STT 3850 : Week 6

Fall 2024

Appalachian State University

Section 1

Outline for the week

By the end of the week: Multiple Linear Regression

- Extra Sums of Squares
- Model selection

Section 2

Extra Sums of Squares

Partition of Total sum of squares

- For the linear regression model:

$$y_i = \beta_0 + \beta_1 x_{i,1} + \ldots + \beta_{p-1} x_{i,p-1} + \epsilon_i$$

- We fit the line:

$$\hat{y}_i = b_0 + b_1 x_{i,1} + \ldots + b_{p-1} x_{i,p-1}$$

- Partition of Total sum of squares:
  - Total sum of squares (SST): $SST = \sum (y_i - \bar{y})^2$.
  - Error sum of squares (SSE): $SSE = \sum (y_i - \hat{y}_i)^2$.
  - Regression sum of squares (SSR): $SSR = \sum (\hat{y}_i - \bar{y})^2$

Extra Sums of Squares

An extra sum of squares measures the marginal reduction in the error sum of squares when one or several predictor variables are added to the regression model, given that the other predictor variables are already in the model.

Term Life Insurance Example

Like all firms, life insurance companies continually seek new ways to deliver products to the market. Those involved in product development want to know who buys insurance and how much they buy. In this example, we examine the Survey of Consumer Finances (SCF), that contains extensive information on assets, liabilities, income, and demographic characteristics of those sampled (potential U.S. customers). We study a random sample of 500 households with positive incomes that were interviewed in the 2004 survey.

Example: Term Life Insurance

- y: FACE amount (log scale)
- x_1: Annual Income (log scale)
- x_2: Education
- x_3: Number of household members

Term Life Insurance Example

```r
library(tidyverse)
library(moderndive)
library(janitor)
Term <- read.csv("TermLife.csv")
term <- Term |>
  clean_names() |>
  filter(face > 0) |>
  mutate(ln_face = log(face), ln_income = log(income)) |>
  select(education, ln_face, ln_income, numhh)
```

Extra Sums of Squares: y on x_1

```r
modelX1 <- lm(ln_face ~ ln_income, data = term)
summary(modelX1)
```

```

Call:
lm(formula = ln_face ~ ln_income, data = term)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.1967 -0.8032 -0.0018  0.8954  6.4711 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  4.23003    0.85985   4.920  1.5e-06 ***
ln_income    0.69604    0.07661   9.086  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.642 on 273 degrees of freedom
Multiple R-squared:  0.2322,	Adjusted R-squared:  0.2294 
F-statistic: 82.55 on 1 and 273 DF,  p-value: < 2.2e-16
```

Extra Sums of Squares: y on x_1

```r
eis <- resid(modelX1)
SSE <- sum(eis^2)
SST <- sum((term$ln_face - mean(term$ln_face))^2)
SSR <- SST - SSE
c(SSE, SSR)
```

```
[1] 736.2671 222.6292
```

```r
knitr::kable(anova(modelX1))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
ln_income	1	222.6292	222.629245	82.54855	0
Residuals	273	736.2671	2.696949	NA	NA

Extra Sums of Squares: y on x_2

```r
modelX2 <- lm(ln_face ~ education, data = term)
summary(modelX2)
```

```

Call:
lm(formula = ln_face ~ education, data = term)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.4395 -1.2698  0.2065  1.2194  4.5559 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  7.90986    0.60499  13.074  < 2e-16 ***
education    0.28095    0.04103   6.847 4.96e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.731 on 273 degrees of freedom
Multiple R-squared:  0.1466,	Adjusted R-squared:  0.1434 
F-statistic: 46.89 on 1 and 273 DF,  p-value: 4.964e-11
```

Extra Sums of Squares: y on x_2

```
get_regression_points(modelX2) -> RT
RT |>
  summarize(SSE = sum(residual^2),
            SST = sum((ln_face - mean(ln_face))^2),
            SSR = SST - SSE) -> ESS
knitr::kable(ESS)
```

SSE	SST	SSR
818.375	958.967	140.5921

```
knitr::kable(anova(modelX2))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
education	1	140.5486	140.548609	46.88688	0
Residuals	273	818.3477	2.997611	NA	NA

Extra Sums of Squares: y on x_1 and x_2

```r
modelX1X2 <- lm(ln_face ~ ln_income + education, data = term)
summary(modelX1X2)
```

```

Call:
lm(formula = ln_face ~ ln_income + education, data = term)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.1266 -1.0284  0.1817  0.9185  5.3403 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.96235    0.87676   3.379 0.000835 ***
ln_income    0.57392    0.07879   7.284 3.50e-12 ***
education    0.18103    0.04003   4.523 9.11e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.587 on 272 degrees of freedom
Multiple R-squared:  0.2859,    Adjusted R-squared:  0.2806 
F-statistic: 54.44 on 2 and 272 DF,  p-value: < 2.2e-16
```

Extra Sums of Squares: y on x_1 and x_2

```
get_regression_points(modelX1X2) -> RT2
RT2 |>
  summarize(SSE = sum(residual^2),
            SST = sum((ln_face - mean(ln_face))^2),
            SSR = SST - SSE) -> ESS2
knitr::kable(ESS2)
```

SSE	SST	SSR
684.7941	958.967	274.1729

```
knitr::kable(anova(modelX1X2))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
ln_income	1	222.6292	222.629245	88.43204	0.0e+00
education	1	51.5022	51.502201	20.45753	9.1e-06
Residuals	272	684.7649	2.517518	NA	NA

Extra Sums of Squares: y on x_1, x_2 and x_3

```r
modelAll <- lm(ln_face ~ ln_income + education + numhh, data = term)
summary(modelAll)
```

```

Call:
lm(formula = ln_face ~ ln_income + education + numhh, data = term)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.7420 -0.8681  0.0549  0.9093  4.7187 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.58408    0.84643   3.053  0.00249 ** 
ln_income    0.49353    0.07754   6.365 8.32e-10 ***
education    0.20641    0.03883   5.316 2.22e-07 ***
numhh        0.30605    0.06333   4.833 2.26e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.525 on 271 degrees of freedom
Multiple R-squared:  0.3425,	Adjusted R-squared:  0.3353 
F-statistic: 47.07 on 3 and 271 DF,  p-value: < 2.2e-16
```

Extra Sums of Squares: y on x_1, x_2 and x_3

```
get_regression_points(modelAll) -> RT3
RT3 |>
  summarize(SSE = sum(residual^2),
            SST = sum((ln_face - mean(ln_face))^2),
            SSR = SST - SSE) -> ESS3
knitr::kable(ESS3)
```

SSE	SST	SSR
630.458	958.967	328.509

```
knitr::kable(anova(modelAll))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
ln_income	1	222.62925	222.629245	95.70075	0.0e+00
education	1	51.50220	51.502201	22.13905	4.0e-06
numhh	1	54.33593	54.335932	23.35717	2.3e-06

Extra Sums of Squares

Independent Variable	SSR	SSE
x_1	222.63	736.27
x_2	140.55	818.35
x_1 and x_2	274.13	684.76
x_1, x_2 and x_3	328.47	630.43

Hence:

$$SSR(x_2|x_1) = SSE(x_1) - SSE(x_1, x_2) = 736.27 - 684.76 = 51.51$$

or

$$SSR(x_2|x_1) = SSR(x_1, x_2) - SSR(x_1) = 274.13 - 222.63 = 51.51$$

Question:

1. Why are they equal?
2. Find $SSR(x_1|x_2)$?

Extra Sums of Squares

Independent Variable	SSR	SSE
x_1	222.63	736.27
x_2	140.55	818.35
x_1 and x_2	274.13	684.76
x_1, x_2 and x_3	328.47	630.43

Similarly:

$$\begin{aligned} SSR(x_3|x_1,x_2) &= SSE(x_1,x_2) - SSE(x_1,x_2,x_3) \\ &= 684.76 - 630.43 = 54.33 \end{aligned}$$

or

$$\begin{aligned} SSR(x_3|x_1,x_2) &= SSR(x_1,x_2,x_3) - SSR(x_1,x_2) \\ &= 328.47 - 274.13 = 54.33 \end{aligned}$$

Problem: Find the value of $SSR(x_2,x_3|x_1)$.

Decomposition

In multiple regression, we can obtain a variety of decompositions of the regression SSR into extra sum of squares. For example,

$$SSR(x_1, x_2) = SSR(x_1) + SSR(x_2|x_1), \quad \text{or}$$
$$SSR(x_1, x_2) = SSR(x_2) + SSR(x_1|x_2).$$

If we have three variables, then:

$$SSR(x_1, x_2, x_3) = SSR(x_1) + SSR(x_2|x_1) + SSR(x_3|x_1, x_2), \quad \text{or}$$
$$SSR(x_1, x_2, x_3) = SSR(x_2) + SSR(x_3|x_2) + SSR(x_1|x_2, x_3), \quad \text{or}$$
$$SSR(x_1, x_2, x_3) = SSR(x_1) + SSR(x_2, x_3|x_1).$$

The ANOVA table is shown below

Source of Variation (Source)	Degrees of Freedom (df)	Sum of Squares (SS)	Mean Square (MS)	F
x_1	1	$SSR(x_1)$	$MSR(x_1) = SSR(x_1)/1$	$F = \frac{MSR((x_1))}{MSE((x_1,x_2,x_3))}$
$x_2 \vert x_1$	1	$SSR(x_2 \vert x_1)$	$MSR(x_2 \vert x_1) = SSR(x_2 \vert x_1)/1$	$F = \frac{MSR((x_2 \vert x_1))}{MSE((x_1,x_2,x_3))}$
$x_3 \vert x_1, x_2$	1	$SSR(x_3 \vert x_1, x_2)$	$MSR(x_3 \vert x_1, x_2) = SSR(x_3 \vert x_1, x_2)/1$	$F = \frac{MSR((x_3 \vert x_1,x_2))}{MSE((x_1,x_2,x_3))}$
Error	$n-4$	$SSE(x_1, x_2, x_3)$	$MSE = SSE(x_1, x_2, x_3)/(n-4)$	
Total	$n-1$	$SST(x_1, x_2, x_3)$		

Analysis of Variance: ANOVA

```r
#For Term Life Insurance Example:
knitr::kable(anova(modelAll))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
ln_income	1	222.62925	222.629245	95.70075	0.0e+00
education	1	51.50220	51.502201	22.13905	4.0e-06
numhh	1	54.33593	54.335932	23.35717	2.3e-06
Residuals	271	630.42897	2.326306	NA	NA

Why are extra sum of squares of interest?

Tests for Regression Coefficients

When we wish to test whether the term $\beta_k x_k$ can be dropped from a multiple regression model, we are interested in:

$$H_0 : \beta_k = 0, \quad vs. \quad H_a : \beta_k \neq 0.$$

Tests for Regression Coefficients

For example, let us consider the first-order regression model

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \epsilon_i.$$

Test:

$$H_0 : \beta_3 = 0, \quad vs. \quad H_a : \beta_3 \neq 0.$$

Under the null, we have the reduced model,

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i.$$

For those two models, the extra sum of squares is

$$SSR(x_3|x_1, x_2) = SSE(x_1, x_2) - SSE(x_1, x_2, x_3)$$

Tests for Regression Coefficients

The general linear test statistic

$$F^* = \frac{SSE_{reduced} - SSE_{full}}{df_{reduced} - df_{full}} \div \frac{SSE_{full}}{df_{full}}$$

becomes:

$$\begin{aligned} F^* &= \frac{SSE(x_1,x_2) - SSE(x_1,x_2,x_3)}{(n-3)-(n-4)} \div \frac{SSE(x_1,x_2,x_3)}{n-4} \\ &= \frac{SSR(x_3|x_1,x_2)}{1} \div \frac{SSE(x_1,x_2,x_3)}{n-4} \\ &= \frac{MSR(x_3|x_1,x_2)}{MSE(x_1,x_2,x_3)} \end{aligned}$$

Term Life Insurance Example

$$F^* = \frac{54.34}{1} \div \frac{630.43}{271} = 23.36$$

```
Fstar <- anova(modelAll)[3, 4]
Fstar
```

```
[1] 23.35717
```

```
## Get p-value for F-statistic
pvalue <- 1 - pf(23.36, 1, 271)
pvalue
```

```
[1] 2.252657e-06
```

Term Life Insurance Example

We can use the $\wp$-value to test $H_0 : \beta_3 = 0$.

```
summary(modelAll)
```

```
Call:
lm(formula = ln_face ~ ln_income + education + numhh, data = term)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.7420 -0.8681  0.0549  0.9093  4.7187 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.58408    0.84643   3.053  0.00249 ** 
ln_income    0.49353    0.07754   6.365 8.32e-10 ***
education    0.20641    0.03883   5.316 2.22e-07 ***
numhh        0.30605    0.06333   4.833 2.26e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.525 on 271 degrees of freedom
Multiple R-squared:  0.3425,	Adjusted R-squared:  0.3353 
F-statistic: 47.07 on 3 and 271 DF,  p-value: < 2.2e-16
```

Testing More Than One Coefficient

Consider testing:

$H_0 : \beta_2 = \beta_3 = 0,$ versus $\quad H_a :$ at least one $\beta_i \neq 0$ for $i = 1, 2, \ldots, p-1.$

Under the null, we have the reduced model,

$$y_i = \beta_0 + \beta_1 x_{i1} + \epsilon_i.$$

This is the `modelX1` we estimated earlier.

Testing More Than One Coefficient

```r
#For Term Life Insurance Example:
anova(modelX1, modelAll)
```

```
Analysis of Variance Table

Model 1: ln_face ~ ln_income
Model 2: ln_face ~ ln_income + education + numhh
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1    273 736.27
2    271 630.43  2    105.84 22.748 7.369e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Is There a Relationship Between the Response and Predictors

Now, we test:

$H_0 : \beta_1 = \beta_2 = \beta_3 = 0,$ versus H_a : at least one $\beta_i \neq 0$ for $i = 1, 2, \ldots, p-1$

Under the null, we have the reduced model,

$$y_i = \beta_0 + \epsilon_i.$$

Is There a Relationship Between the Response and Predictors

```r
modelInt <- lm(ln_face ~ 1, data = term)
summary(modelInt)
```

```
Call:
lm(formula = ln_face ~ 1, data = term)

Residuals:
    Min      1Q  Median      3Q     Max
-5.3057 -1.1705 -0.0719  1.2974  4.4643

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  11.9903     0.1128   106.3   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Is There a Relationship Between the Response and Predictors

```r
anova(modelInt, modelAll)
```

```
Analysis of Variance Table

Model 1: ln_face ~ 1
Model 2: ln_face ~ ln_income + education + numhh
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1    274 958.90
2    271 630.43  3    328.47 47.066 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Is There a Relationship Between the Response and Predictors

```r
# Or
summary(modelAll)
```

```
Call:
lm(formula = ln_face ~ ln_income + education + numhh, data = term)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.7420 -0.8681  0.0549  0.9093  4.7187 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.58408    0.84643   3.053  0.00249 ** 
ln_income    0.49353    0.07754   6.365 8.32e-10 ***
education    0.20641    0.03883   5.316 2.22e-07 ***
numhh        0.30605    0.06333   4.833 2.26e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.525 on 271 degrees of freedom
Multiple R-squared:  0.3425,	Adjusted R-squared:  0.3353 
F-statistic: 47.07 on 3 and 271 DF,  p-value: < 2.2e-16
```

Section 3

Model selection

Model selection

The general multiple linear regression model with response y and terms $x_1, \ldots, x_{p-1}$ will have the form:

$$y = \beta_0 + \beta_1 x_1 + \ldots + \beta_{p-1} x_{p-1} + \epsilon.$$

- How many alternative models:
  - $y = \beta_0 + \epsilon$
  - $y = \beta_0 + \beta_1 x_1 + \epsilon$
  - $y = \beta_0 + \beta_2 x_2 + \epsilon$
  - $\vdots$
  - $y = \beta_0 + \beta_{p-1} x_{p-1} + \epsilon$
  - $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$
  - $\vdots$
  - $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_{p-1} x_{p-1} \epsilon$

One can construct a total of 2^{p-1} models! Question: How to select the "best" model?

Partition of Total sum of squares

- For the linear regression model:

$$y_i = \beta_0 + \beta_1 x_{i,1} + \ldots + \beta_{p-1} x_{i,p-1} + \epsilon_i.$$

- We have fitted the line: $\hat{y}_i = b_0 + b_1 x_{i,1} + \ldots + b_{p-1} x_{i,p-1}$.
- Partitioning the sum of squares is the same:
  - Total sum of squares (SST): $SST = \sum (y_i - \bar{y})^2$.
  - Error sum of squares (SSE): $SSE = \sum (y_i - \hat{y}_i)^2$.
  - Regression sum of squares (SSR): $SSR = \sum (\hat{y}_i - \bar{y})^2$

R^2 and Adjusted R^2 (R^2_{adj})

The coefficient of determination of the regression model, is defined as the proportion of the total sample variability in the y's explained by the regression model, that is,

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

We can also write:

$$R^2 = \frac{var(\hat{y})}{var(y)}$$

R^2 and Adjusted R^2 (R^2_{adj})

Caution: adding irrelevant predictor variables to the regression equation often increases R^2.

Q: Why do we use it?

A: The intent in using R^2 criterion is to find the point where adding more x variables is not worthwhile because it leads to a very small increase in R^2. Often this point is reached when only a limited number of x variables are included in the regression model.

R^2 and Adjusted R^2 (R^2_{adj})

One can define an adjusted coefficient of determination

$$R^2_{\text{adj}} = 1 - \frac{SSE/(n-p)}{SST/(n-1)} = 1 - \frac{MSE}{SST/n-1}$$

where p is the number of predictors in the current model. This coefficient takes the number of parameters in the regression model into account using degrees of freedom.

Users of the R^2_{adj} criterion seek to find a few subsets for which R^2_{adj} is at the maximum or that adding more variables is not worthwhile.

Needed packages

Let's load all the packages needed for this chapter.

```
library(tidyverse)
library(moderndive)
library(skimr)
library(ISLR)
evals_ch6 <- evals |>
  select(ID, score, age, gender)
```

Compare Interaction and Parallel slopes model

```r
# Fit interaction model:
score_model_interaction <- lm(score ~ age + gender + age:gender, data = evals_ch6)
summary(score_model_interaction)
```

```
Call:
lm(formula = score ~ age + gender + age:gender, data = evals_ch6)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.86453 -0.34815  0.09863  0.40661  0.96327 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)     4.882989   0.205210  23.795  < 2e-16 ***
age            -0.017523   0.004472  -3.919 0.000103 ***
gendermale     -0.446044   0.265407  -1.681 0.093520 .  
age:gendermale  0.013531   0.005531   2.446 0.014803 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5314 on 459 degrees of freedom
Multiple R-squared:  0.05138,	Adjusted R-squared:  0.04518 
F-statistic: 8.288 on 3 and 459 DF,  p-value: 2.227e-05
```

Compare Interaction and Parallel slopes model

```r
# Fit parallel slopes model:
score_model_parallel_slopes <- lm(score ~ age + gender, data = evals_ch6)
summary(score_model_parallel_slopes)
```

```

Call:
lm(formula = score ~ age + gender, data = evals_ch6)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.82833 -0.33494  0.09391  0.42882  0.91506 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  4.484116   0.125284  35.792  < 2e-16 ***
age         -0.008678   0.002646  -3.280 0.001117 ** 
gendermale   0.190571   0.052469   3.632 0.000313 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5343 on 460 degrees of freedom
Multiple R-squared:  0.03901,	Adjusted R-squared:  0.03484 
F-statistic: 9.338 on 2 and 460 DF,  p-value: 0.0001059
```

Compare Interaction and Parallel slopes model

```r
# Get summaries of models:
get_regression_summaries(score_model_interaction)
```

```
# A tibble: 1 x 9
  r_squared adj_r_squared   mse  rmse sigma statistic p_value    df  nobs
      <dbl>         <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl> <dbl> <dbl>
1     0.051         0.045 0.280 0.529 0.531      8.29       0     3   463
```

```r
get_regression_summaries(score_model_parallel_slopes)
```

```
# A tibble: 1 x 9
  r_squared adj_r_squared   mse  rmse sigma statistic p_value    df  nobs
      <dbl>         <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl> <dbl> <dbl>
1     0.039         0.035 0.284 0.533 0.534      9.34       0     2   463
```

AIC: Akaike Information Criterion

Akaike's information criterion (AIC) can be motivated in two ways. The most popular motivation seems to be based on balancing goodness of fit and a penalty for model complexity. **AIC is defined such that the smaller the value of AIC the better the model.**

$$AIC = n \log(SSE_m/n) + 2m.$$

Recall that m is the number of parameters in your subset model. For example, if your model includes only $\beta_0, \beta_1, \beta_2$, then $m = 3$.

Caution: When the sample size is small, or when the number of parameters estimated is a moderate to large fraction of the sample size, it is well-known that AIC has a tendency for over-fitting since the penalty for model complexity is not strong enough.

BIC: Bayes Information Criterion

BIC is defined such that the smaller the value of BIC the better the model.

$$BIC = n\log(SSE_m/n) + m\log(n)$$

BIC penalizes complex models more heavily than AIC, thus favoring simpler models than AIC.

Compare Interaction and Parallel slopes model

```r
# AIC
AIC(score_model_interaction)
```

```
[1] 734.5273
```

```r
AIC(score_model_parallel_slopes)
```

```
[1] 738.5253
```

```r
# BIC
BIC(score_model_interaction)
```

```
[1] 755.2159
```

```r
BIC(score_model_parallel_slopes)
```

```
[1] 755.0762
```

"Best" Subset Algorithm

Time-saving algorithms have been developed in which the best subsets according to a specified criterion are identified without requiring the fitting of all of possible subset regression models.

Example: For the eight predictors, we know there are $2^8 = 256$ possible models.

Forward selection

Forward selection starts with no variables in the model and then adds the $x-$variable that produces the smallest $\wp$-value below α_{crit} when included in the model. This procedure is continued until no new predictors can be added. The user can determine the variable that produces the smallest $\wp$-value by regressing the response variable on the x_is one at a time using `lm()` and `summary()` or using the `add1()` function.

Stepwise Procedures

Backward elimination

Backward elimination begins with a model contining all potential $x-$ variables and identifies the one with the largest $\wp$-value. This can be done by looking at the $\wp$-values for the $t-$ values of the $\hat{\beta}_i, i = 1, \ldots, p-1$ using the function `summary()` or by using the $\wp$-values from the function `drop1()`. If the variable with the largest $\wp$-value is a above a predetermined value, α_{crit}, that variable is dropped. A model with the remaining $x-$variables is then fit and the procedure continues until all the $\wp$-values for the remaining variables in the model are below the predetermined α_{crit}. The α_{crit} is sometimes referred to as the "$\wp$-value-to-remove" and is typically set to 15 or 20%.

Term Life Insurance Example

Like all firms, life insurance companies continually seek new ways to deliver products to the market. Those involved in product development want to know who buys insurance and how much they buy. In this example, we examine the Survey of Consumer Finances (SCF), that contains extensive information on assets, liabilities, income, and demographic characteristics of those sampled (potential U.S. customers). We study a random sample of 500 households with positive incomes that were interviewed in the 2004 survey.

Term Life Insurance Example

```r
####forward selection based on AIC #####
library(MASS)
null <- lm(ln_face  ~ 1, data = term)
full <- lm(ln_face ~ ., data = term)
mod_fs <- stepAIC(null, scope = list(lower= null, upper= full),
                  direction = "forward", trace = 0)
summary(mod_fs)
```

```
Call:
lm(formula = ln_face ~ ln_income + education + numhh, data = term)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.7420 -0.8681  0.0549  0.9093  4.7187 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.58408    0.84643   3.053  0.00249 ** 
ln_income    0.49353    0.07754   6.365 8.32e-10 ***
education    0.20641    0.03883   5.316 2.22e-07 ***
numhh        0.30605    0.06333   4.833 2.26e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.525 on 271 degrees of freedom
Multiple R-squared:  0.3425,	Adjusted R-squared:  0.3353 
F-statistic: 47.07 on 3 and 271 DF,  p-value: < 2.2e-16
```

Term Life Insurance Example

```r
#### backward elimination based on AIC #####
mod_be <- stepAIC(full, scope = list(lower = null, upper = full),
                  direction = "backward", trace = 0)
summary(mod_be)
```

```
Call:
lm(formula = ln_face ~ education + ln_income + numhh, data = term)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.7420 -0.8681  0.0549  0.9093  4.7187 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.58408    0.84643   3.053  0.00249 ** 
education    0.20641    0.03883   5.316 2.22e-07 ***
ln_income    0.49353    0.07754   6.365 8.32e-10 ***
numhh        0.30605    0.06333   4.833 2.26e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.525 on 271 degrees of freedom
Multiple R-squared:  0.3425,	Adjusted R-squared:  0.3353 
F-statistic: 47.07 on 3 and 271 DF,  p-value: < 2.2e-16
```

Section 4

Multicollinearity

Multicollinearity and its Effects

- **Multicollinearity** exists when two or more of the predictors in a regression model are moderately or highly correlated with one another.
  - Unfortunately, when it exists, it can wreak havoc on our analysis and thereby limit the research conclusions we can draw.
- When multicollinearity exists, any of the following outcomes can be exacerbated:
  - The estimated regression coefficient of any one variable depends on which other predictors are included in the model.
  - The standard errors and hence the variances of the estimated coefficients are inflated when multicollinearity exists.
  - Inflated variances impact the conclusion for hypothesis tests for $\beta_k = 0$.

Variance Inflation Factor

- The **variance inflation factors (VIF)** quantifies how much the variance of the estimated coefficients are inflated.
- Hence, the variance inflation factor for the estimated regression coefficient b_j, denoted VIF_j is just the factor by which the variance of b_j is "inflated" by the existence of correlation among the predictor variables in the model.

Variance Inflation Factor

In particular, the variance inflation factor for the jth predictor is

$$VIF_j = \frac{1}{1 - R_j^2}$$

where R_j^2 is the R^2-value obtained by regressing the jth predictor on the remaining predictors.

- How do we interpret the variance inflation factors for a regression model?
  - A VIF of 1 means that there is no correlation among the jth predictor and the remaining predictor variables, and hence the variance of b_j is not inflated at all.
  - The general rule of thumb is that VIFs exceeding 4 warrant further investigation, while VIFs exceeding 10 are signs of serious multicollinearity requiring correction.

High Blood Pressure Example

- The researchers were interested in determining if a relationship exists between blood pressure and age, weight, body surface area, duration, pulse rate and stress level.
  - blood pressure (y = bp, in mm Hg)
  - age (x1 = age, in years)
  - weight (x2 = weight, in kg)
  - body surface area (x3 = bsa, in sq m)
  - duration of hypertension (x4 = dur, in years)
  - basal pulse (x5 = pulse, in beats per minute)
  - stress index (x6 = stress)

High Blood Pressure Example

High correlation between `weight` and `bsa`

```
bloodpress <- read.csv("bloodpress.csv")
bloodpress <- bloodpress |>
  clean_names()
bloodpress <- bloodpress[, -1]
cor(bloodpress, use = "complete.obs")
```

```
              bp       age     weight        bsa       dur     pulse     stress
bp     1.0000000 0.6590930 0.95006765 0.86587887 0.2928336 0.7214132 0.16390139
age    0.6590930 1.0000000 0.40734926 0.37845460 0.3437921 0.6187643 0.36822369
weight 0.9500677 0.4073493 1.00000000 0.87530481 0.2006496 0.6593399 0.03435475
bsa    0.8658789 0.3784546 0.87530481 1.00000000 0.1305400 0.4648188 0.01844634
dur    0.2928336 0.3437921 0.20064959 0.13054001 1.0000000 0.4015144 0.31163982
pulse  0.7214132 0.6187643 0.65933987 0.46481881 0.4015144 1.0000000 0.50631008
stress 0.1639014 0.3682237 0.03435475 0.01844634 0.3116398 0.5063101 1.00000000
```

High Blood Pressure Example

```r
model_bp <- lm(bp ~ ., data = bloodpress)
summary(model_bp)
```

```
Call:
lm(formula = bp ~ ., data = bloodpress)

Residuals:
     Min       1Q   Median       3Q      Max
-0.93213 -0.11314  0.03064  0.21834  0.48454

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -12.870476   2.556650  -5.034 0.000229 ***
age           0.703259   0.049606  14.177 2.76e-09 ***
weight        0.969920   0.063108  15.369 1.02e-09 ***
bsa           3.776491   1.580151   2.390 0.032694 *
dur           0.068383   0.048441   1.412 0.181534
pulse        -0.084485   0.051609  -1.637 0.125594
stress        0.005572   0.003412   1.633 0.126491
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4072 on 13 degrees of freedom
Multiple R-squared:  0.9962,    Adjusted R-squared:  0.9944
F-statistic: 560.6 on 6 and 13 DF,  p-value: 6.395e-15
```

High Blood Pressure Example

```r
library(car)
vif(model_bp)
```

```
     age   weight      bsa      dur    pulse   stress
1.762807 8.417035 5.328751 1.237309 4.413575 1.834845
```

- The VIF_j for a predictor x_j can be interpreted as the factor ($\sqrt{\mathrm{VIF}_j}$) by which the standard error of $\hat{\beta}_j$ is increased due to the presence of multicollinearity.
- Regressing `weight` on the remaining five predictors, gives $R^2_{\text{weight}} = 88.12\%$.

$$VIF_{\text{weight}} = \frac{1}{1 - R^2_{\text{weight}}} = \frac{1}{1 - 0.8812} = 8.42$$

Variance Inflation Factor

```
r2age <- summary(lm(age ~ weight + bsa + dur + pulse +
                      stress, data = bloodpress))$r.squared
r2weight <- summary(lm(weight ~ age + bsa + dur + pulse +
                         stress, data = bloodpress))$r.squared
r2bsa <- summary(lm(bsa ~ age + weight + dur + pulse +
                      stress, data = bloodpress))$r.squared
r2dur <- summary(lm(dur ~ age + weight + bsa + pulse +
                      stress, data = bloodpress))$r.squared
r2pulse <- summary(lm(pulse ~ age + weight + bsa + dur +
                        stress, data = bloodpress))$r.squared
r2stress <- summary(lm(stress ~ age + weight + bsa + dur +
                         pulse, data = bloodpress))$r.squared
c(r2age, r2weight, r2bsa, r2dur, r2pulse, r2stress, r2age) -> r2s
r2s
```

```
[1] 0.4327228 0.8811933 0.8123388 0.1917947 0.7734263 0.4549949 0.4327228
```

```
(VIFs <- 1 /(1 - r2s))
```

```
[1] 1.762807 8.417035 5.328751 1.237309 4.413575 1.834845 1.762807
```

```
sqrt(VIFs)
```

```
[1] 1.327707 2.901213 2.308409 1.112344 2.100851 1.354565 1.327707
```

Variance Inflation Factor

```r
cor(bloodpress, use = "complete.obs")
```

```
              bp       age     weight        bsa       dur     pulse     stress
bp     1.0000000 0.6590930 0.95006765 0.86587887 0.2928336 0.7214132 0.16390139
age    0.6590930 1.0000000 0.40734926 0.37845460 0.3437921 0.6187643 0.36822369
weight 0.9500677 0.4073493 1.00000000 0.87530481 0.2006496 0.6593399 0.03435475
bsa    0.8658789 0.3784546 0.87530481 1.00000000 0.1305400 0.4648188 0.01844634
dur    0.2928336 0.3437921 0.20064959 0.13054001 1.0000000 0.4015144 0.31163982
pulse  0.7214132 0.6187643 0.65933987 0.46481881 0.4015144 1.0000000 0.50631008
stress 0.1639014 0.3682237 0.03435475 0.01844634 0.3116398 0.5063101 1.00000000
```

- We see that:
    - `weight` and `bsa` are highly correlated $(r = 0.875)$.
    - `pulse` also appears to exhibit fairly strong marginal correlations with several of the predictors, including `age` $(r = 0.619)$, `weight` $(r = 0.659)$ and `stress` $(r = 0.506)$
- We will remove `bsa` and `pulse` from the data.

Variance Inflation Factor

```r
model_bp_new <- lm(bp ~ age + weight + dur + stress, data = bloodpress)
summary(model_bp_new)
```

```

Call:
lm(formula = bp ~ age + weight + dur + stress, data = bloodpress)

Residuals:
     Min       1Q   Median       3Q      Max
-1.11359 -0.29586  0.01515  0.27506  0.88674

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -15.869829   3.195296  -4.967 0.000169 ***
age           0.683741   0.061195  11.173 1.14e-08 ***
weight        1.034128   0.032672  31.652 3.76e-15 ***
dur           0.039889   0.064486   0.619 0.545485
stress        0.002184   0.003794   0.576 0.573304
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5505 on 15 degrees of freedom
Multiple R-squared:  0.9919,    Adjusted R-squared:  0.9897
F-statistic: 458.3 on 4 and 15 DF,  p-value: 1.764e-15
```

```r
vif(model_bp_new)
```

```
     age   weight      dur   stress
1.468245 1.234653 1.200060 1.241117
```