1} \subset \Omega_{d_2} \subset \cdots \subset \Omega_\infty$$

Equivalence between Ivanov regularization and Tikhonov regularization. Consider Ivanov regularization.

$$\begin{aligned}\hat{\boldsymbol{\beta}}_{A,R} &= \underset{\boldsymbol{\beta}_A \in \mathbb{R}^p}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A \boldsymbol{\beta}_A\|^2 \quad \text{subject to } \|\boldsymbol{\beta}_A\|^2 \leq d \\ &= \underset{\boldsymbol{\beta}_A \in \Omega_d}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A \boldsymbol{\beta}_A\|^2\end{aligned}$$

If $\boldsymbol{\beta}_{A,d_1} \in \Omega_d$ and $\boldsymbol{\beta}_{A,d_2}$ with $d_1 < d_2$, then

$$\|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A \boldsymbol{\beta}_{A,d_1}\|^2 \leq \|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A \boldsymbol{\beta}_{A,d_2}\|^2$$

by construction. It follows that

$$\begin{aligned}\hat{\boldsymbol{\beta}}_{A,R} &= \underset{\boldsymbol{\beta}_A \in \partial\Omega_d}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A \boldsymbol{\beta}_A\|^2 \\ &= \underset{\boldsymbol{\beta}_A \in \mathbb{R}^p}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A \boldsymbol{\beta}_A\|^2 \quad \text{subject to } \|\boldsymbol{\beta}_A\|^2 = d\end{aligned} \tag{4.33}$$

We now have minimization problem with constraint

$$\|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A \boldsymbol{\beta}_A\|^2 \quad \text{subject to } \|\boldsymbol{\beta}_A\|^2 - d = 0$$

From Lagrangian multipliers 4.3,

$$\exists\, k \in \mathbb{R} : \quad \frac{\partial}{\partial \boldsymbol{\beta}_A} \|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A \boldsymbol{\beta}_A\|^2 + k \frac{\partial}{\partial \boldsymbol{\beta}_A} (\|\boldsymbol{\beta}_A\|^2 - d) = \mathbf{0}$$

Then

$$-2\widetilde{\mathbb{X}}_A^T \mathbf{Y} + 2\widetilde{\mathbb{X}}_A^T \widetilde{\mathbb{X}}_A \boldsymbol{\beta}_A + 2k\boldsymbol{\beta}_A = \mathbf{0}$$

and so

$$(\widetilde{\mathbb{X}}_A^T \widetilde{\mathbb{X}}_A + kI)\hat{\boldsymbol{\beta}}_{A,R} = \widetilde{\mathbb{X}}_A^T \mathbf{Y}$$

which is same as normal equation dervied from Tikhonov regularization. This completes the proof. □

Corollary 4.4. *κ of Tikhonov regualrization and d of Ivanov regularization have ono-to-one correspondence.*

Proof. From our previous result, we have

$$\begin{aligned}
\hat{\boldsymbol{\beta}}_{A,R} &= \underset{\boldsymbol{\beta}_A \in \mathbb{R}^p}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A \boldsymbol{\beta}_A\|^2 \quad \text{subject to } \|\boldsymbol{\beta}_A\|^2 \leq d \\
&= \underset{\boldsymbol{\beta}_A \in \mathbb{R}^p}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A \boldsymbol{\beta}_A\|^2 \quad \text{subject to } \|\boldsymbol{\beta}_A\|^2 = d \\
&= \underset{\boldsymbol{\beta}_A \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ \|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A \boldsymbol{\beta}_A\|^2 + \kappa \|\boldsymbol{\beta}_A\|^2 \right\} \\
&= (\widetilde{\mathbb{X}}_A^T \widetilde{\mathbb{X}}_A + \kappa I)^{-1} \widetilde{\mathbb{X}}_A^T \mathbf{Y} \\
&= V diag\left(\frac{\sigma_j}{\sigma_j^2 + \kappa} \right) U^T \mathbf{Y}
\end{aligned}$$

Then

$$\|\hat{\boldsymbol{\beta}}_{A,R}\|^2 = \mathbf{Y}^T U diag\left(\frac{\sigma_j^2}{(\sigma_j^2 + \kappa)^2} \right) U^T \mathbf{Y} = d \tag{4.34}$$

Suppose that $\kappa_1 \neq \kappa_2$ corresponding to each d_1 and d_2. Then

$$\begin{aligned}
d_1 - d_2 &= \mathbf{Y}^T U diag\left(\frac{\sigma_j^2}{(\sigma_j^2 + \kappa_1)^2} \right) U^T \mathbf{Y} - \mathbf{Y}^T U diag\left(\frac{\sigma_j^2}{(\sigma_j^2 + \kappa_2)^2} \right) U^T \mathbf{Y} \\
&= \mathbf{Y}^T U diag\left(\frac{\sigma_j^2}{(\sigma_j^2 + \kappa_1)^2} - \frac{\sigma_j^2}{(\sigma_j^2 + \kappa_2)^2} \right) U^T \mathbf{Y} \\
&= \mathbf{Y}^T U D diag\left(\frac{1}{(\sigma_j^2 + \kappa_1)^2} - \frac{1}{(\sigma_j^2 + \kappa_2)^2} \right) D U^T \mathbf{Y}
\end{aligned}$$

Since

$$DU^T \mathbf{Y} = \begin{bmatrix} \sigma_1 \mathbf{u}_1^T \mathbf{Y} & \cdots & \sigma_p \mathbf{u}_p^T \mathbf{Y} \end{bmatrix}$$

and

$$\sigma_j \mathbf{u}_j = \mathbf{z}_j \quad : j\text{-th principal component score}$$

$$d_1 - d_2 = \sum_{j=1}^{p} \left\{ \frac{1}{(\sigma_j^2 + \kappa_1)^2} - \frac{1}{(\sigma_j^2 + \kappa_2)^2} \right\} (\mathbf{z}_j^T \mathbf{Y})^2$$

Assuming that

$$\hat{\alpha}_j = \mathbf{z}_j^T \mathbf{Y} \neq 0 \quad \text{for some } j$$

we have

$$d_1 - d_2 \neq 0$$

Hence, κ and d has one-to-one correspondence. □

Consider the decomposition of sums of squares.

$$\begin{aligned}
\|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A \boldsymbol{\beta}_A\|^2 &= (\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A \boldsymbol{\beta}_A)^T (\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A \boldsymbol{\beta}_A) \\
&= \left(\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A (\boldsymbol{\beta}_A - \hat{\boldsymbol{\beta}}_A) - \widetilde{\mathbb{X}}_A \hat{\boldsymbol{\beta}}_A\right)^T \left(\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A (\boldsymbol{\beta}_A - \hat{\boldsymbol{\beta}}_A) - \widetilde{\mathbb{X}}_A \hat{\boldsymbol{\beta}}_A\right) \\
&= \|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A \hat{\boldsymbol{\beta}}_A\|^2 + (\boldsymbol{\beta}_A - \hat{\boldsymbol{\beta}}_A)^T \widetilde{\mathbb{X}}_A^T \widetilde{\mathbb{X}}_A (\boldsymbol{\beta}_A - \hat{\boldsymbol{\beta}}_A) \\
&\quad - (\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A \hat{\boldsymbol{\beta}}_A)^T \widetilde{\mathbb{X}}_A (\boldsymbol{\beta}_A - \hat{\boldsymbol{\beta}}_A) - \widetilde{\mathbb{X}}_A^T (\boldsymbol{\beta}_A - \hat{\boldsymbol{\beta}}_A)^T (\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A \hat{\boldsymbol{\beta}}_A) \\
&= \|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A \hat{\boldsymbol{\beta}}_A\|^2 + (\boldsymbol{\beta}_A - \hat{\boldsymbol{\beta}}_A)^T \widetilde{\mathbb{X}}_A^T \widetilde{\mathbb{X}}_A (\boldsymbol{\beta}_A - \hat{\boldsymbol{\beta}}_A)
\end{aligned}$$

Write the function of error by

$$f(\boldsymbol{\beta}_A) := (\boldsymbol{\beta}_A - \hat{\boldsymbol{\beta}}_A)^T \widetilde{\mathbb{X}}_A^T \widetilde{\mathbb{X}}_A (\boldsymbol{\beta}_A - \hat{\boldsymbol{\beta}}_A) \tag{4.35}$$

Denote that ridge regression gives a constraint to this error such that

$$\|\boldsymbol{\beta}_A\|^2 \leq d$$

As in Definition 4.9, write the constraint region by

$$\Omega_d = \{\boldsymbol{\beta}_A : \|\boldsymbol{\beta}_A\|^2 \leq d\}$$

By construction, this is a *closed ball*.

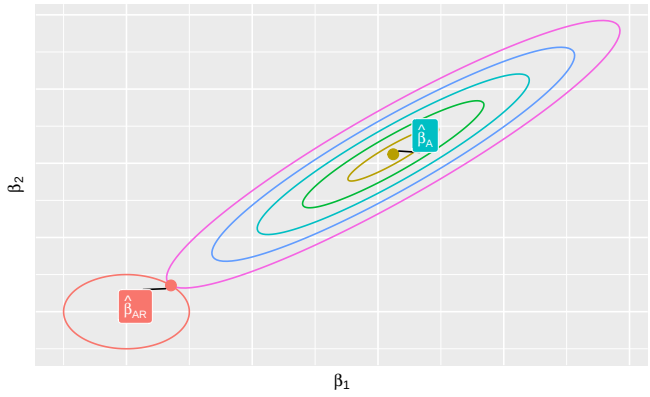


Figure 4.12: Contours of the error and constraint function for ridge regression

See Figure 4.12. In two dimensional situation, the constraint space, i.e.

$$\Omega_d = \{\boldsymbol{\beta}_A : \beta_1^2 + \beta_2^2 \leq d\}$$

the closed ball Ω_d becomes a circle. It can be shown that OLS $\hat{\boldsymbol{\beta}}_A$ is outside of Ω_d, i.e.

$$\|\hat{\boldsymbol{\beta}}_A\|^2 > d$$

Equation (4.32) implies that $\hat{\boldsymbol{\beta}}_{A,R}$ is the minimizer of

$$\|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A\boldsymbol{\beta}_A\|^2 + k(\|\boldsymbol{\beta}_A\|^2 - d)$$

Then we have

$$\begin{aligned}
\|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A\boldsymbol{\beta}_A\|^2 &\geq \|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A\boldsymbol{\beta}_A\|^2 + k(\underbrace{\|\boldsymbol{\beta}_A\|^2 - d}_{<0}) \\
&\geq \|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A\hat{\boldsymbol{\beta}}_{A,R}\|^2 + k(\|\hat{\boldsymbol{\beta}}_{A,R}\|^2 - d) \\
&= \|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A\hat{\boldsymbol{\beta}}_{A,R}\|^2
\end{aligned}$$

Now *claim that* $\|\hat{\boldsymbol{\beta}}_A\|^2 > d$. Hence from Equation (4.34),

$$
\begin{aligned}
d &= \|\hat{\boldsymbol{\beta}}_{A,R}\|^2 \\
&= \mathbf{Y}^T U diag\left(\frac{\sigma_j^2}{(\sigma_j^2+\kappa)^2}\right) U^T \mathbf{Y} \\
&= \sum_{j=1}^{p} \left\{\frac{\sigma_j^2}{(\sigma_j^2+\kappa)^2}\right\} (\mathbf{u}_j^T \mathbf{Y})^2 \\
&< \sum_{j=1}^{p} \left\{\frac{1}{\sigma_j^2}\right\} (\mathbf{u}_j^T \mathbf{Y})^2 \quad \leftarrow \text{if } \sigma_j \neq 0 \\
&= \mathbf{Y}^T U diag\left(\frac{1}{\sigma_j^2}\right) U^T \mathbf{Y} \\
&= \mathbf{Y}^T U D^{-1} V^T V D^{-1} U^T \mathbf{Y} \\
&= \hat{\boldsymbol{\beta}}_A^T \hat{\boldsymbol{\beta}}_A \\
&= \|\hat{\boldsymbol{\beta}}_A\|^2
\end{aligned}
$$

Figure 4.12 presents these relationship of OLS and ridge estimator. OLS is optimized outside of the closed ball, ridge constraint. The contours are of OLS error $f(\boldsymbol{\beta}_A)$ (4.35). These are contours of same RSS values. $\hat{\boldsymbol{\beta}}_A$ makes it the smallest and it locate in the smallest RSS. $\hat{\boldsymbol{\beta}}_{A,R}$, however, must be in Ω_d, while optimizing $f(\boldsymbol{\beta}_A)$ as possible as it can. As a result, $\hat{\boldsymbol{\beta}}_{A,R}$ *minimizes the penalized RSS at the contact point of the circle and ellipse.*

Think about smaller and smaller d. Then we get smaller size of $\hat{\boldsymbol{\beta}}_{A,R}$. Thus, smaller value of d shrinks $\hat{\boldsymbol{\beta}}_{A,R}$ toward $\mathbf{0}$. Since it is larger than 0, it may not be exactly zero.

4.4.5 Ridge regression in R

In general, we use `glmnet` package to fit ridge regression. In fact this implements maximum likelihood, not least squares. `glmnet::glmnet()` can fit a penalization model called *elasticnet* which has both l_2 penalty and l_1 penalty (of LASSO). The penalty weight is controlled by `alpha = 1`. Here, we use ridge regression, so we should change it to `alpha = 0`. Furthermore, this function does not accept `data.frame` as input. We should give `matrix` to `x` and `y`. In `lambda`, κ can be specified.

```
eeo_mat <-
  eeo %>%
  scale() %>%
  as_tibble() %>%
  model.matrix(ACHV ~ .-1, data = .)
#-------------------------------------
eeo_ridge <- glmnet::glmnet(x = eeo_mat, y = eeo$ACHV, alpha = 0)
```

`coef(glmnet)` gives a sparse Matrix object `dgCMatrix` which contains ridge estimates for each `lambda`. Rows are variables.

```
coef(eeo_ridge) %>%
  Matrix::t() %>%
  as.matrix() %>%
  as_tibble() %>%
  add_column(s = eeo_ridge$lambda) %>%
  rename_all(.funs = list(~str_remove_all(., pattern = "\\(|\\)"))) %>% # (Intercept)
  select(-Intercept) %>% # Intercept = average of y
  gather(-s, key = "coeff", value = "b") %>%
  ggplot(aes(x = s, y = b, colour = coeff)) +
  geom_ref_line(h = 0) +
  geom_path() +
  scale_x_log10() +
  labs(
    colour = "Variables",
    x = expression(Log ~ kappa),
    y = expression(hat(beta)[AR])
  )
```

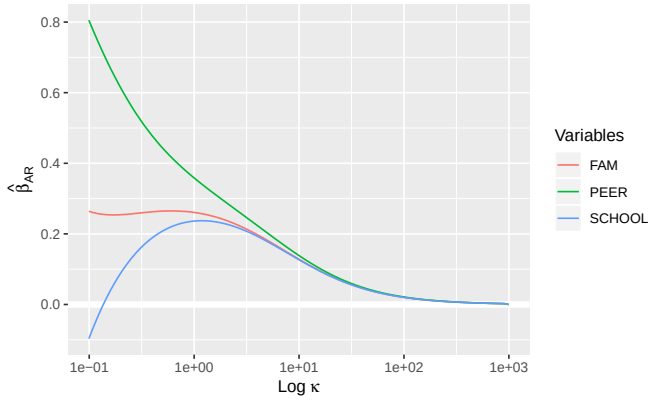


Figure 4.13: Ridge regression path along $\log \kappa$

Observe that each coefficient is shrinking as κ enlarging in Figure 4.13.

In the above code, I manually made coefficient matrix to draw a plot. Actually, `broom::tidy()` perform this all at once. If `return_zeros =TRUE` is added, zero estimates are also included in the output.

```
broom::tidy(eeo_ridge, return_zeros = TRUE)
#> # A tibble: 400 x 5
#>   term         step estimate lambda dev.ratio
```

```
#>   <chr>       <dbl>    <dbl>  <dbl>     <dbl>
#> 1 (Intercept)     1 1.92e- 2   992.  1.10e-36
#> 2 FAM             1 9.63e-37   992.  1.10e-36
#> 3 PEER            1 1.01e-36   992.  1.10e-36
#> 4 SCHOOL          1 9.60e-37   992.  1.10e-36
#> 5 (Intercept)     2 1.92e- 2   904.  2.69e- 3
#> 6 FAM             2 2.36e- 3   904.  2.69e- 3
#> # ... with 394 more rows
```

Chapter 5

Variable Selection

5.1 Motivation of Variable Selection

Large number of variables causes problem. Extremely, consider

$$n < p$$

Then we have

$$rank(X) \leq n$$

In this case, $(X^T X)^{-1}$ does not exist and OLS becomes to have no unique solution. This situation gives us

$$Var\hat{\boldsymbol{\beta}} \to \infty$$

5.1.1 Full model

Subsetting variables among moderate number of variables gives similar result. Assume that true regression model of no intercept with two covariates X_1 and X_2 for the data $\{(x_{i1}, x_{i2}, Y_i) : i = 1, \ldots, n\}$

$$Y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i \tag{5.1}$$

For simplicity, let the model satisfy that

$$\sum_i x_{i1} = \sum_i x_{i2} = 0 \quad \text{and} \quad \sum_i x_{i1}^2 = \sum_i x_{i2}^2 = 1$$

i.e. *inputs are centered* and $s_{11} = s_{22} = 1$. In the full model (5.1), LSE $\hat{\boldsymbol{\beta}}_F = (\hat{\beta}_{F,1}, \hat{\beta}_{F,2})^T$ is

$$\hat{\boldsymbol{\beta}}_F = (X^T X)^{-1} X^T \mathbf{Y}$$

with design matrix X of the model. Denote that by the condition,

$$X \perp \mathbf{1}$$

Equation (4.3) implies that for each $j = 1, 2$

$$\mathbf{x}_1^T \mathbf{x}_1 = 1 - R_j^2$$

and that

$$\begin{aligned} Var(\hat{\beta}_{F,j}) &= \frac{\sigma^2}{1 - R_j^2} \\ &= \frac{\sigma^2}{1 - r_{12}^2} \end{aligned} \tag{5.2}$$

with r_{12} is sample correlation coefficient. Furthermore, each coefficient estimate is unbiased.

$$E(\hat{\boldsymbol{\beta}}_F) = (X^T X)^{-1} X^T X \boldsymbol{\beta}_F = \boldsymbol{\beta}_F$$

5.1.2 Subset model

Now consider the subset model using only x_{i1}.

$$Y_i = \beta_1 x_{i1} + \epsilon_i \tag{5.3}$$

By solving normal equation, OLS gives estimate for β_1

$$\hat{\beta}_{S,1} = \sum_{i=1}^{n} x_{i1} Y_i$$

Recall that our true model is Full model (5.1). Then

$$\begin{aligned} E(\hat{\beta}_{S,1}) &= \sum_i x_{i1} E(Y_i) \\ &= \sum_i x_{i1}(\beta_1 x_{i1} + \beta_2 x_{i2}) \\ &= \beta_1 + r_{12}\beta_2 \quad \because \text{assumption for inputs} \\ &\neq \beta_1 \end{aligned} \tag{5.4}$$

i.e. $\hat{\beta}_{S,1}$ is *biased*. Additionally,

$$Var(\hat{\beta}_{S,1}) = \sum_i x_{i1}^2 \sigma^2 = \sigma^2 \tag{5.5}$$

5.1.3 Comparison

Now compare two different estimators of β_1 from the full and the subset models. Compute *mean squared error* (MSE). The following lemma about variance and bias can be used here.

Lemma 5.1 (Bias-variance trade-off)**.** *For any estimator $\hat{\beta}$ for β,*

$$\begin{aligned} MSE(\hat{\beta}) &= \left(E\hat{\beta} - \beta\right)^2 + E(\hat{\beta} - E\hat{\beta})^2 \\ &= bias(\hat{\beta})^2 + Var(\hat{\beta}) \end{aligned}$$

Proof. Plus and minus $E\hat{\beta}$.

$$\begin{aligned} MSE(\hat{\beta}) &= E(\hat{\beta} - \beta)^2 \\ &= E(\hat{\beta} - E\hat{\beta} + E\hat{\beta} - \beta)^2 \\ &= E\Big[\underbrace{(\hat{\beta} - E\hat{\beta})^2}_{\text{r.v.}} + (E\hat{\beta} - \beta)^2 + 2\underbrace{(\hat{\beta} - E\hat{\beta})}_{\text{r.v.}}(E\hat{\beta} - \beta)\Big] \\ &= E(\hat{\beta} - E\hat{\beta})^2 + (E\hat{\beta} - \beta)^2 + 2(E\hat{\beta} - \beta)\underbrace{(E\hat{\beta} - E\hat{\beta})}_{=0} \\ &= Var(\hat{\beta}) + bias(\hat{\beta})^2 \end{aligned} \tag{5.6}$$

□

We now apply this lemma to each model. Note that $\hat{\beta}_{F,1}$ of the full model (5.1) is unbiased, i.e.

$$E\hat{\beta}_{F,1} = \beta_1$$

Then from Equation (5.2),

$$\begin{aligned} MSE(\hat{\beta}_{F,1}) &= Var(\hat{\beta}_{F,1}) \\ &= \frac{\sigma^2}{1 - r_{12}^2} \end{aligned} \tag{5.7}$$

On the other hand, $\hat{\beta}_{S,1}$ of the subset model (5.3) is biased. Then from Equations (5.4) and (5.5),

$$\begin{aligned} MSE(\hat{\beta}_{S,1}) &= Var(\hat{\beta}_{S,1}) + \left(E\hat{\beta}_{S,1} - \beta_1\right)^2 \\ &= \sigma^2 + (r_{12}\beta_2)^2 \end{aligned} \tag{5.8}$$

The less MSE is, the better. If

$$1 + \frac{|\beta_2|}{\sigma} < \frac{1}{\sqrt{1 - r_{12}^2}} \tag{5.9}$$

then the subset model will estimate more precisely. Note that

$$\frac{1}{\sqrt{1 - r_{12}^2}} > 1$$

Remark. We can divide in two case for β_2.

1. If $|\beta_2| = 0$, then the subset model is understood as the true model so that it is preferred.
2. Even if $\beta_2 \neq 0$, $r_{12}^2 \approx 1$ (multicollinearity) results in the superiority of the subset model over the full model.

Sometimes we try to remove some variables. If the β_js and σ^2 were known, deletion of variables with small $|\beta_j|$ compared to σ, i.e. $\frac{|\beta_j|}{\sigma}$ would be desirable.

5.2 Criteria for Selecting Subsets

From now on, we subset variables using various criteria. This requires procedures deciding if subset is better than another, and some criteria are implemented. This is often called **model selection**. These criteria helps to choose the optimal subset size.

A general approach is to implement assumed Gaussian likelihood (Hastie et al., 2013).

$$\text{likelihood} + \text{penalty for the number of parameters}$$

The following correspond to likelihood term.

Remark. Likelihood means training error based on normal distribution error.

- Coefficient of determination $R^2 := \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$
- Residual sum of squares $SSE := \sum(y_i - \hat{y}_i)^2$
- Deviance $-2l(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2)$

As the number of parameters grows, each error

- $1 - R^2$
- SSE
- $-2l(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2)$

becomes smaller, i.e. we can regard the model better than before. However, previous section has shown that this is not the case, so we add or multiply a penalty for the number.

The following is the example dataset from Chatterjee and Hadi (2015). The reference had studied relation between total mortality (`MORT`) and climate, socioeconomic, and pollution variables (other 15 predictors). Response is total age-adjusted mortality from all causes. There are quite many predictors.

```
(death <- bestglm::mcdonald %>% as_tibble())
#> # A tibble: 60 x 16
#>    PREC  JANT  JULT OVR65  POPN  EDUC  HOUS  DENS  NONW WWDRK  POOR    HC
#>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#> 1    36    27    71   8.1  3.34  11.4  81.5  3243   8.8  42.6  11.7    21
#> 2    35    23    72  11.1  3.14  11    78.8  4281   3.5  50.7  14.4     8
#> 3    44    29    74  10.4  3.21   9.8  81.6  4260   0.8  39.4  12.4     6
#> 4    47    45    79   6.5  3.41  11.1  77.5  3125  27.1  50.2  20.6    18
#> 5    43    35    77   7.6  3.44   9.6  84.6  6441  24.4  43.7  14.3    43
#> 6    53    45    80   7.7  3.45  10.2  66.8  3325  38.5  43.1  25.5    30
#> # ... with 54 more rows, and 4 more variables: NOX <dbl>, SOx <dbl>,
#> #   HUMID <dbl>, MORT <dbl>
```

For each predictor, see the below description. 60 observations indicate 60 SMSAs.

predictors	description
`PREC`	mean annual precipitation, in inches
`JANT`	mean January temperature, degrees Farenheit
`JULT`	mean July temperature, degrees Fareheit
`OVR65`	percent of population aged 65 or older than 65
`POPN`	population per household

predictors	description
EDUC	median school years completed
HOUS	percent of housing units that are sound
DENS	population per square mile
NONW	percent of non-white population
WWDRK	percent employment of white-collor job
POOR	percent of families with income less than $3000
HC	relative pollution potential of hydrocarbon
NOX	of nitric oxides
SOx	of sulphur dioxide
HUMID	percent relative humidity

```
(death_fit <- lm(MORT ~ ., data = death)) %>%
  summary()
#>
#> Call:
#> lm(formula = MORT ~ ., data = death)
#>
#> Residuals:
#>    Min     1Q Median     3Q    Max
#> -68.07 -18.02   0.91  19.22  86.96
#>
#> Coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  1.76e+03   4.37e+02    4.03  0.00022 ***
#> PREC         1.91e+00   9.24e-01    2.06  0.04507 *
#> JANT        -1.94e+00   1.11e+00   -1.75  0.08741 .
#> JULT        -3.10e+00   1.90e+00   -1.63  0.11016
#> OVR65       -9.07e+00   8.49e+00   -1.07  0.29123
#> POPN        -1.07e+02   6.98e+01   -1.53  0.13295
#> EDUC        -1.72e+01   1.19e+01   -1.45  0.15508
#> HOUS        -6.51e-01   1.77e+00   -0.37  0.71439
#> DENS         3.60e-03   4.03e-03    0.89  0.37615
#> NONW         4.46e+00   1.33e+00    3.36  0.00162 **
#> WWDRK       -1.87e-01   1.66e+00   -0.11  0.91088
#> POOR        -1.68e-01   3.23e+00   -0.05  0.95881
#> HC          -6.72e-01   4.91e-01   -1.37  0.17799
#> NOX          1.34e+00   1.01e+00    1.33  0.18951
#> SOx          8.62e-02   1.48e-01    0.58  0.56175
#> HUMID        1.07e-01   1.17e+00    0.09  0.92764
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 34.9 on 44 degrees of freedom
```

```
#> Multiple R-squared:  0.765,  Adjusted R-squared:  0.685
#> F-statistic: 9.54 on 15 and 44 DF,  p-value: 2.19e-09
```

As we can see, some standard errors are very large.

```
car::vif(death_fit)
#>   PREC   JANT   JULT  OVR65   POPN   EDUC   HOUS   DENS   NONW  WWDRK
#>   4.11   6.14   3.97   7.47   4.31   4.86   3.99   1.66   6.78   2.84
#>   POOR     HC    NOX    SOx  HUMID
#>   8.72  98.64 104.98   4.23   1.91
```

We can observe multicollinearity by $VIF_j > 1$.

5.2.1 Adjusted R^2

First consider $R^2 = \frac{SSR}{SST}$. Denote that this is *nondecreasing as a new predictor enters the model*. It it just matter of how it increases.

We want to select a subset S of the index set $\{1, \dots p\}$. Write $|S|$ as the cardinality of the set and set q by the number of regression coeffcients.

$$q := |S| + 1$$

Let $SSE(S)$ and $R^2(S)$ be residual sum of squares and coefficient of determination corresponding to the model Y regressed on $\{x_j : j \in S\}$ including intercept term. Recall that

$$R^2(S) = \frac{SSR(S)}{SST} = 1 - \frac{SSE(S)}{SST}$$

Just compute adjusted R^2 for this S using Definition 2.3.

Definition 5.1 (Adjusted R^2)**.** Let S be a subset of variables and let R^2 be corresponding R^2. Then the adjusted R^2 for S is

$$R_a^2(S) := 1 - \left(\frac{n-1}{n-q}\right)(1 - R^2(S)) = 1 - \frac{SSE/(n-q)}{SST/(n-1)}$$

Different with $R^2(S)$, $R_a^2(S)$ increases and decreases at some point as $|S|$ increases. The point that maximizes $R_a^2(S)$ can be said to be an optimal subset.

Conjecture 5.1 (Optimal number of variables w.r.t. $R_a^2(S)$)**.** *Choose S that maximizes $R_a^2(S)$.*

5.2.2 Residual mean square

Residual mean square $MSE(S)$ is related to $R_a^2(S)$.

Definition 5.2 (Residual mean square)**.** Let S be a subset of variables and let $q = |S| + 1$ be the degrees of freedom of $SSE(S)$. Then the residual mean square according to S is

$$MSE(S) := \frac{SSE(S)}{n - q}$$

By construction, $R_a^2(S)$ can be re-expressed by $MSE(S)$.

Remark. From Definition 5.2,

$$R_a^2(S) = 1 - \frac{MSE(S)}{SST/(n-1)}$$

Since $\frac{SST}{n-1}$ is constant, maximimizing $R_a^2(S)$ as in Conjecture 5.1 is equivalent to minimizing $MSE(S)$.

Conjecture 5.2 (Optimal number of variables w.r.t. $MSE(S)$)**.** *Choose S that minimizes $MSE(S)$.*

5.2.3 Mallow's C_p

Previously, we multiplied penalty. From now on, we add it. Consider $SSE(S)$.

Definition 5.3 (Mallow's C_p)**.** Let $F = \{1, \ldots, p\}$ be a full set of variables, let S be a subset of variables, and let $q = |S| + 1$. Then the Mallow's C_p for S is

$$C_p(S) := \frac{SSE(S)}{\hat{\sigma}^2} + (2q - n)$$

where $\hat{\sigma}^2 = MSE(F) = \frac{SSE(F)}{n-p-1}$, i.e. MSE obtained from the full model.

$\frac{SSE(S)}{\hat{\sigma}^2}$ measures $SSE(S)$ compared to MSE of full model. When the model becomes closed to the full model, $SSE(S)$ gets smaller and goes to $SSE(F)$. In the full model,

$$\frac{SSE(F)}{\hat{\sigma}^2} = n - p - 1$$

Since $q = p + 1$ in this model, we have

$$C_p(F) = n - p - 1 + 2(p + 1) - n = p + 1$$

i.e. same value as q. In fact, this occurs when the model is unbiased.

Remark. Suppose that the fitted values of subset model are unbiased. Then

$$E(C_p(S)) = q$$

Proof. From Lemma 5.1, mean squared error can be decomposed into bias and variance.

$$\sum_i E(\hat{Y}_{S,i} - \mu_{S,i}) = \sum_i \left(E\hat{Y}_{S,i} - \mu_{S,i}\right)^2 + \sum_i E(\hat{Y}_{S,i} - E\hat{Y}_{S,i})^2$$

Note that the left hand side is constant. If there is no bias in the model, we have

$$MSE(F) = MSE(S)$$

Thus,

$$C_p = \frac{(n-q)MSE(S)}{MSE(F)} + 2q - n = q$$

□

This remark implies that the good model has small value of

$$C_p(S) \approx q$$

We need to find this model.

Conjecture 5.3 (Optimal number of variables w.r.t. C_p)**.** *Choose S with the smallest $q = |S| + 1$ that has $C_p(S) \approx q$.*

5.2.4 Akaike Information criterion

Information criteria implements deviance -2log-likelihood. There are two popular criteria called **Akaike information criterion (AIC)** and **Bayesian information criterion (BIC)**. Each adds penalty of form

$$-2l(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2) + \text{constant} \times q$$

AIC gives 2 as a penalty factor.

Definition 5.4 (Akaike information criterion)**.** Let S be a subset of variables and let $q = |S| + 1$ be the number of parameters. Then AIC for S is

$$AIC(S) := -\frac{2}{n} l(\hat{\boldsymbol{\beta}}_S, \hat{\sigma}^2_S) + \frac{2}{n} q$$

This definition is actually quite general formulation. In case of OLS, Gaussian error term is given. Since the form of $l(\hat{\boldsymbol{\beta}}_S, \hat{\sigma}^2)$ is known, we can compute it.

Corollary 5.1 (AIC for OLS with Gaussian error)**.** *Suppose that* $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$*. Then*

$$AIC(S) = \ln SSE(S) + \frac{2|S|}{n}$$

Proof. Let

$$\hat{\sigma}_S^2 = \frac{1}{n}\|\mathbf{Y} - X\hat{\boldsymbol{\beta}}_S\|^2$$

be the MLE of σ^2. Then we now have

$$\begin{aligned} l(\hat{\boldsymbol{\beta}}_S, \hat{\sigma}_S^2) &= -\frac{n}{2}\ln(2\pi\hat{\sigma}_S^2) - \frac{1}{2\hat{\sigma}_S^2}\|\mathbf{Y} - X\hat{\boldsymbol{\beta}}_S\|^2 \\ &= -\frac{n}{2}\ln(2\pi) + \frac{n}{2}\ln n - \frac{n}{2}\ln SSE(S) - \frac{n}{2} \end{aligned}$$

Thus,

$$\begin{aligned} AIC(S) &= \ln SSE(S) + \ln(2\pi) - \ln n + 1 + \frac{2|S|}{n} + \frac{2}{n} \\ &\propto \ln SSE(S) + \frac{2|S|}{n} \quad \leftarrow \text{ignore constants} \end{aligned}$$

and so $AIC(S)$ is equivalent to

$$\ln SSE(S) + \frac{2|S|}{n}$$

□

SSE is involved in OLS, so smaller AIC is preferred.

Conjecture 5.4 (Optimal number of variables w.r.t. AIC)**.** *Choose* S *that minimizes* $AIC(S)$*.*

5.2.5 Bayesian information criterion

Instead of 2, Bayesian information criterion (BIC) uses $\ln n$.

Definition 5.5 (Bayesian information criterion)**.** Let S be a subset of variables and let $q = |S| + 1$ be the number of parameters. Then BIC for S is

$$BIC(S) := -\frac{2}{n} l(\hat{\boldsymbol{\beta}}_S, \hat{\sigma}_S^2) + \frac{\ln n}{n} q$$

One proceeds in a similar way for OLS likelihood.

Corollary 5.2 (BIC for OLS with Gaussian error)**.** *Suppose that* $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$. *Then*

$$BIC(S) = \ln SSE(S) + \frac{|S| \ln n}{n}$$

Of course it is the same form, so we might choose the small one.

Conjecture 5.5 (Optimal number of variables w.r.t. BIC)**.** *Choose* S *that minimizes* $BIC(S)$.

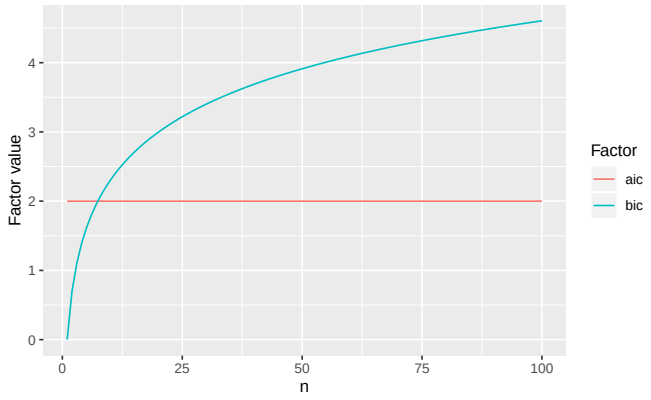


Figure 5.1: Penalties of AIC and BIC

Figure 5.1 presents the difference between AIC and BIC, 2 and $\ln n$. In the most of the domain,

$$\ln n > 2$$

In this case, BIC penalizes larger models more heavily than AIC. In turn, it prefers *smaller models in comparison with AIC*.

5.3 Computational Techniques

Using only criteria in the previous section, it is hard to find the best subset of the variables. We should know how to effectively use it, i.e. algorithm.

5.3.1 All possible regressions

The most basic way is investigating all possible models. Each submodel is

$$Y_i = \beta_0 + \sum_{j \in S} \beta_j x_{ij} + \epsilon_i, \quad S \subseteq \{1, \dots p\} \tag{5.10}$$

For example, if $p = 3$,

$\lvert S \rvert$	S	submodels
0	$\varnothing$	$Y = \beta_0 + \epsilon$
1	$\{1\}$	$Y = X_1 + \epsilon$
1	$\{2\}$	$Y = X_2 + \epsilon$
1	$\{3\}$	$Y = X_3 + \epsilon$
2	$\{1,2\}$	$Y = X_1 + X_2 + \epsilon$
2	$\{1,3\}$	$Y = X_1 + X_3 + \epsilon$
2	$\{2,3\}$	$Y = X_2 + X_3 + \epsilon$
3	$\{1,2,3\}$	$Y = X_1 + X_2 + X_3 + \epsilon$

For each submodel, compute one of R_a^2, C_p, AIC, and BIC. Submodel with the best value of criterion would be chosen. This requires fitting

$$\binom{p}{0} + \binom{p}{1} + \cdots \binom{p}{p} = 2^p$$

James et al. (2013) summarizes the procedure as follows.

Algorithm 5: All possible regressions
Data: Y_i and every predictor $x_{i1}, \dots x_{ip}$
1 Initialize null model $\mathcal{M}_0$ by

$$Y_i = \beta_0 + \epsilon_i$$

;
2 **for** $k \leftarrow 1$ **to** p **do**
3 Fit all $\binom{p}{k}$ models with k predictors

$$Y_i = \beta_0 + \sum_{j \in S} \beta_j x_{ij} + \epsilon_i$$

with $|S| = k$;
4 Denote $\mathcal{M}_k$ a model with the smallest SSE;
5 **end**
6 Select a single best among $\mathcal{M}_0, \mathcal{M}_1, \dots, \mathcal{M}_p$ using R_a^2, C_p, AIC, or BIC;
output: $\mathcal{M}_{k'}$ with the best criterion value

However, as the full number of variables p increases, the number of submodels to be conducted 2^p increases rapidly. It is computational burden. Against this kind of efficiency problem, some simpler methods are developed.

5.3.2 Forward selection

Beginning with the null model, forward selection *adds predictors one-at-a-time*. We might add the most important predictor at each step by a settled condition.

Table 5.3: Illustration of Forward selection

x1	x2	x3
0	0	0
p=1		
1	0	0
0	1	0
0	0	1
p=2		
2	1	0
0	1	2
p=3		
3	1	1

See Table 5.3. In each step, i.e. $p = k$, the most relavant variable is added to the model. It is remained to the end. Next, we find the most relavant variables given the previously chosen variables already in the model.

Here we need to decide some measures.

- that finds the most relavant variables among the remains
- that determines stopping rules of the algorithm

The algorithm starts with null model, which has no predictors. Set

$$S_0 = \varnothing \quad \text{and} \quad C_0 = Col(\mathbf{1}) = \{\beta_0 \mathbf{1} : \beta_0 \in \mathbb{R}\}$$

Next for each step $k = 1, 2, \ldots$, let C_{k-1} be the design space in the previous step. We now add to the given model the predictor that has *the highest size of sample partial correlation* with the response. Recall that from Theorem 2.11, sample correlation between $\mathbf{x}_j$ and $\mathbf{Y}$ is same as cosine of the angle between the two centered vectors, i.e.

$$r_j = \frac{(\mathbf{x}_j - \overline{x}_j \mathbf{1})^T (\mathbf{Y} - \overline{Y}\mathbf{1})}{\|\mathbf{x}_j - \overline{x}_j \mathbf{1}\| \|\mathbf{Y} - \overline{Y}\mathbf{1}\|}$$

In fact, each $\overline{x}_j \mathbf{1}$ and $\overline{Y}\mathbf{1}$ indicates $\Pi(\mathbf{x}_j \mid \mathbf{1}) = \Pi(\mathbf{x}_j \mid C_0)$ and $\Pi(\mathbf{Y} \mid \mathbf{1}) = \Pi(\mathbf{Y} \mid C_0)$. In step $k = 1$, we might find the variable that maximizes the size of correlation. However, in the other step, the former variables are already in the model. Variables would be compared with respect to partial correlation, not correlation. Thus, each $\mathbf{x}_j$ and $\mathbf{Y}$ is projected to C_{k-1}.

$$r_{k,j} = \frac{\Big(\mathbf{x}_j - \Pi(\mathbf{x}_j \mid C_{k-1})\Big)^T \Big(\mathbf{Y} - \Pi(\mathbf{Y} \mid C_{k-1})\Big)}{\|\mathbf{x}_j - \Pi(\mathbf{x}_j \mid C_{k-1})\| \|\mathbf{Y} - \Pi(\mathbf{Y} \mid C_{k-1})\|} \tag{5.11}$$

Find the variable with an index $j = j_k$ that maximizes $r_{k,j}^2$ and update the subset

$$S_k = S_{k-1} \cup \{j_k\}$$

and the column space

$$C_k = Col(\mathbf{1}, \mathbf{x}_l : l \in S_k)$$

This procedure continues until it stops by our rule. For instance,

1. $|S_k|$ exceeds a predetermined size p^*.
2. Partial F-statistic $F_{k,j}$ testing $H_0 : \beta_{j,k} = 0$ versus predetermined number F_{IN} called *F-to-enter*.

See the second one. From the chosen k-th submodel, saying $\mathcal{M}_k$, we can compute partial F-statistic.

$$F_{k,j} = \frac{SSR(X_j \mid X_l : l \in S_{k-1})}{SSE(X_l : l \in S_k)/(n-k-1)} < F_{IN} \tag{5.12}$$

Should we compute both $r_{k,j}$ and $F_{k,j}$ in each step?

Remark. $F_{k,j}$ is a monotonic function of $r^2_{k,j}$, that is,

$$F_{k,j} = (n-k-1)\frac{r^2_{k,j}}{1-r^2_{k,j}}$$

with $0 \le r^2_{k,j} \le 1$

Proof. Consider the model with S_k and S_{k-1}. It can be shown that

$$r^2_{k,j} = \frac{SSR(X_j \mid X_l : l \in S_{k-1})}{SSE(S_{k-1})} \tag{5.13}$$

It follows that

$$\begin{aligned} F_{k,j} &= \frac{SSR(X_j \mid X_l : l \in S_{k-1})}{SSE(S_k)/(n-k-1)} \\ &= (n-k-1)\frac{SSR(X_j \mid X_l : l \in S_{k-1})/SSE(S_{k-1})}{SSE(S_k)/SSE(S_{k-1})} \\ &= (n-k-1)\frac{r^2_{k,j}}{(SSE(S_{k-1}) - SSR(X_j \mid X_l : l \in S_{k-1}))/SSE(S_{k-1}))} \\ &= (n-k-1)\frac{r^2_{k,j}}{1-r^2_{k,j}} \end{aligned}$$

□

Thus, choosing j that maximizes $r^2_{k,j}$ is equivalent to that maximizes $F_{k,j}$. It also equivalent to choosing j maximizing the coefficient of determination R^2 when the variable X_j is added to the existing group S_{k-1}. The procedure is

enough to use only $F_{k,j}$ in each step.

Algorithm 6: Forward Selection
Data: Y_i and every predictor $x_{i1}, \dots x_{ip}$
input : F-to-enter F_{IN}
Initialize subset of variables $S_0 = \varnothing$;
Initialize design column space $C_0 = Col(\mathbf{1})$;
Initialize null model $\mathcal{M}_0$ by $Y_i = \beta_0 + \epsilon_i$;
for $k \leftarrow 1$ **to** p **do**
  For remained $(p - k + 1)$ predictors, compute partial F-statistic

$$F_{k,j} = \frac{SSR(X_j \mid X_l : l \in S_{k-1})}{SSE(X_l : l \in S_k)/(n-k-1)}$$

  ;
  $j_k = \text{argmax}_j \, F_{k,j}$;
  if $F_{k,j_k} \geq F_{IN}$ **then**
    Update the subset $S_k = S_{k-1} \cup \{j_k\}$;
    Update the column space $C_k = Col(\mathbf{1}, \mathbf{x}_l : l \in S_k)$;
  else
    Stop the procedure;
  end
end
output: $\mathcal{M}_k$ corresponding to selected variables S_k

Even if Algorithm 6 goes to the end p, the number of models reviewed is less than of all possible regressions.

$$\begin{aligned} \underset{\mathcal{M}_0}{1} + p + (p-1) + \cdots + (p-(p-1)+1) + (p-p+1) &= 1 + \sum_{k=1}^{p} p \\ &= 1 + \frac{p(p+1)}{2} \\ &\leq 2^p \end{aligned}$$

Thus, it has computational advantage over all possible regressions. Moreover, this method is even possible for high-dimensional setting $n < p$. Even though OLS cannot be computed in full model, forward selection procedure is able to go on until $n - 1$ variable-subset model.

Since this does not look at the whole models but *nested models*, it is not guaranteed to find the best model which comes from all possible regressions.

5.3.3 Backward elimination

Backward elimination is just the reverse of forward selection. Beginning with the full model, it iteratively removes the least relevant predictor one-at-a-time.

Table 5.4: Illustration of Backward elimination

x1	x2	x3
3	3	3
p=2		
2	3	3
3	2	3
3	3	2
p=1		
1	2	3
3	2	1
p=0		
0	2	1

Similarly, we use partial correlation $r_{k,j}$ or equivalent partial F-statistic $F_{k,j}$. Since we are looking for the least important one, we find the smallest value. We can remove the variable with the smallest changes in R^2. Stopping rules are also similar.

1. Stop when $|S_k|$ get to a predetermined size p^*.
2. Partial F-statistic $F_{k,j}$ is larger than or equal to predetermined F_{OUT} called *F-to-leave*.

We have stopped if $F_{k,j} < F_{IN}$ in forward selection 6. On the other hand,

backward selection stops if $F_{k,j} \geq F_{OUT}$.

Algorithm 7: Backward elimination
Data: Y_i and every predictor $x_{i1}, \dots x_{ip}$
input : F-to-leave F_{OUT}
Initialize subset of variables $S_0 = \varnothing$;
Initialize design column space $C_0 = Col(\mathbf{1})$;
Initialize null model $\mathcal{M}_0$ by $Y_i = \beta_0 + \epsilon_i$;
for $k \leftarrow 1$ **to** p **do**
    For non-removed $(p - k + 1)$ predictors, compute partial F-statistic

$$F_{k,j} = \frac{SSR(X_j \mid X_l : l \in S_{k-1})}{SSE(X_l : l \in S_k)/(n-k-1)}$$

    ;
    $j_k = \text{argmin}_j \, F_{k,j}$;
    if $F_{k,j_k} < F_{OUT}$ **then**
        Update the subset $S_k = S_{k-1} - \{j_k\}$;
        Update the column space $C_k = Col(\mathbf{1}, \mathbf{x}_l : l \in S_k)$;
    else
        Stop the procedure;
    end
end
output: $\mathcal{M}_k$ corresponding to selected variables S_k

As forward selection, backward elimination 7 investigates $1 + \frac{p(p+1)}{2}$ models. It has computational advantages, but does not guarantee the best model among every model. Since this procedure starts from the full model, it is not available when $n < p$, while forward selection is.

5.3.4 Stepwise regression

Since each forward and backward method has own problem, a hybrid approach has been made. Beginning with null model, add one predictor as in forward selection, and remove any predictor that is useless in that step after adding one

predictor.

Algorithm 8: Stepwise regression
Data: Y_i and every predictor $x_{i1}, \dots x_{ip}$
input : F-to-enter F_{IN} and F-to-leave F_{OUT}
Initialize subset of variables $S_0 = \varnothing$;
Initialize design column space $C_0 = Col(\mathbf{1})$;
Initialize null model $\mathcal{M}_0$ by $Y_i = \beta_0 + \epsilon_i$;
for $k \leftarrow 1$ **to** p **do**
    For each remained predictor, compute partial F-statistic

$$F_{k,j} = \frac{SSR(X_j \mid X_l : l \in S_{k-1})}{SSE(X_l : l \in S_k)/(n-k-1)}$$

    ;
    $j_k = \text{argmax}_j \, F_{k,j}$;
    if $F_{k,j_k} \geq F_{IN}$ **then**
        Compute partial F-statistic including the new variable

$$F^*_{k,i} = \frac{SSR(X_i \mid (X_l : l \in S_{k-1}, l \neq i), X_{j_k})}{SSE((X_l : l \in S_k), X_{jk})/(n-k-1)}$$

        ;
        Identify a set of $i \in S_{k-1}$ denoted by D_{k-1} s.t.

$$F^*_{k,i} < F_{OUT}$$

        ;
        Update the subset $S_k = S_{k-1} \cup \{j_k\} - D_{k-1}$;
        Update the column space $C_k = Col(\mathbf{1}, \mathbf{x}_l : l \in S_k)$;
    else
        Stop the procedure;
    end
end
output: $\mathcal{M}_k$ corresponding to selected variables S_k

In this process, adding and removal can happen simultaneously. This makes iteration more complex than forward or backward method.

5.3.5 Computational Techniques in R

Chapter 6

The LASSO

Consider centered model as in Equation (4.17) or ridge regression.

$$X\boldsymbol{\beta} = \mathbf{1}\beta_0^* + \widetilde{\mathbb{X}}_A\boldsymbol{\beta}_A \tag{6.1}$$

with $\widetilde{\mathbb{X}}_A = \mathbb{X}_{A,\perp}$ and $\boldsymbol{\beta} = (\beta_0, \boldsymbol{\beta}_A^T)^T$. Recall that $\hat{\beta}_0^* = \overline{Y}$.

6.1 LASSO Estimator

Least Absolute Shrinkage Selection Operator (LASSO) adds l_1 penalty to sum of squares in OLS problem.

Definition 6.1 (LASSO penalty)**.** Let $\beta_1, \ldots, \beta_p$ be the regression coefficients. Then ***LASSO penalty*** is l_1 norm compted by

$$\|\boldsymbol{\beta}_A\|_1 = \sum_{j=1}^{p} |\beta_j|$$

Then LASSO estimator $\hat{\boldsymbol{\beta}}_{A,L}$ becomes

$$\hat{\boldsymbol{\beta}}_{A,L} \equiv \hat{\boldsymbol{\beta}}_{A,L}(\lambda) = \underset{\boldsymbol{\beta}_A}{\operatorname{argmin}} \left\{ \|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A\boldsymbol{\beta}_A\|^2 + \lambda\|\boldsymbol{\beta}_A\|_1 \right\} \tag{6.2}$$

for some $\lambda > 0$. It is written as $\hat{\boldsymbol{\beta}}_{A,L}(\lambda)$ in that it changes along λ values. Obviously, $\lambda = 0$ produces OLS estimator. Choosing $\lambda = \infty$ would give $\hat{\boldsymbol{\beta}}_{A,L} = \mathbf{0}$, i.e. exact 0's. This is the difference with ridge regression.

LASSO estimator has equivalent Ivanov regularization.

Remark (Ivavnov regularization). An equivalent way to write the lasso problem is

$$\hat{\boldsymbol{\beta}}_{A,L} = \underset{\boldsymbol{\beta}_A}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A \boldsymbol{\beta}_A\|^2 \quad \text{subject to } \|\boldsymbol{\beta}_A\|_1 \leq d \tag{6.3}$$

where $d = \|\hat{\boldsymbol{\beta}}_{A,L}(\lambda)\|_1$.

As in ridge regression, OLS $\hat{\boldsymbol{\beta}}_A$ is out of the constraint region

$$\Omega_d = \{\boldsymbol{\beta}_A : \|\boldsymbol{\beta}_A\|_1 \leq d\}$$

i.e.

$$\|\hat{\boldsymbol{\beta}}_A\|_1 > \|\hat{\boldsymbol{\beta}}_{A,L}(\lambda)\|_1 = d$$

6.2 Geometry of LASSO

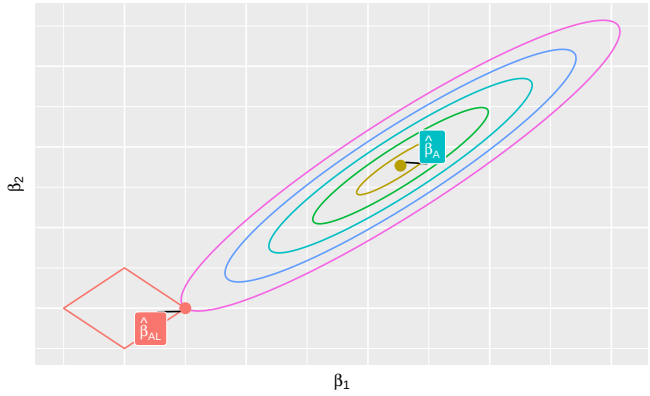


Figure 6.1: Contours of the error and constraint function for LASSO

See Figure 6.1. This shows the two properties of LASSO. *shrinkage and selection.* The principle is same. As λ becomes smaller, $\hat{\boldsymbol{\beta}}_{A,L}$ shrinks toward $\mathbf{0}$. However, the shape of Ω_d is different. Ridge regression with l_2 penalty does not meet exactly zero, while LASSO with l_1 penalty does. Moreover, as the number of variables is large, there are more sharp corners. So it is highly possible to meet $\hat{\beta}_{j,L} = 0$.

In this sense, *LASSO can perform both the model selection and estimation of the regression parameters in one step.*

6.3 LASSO for Orthogonal Design

l_1 Penalty 6.1 restricts the size of $\hat{\beta}_j$ like ridge regression. However, it is non-linear, so the estimator does not have a closed form. It requires numerical methods. In some special case, we can get an analytical solution.

Consider design matrix $\mathbb{X}_{A,\perp} = \widetilde{\mathbb{X}}_A$. When *every column of design matrix is orthogonal*, i.e.

$$\forall j \neq k: \quad \sum_{i=1}^{n}(x_{ij} - \overline{x}_j)(x_{ik} - \overline{x}_k) = 0$$

Hastie et al. (2013) shows that *the LASSO estimator has the explicit solution.*

Theorem 6.1 (Explicit solutions in orthogonal design)**.** *Suppose that the design matrix $\mathbb{X}_{A,\perp} = \widetilde{\mathbb{X}}_A$ has orthogonal columns. Set*

$$\kappa_j := \sum_{i=1}^{n}(x_{ij} - \overline{x}_j)^2$$

Then

Estimator	*Formula*
Best subset of size k	$\hat{\beta}_j I(\lvert\hat{\beta}_j\rvert \geq \lvert\hat{\beta}_{k,j}\rvert)$ *when orthonormal*
Ridge regression	$\frac{\kappa_j \hat{\beta}_j}{\kappa_j + \lambda}$
LASSO	$sgn(\hat{\beta}_j)\left(\lvert\hat{\beta}_j\rvert - \frac{\lambda}{2\kappa_j}\right)_+$

LASSO for orthogonal design. Write the length of each column vector by

$$\kappa_j := \mathbf{x}_{j,A,\perp}^T \mathbf{x}_{j,A,\perp} = \sum_{i=1}^{n}(x_{ij} - \overline{x}_j)^2$$

so that

$$\mathbb{X}_{A,\perp}^T \mathbb{X}_{A,\perp} = diag(\kappa_1, \ldots, \kappa_p)$$

Let $\hat{\boldsymbol{\beta}}_A = (\hat{\beta}_1, \ldots, \hat{\beta}_p)^T$ be the OLS. Recall that from Equation (4.35), the loss function for OLS becomes

$$
\begin{aligned}
f(\boldsymbol{\beta}_A) &= (\boldsymbol{\beta}_A - \hat{\boldsymbol{\beta}}_A)^T \mathbb{X}_{A,\perp}^T \mathbb{X}_{A,\perp} (\boldsymbol{\beta}_A - \hat{\boldsymbol{\beta}}_A) \\
&= (\boldsymbol{\beta}_A - \hat{\boldsymbol{\beta}}_A)^T diag(\kappa_1, \ldots, \kappa_p)(\boldsymbol{\beta}_A - \hat{\boldsymbol{\beta}}_A) \\
&= \sum_{j=1}^{p} \kappa_j (\beta_j - \hat{\beta}_j)^2
\end{aligned} \tag{6.4}
$$

It follows that LASSO objective function becomes

$$
\begin{aligned}
L(\boldsymbol{\beta}_A) &= \underbrace{\left(\|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A \hat{\boldsymbol{\beta}}_A\|^2 + f(\boldsymbol{\beta}_A) \right)}_{\text{LS criterion}} + \lambda \sum_{j=1}^{p} |\beta_j| \\
&= \underbrace{\|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A \hat{\boldsymbol{\beta}}_A\|^2}_{\text{constant term}} + \sum_{j=1}^{p} \kappa_j (\beta_j - \hat{\beta}_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \\
&= \|\mathbf{Y} - \mathbf{1}\overline{Y} - \widetilde{\mathbb{X}}_A \hat{\boldsymbol{\beta}}_A\|^2 + \sum_{j=1}^{p} \left(\kappa_j (\beta_j - \hat{\beta}_j)^2 + \lambda |\beta_j| \right) \\
&\propto \sum_{j=1}^{p} \left(\kappa_j (\beta_j - \hat{\beta}_j)^2 + \lambda |\beta_j| \right)
\end{aligned} \tag{6.5}
$$

Therefore the minimization of the above LASSO objective function (6.5) can be done by componentwise minimization of

$$
\kappa_j (\beta_j - \hat{\beta}_j)^2 + \lambda |\beta_j| \tag{6.6}
$$

What β_j does minimize the componentwise objective (6.6)? Note that

$$
\begin{aligned}
\kappa_j (\beta_j - \hat{\beta}_j)^2 + \lambda |\beta_j| &= \kappa_j \left((\beta_j - \hat{\beta}_j)^2 + \frac{\lambda}{\kappa_j} |\beta_j| \right) \\
&= \kappa_j \left[\hat{\beta}_j^2 - 2\hat{\beta}_j \beta_j + \beta_j^2 + \frac{\lambda}{\kappa_j} sgn(\beta_j)\beta_j \right] \quad \leftarrow |\beta_j| = sgn(\beta_j)\beta_j \\
&= \kappa_j \left[\beta_j^2 - 2\left(\hat{\beta}_j - \frac{\lambda}{2\kappa_j} sgn(\beta_j) \right) \beta_j + \hat{\beta}_j^2 \right] \\
&= \kappa_j \left[\left(\beta_j - \left(\hat{\beta}_j - \frac{\lambda}{2\kappa_j} sgn(\beta_j) \right) \right)^2 + \text{constant} \right] \\
&\propto \begin{cases} \left(\beta_j - \left(\hat{\beta}_j - \frac{\lambda}{2\kappa_j} \right) \right)^2 & \beta_j \geq 0 \\ \left(\beta_j - \left(\hat{\beta}_j + \frac{\lambda}{2\kappa_j} \right) \right)^2 & \beta_j < 0 \end{cases}
\end{aligned}
$$

Hence, the objective function is minimized at

$$\beta_j = \begin{cases} \hat{\beta}_j - sgn(\beta_j)\frac{\lambda}{2\kappa_j} & |\hat{\beta}_j| > \frac{\lambda}{2\kappa_j} \\ 0 & |\hat{\beta}_j| \leq \frac{\lambda}{2\kappa_j} \end{cases}$$

□

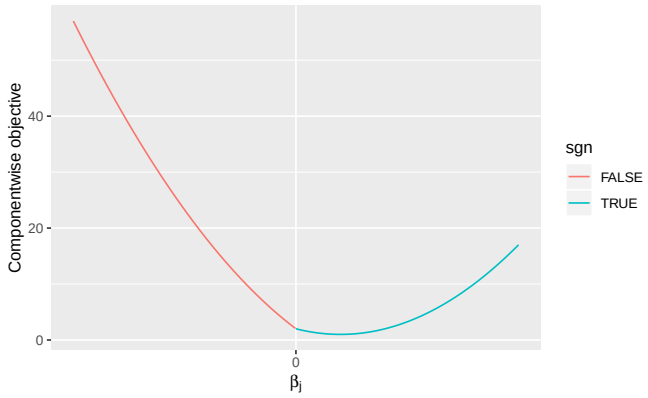


Figure 6.2: $\hat{\beta}_j > \frac{\lambda}{2\kappa_j}$

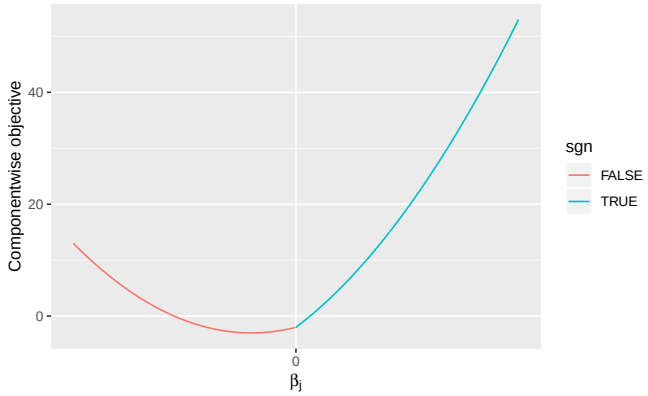


Figure 6.3: $\hat{\beta}_j < -\frac{\lambda}{2\kappa_j}$

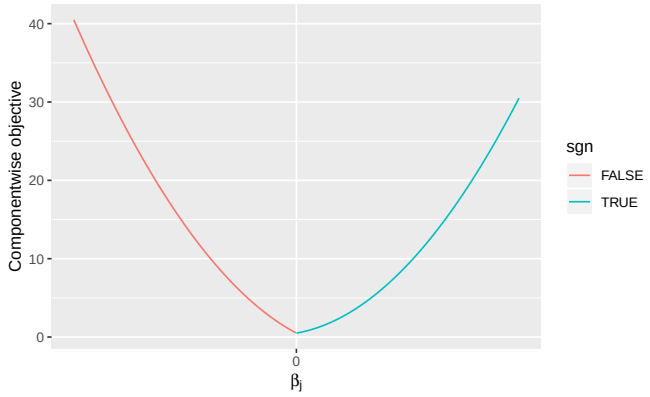


Figure 6.4: $-\frac{\lambda}{2\kappa_j} < \hat{\beta}_j < \frac{\lambda}{2\kappa_j}$

Look at the above three Figures 6.2 to 6.4. First two figures are the case when $|\hat{\beta}_j| > \frac{\lambda}{2\kappa_j}$. Minimum occurs at each $\hat{\beta}_j \pm \frac{\lambda}{2\kappa_j}$. The last figure is the case of $|\hat{\beta}_j| \leq \frac{\lambda}{2\kappa_j}$ so that it is minimized by $\beta_j = 0$.

Ridge regression for orthogonal design. Since ridge regression has analytical solution, we can prove its part of 6.1 more easily. Recall that

$$\hat{\boldsymbol{\beta}}_{A,R} = (\mathbb{X}_{A,\perp}^T \mathbb{X}_{A,\perp} + \lambda I)^{-1} \mathbb{X}_{A,\perp}^T \mathbf{Y}$$

Since $\mathbb{X}_{A,\perp}^T \mathbb{X}_{A,\perp} = diag(\kappa_1, \ldots, \kappa_p)$,

$$\hat{\boldsymbol{\beta}}_A = diag\left(\frac{1}{\kappa_j}\right) \mathbb{X}_{A,\perp}^T \mathbf{Y}$$

and

$$\hat{\boldsymbol{\beta}}_{A,R} = diag\left(\frac{1}{\kappa_j + \lambda}\right) \mathbb{X}_{A,\perp}^T \mathbf{Y}$$

Hence,

$$\hat{\beta}_{j,R} = \frac{\kappa_j \hat{\beta}_j}{\kappa_j + \lambda}$$

□

In sum, we can see the relationship between OLS and each shrinked estimator.

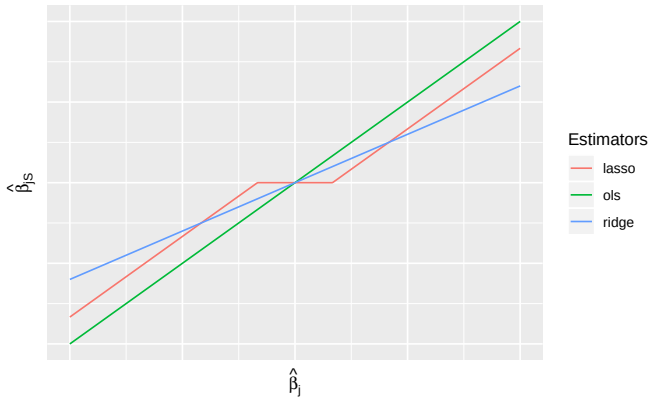


Figure 6.5: OLS and the other estimators

See Figure 6.5. Green line indicates OLS. As we can see, the other ridge estimator and lasso estimator is shrinking it. In case of ridge regression, it shrinks OLS proportionally by $\frac{\kappa_j}{\kappa_j+\lambda}$. This is why we cannot see exact zero. LASSO, on the other hand, moves the line up and down by $\frac{\lambda}{2\kappa_j}$. We can see entire zeros near $\hat{\beta}_j \approx 0$. LASSO is doing feature selection. This is called *soft-thresholding*.

On the other hand, the best subset regression conduct *hard-thresholding*. This is because model selection criteria are related to l_0 penalty

$$\|\boldsymbol{\beta}_A\|_0 = \sum_{j=1}^{p} I(\beta_j \neq 0)$$

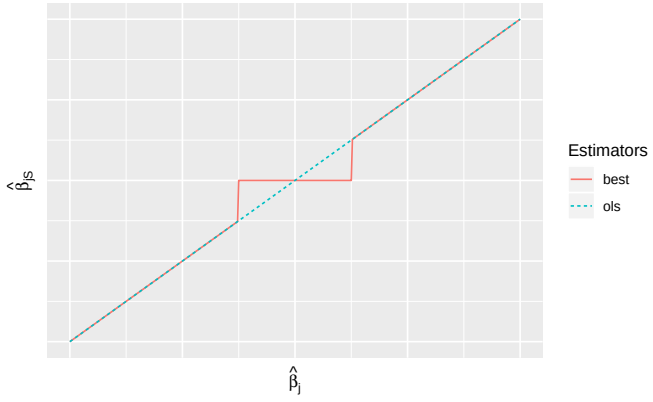


Figure 6.6: OLS and the best subset estimator

In Figure 6.6, we can see that the best subset regression is finding estimators larger than some threshold in orthonormal setting.

6.4 Numerical Methods

Contrary to ridge estimation, the LASSO estimator does not have a closed form except in the orthogonal design case. So we should get the solution numerically. However, most of the numerical methods give unstable one. Some algorithm have overcome this problem and is widely used, e.g. *LARS* and *glmnet*.

6.4.1 Least Angle Regression

The **Least Angle Regression (LAR)** algorithm (Hastie et al., 2013) give the complete set of the LASSO estimates for all $0 < \lambda < \infty$. In `R`, `lars::lars(x, y)` can perform this algorithm.

```
death_mat <-
  death %>%
  scale() %>%
  as_tibble() %>%
  model.matrix(MORT ~ .-1, data = .)
#-------------
death_lars <- lars::lars(x = death_mat, y = death$MORT, type = "lasso", normalize = TRU
```

There exists `plot` method, but we try `ggplot2`.

```
l1 <-
  apply(death_lars$beta, 1, function(x) {
    sum(abs(x))
  })
#-------------------
death_lars$beta %>%
  as_tibble() %>%
  mutate(
    l1 = l1,
    l1 = l1 / max(l1) # normalize
  ) %>%
  gather(-l1, key = "var", value = "value") %>%
  mutate(label = ifelse(l1 == 1, var, NA_character_)) %>%
  ggplot(aes(x = l1, y = value, colour = var)) +
  geom_vline(aes(xintercept = l1), col = "grey70", alpha = .5) +
  geom_point(shape = 4, alpha = .5) +
  geom_path() +
  ggrepel::geom_label_repel(
    aes(label = label),
    nudge_x = .1,
```

```
    na.rm = TRUE
  ) +
  labs(
    x = expression(l[1] / max(l[1])),
    y = expression(beta[jL])
  ) +
  theme(legend.position = "none")
```

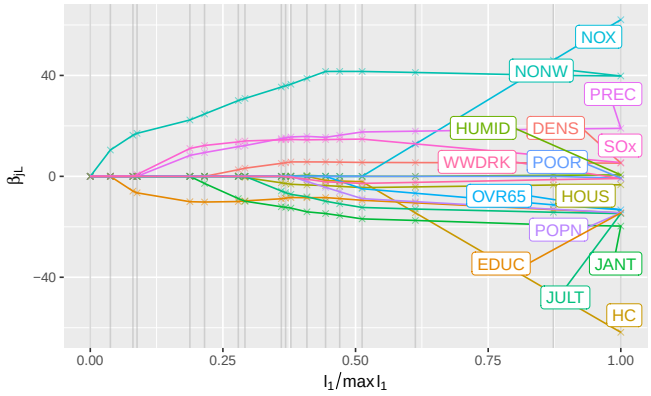


Figure 6.7: Shrinkage in LASSO fitted by LARS

6.4.2 glmnet

Actually, `glmnet::glmnet()` is more widely used in R. This algorithm enables to fit l_1 and l_2 penalty model very fast.

$$\lambda \sum_{j=1}^{p} \left((1-\alpha)\beta_j^2 + \alpha|\beta_j| \right)$$

with $\alpha \in [0, 1]$. Setting $\alpha = 1$ and $\alpha = 0$ each, LASSO and ridge regression can be solved.

```
death_lasso <- glmnet::glmnet(x = death_mat, y = death$MORT, alpha = 1)
```

As λ grows, each coefficient shrinks.

```
coef(death_lasso) %>%
  Matrix::t() %>%
  as.matrix() %>%
  as_tibble() %>%
  add_column(s = death_lasso$lambda) %>%
  rename_all(.funs = list(~str_remove_all(., pattern = "\\(|\\)"))) %>% # (Intercept) to Intercep
  select(-Intercept) %>% # Intercept = average of y
```

```
  gather(-s, key = "coeff", value = "b") %>%
  mutate(label = ifelse(s == min(s), coeff, NA_character_)) %>%
  ggplot(aes(x = s, y = b, colour = coeff)) +
  geom_ref_line(h = 0) +
  geom_path() +
  ggrepel::geom_label_repel(
    aes(label = label),
    nudge_x = -.1,
    na.rm = TRUE
  ) +
  scale_x_log10() +
  labs(
    x = expression(Log ~ lambda),
    y = expression(hat(beta)[AL])
  ) +
  theme(legend.position = "none")
```

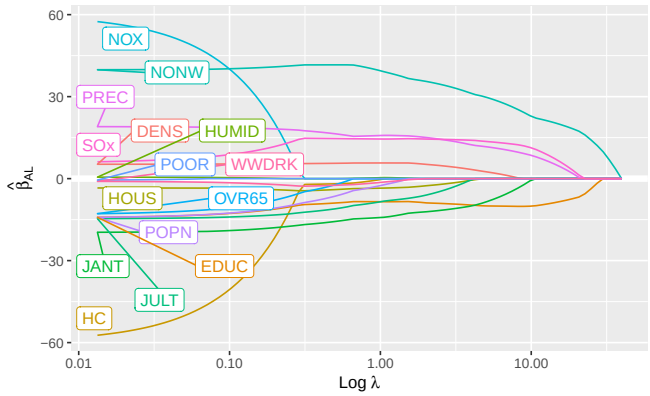


Figure 6.8: LASSO path along $\log \lambda$

6.4.3 LASSO for high-dimensional data

In ordinary linear regression problem, it is assumed that $n > p$. However, sometimes p becomes too large and even $n < p$. This is called **high-dimensionality**. This kind of problems occurs in the field of gene expression data and econometrics, et cetera.

If the number of variables, i.e. the number of columns is larger than the number of rows, then OLS method breaks down. LASSO is very useful here. It can give a solution, and also conduct variable selection. See Figure 6.8. It is considered as *continuous subset selection*.

Chapter 7

Further Issues in Parametric Regression

7.1 Non-linear Relationship

7.1.1 Tensile strength of kraft paper

The data set is an excerpt from Ekstrom and Sorensen (2014).

```
data(paperstr, package = "isdals")
(paperstr <- as_tibble(paperstr))
#> # A tibble: 19 x 2
#>   hardwood strength
#>      <dbl>    <dbl>
#> 1      1        6.3
#> 2      1.5     11.1
#> 3      2       20
#> 4      3       24
#> 5      4       26.1
#> 6      4.5     30
#> # ... with 13 more rows
```

- `hardwood`: amounts of hardwood contents in the paper pulp
- `strength`: tensile strength of kraft paper (in pound-force per square inch)

```
paperstr %>%
  ggplot(aes(x = hardwood, y = strength)) +
  geom_smooth(method = "lm") +
  geom_point()
```

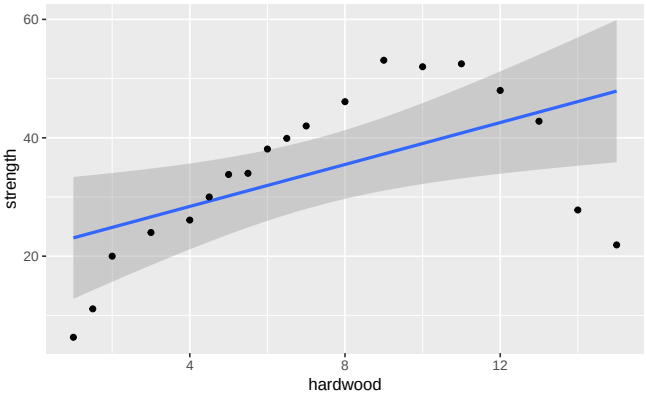


Figure 7.1: Tensile strength of kraft paper

In Figure 7.1, it seems that two variables have *non-linear relationship*. Linear regression model

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

might not explain the data well. Instead of linear model, we should find some other function f such that

$$Y_i = f(x_i) + \epsilon_i \tag{7.1}$$

7.1.2 Polynomial regression model

A real-valued function f is called *analytic* on (a, b) if and only if there exists $\{a_k\}_0^\infty$

$$f(x) = \sum_{k=0}^{\infty} a_k (x - x_0)^k$$

For more detailed definition, see Wade (2017). It can be shown that if $f \in C^\infty(a, b)$, then f is analytic and its coefficients are defined by Talyor expansion.

Theorem 7.1 (Taylor expansion)**.** *Let $f \in C^\infty(a, b)$. Suppose that*

$$\forall x \in (a, b) \quad \forall n \in \mathbb{N} \quad \exists M > 0 : |f^{(n)}| \leq M^n$$

Then f is analytic on (a, b). In fact, for each $x_0 \in (a, b)$,

$$f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k \tag{7.2}$$

for every $x \in (a, b)$.

This Theorem suggests a lot in regression context. Rewrite Equation (7.2).

$$\begin{aligned} f(x) &= E(Y \mid x) \\ &= \underbrace{\underbrace{f(x_0) + f^{(1)}(x_0)(x - x_0)}_{\text{linearity assumption}} + \frac{f^{(2)}(x_0)}{2!}(x - x_0)^2 + \frac{f^{(3)}(x_0)}{3!}(x - x_0)^3 + \cdots}_{\text{cubic model}} \end{aligned} \tag{7.3}$$

In the taylor expansion, we have used only first order part. It is often good enough and gives interpretability. However, as we can see in Figure 7.1, sometimes this kind of approximation cannot be made. In this case, we should add terms, i.e. *polynomial function*. In case of single variable, we can construct k-th order polynomial model as follows.

$$Y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_k x_i^k + \epsilon_i \tag{7.4}$$

If we have several variables, e.g. two, the second-order polynomial model can be constructed as follows.

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}^2 + \beta_{11} x_{i1}^2 + \beta_{22} x_{i2}^2 + \beta_{12} x_{i1} x_{i2} + \epsilon_i \tag{7.5}$$

As we can see, the model is built by *transformation* of the original predictors. It is called **linear basis expansion**. For examle,

```
(cubic_trans <-
  paperstr %>%
  transmute(
    x = hardwood,
    x2 = hardwood^2,
    x3 = hardwood^3
  ))
#> # A tibble: 19 x 3
#>       x    x2    x3
#>   <dbl> <dbl> <dbl>
#> 1   1    1     1
#> 2   1.5  2.25  3.38
#> 3   2    4     8
```

```
#> 4   3    9    27
#> 5   4   16    64
#> 6   4.5 20.2  91.1
#> # ... with 13 more rows
```

However, this kind of transformation carries multicollinearity problems.

```
cor(cubic_trans)
#>        x    x2    x3
#> x  1.000 0.970 0.921
#> x2 0.970 1.000 0.987
#> x3 0.921 0.987 1.000
```

Thus in general, we orthogonalize these bases. Refer to Gram-Schmidt process 2.5. This can be applied to any Hilbert space, for instance, L^2 space (Kreyszig, 2007).

Lemma 7.1 (Legendre polynomial)**.** *Let $L^2[-1,1]$ be the completion of the innter product space X of all continuous real-valued functions on $[-1,1]$ with inner product defined by*

$$< x, y >= \int_{-1}^{1} x(t)y(t)dt$$

Then there is a total orthonormal set $\{e_k\}$ in $L^2[-1,1]$. Furthermore,

$$e_n(t) = \sqrt{2n+12}P_n(t), \quad n = 0, 1, \ldots$$

where

$$P_n(t) = \frac{1}{2^n n!}\frac{d^n}{dt^n}(t^2-1)^n$$

By applying the binomial theorem to $(t^2-1)^n$, we obtain the result of differentiation.

Corollary 7.1. *$P_0, P_1, P_2 \ldots$ are orthogonal.*

$$P_n(t) = \sum_{j=0}^{N}(-1)^j \frac{(2n-2j)!}{2^n j!(n-2j)!}t^{n-2j}$$

$$\text{with} \begin{cases} N = \frac{n}{2} & n \text{ is even} \\ N = \frac{n-1}{2} & n \text{ is odd} \end{cases}$$

Similarly, we can apply G-S process to our expanded bases $\{1, x, x^2, x^3 \ldots\}$ using L_2 norm defined above.

$$
\begin{aligned}
P_0(x) &= 1 \\
P_1(x) &= x - \frac{\int x dx}{\int 1^2 dx} 1 \\
P_2(x) &= x^2 - \frac{\int x^2 dx}{\int 1^2 dx} 1 - \frac{\int x^2 P_1(x) dx}{\int P_1^2(x) dx} P_1(x) \\
P_3(x) &= x^3 - \frac{\int x^3 dx}{\int 1^2 dx} 1 - \frac{\int x^3 P_1(x) dx}{\int P_1^2(x) dx} P_1(x) - \frac{\int x^3 P_2(x) dx}{\int P_2^2(x) dx} P_2(x) \\
&\vdots
\end{aligned}
$$

and hence,

$$\int P_j(x) P_k(x) dx = 0 \quad \forall j \neq k \quad \Leftrightarrow P_j \perp P_k$$

In `R`, `poly()` function makes orthogonal polynomials with `degree` by default otherwise we specify `simple = TRUE`. If we provide this as predictor in `lm()`, polynomial fit is estimated.

```
(hardwood_fit <- lm(strength ~ poly(hardwood, degree = 2), data = paperstr))
#>
#> Call:
#> lm(formula = strength ~ poly(hardwood, degree = 2), data = paperstr)
#>
#> Coefficients:
#>                   (Intercept)  poly(hardwood, degree = 2)1
#>                          34.2                         32.3
#> poly(hardwood, degree = 2)2
#>                         -45.4
```

See the difference between Figure 7.1 and Figure 7.2. We can see how polynomial regression improve approximation in the eye.

```
paperstr %>%
  ggplot(aes(x = hardwood, y = strength)) +
  geom_smooth(formula = y ~ poly(x, degree = 2), method = "lm") +
  geom_point()
```

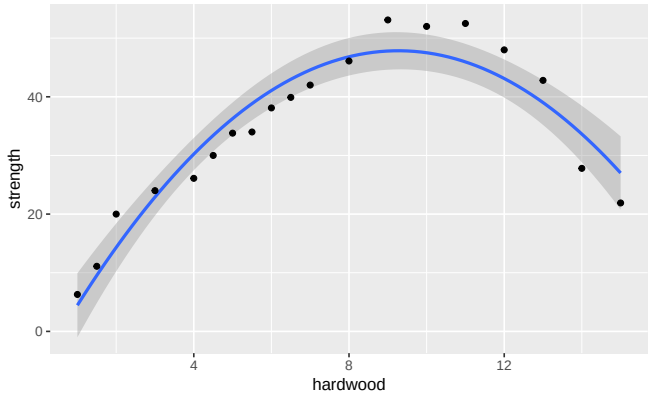


Figure 7.2: Quadratic regression to the hardwood data set

7.2 Variable Selection Issue

We have covered variable selection topic. This is important in that it affects stable estimation. It is related to the complexity of the model.

- Underfit: miss one of the important variables
- Correct fit: include all the necessary variables exactly
- Overfit: include all the necessary variables and some of unnecessary variables

Among the above three concept, we should find *correct fit*. If the model is too complex, the model overfit the data.

7.2.1 Simulated example

First consider true mode

$$Y = -2 + 3X - 2X^2$$

```
simul_quad <-
  tibble(
    x = runif(20, -2, 4),
    y = -2 + 3 * x - 2 * x^2 + rnorm(20, sd = 2)
  )
#-----------------------
underfit <- lm(y ~ x, data = simul_quad)
correctfit <- lm(y ~ poly(x, 2), data = simul_quad)
overfit <- lm(y ~ poly(x, 12), data = simul_quad)
simul_quad %>%
  gather_predictions(underfit, correctfit, overfit) %>%
```

```
  ggplot(aes(x = x)) +
  geom_line(aes(y = pred, colour = model)) +
  geom_point(aes(y = y))
```

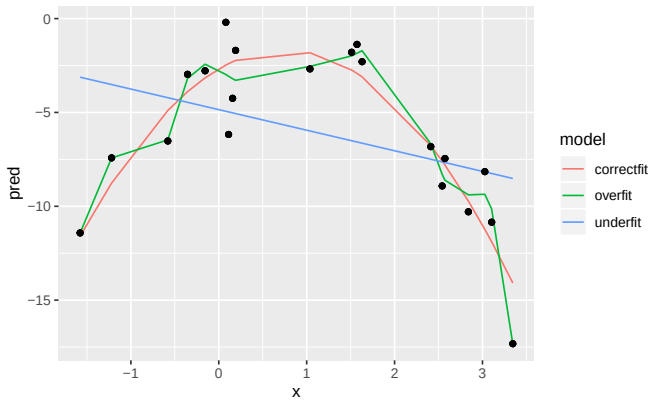


Figure 7.3: Simulation - Polynomial orders

See Figure 7.3. If we choose too large order for polynomial regression, we would get wiggly fit.

7.2.2 Penalization

To solve this kind of overfitting issues, we implement penalization method, like **ridge regression and lasso**. Adding l_1 penalty, we have seen that we can perform both variable selection and estimation.

7.3 Moving Beyond Linearity

When using non-linear model, polynomial model is not enough, sometimes. Moreover, polynomial regression gives unstable estimator near boundary. There are many other non-linear models such as local regression, splines et cetera.

```
paperstr %>%
  ggplot(aes(x = hardwood, y = strength)) +
  geom_smooth(method = "loess") +
  geom_point()
```

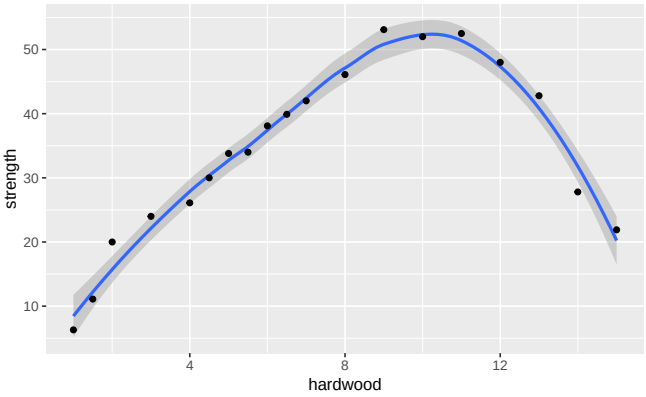


Figure 7.4: Local polynomial to the hardwood data set

Bibliography

Chatterjee, S. and Hadi, A. S. (2015). *Regression Analysis by Example*. John Wiley & Sons.

Ekstrom, C. T. and Sorensen, H. (2014). *Introduction to Statistical Data Analysis for the Life Sciences, Second Edition*. CRC Press.

Hastie, T., Tibshirani, R., and Friedman, J. (2013). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Science & Business Media.

Hogg, R. V., McKean, J. W., and Craig, A. T. (2018). *Introduction to Mathematical Statistics*. Pearson College Division, 8 edition.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer Science & Business Media.

Johnson, R. A. and Wichern, D. W. (2013). *Applied Multivariate Statistical Analysis*.

Kreyszig (2007). *Introductory Functional Analysis with Applications*. John Wiley & Sons.

Leon, S. (2014). *Linear Algebra with Applications*. Pearson Higher Ed.

Montgomery, D. C. (2012). *Design and Analysis of Experiments, 8th Edition*. Wiley Global Education.

Montgomery, D. C., Peck, E. A., and Vining, G. G. (2015). *Introduction to Linear Regression Analysis*. John Wiley & Sons.

Rawlings, J. O., Pantula, S. G., and Dickey, D. A. (2006). *Applied Regression Analysis: A Research Tool*. Springer Science & Business Media.

Wade, W. (2017). *An Introduction to Analysis*. Pearson, 4 edition.