Chapter 12: Simple Linear Regression

Example 2

```
t.test(winnipeg_apt$price, conf = 0.95)
```

Example 4

```
lm(price ~ sqft, data = winnipeg_apt)
```

```
lm(winnipeg_apt$price ~ winnipeg_apt$sqft)
```

Example 6

```
cor(winnipeg_apt$price, winnipeg_apt$sqft)
```

Example 7

```
qt(0.025, 12, lower.tail = FALSE)
```

```
qt(0.025, 12)
```

Example 8

```
pt(3.910, 12, lower.tail = FALSE) + pt(-3.910, 12)
```

Example 9

```
slr <- lm(price ~ sqft, data = winnipeg_apt)

summary(slr)
```

Example 10

```
fitted(slr) -> price_predicted

cbind(winnipeg_apt, price_predicted) -> winnipeg_apt_new

head(winnipeg_apt_new)
```

Example 11

```
price_new <- data.frame(sqft <- c(800, 900, 1000, 1100, 1200))

predict(slr, price_new)
```

Example 12

```
confint(slr, level = 0.95)
```

12.8 Assumptions: How Are They Validated?

```
plot(winnipeg_apt$sqft, resid(slr), pch = 19, abline(h = 0), xlab =
      'Square Feet of Living Area', ylab = 'Residuals', main =
      'Winnipeg Homes: Residuals Against the Independent
      Variable')
```

```
plot(winnipeg_apt$sqft, winnipeg_apt$price, pch = 19, xlab =
      'Square Feet of Living Area', ylab = 'Price (in Canadian
      Dollars)')


slr <- lm(price ~ sqft, data = winnipeg_apt)


plot(winnipeg_apt$sqft, resid(slr), pch = 19, abline(h = 0), xlab =
      'Square Feet of Living Area', ylab = 'Residuals', main =
      'Winnipeg Homes: Residuals Against the Independent
      Variable')


summary(slr)


price_new <- data.frame(sqft <- c(800, 900, 1000, 1100, 1200))


predict(slr, price_new)
```

End-of-Chapter 12 Exercises

Exercise 1

```
city <- c('Auckland', 'Beijing', 'Cairo', 'Lagos', 'London',
        'Mexico City', 'Mumbai', 'Paris', 'Rio de Janeiro',
        'Sydney', 'Tokyo', 'Toronto', 'Vancouver', 'Zurich')



high <- c(71, 45, 65, 91, 46, 67, 88, 44, 92, 88, 57, 20, 42, 40)



low <- c(56, 23, 48, 76, 37, 45, 71, 35, 73, 65, 39, 15, 39, 29)



WorldTemps <- data.frame(City = city, High = high, Low = low)



plot(WorldTemps$Low, WorldTemps$High, pch = 19, xlab =
      "Low", ylab = "High", main = "High and Low Intraday
      Temperatures")



reg_eq_temps <- lm(High ~ Low, data = WorldTemps)



summary(reg_eq_temps)
```

Exercise 2

```
2 * pt(-8.75, 30)
```

Exercise 3

```
plot(mtcars$hp, mtcars$qsec, pch = 19, xlab = "Gross Horse
        Power", ylab = "Quarter Mile Time (seconds)")
```

```
reg_eq_mtcars <- lm(qsec ~ hp, data = mtcars)
```

```
summary(reg_eq_mtcars)
```

Exercise 4

```
tail(fitted(reg_eq_mtcars) , 4)
```

```
new_values <- data.frame(hp <- c(100, 125, 160, 225, 250))
```

```
predict(reg_eq_mtcars, new_values)
```

```
min(mtcars$hp)
```

```
max(mtcars$hp)
```

Exercise 5

```
plot(polling$x1, polling$x3, xlab = "Age", ylab = "Views of
      Same-Sex Marriage", pch=19)


reg_eq_polling <- lm(x3 ~ x1, data = polling)


summary(reg_eq_polling)


confint(reg_eq_polling, level = 0.95)
```

R Functions

. abline() Introduces a line to a plot. In this chapter, we use this function to include both the line of the estimated regression equation and a line representing the mean.

. confint() An extractor function that provides confidence interval

estimates of the parameters of a fitted regression model.

. fitted() An extractor function that provides the predicted (or fitted) values of the dependent variable for the model object. Only those values included in the original sample are fit to the model.

. lm(y~x, data=) Provides the intercept term b_0 and the regression coefficient b_1 for an estimated simple linear regression equation of the form $\hat{y} = b_0 + b_1x$.

. predict() An extractor function that predicts values of the independent variable not included in the original sample. If the new independent variable values are omitted as an argument, predict() provides the same predictions as fitted().

. resid() An extractor function that reports residuals; that is, the difference between the actual and predicted values, $y - \hat{y}$, for all values from original sample.

. summary() An extractor function that extracts (from the model object) the parameters, goodness-of-fit measures, and p-values.