

Chapter 9

MLR Estimation, Prediction & Model Assumptions

Chanwoo Yoo, Division of Advanced Engineering,
Korea National Open University

This work is a derivative of '[Regression Methods](#)' by Iain Pardoe, Laura Simon and Derek Young, used under [CC BY-NC](#).

Contents

1. MLR Estimation and Prediction
2. Assessing Model Assumptions using Plots
3. Assessing Model Assumptions using Hypothesis Tests

1. MLR Estimation and Prediction

1. Confidence Interval for the Mean Response

- 100(1 - α) percent confidence interval for μ_Y when the predictor values are $X_h = (1, X_{h,1}, X_{h,2}, \dots, X_{h,p-1})^T$
 - sample estimate \pm (t-multiplier \times standard error)
 - $\hat{y}_h \pm t_{(\frac{\alpha}{2}, n-p)} \times \sqrt{MSE(X_h^T (X^T X)^{-1} X_h)}$

2. Factors affecting the width of Confidence Interval for the Mean Response

- $\hat{y}_h \pm t_{\left(\frac{\alpha}{2}, n-p\right)} \times \sqrt{MSE(X_h^T (X^T X)^{-1} X_h)}$
 - As the mean squared error(MSE) decreases, the width of the interval decreases.
 - As we decrease the confidence level, the t-multiplier decreases, and hence the width of the interval decreases.

2. Factors affecting the width of Confidence Interval for the Mean Response

- $\hat{y}_h \pm t_{\left(\frac{\alpha}{2}, n-p\right)} \times \sqrt{MSE(X_h^T (X^T X)^{-1} X_h)}$
 - As we increase the sample size n , the width of the interval decreases.
 - The closer X_h is to the average of the sample's predictor values, the narrower the interval.

3. Prediction Interval for a New Response

- 100(1 - α) percent confidence interval for y_{new}

- $\hat{y}_h \pm t_{\left(\frac{\alpha}{2}, n-p\right)} \times \sqrt{MSE + (se(\hat{y}_h))^2}$

- $\hat{y}_h \pm t_{\left(\frac{\alpha}{2}, n-p\right)} \times \sqrt{MSE + MSE(X_h^T(X^T X)^{-1}X_h)}$

4. Standard Error of the Prediction

- $\sqrt{MSE + (se(\hat{y}_h))^2}$
 - Confidence Interval for the Mean Response will always be narrower than the corresponding prediction interval

5. Code: Confidence Interval

```
iqsize <- read.table("iqsize.txt", header=T)
attach(iqsize)
model <- lm(PIQ ~ Brain + Height)

predict(model, interval="confidence", se.fit=T,
        newdata=data.frame(Brain=90, Height=70))
```

6. Results: Confidence Interval

```
> predict(model, interval="confidence", se.fit=T,  
+         newdata=data.frame(Brain=90, Height=70))  
$fit
```

	fit	lwr	upr
1	105.6391	98.23722	113.041

```
$se.fit  
[1] 3.646064
```

```
...
```

7. Code: Prediction Interval

```
iqsize <- read.table("iqsize.txt", header=T)
attach(iqsize)
model <- lm(PIQ ~ Brain + Height)

predict(model, interval="prediction",
        newdata=data.frame(Brain=90, Height=70))
```

8. Results: Prediction Interval

```
> predict(model, interval="prediction",  
+         newdata=data.frame(Brain=90, Height=70))  
      fit      lwr      upr  
1 105.6391 65.34688 145.9314
```

9. Standard error of the fit at X_h

- $se(\hat{y}_h) = \sqrt{MSE(X_h^T(X^T X)^{-1}X_h)}$

10. Assumptions in MLR

- $y = X\beta + \varepsilon$
 - $X: n \times p$ matrix, $\beta: p \times 1$ vector, $\varepsilon: n \times 1$ vector, $y: n \times 1$ vector
 - Linearity: $E(\varepsilon) = 0$
 - Independence, Equal Variance: $E(\varepsilon\varepsilon^T) = \sigma^2 I_n$
 - Normality: $\varepsilon \sim N(0, \sigma^2 I_n)$
 - $\text{Rank}(X) = p$

11. Variance of \hat{y}_h

$$V(\hat{y}_h)$$

$$= V(X_h^T b)$$

$$= X_h^T V(b) X_h$$

$$= \sigma^2 X_h^T (X^T X)^{-1} X_h$$

12. Covariance Matrix of b

$$\begin{aligned} V(b) &= E[(b - E(b))(b - E(b))^T] \\ &= E[(b - \beta)(b - \beta)^T] \\ &= E[(X^T X)^{-1} X^T \varepsilon \varepsilon^T X (X^T X)^{-1}] \\ &= (X^T X)^{-1} X^T E(\varepsilon \varepsilon^T) X (X^T X)^{-1} \\ &= \sigma^2 (X^T X)^{-1} X^T I X (X^T X)^{-1} \\ &= \sigma^2 (X^T X)^{-1} \end{aligned}$$

13. Estimation Error of b

$$\begin{aligned} b - \beta &= (X^T X)^{-1} X^T \mathbf{y} - \beta \\ &= (X^T X)^{-1} X^T (X\beta + \varepsilon) - \beta \\ &= (X^T X)^{-1} X^T \varepsilon \end{aligned}$$

14. Bias

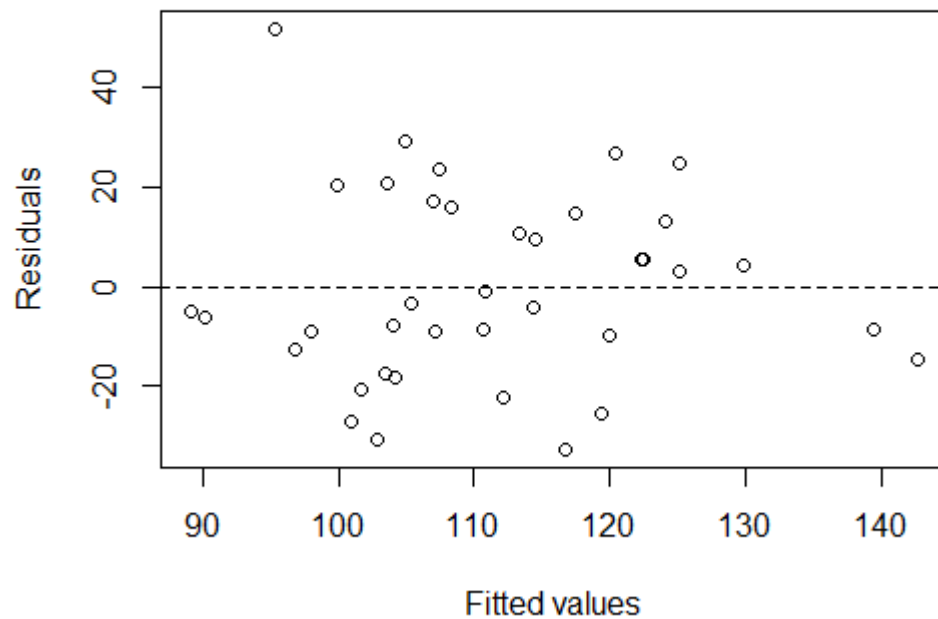
- $E(b - \beta) = (X^T X)^{-1} X^T E(\varepsilon) = 0$
 - $E(b) = \beta$

2. Assessing Model Assumptions using Plots

1. Code: Residuals vs Fits Plot

```
plot(x=fitted(model), y=residuals(model),  
     xlab="Fitted values", ylab="Residuals",  
     panel.last = abline(h=0, lty=2))
```

2. Results: Residuals vs Fits Plot

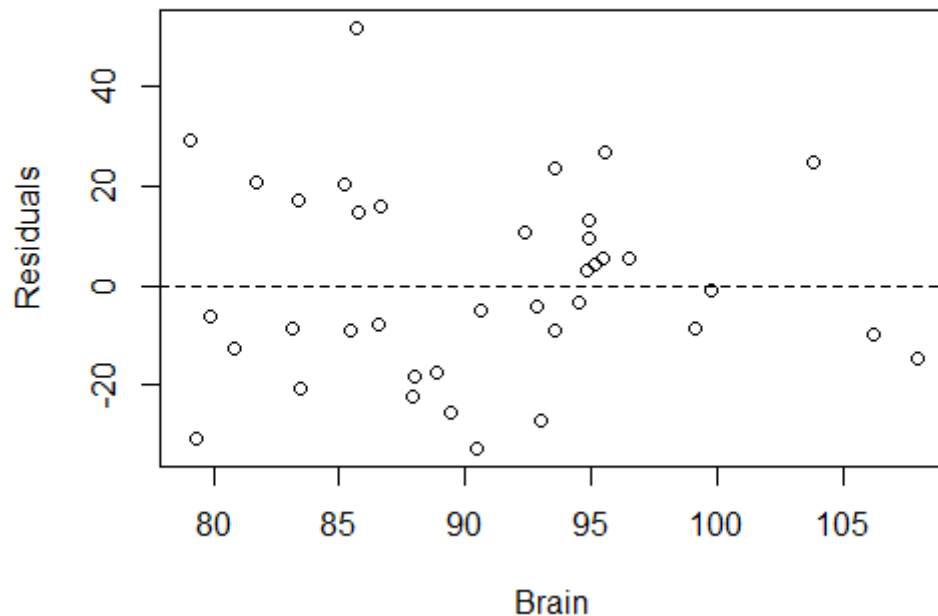


- The average of the residuals remains approximately 0.
- The variation of the residuals appears to be roughly constant.
- There are no excessively outlying points (except the observation with a residual of about 50).

3. Code: Residuals vs Brain

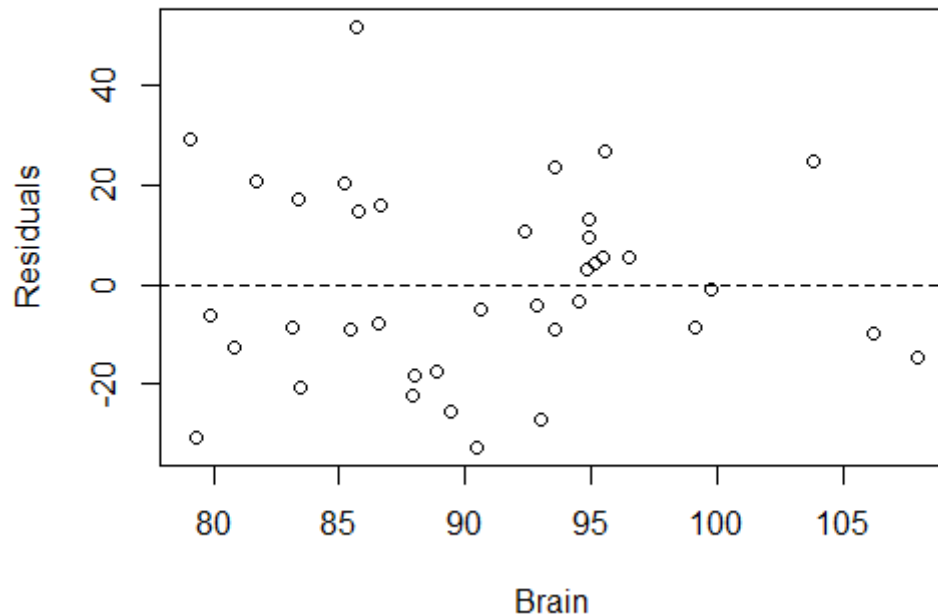
```
plot(x=Brain, y=residuals(model),  
     ylab="Residuals",  
     panel.last = abline(h=0, lty=2))
```

4. Results: Residuals vs Brain



- The average of the residuals remains approximately 0.
- The variation of the residuals appears to be roughly constant.
- There are no excessively outlying points.

4. Results: Residuals vs Brain

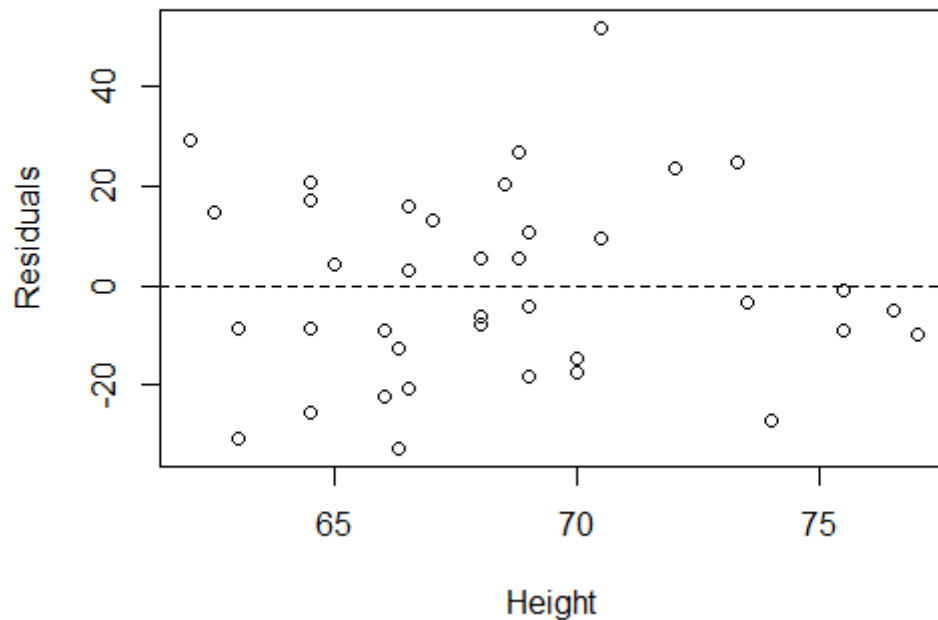


- There is no strong nonlinear trend in this plot that might suggest a transformation of PIQ or Brain in this model.

5. Code: Residuals vs Height

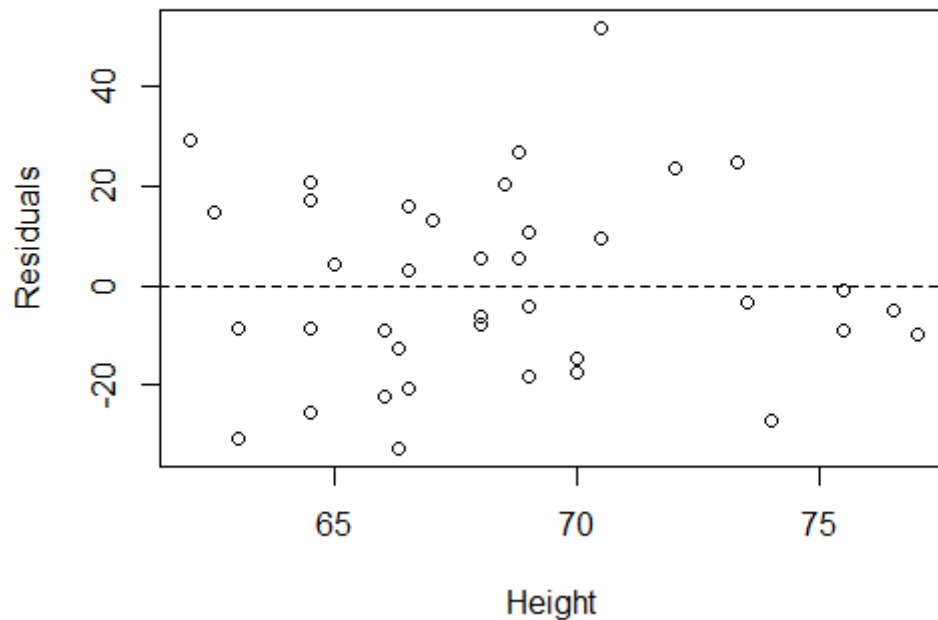
```
plot(x=Height, y=residuals(model),  
     ylab="Residuals",  
     panel.last = abline(h=0, lty=2))
```

6. Results: Residuals vs Height



- The average of the residuals remains approximately 0.
- The variation of the residuals appears to be roughly constant.
- There are no excessively outlying points.

6. Results: Residuals vs Height

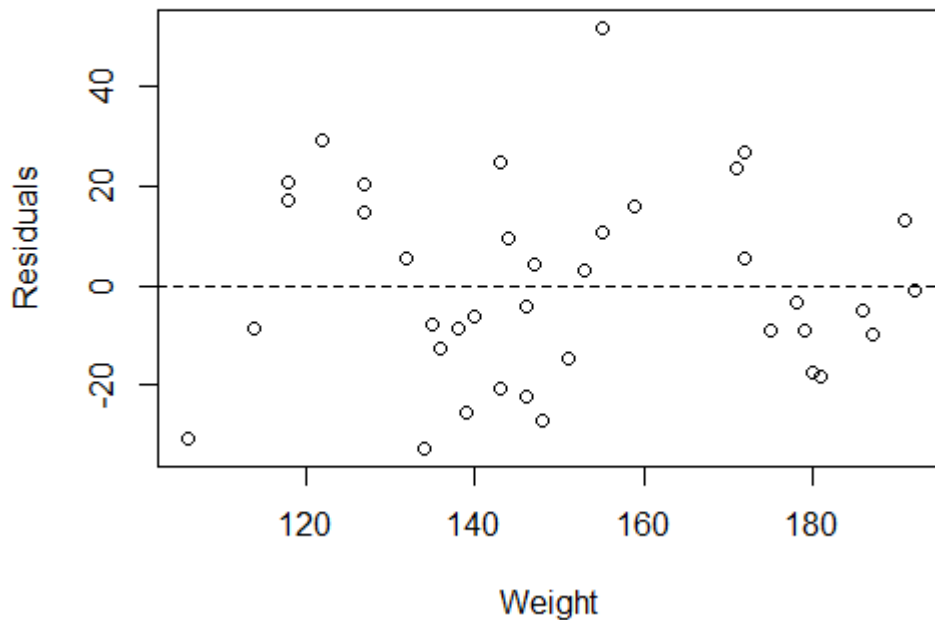


- There is no strong nonlinear trend in this plot that might suggest a transformation of PIQ or Brain in this model.

7. Code: Residuals vs Weight

```
plot(x=Weight, y=residuals(model),  
     ylab="Residuals",  
     panel.last = abline(h=0, lty=2))
```

8. Results: Residuals vs Weight

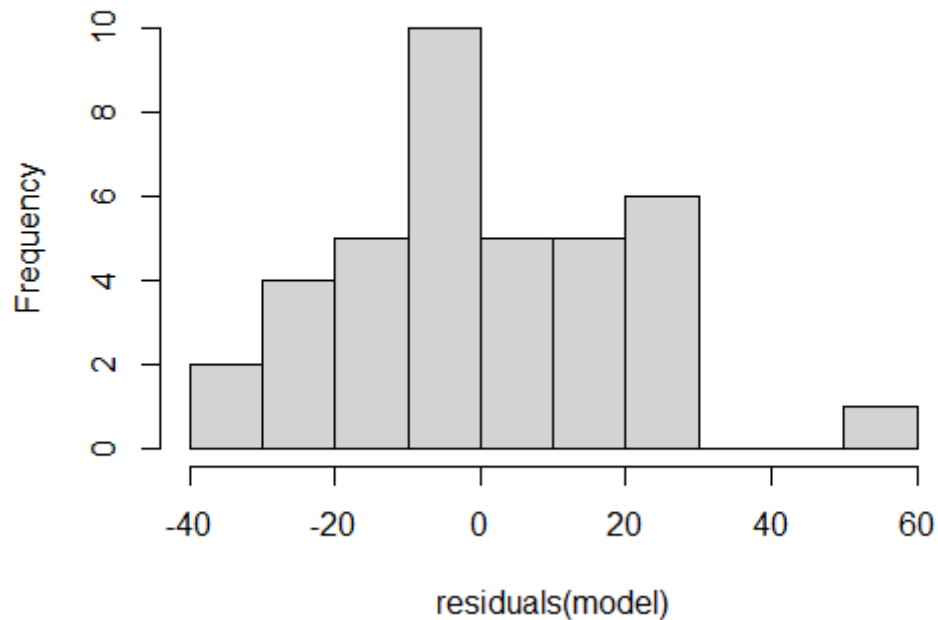


- Since there is no strong linear or simple nonlinear trend in this plot, there is nothing to suggest that Weight might be usefully added to the model.

9. Code: Histogram

```
hist(residuals(model), main="")
```

10. Results: Histogram

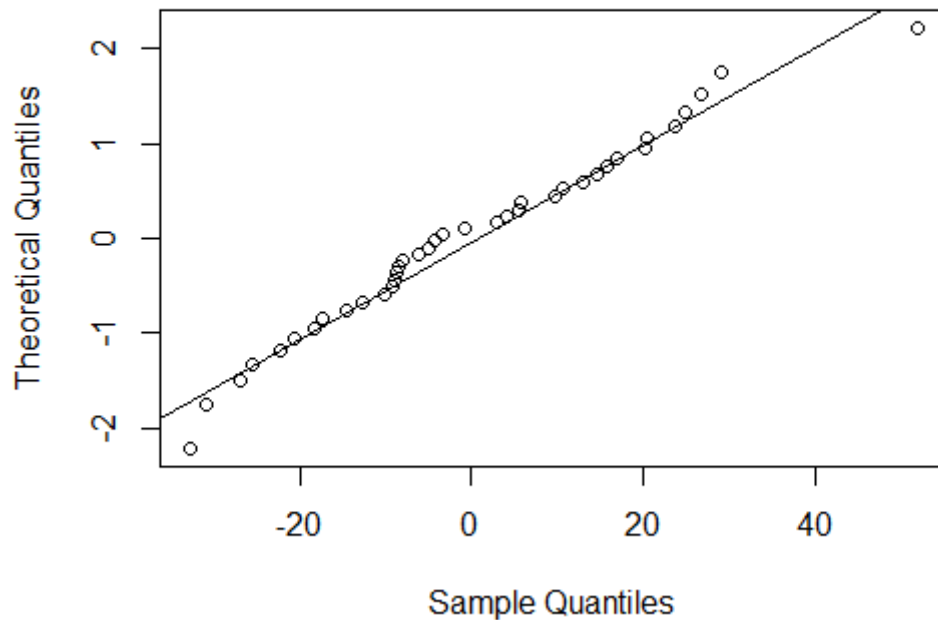


- Although this doesn't have the ideal bell-shaped appearance, given the small sample size there's little to suggest violation of the normality assumption.

11. Code: Normal Probability Plot

```
qqnorm(residuals(model), main="", datax=TRUE)  
qqline(residuals(model), datax=TRUE)
```


12. Results: Normal Probability Plot



- Given the small sample size there's little to suggest violation of the normality assumption.

3. Assessing Model Assumptions using Hypothesis Tests

1. Tests for Error Normality

- To complement the graphical methods just considered for assessing residual normality, we can perform a hypothesis test in which the null hypothesis is that the errors have a normal distribution.

2. Shapiro-Wilk Test

- Normality test which is used to determine if a dataset is normally distributed.
 - H_0 : the errors follow a normal distribution
 - H_1 : the errors do not follow a normal distribution
- It is a little bit more powerful than Anderson - Darling test and Kolmogorov - Smirnov test [1].

2. Shapiro-Wilk Test

- $W = \frac{\text{variance if distribution is normal}}{\text{actual variance}} = \frac{(\sum_{i=1}^n a_i x_{(i)})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}$
 - $x_{(i)}$: i th largest value
 - a_i : calculated value using the means, variances, and covariances of the x_i
 - Small values of W will lead to rejection of the null hypothesis.

3. Code: Tests for Error Normality

```
shapiro.test(residuals(model))
```

4. Results: Tests for Error Normality

```
> shapiro.test(residuals(model))
```

Shapiro-Wilk normality test

```
data: residuals(model)
```

```
W = 0.976, p-value = 0.5764
```

5. Shapiro-Wilk Test: Cautions

- Null hypothesis is rarely rejected when sample size is small.
- Null hypothesis is almost always rejected when sample size is large.

6. Tests for Constant Error Variance

- There are various tests that may be performed on the residuals for testing if the regression errors have constant variance. It is usually sufficient to "visually" interpret a residuals versus fitted values plot. However, the tests we discuss can provide an added layer of justification to your analysis.
 - H_0 : the variance is constant
 - H_1 : the variance is not constant

7. F-Test

- Suppose we partition the residuals of observations into two groups
 - one consisting of residuals associated with the lowest fitted values and the other consisting of those belonging to the highest fitted values. Treating these two groups as if they could represent two different populations, we can use F-statistic to test
 - $H_0: \sigma_1^2 = \sigma_2^2$
 - $H_A: \sigma_1^2 \neq \sigma_2^2$

7. F-Test

- $F^* = \frac{s_1^2}{s_2^2}$
 - This test statistic is distributed according to a F_{n_1-1, n_2-1} distribution.
 - $s_i^2 = \frac{\sum_{j=1}^{n_i} (e_{i,j} - \bar{e}_i)^2}{n_i - 1}$

8. Code: Tests for Constant Error Variance

```
fits <- fitted(model)
n_sample <- length(fits)
mid_index <- round(n_sample/2)
group1_index <- order(fits)[1:mid_index]
group1_error <- residuals(model)[group1_index]
group2_index <- order(fits)[(mid_index+1):n_sample]
group2_error <- residuals(model)[group2_index]

f <- var(group1_error)/var(group2_error)
pf(f, (length(group1_error)-1), (length(group2_error)-1), lower.tail=FALSE)
```

9. Results: Tests for Constant Error Variance

```
> pf(f, (length(group1_error)-1), (length(group2_error)-  
1), lower.tail=FALSE)  
[1] 0.1028362
```

Next

Chapter 10

Categorical Predictors

References

[1] Razali, Nornadiah Mohd, and Yap Bee Wah. "Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests." *Journal of Statistical Modeling and Analytics* 2.1 (2011): 21–33.