MTH 3270 Notes 5

Tobias Boggess

2/28/2022

Section 9.3 Exercises

Exercise 1: Do the following.

a) Using a for() loop and rnorm(), simulate 1,000 random samples of size n = 10 from a N(50, 15) population (i.e. = 50 and = 15), compute the sample mean $\bar{X}$ of each sample, and store the $\bar{X}$ values in a 1,000-element vector named, say, sim.sample_means. Report your R command(s). Code:

```
sim.sample_means <- rep(NA, 1000)
for(i in 1:1000) {
  sim.sample <- rnorm(n = 10, mean = 50, sd = 15)
  sim.sample_means[i] <- mean(sim.sample)
}
head(sim.sample_means, 5)
```

```
## [1] 48.84105 51.49588 43.95355 53.74380 55.76600
```

b) Now use mean() and sd() to compute the mean and standard error of the 1,000 $\bar{X}$ values. Report these two values. Code:

```
sim.mean <- mean(sim.sample_means)
sim.mean
```

```
## [1] 49.95189
```

```
sim.sd <- sd(sim.sample_means)
sim.sd
```

```
## [1] 4.638603
```

c) Recall that if a random sample of size n is drawn from a N(,) population, the sampling distribution of $\bar{X}$ (obtained via mathematical theory) is N(, /√n).Compare the two values of Part a to the theoretical mean and standard error, and /√n, of the sampling distribution of $\bar{X}$. Code:

```
diff.x_bar <- sim.mean - 50
diff.x_bar
```

```
## [1] -0.04810946
```

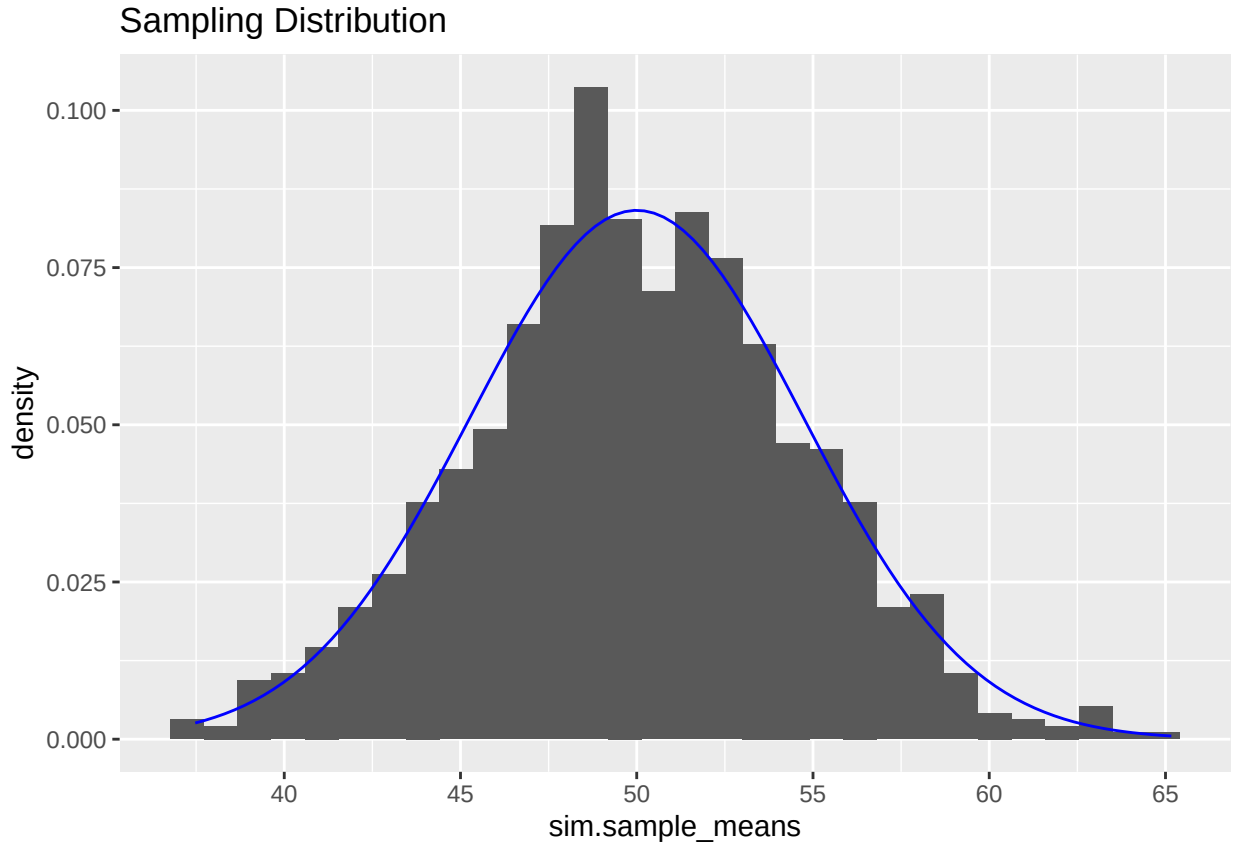
```
diff.x_sd <- sim.sd - (15 / sqrt(1000))
diff.x_sd
```

```
## [1] 4.164261
```

d) Make a histogram of the 1,000 simulated $\bar{X}$ values. Compare the shape, center, and spread of the histogram to the theoretical $N(\ ,\ /\sqrt{n})$ sampling distribution of $\bar{X}$. Code:

```
library(ggplot2)

ggplot(data = data.frame(sim.sample_means)) +
  geom_histogram(mapping = aes(x = sim.sample_means, y = stat(density))) +
  geom_function(fun = dnorm,
                args = list(mean = 50, sd = 15 / sqrt(10)),
                color = "blue") +
  labs(title = "Sampling Distribution")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Exercise 2: Simulate 1,000 random samples of size n = 5 from a N(50, 15) population (i.e. = 50 and = 15), and compute the four statistics below for each sample.In each case: 1) Report the mean and standard error of the simulated statistic values, and 2) Plot the simulated values in a histogram and describe the shape, center, and spread of this sampling distribution.

a) The sample median˜X (use median()).

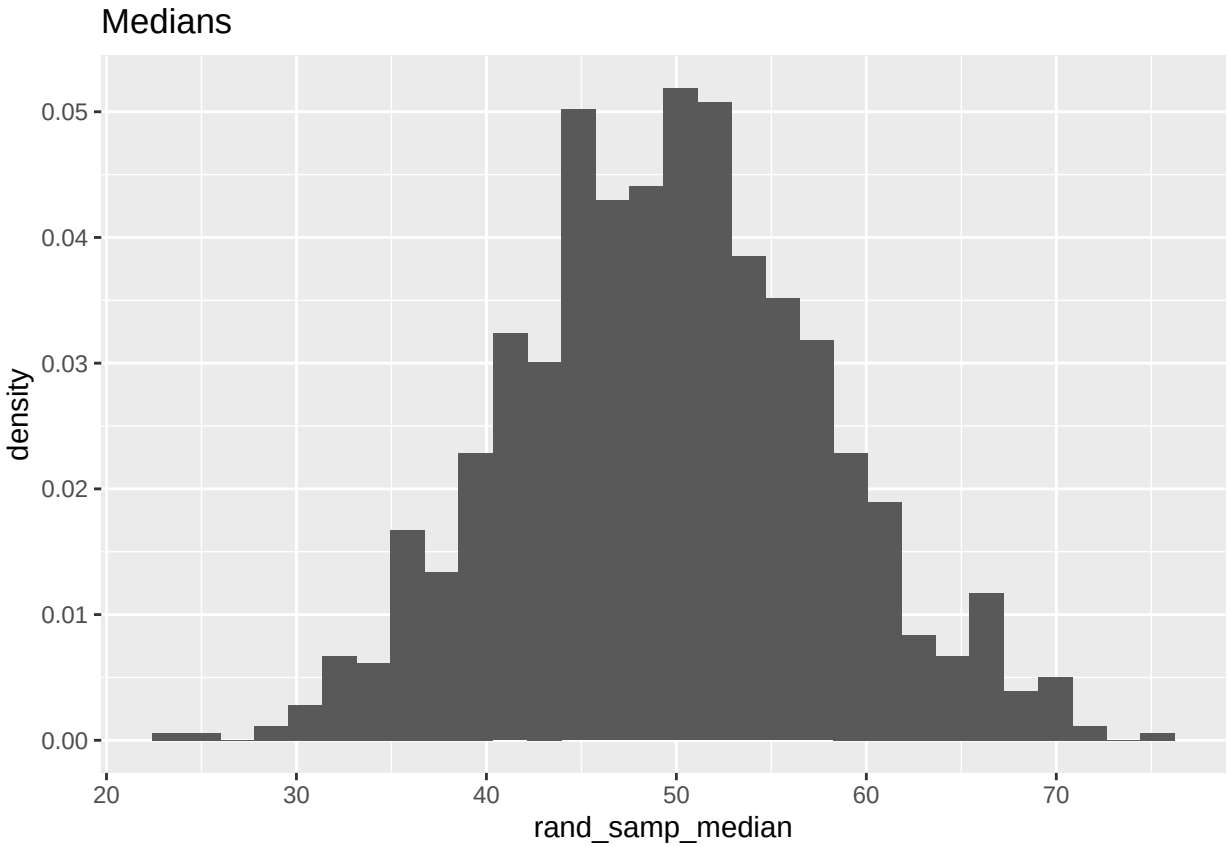
b) The sample standard deviation S (use sd()).

c) The sample minimum X(1) (use min()).

d) The sample maximum X(n) (use max()). Code:

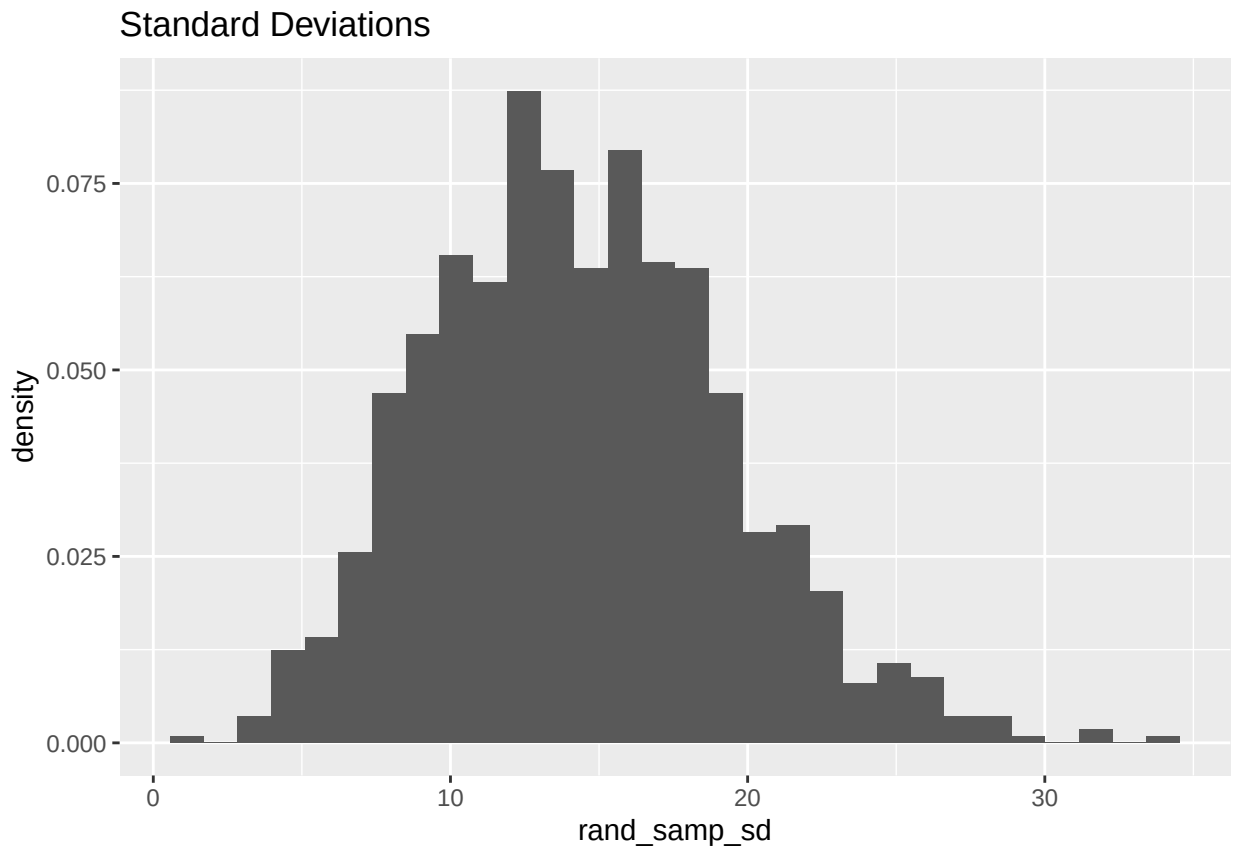
```
rand_samp_median <- rep(NA, 1000)
rand_samp_min <- rep(NA, 1000)
rand_samp_max <- rep(NA, 1000)
rand_samp_sd <- rep(NA, 1000)

for (i in 1:1000) {
  rand_samp <- rnorm(n = 5, mean = 50, sd = 15)
  rand_samp_median[i] <- median(rand_samp)
  rand_samp_sd[i] <- sd(rand_samp)
  rand_samp_min[i] <- min(rand_samp)
  rand_samp_max[i] <- max(rand_samp)
}

mean.of_median <- mean(rand_samp_median)
mean.of_sd <- mean(rand_samp_sd)
mean.of_min <- mean(rand_samp_min)
mean.of_max <- mean(rand_samp_max)

sd.of_median <- sd(rand_samp_median)
sd.of_sd <- sd(rand_samp_sd)
sd.of_min <- sd(rand_samp_min)
sd.of_max <- sd(rand_samp_max)


ggplot(data = data.frame(rand_samp_median)) +
  geom_histogram(mapping = aes(x = rand_samp_median, y = stat(density))) +
  labs(title = "Medians")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


Medians
density
0.05
0.04
0.03
0.02
0.01
0.00
20
30
40
50
60
70
rand_samp_median

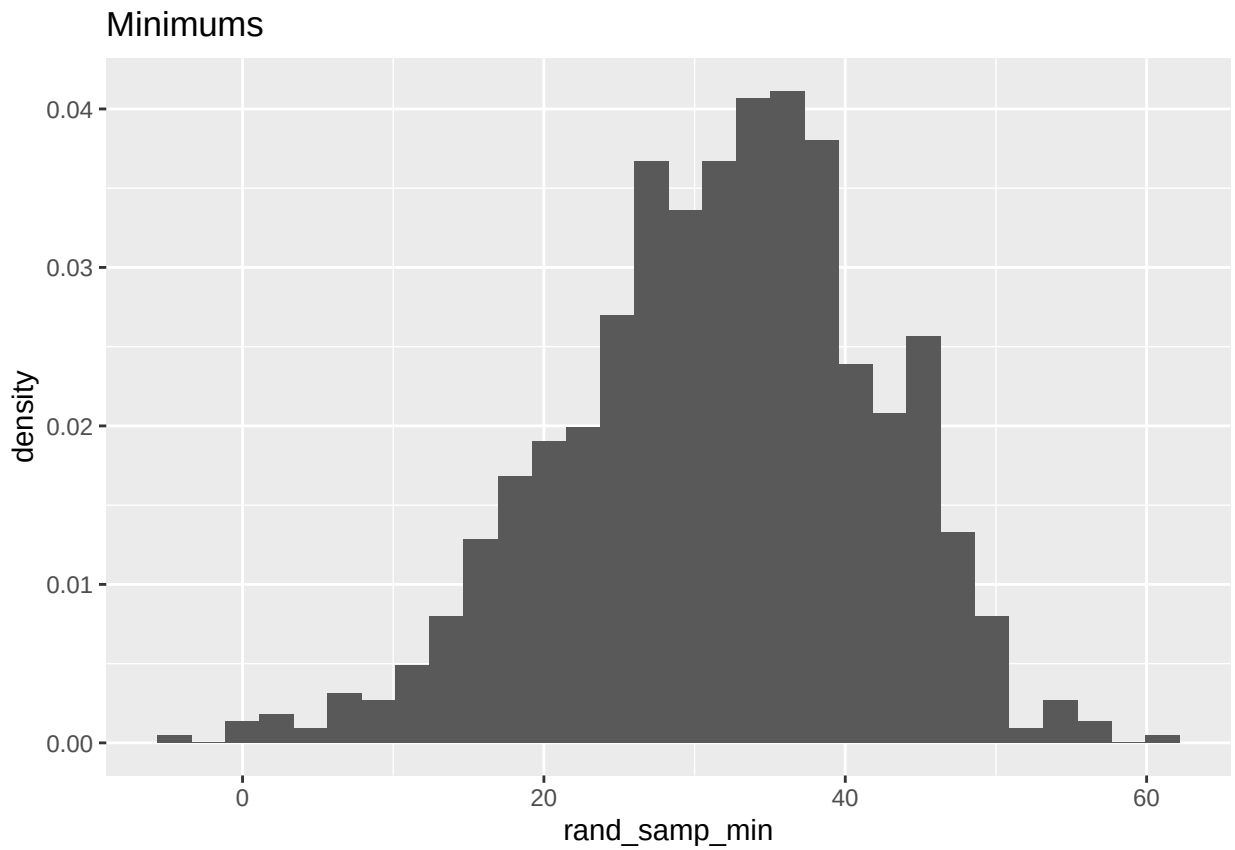
```
ggplot(data = data.frame(rand_samp_sd)) +
  geom_histogram(mapping = aes(x = rand_samp_sd, y = stat(density))) +
  labs(title = "Standard Deviations")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



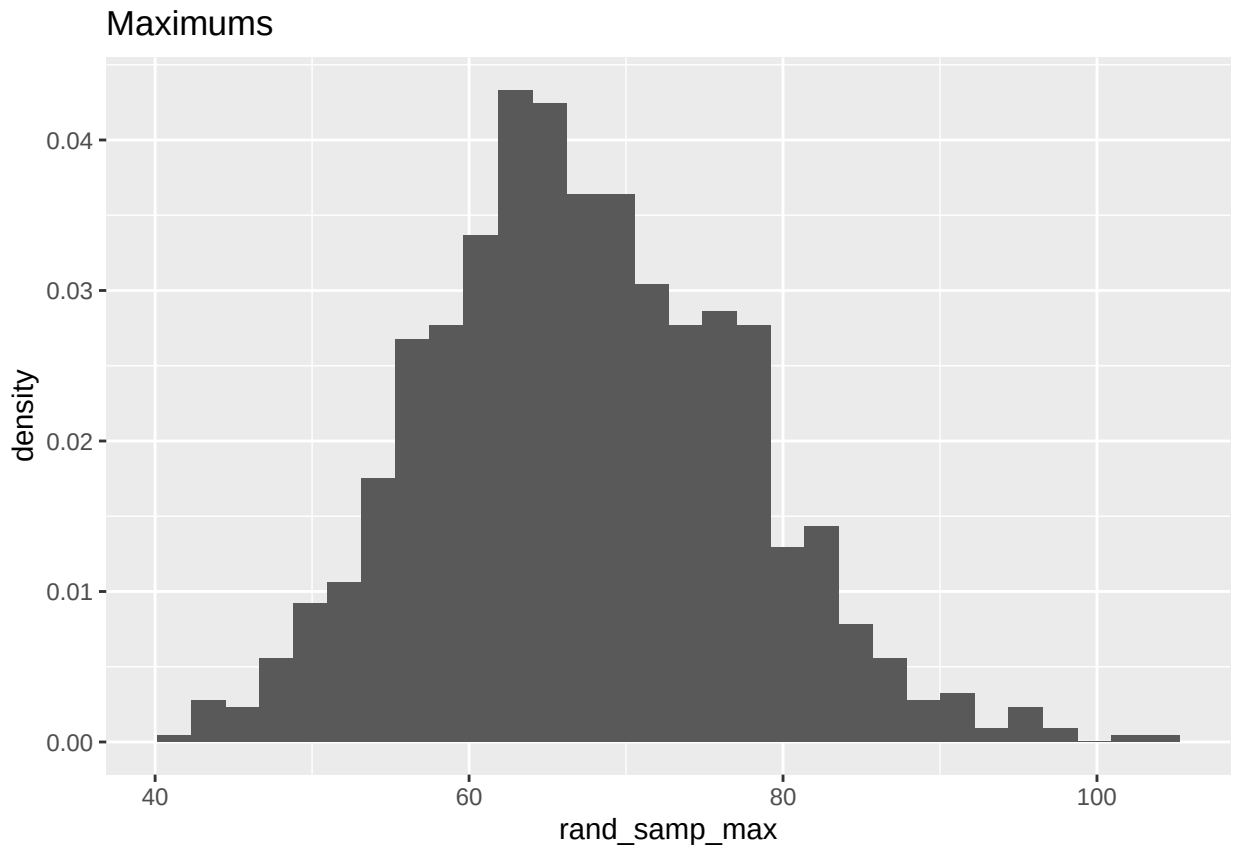
```
ggplot(data = data.frame(rand_samp_min)) +
  geom_histogram(mapping = aes(x = rand_samp_min, y = stat(density))) +
  labs(title = "Minimums")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
ggplot(data = data.frame(rand_samp_max)) +
  geom_histogram(mapping = aes(x = rand_samp_max, y = stat(density))) +
  labs(title = "Maximums")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
mean.of_median
```

```
## [1] 49.57514
```

```
mean.of_sd
```

```
## [1] 14.37469
```

```
mean.of_min
```

```
## [1] 31.76712
```

```
mean.of_max
```

```
## [1] 67.32334
```

```
sd.of_median
```

```
## [1] 8.204169
```

```
sd.of_sd
```

```
## [1] 4.999422
```

```
sd.of_min
```

```
## [1] 10.08694
```

```
sd.of_max
```

```
## [1] 9.944542
```

Section 9.4 Exercises

Exercise 3: Use the bootstrap method to simulate B = 1,000 resamples each of size n = 150 from the original iris data set, and using the Petal.Width variable, compute the four statistics below for each bootstrap resample.

Code:

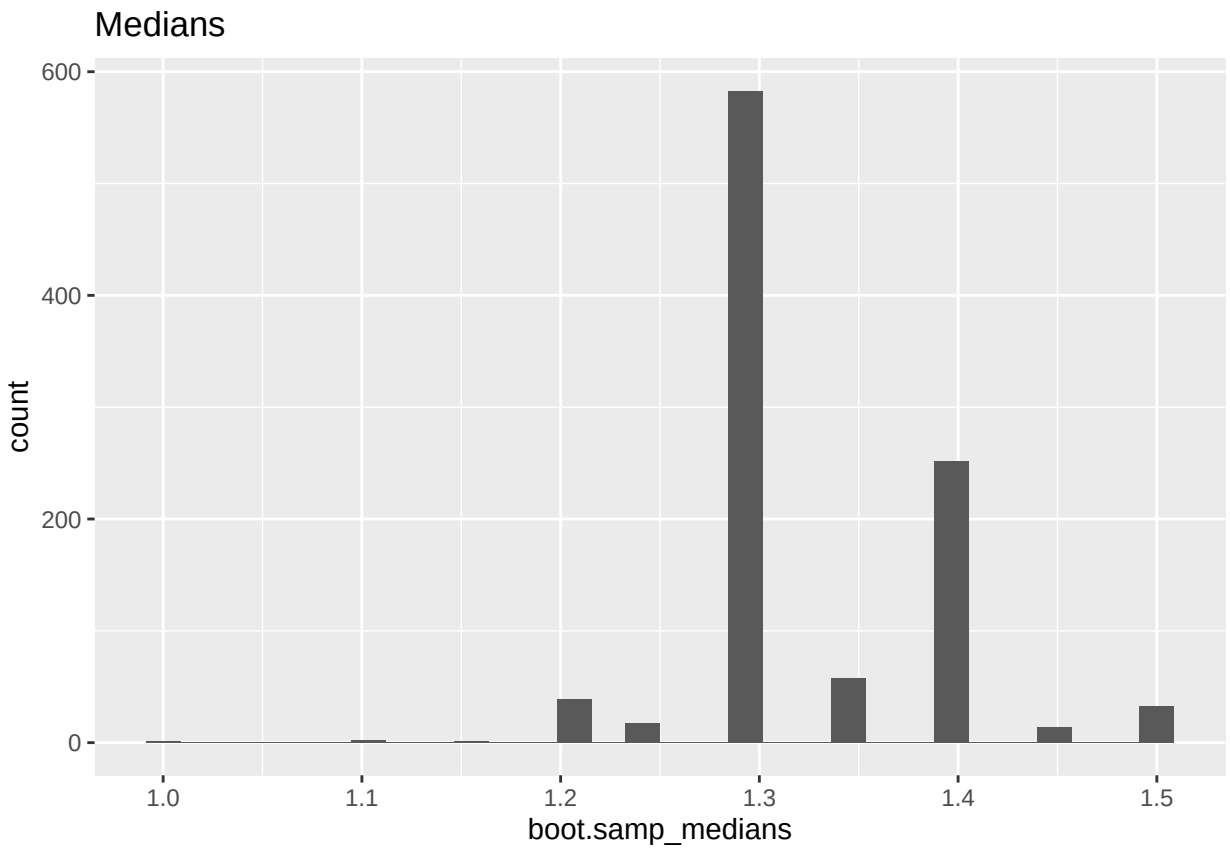
```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
n.rows <- nrow(iris)
B <- 1000
boot.samp_medians <- rep(NA, B)
boot.samp_sds <- rep(NA, B)
boot.samp_mins <- rep(NA, B)
boot.samp_maxs <- rep(NA, B)

set.seed(21)
```

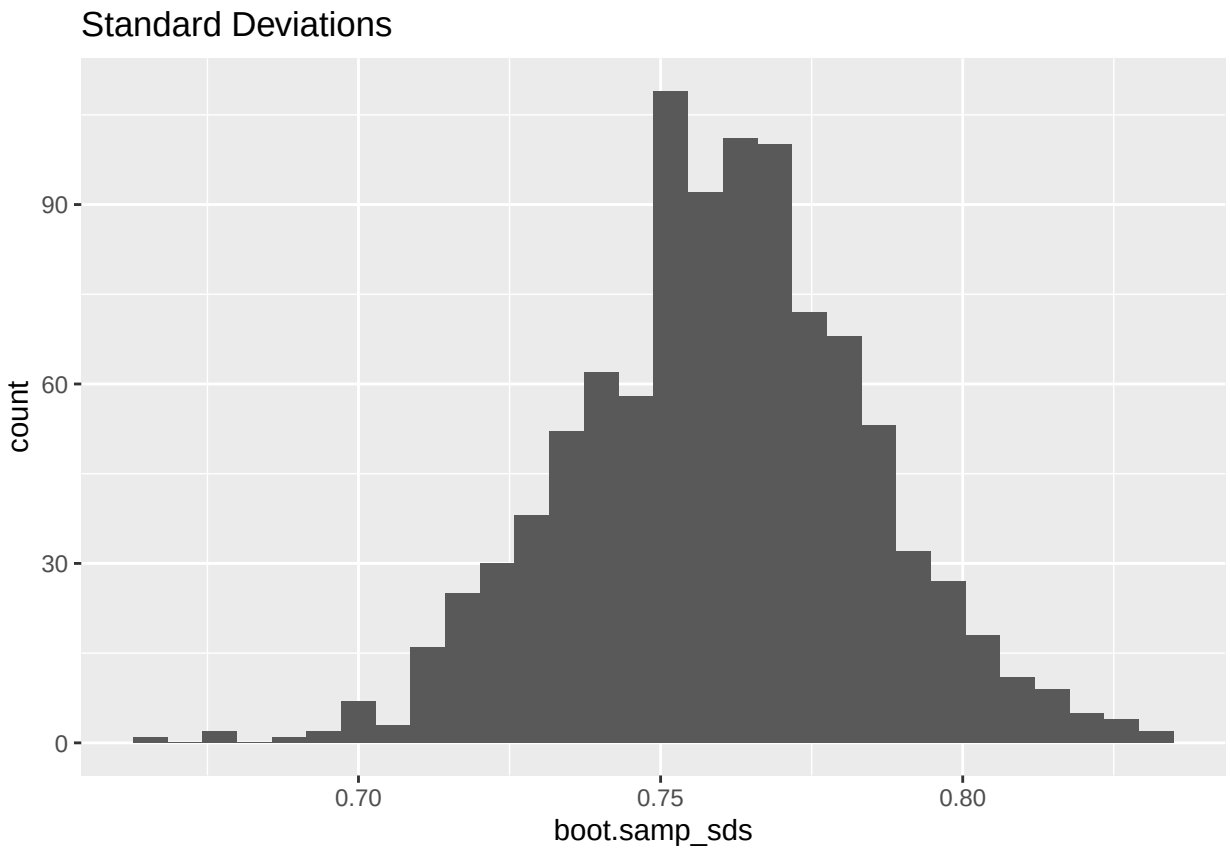
```r
for (i in 1:B) {
  resamp <- slice_sample(.data = iris,
                         n = n.rows,
                         replace = TRUE)
  boot.samp_medians[i] <- median(resamp$Petal.Width)
  boot.samp_sds[i] <- sd(resamp$Petal.Width)
  boot.samp_mins[i] <- min(resamp$Petal.Width)
  boot.samp_maxs[i] <- max(resamp$Petal.Width)
}

ggplot(data = data.frame(boot.samp_medians)) +
  geom_histogram(mapping = aes(x = boot.samp_medians)) +
  labs(title = "Medians")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



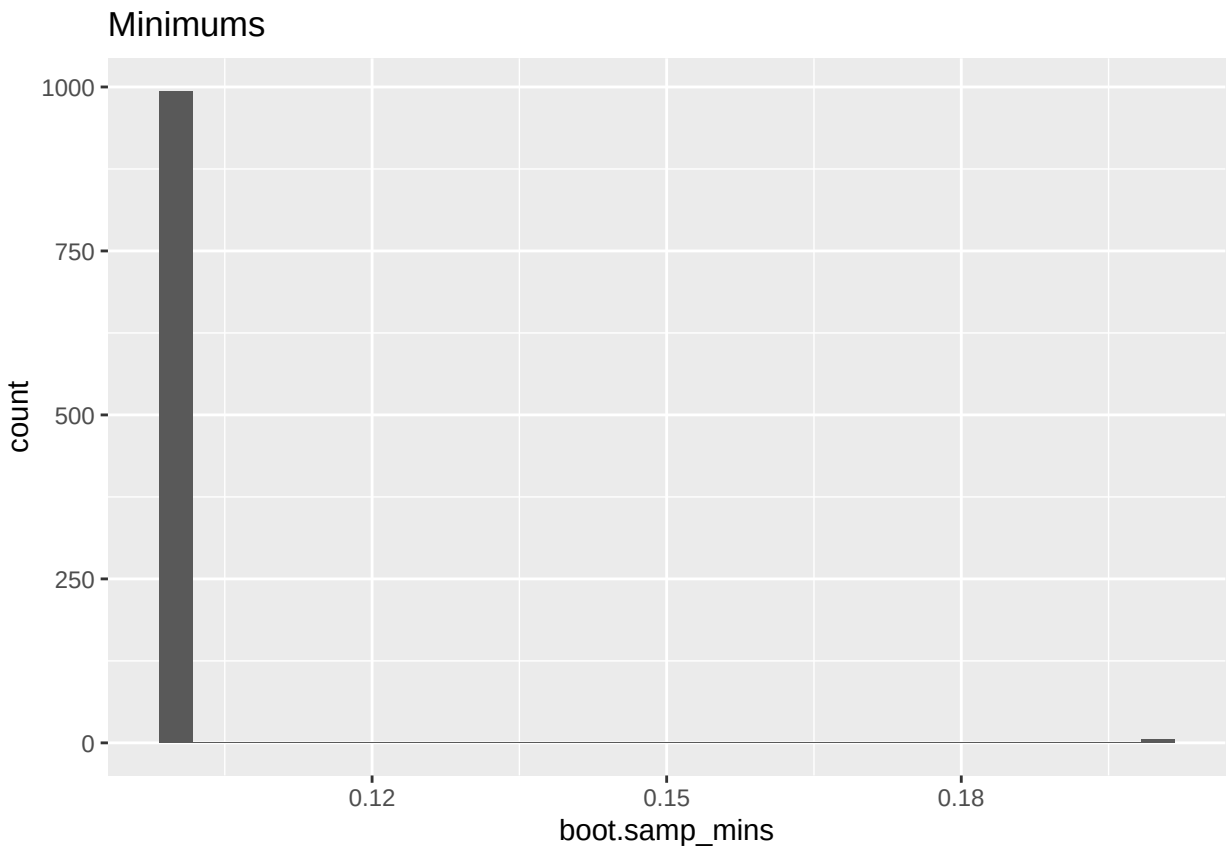
```
ggplot(data = data.frame(boot.samp_sds)) +
  geom_histogram(mapping = aes(x = boot.samp_sds)) +
  labs(title = "Standard Deviations")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



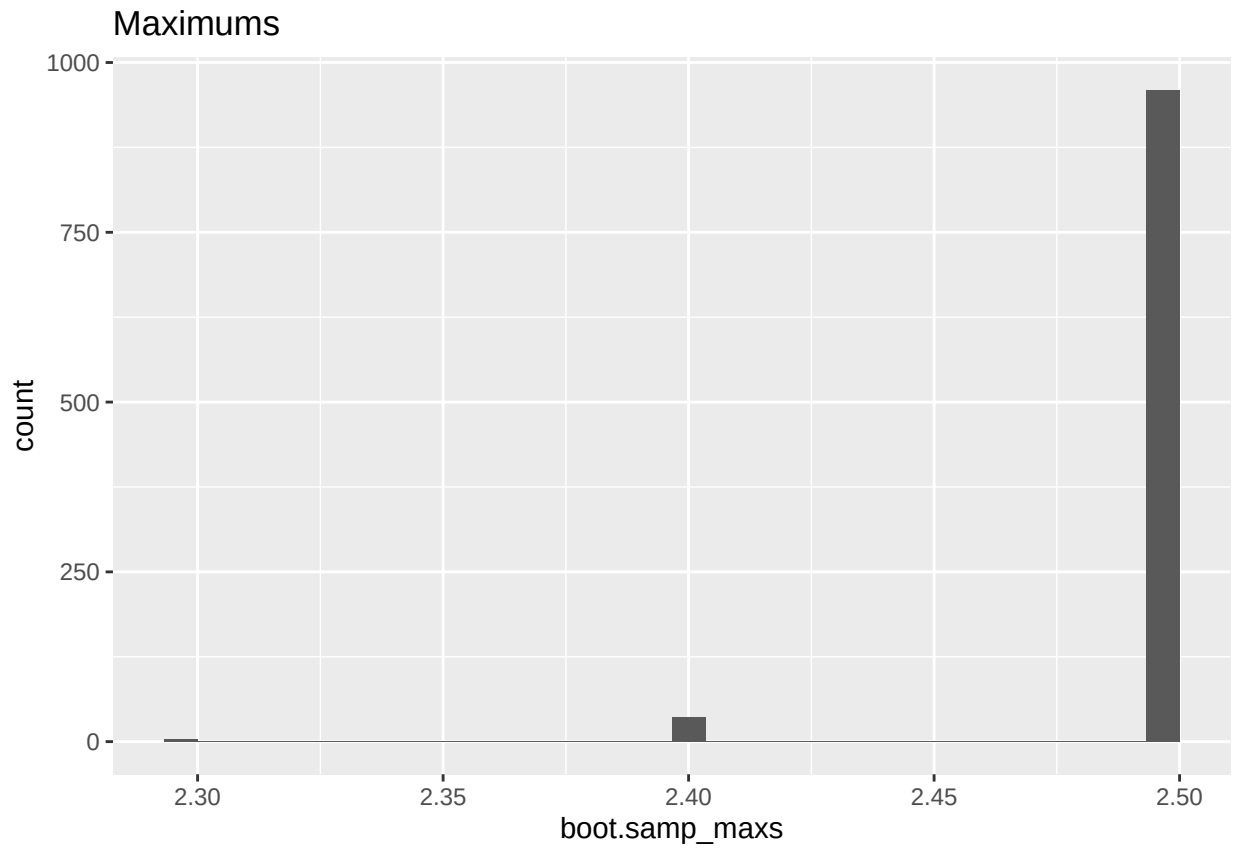
```
ggplot(data = data.frame(boot.samp_mins)) +
  geom_histogram(mapping = aes(x = boot.samp_mins)) +
  labs(title = "Minimums")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
ggplot(data = data.frame(boot.samp_maxs)) +
  geom_histogram(mapping = aes(x = boot.samp_maxs)) +
  labs(title = "Maximums")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
median.mean <- mean(boot.samp_medians)
sd.mean <- mean(boot.samp_sds)
min.mean <- mean(boot.samp_mins)
max.mean <- mean(boot.samp_maxs)

median.sd <- sd(boot.samp_medians)
sd.sd <- sd(boot.samp_sds)
min.sd <- sd(boot.samp_mins)
max.sd <- sd(boot.samp_maxs)

median.mean
```

```
## [1] 1.3312
```

```
sd.mean
```

```
## [1] 0.7598777
```

```
min.mean
```

```
## [1] 0.1006
```

```
max.mean
```

```
## [1] 2.4956
```

```
median.sd
```

```
## [1] 0.06289289
```

```
sd.sd
```

```
## [1] 0.02445273
```

```
min.sd
```

```
## [1] 0.007726558
```

```
max.sd
```

```
## [1] 0.02238618
```

Section 9.5 Exercises

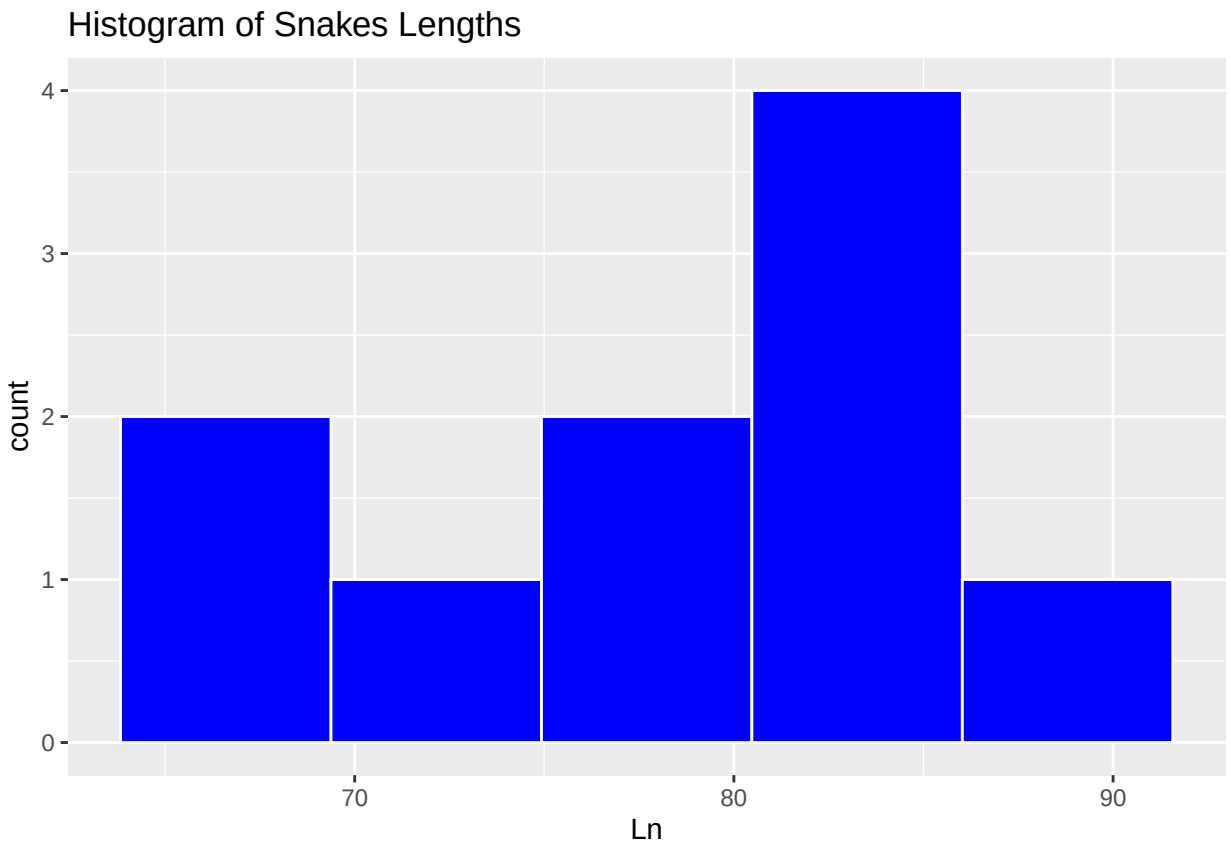
Exercise 4: Do the following.

Dataframe:

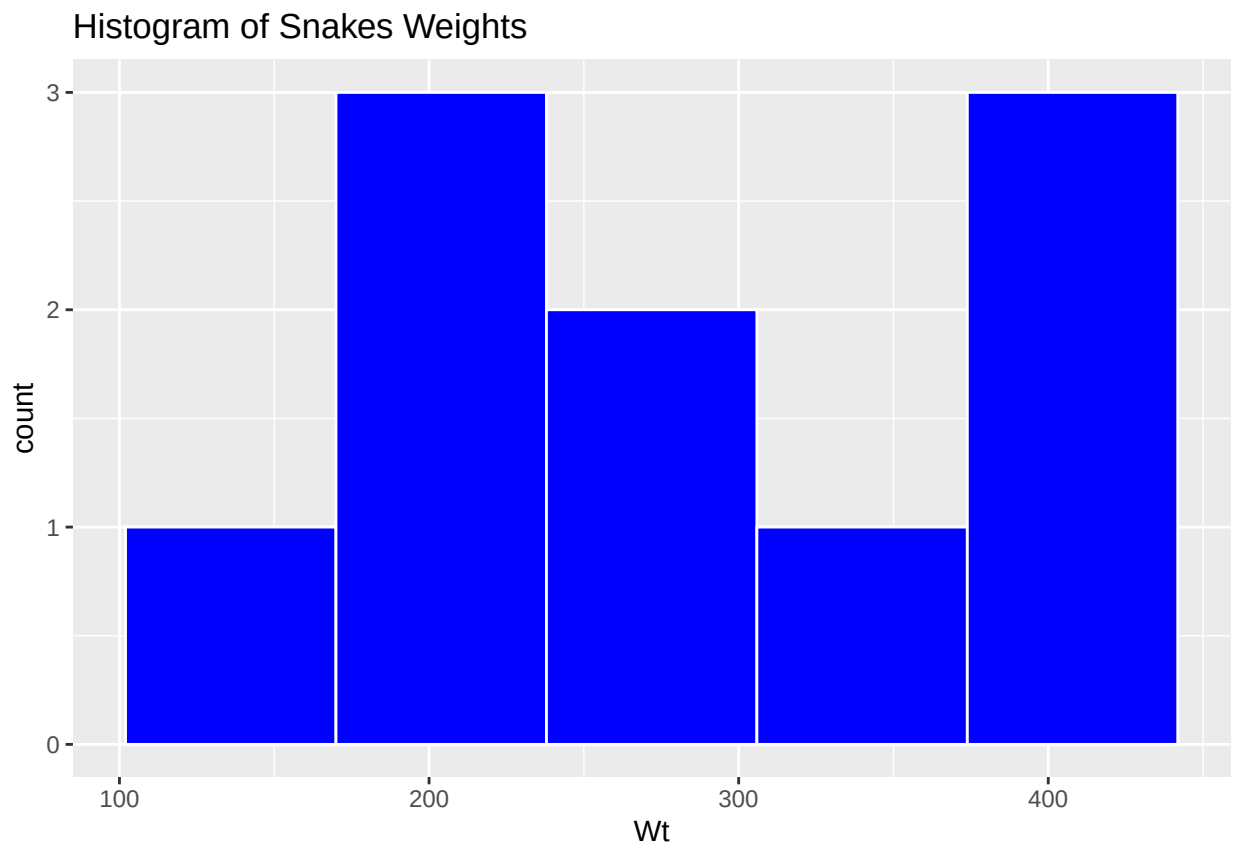
```
SnakeID <- 1:10
Ln <- c(85.7, 64.5, 84.1, 82.5, 78.0, 65.9, 81.3, 71.0, 86.7, 78.7)
Wt <-
  c(331.9,
    121.5,
    382.2,
    287.3,
    224.3,
    380.4,
    245.2,
    208.2,
    393.4,
    228.3)
Snakes <- data.frame(SnakeID, Ln, Wt)
```

a) Can you identify the outlier in either of these univariate graphs (histograms)? Code:

```
ggplot(data = Snakes) +
  geom_histogram(
    mapping = aes(x = Ln),
    fill = "blue",
    color = "white",
    bins = 5
  ) +
  ggtitle("Histogram of Snakes Lengths")
```



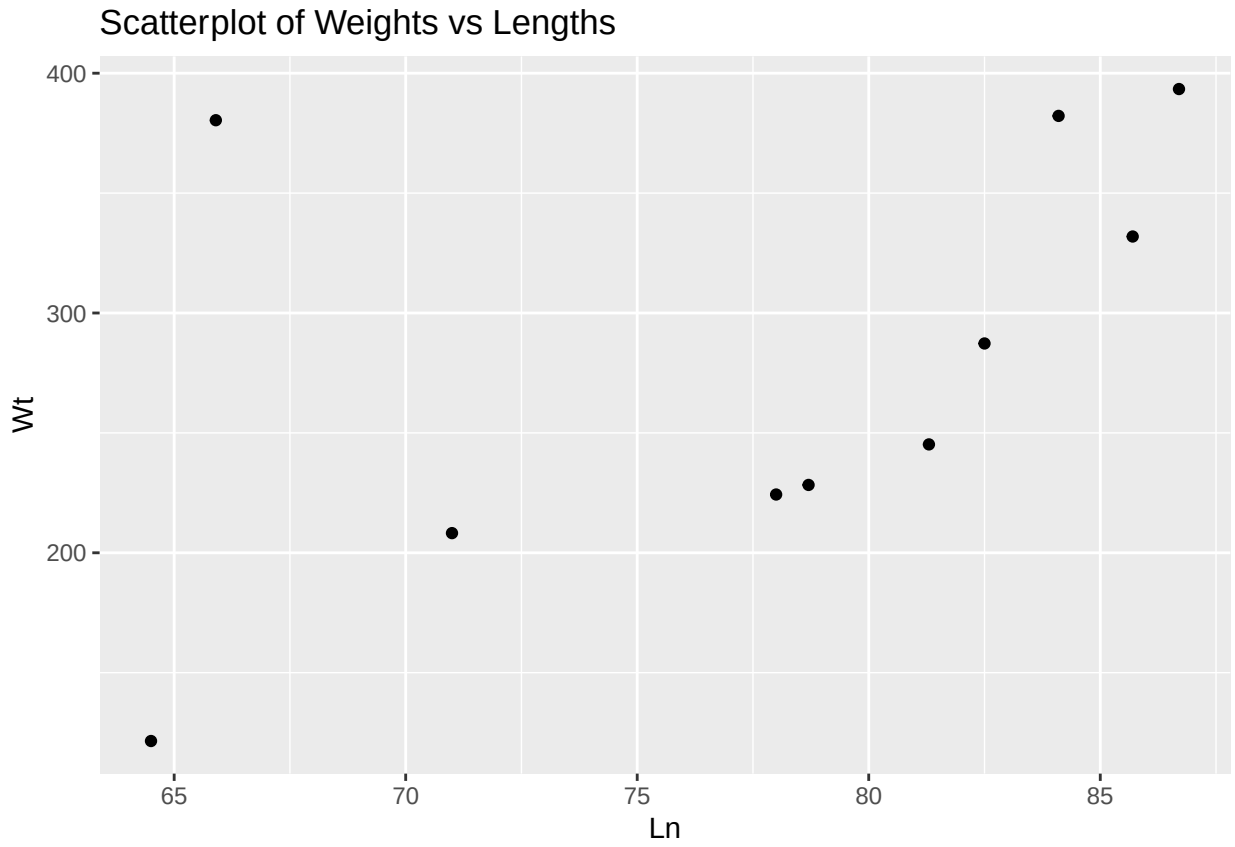
```
ggplot(data = Snakes) +
  geom_histogram(
    mapping = aes(x = Wt),
    fill = "blue",
    color = "white",
    bins = 5
  ) +
  ggtitle("Histogram of Snakes Weights")
```



The only outlier in the two graphs is the weight of the snakes. The outlier value is 121.5.

b) Can you identify the outlier in this bivariate graph (scatterplot)? If so, which snake (SnakeID) is the outlier? Code:

```
ggplot(data = Snakes) +
  geom_point(mapping = aes(x = Ln, y = Wt)) +
  ggtitle("Scatterplot of Weights vs Lengths")
```



The major outlier is the snake whose length is 65.9 with a weight of 380.4 and the second most significant outlier is the snake with a 64.5 length and a 121.5 weight.

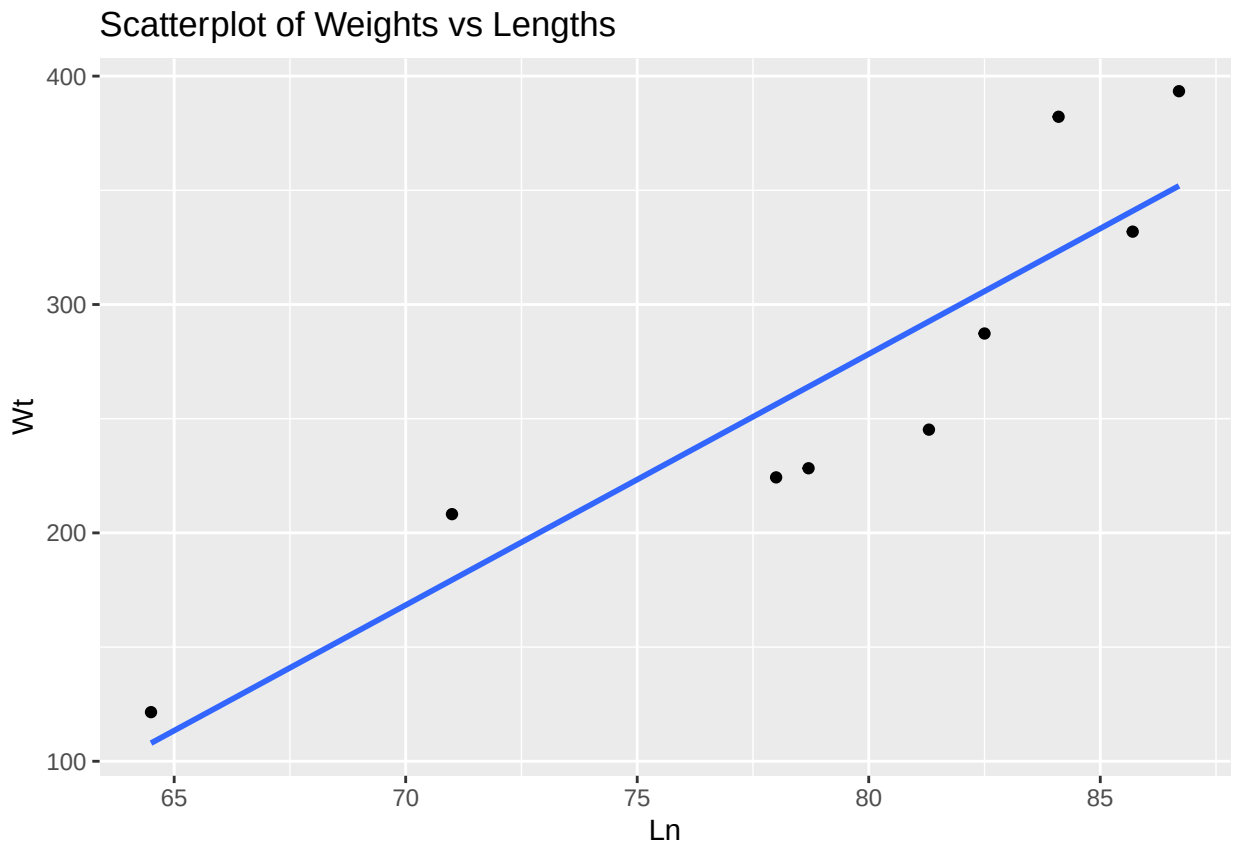
Section 9.6 Exercises

Exercise 5: Do the following.

Data frame:

```
SnakeID <- 1:9
Ln <- c(85.7, 64.5, 84.1, 82.5, 78.0, 81.3, 71.0, 86.7, 78.7)
Wt <-
  c(331.9, 121.5, 382.2, 287.3, 224.3, 245.2, 208.2, 393.4, 228.3)
Snakes <- data.frame(SnakeID, Ln, Wt)

ggplot(data = Snakes, mapping = aes(x = Ln, y = Wt)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  ggtitle("Scatterplot of Weights vs Lengths")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
my.reg <- lm(Wt ~ Ln, data = Snakes)
summary(my.reg)
```

```
##
## Call:
## lm(formula = Wt ~ Ln, data = Snakes)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.395 -32.020  -9.061  28.826  58.827
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -601.083    154.333  -3.895 0.005939 **
## Ln            10.992      1.942   5.660 0.000767 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.94 on 7 degrees of freedom
## Multiple R-squared:  0.8207, Adjusted R-squared:  0.795
## F-statistic: 32.03 on 1 and 7 DF,  p-value: 0.0007667
```

a) Obtain the predicted weight for a snake whose length is 80 cm in two ways:

1. By plugging 80 into the equation for X.
2. By using predict(). Report both sets of your R commands for obtaining the predicted weight.
   Code:

```
pred_val <- -601.08 + 10.99 * 80
pred_val
```

```
## [1] 278.12
```

```
newLen <- data.frame(Ln = 80)
predict(my.reg, newdata = newLen)
```

```
##        1
## 278.3047
```

b) What's a typical change in weight for each 1 cm elongation? What about for a 5 cm elongation?
Code:

```
newLen <- data.frame(Ln = c(80, 81))
predict(my.reg, newdata = newLen)
```

```
##        1        2
## 278.3047 289.2971
```

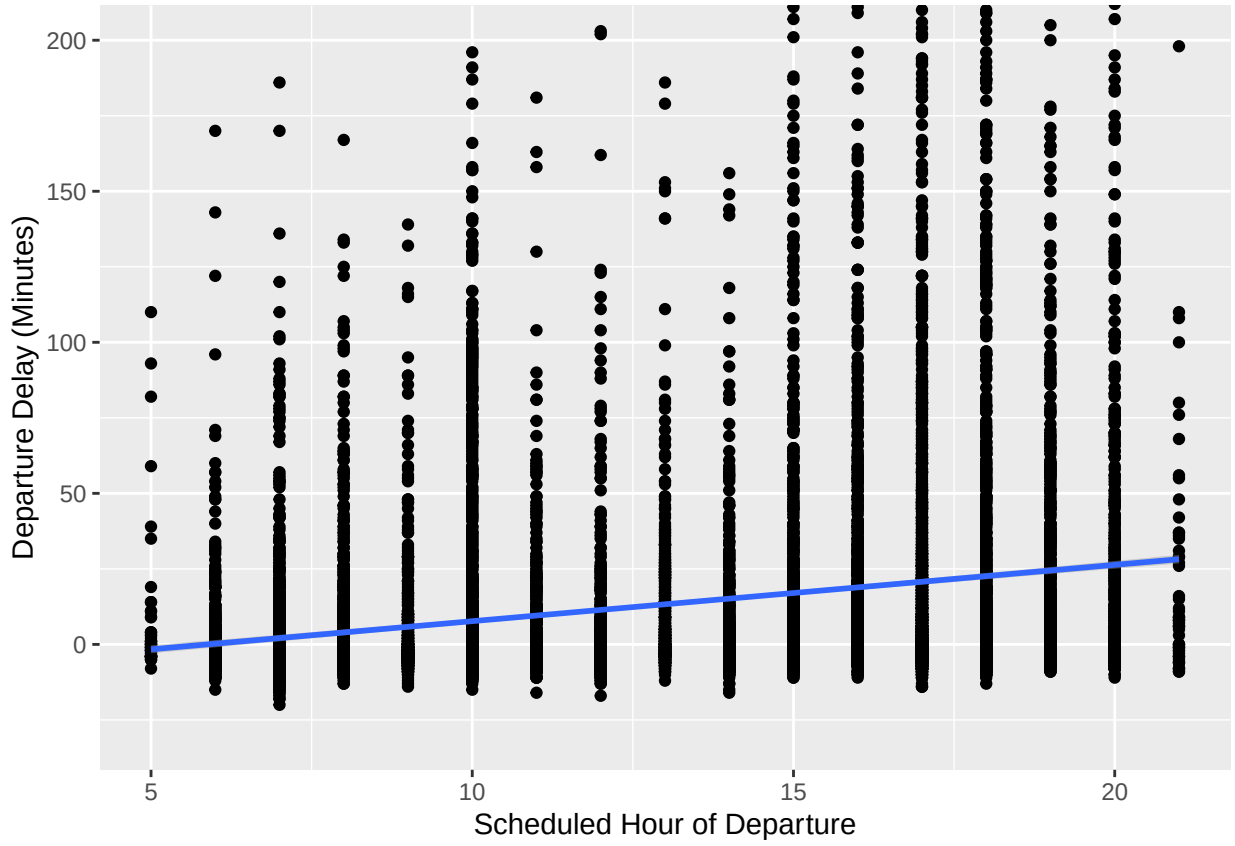
For every centimeter change in length, the weight should increase by approximately 11 and for five cm it should be approximately 55.

Exercise 6: Do the following.

Data frame:

```
library(nycflights13)
SF <- filter(.data = flights, dest == "SFO", !is.na(arr_delay))

ggplot(data = SF, mapping = aes(x = hour, y = dep_delay)) +
  geom_point() +
  geom_smooth(method = "lm") +
  xlab("Scheduled Hour of Departure") +     ylab("Departure Delay (Minutes)") +
  coord_cartesian(ylim = c(-30, 200))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



a) Use lm() and summary() to obtain the equation of the fitted regression line, with with dep_delay as the response (Y) and hour as the explanatory variable (X). Report the equation of the fitted line.
Code:

```
my.reg2 <- lm(dep_delay ~ hour, data = SF)
summary(my.reg2)
```

```
##
## Call:
## lm(formula = dep_delay ~ hour, data = SF)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -37.33 -17.42  -8.74  -1.10 991.39
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -10.95593    1.03591  -10.58   <2e-16 ***
## hour          1.86451    0.07692   24.24   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.34 on 13171 degrees of freedom
## Multiple R-squared:  0.0427, Adjusted R-squared:  0.04263
## F-statistic: 587.5 on 1 and 13171 DF,  p-value: < 2.2e-16
```

The equation of the fitted line is Y = -10.95593 + 1.86451X.

b) Obtain the predicted departure delay for a flight whose departure hour is 15 in two ways:

1. By plugging 15 into the equation for X.
2. By using predict(). Report both sets of your R commands for obtaining the predicted departure delay. Code:

```
-10.95593 + 1.86451 * 15
```

```
## [1] 17.01172
```

```
newHour <- data.frame(hour = 15)
predict(my.reg2, newdata = newHour)
```

```
##        1
## 17.01166
```

The departure delay will increase by 1.86451 for each hour that is additional.

Section 9.6 Exercises

Exercise 7: Do the following.

Data frame:

```
SnakeID <- 1:9
Ln <- c(85.7, 64.5, 84.1, 82.5, 78.0, 81.3, 71.0, 86.7, 78.7)
Wt <-
  c(331.9, 121.5, 382.2, 287.3, 224.3, 245.2, 208.2, 393.4, 228.3)
Snakes <- data.frame(SnakeID, Ln, Wt)
my.reg <- lm(Wt ~ Ln, data = Snakes)
```

a) What class of object is returned by lm()? Find out by typing:
Code:

```
class(my.reg)
```

```
## [1] "lm"
```

b) The "lm" class of objects is a special case of the "list" class. What does the following return?
Code:

```
is.list(my.reg)
```

```
## [1] TRUE
```

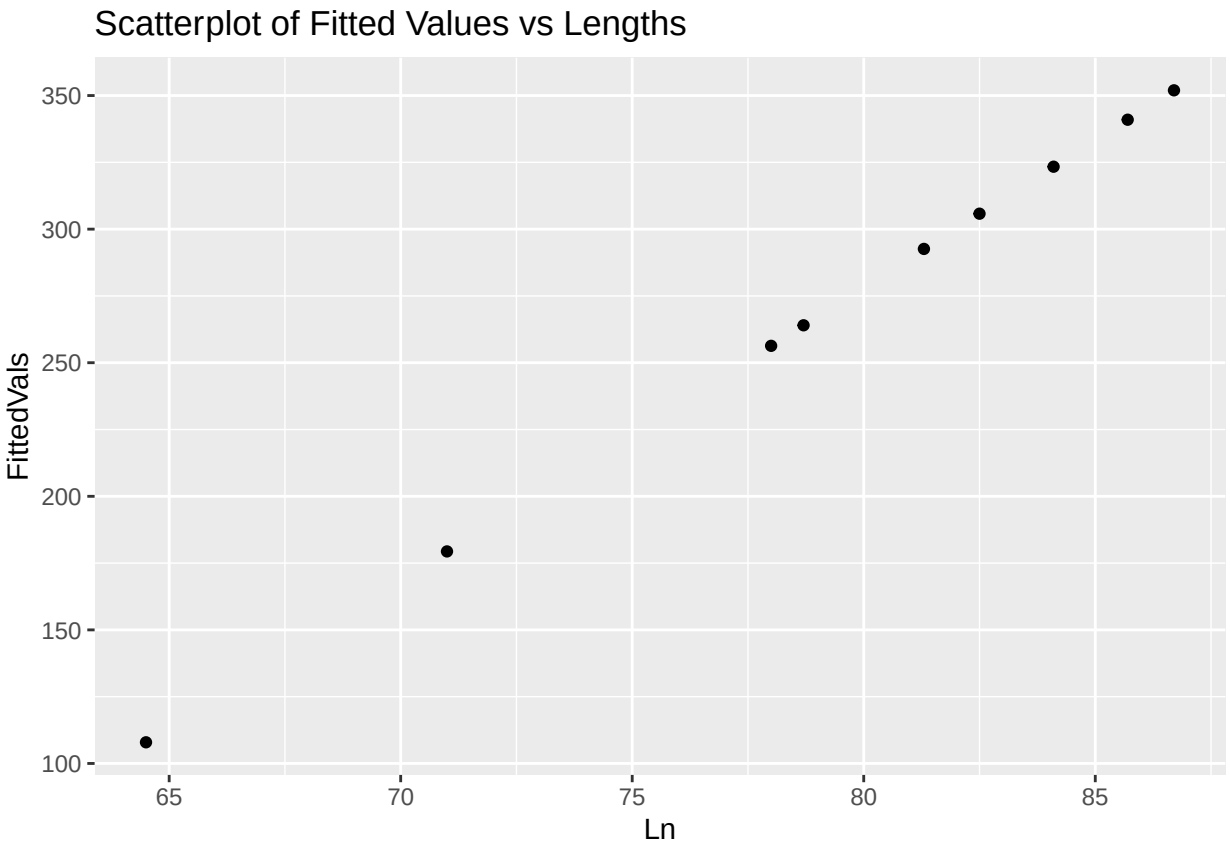
c) How many objects are contained in the my.reg list? Find out by looking at their names:
Code:

```
names(my.reg)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

d) What would a plot of the fitted values versus the lengths look like? Try it, and describe the result:
Code:

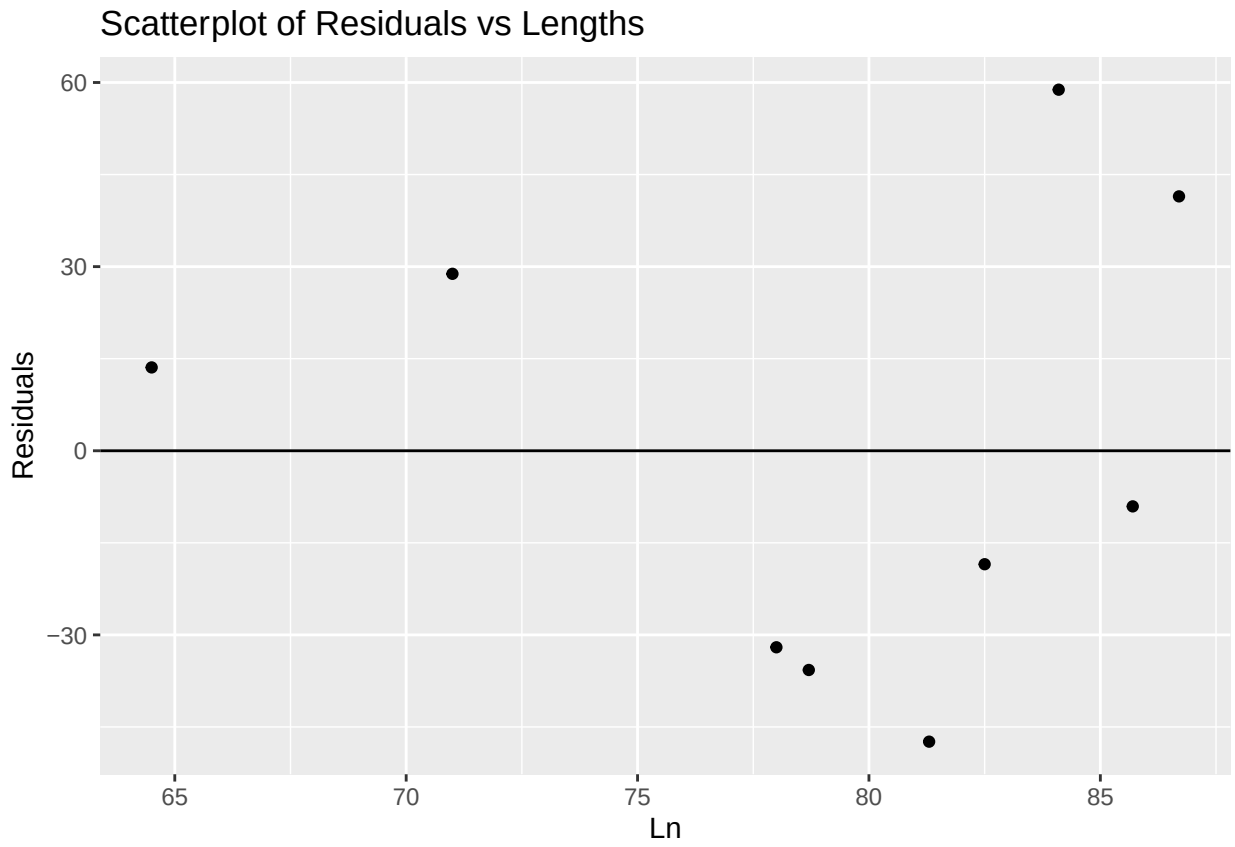
```
library(dplyr) # For mutate()
Snakes <- mutate(Snakes, FittedVals = my.reg$fitted.values)
ggplot(data = Snakes, mapping = aes(x = Ln, y = FittedVals)) +
  geom_point() +
  ggtitle("Scatterplot of Fitted Values vs Lengths")
```



The output is almost a perfect linear representation of the snakes data.

e) What would a plot of the residuals versus the lengths look like? Try it (with a horizontal line at y = 0) and describe the result:
Code:

```
Snakes <- mutate(Snakes, Residuals = my.reg$residuals)
ggplot(data = Snakes, mapping = aes(x = Ln, y = Residuals)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  ggtitle("Scatterplot of Residuals vs Lengths")
```



The residuals seem pretty far off compared to the horizontal line.

f) Show that the residuals sum to zero:
Code:

```
sum(Snakes$Residuals)
```

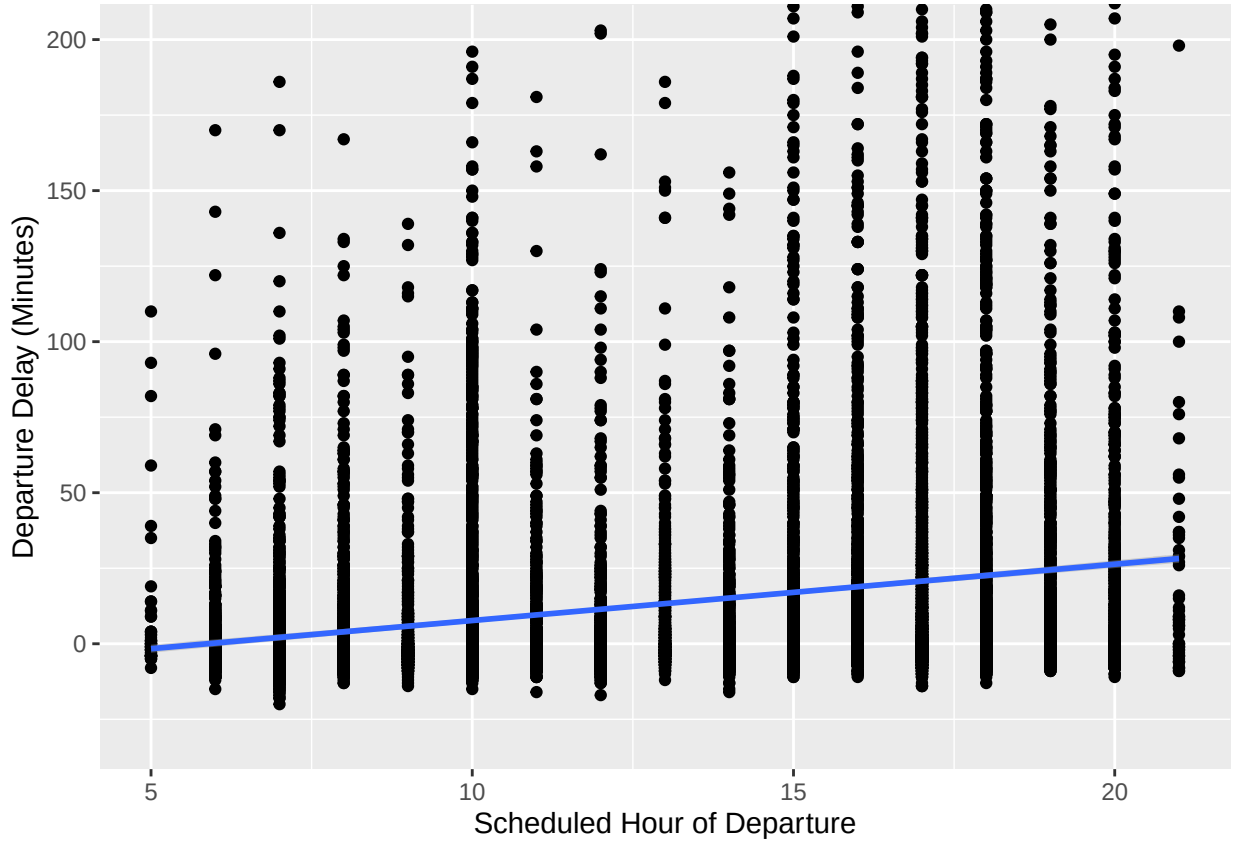
```
## [1] -1.065814e-14
```

Exercise 8: Do the following.

Code:

```
ggplot(data = SF, mapping = aes(x = hour, y = dep_delay)) +
  geom_point() +
  geom_smooth(method = "lm") +
  xlab("Scheduled Hour of Departure") + ylab("Departure Delay (Minutes)") +
  coord_cartesian(ylim = c(-30, 200))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```r
my.reg <- lm(dep_delay ~ hour, data = SF)
summary(my.reg)
```

```
##
## Call:
## lm(formula = dep_delay ~ hour, data = SF)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -37.33 -17.42  -8.74  -1.10 991.39
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -10.95593    1.03591  -10.58   <2e-16 ***
## hour          1.86451    0.07692   24.24   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.34 on 13171 degrees of freedom
## Multiple R-squared:  0.0427, Adjusted R-squared:  0.04263
## F-statistic: 587.5 on 1 and 13171 DF,  p-value: < 2.2e-16
```

a) From the output of summary(), what's the value of the residual standard error?
The residual standard error is: 39.34.

b) From the output of summary(), what's the value of R2 (labeled Multiple R-squared)?
The value of R2 is 0.0427.

c) Using the criteria below (and the R2 from part b), how well does the linear model fit the data (poor, medium, or good)?
The model fit based on the R2 section shows the fit is poor.

Exercise 9: Do the following.

Data frame:

```
Sales <- c(
  174.4,
  164.4,
  244.2,
  154.6,
  181.6,
  207.5,
  152.8,
  163.2,
  145.4,
  137.2,
  241.9,
  191.1,
  232.0,
  145.3,
  161.1,
  209.7,
  146.4,
  144.0,
  232.6,
  224.1,
  166.5
)
Under16 <- c(
  68.5,
  45.2,
  91.3,
  47.8,
  46.9,
  66.1,
  49.5,
  52.0,
  48.9,
  38.4,
  87.9,
  72.8,
  88.4,
  42.9,
  52.5,
  85.7,
  41.3,
  51.7,
  89.6,
  82.7,
  52.3
)
```

```
Income <-
  c(
    16.7,
    16.8,
    18.2,
    16.3,
    17.3,
    18.2,
    15.9,
    17.2,
    16.6,
    16.0,
    18.3,
    17.1,
    17.4,
    15.8,
    17.8,
    18.4,
    16.5,
    16.3,
    18.1,
    19.1,
    16.0
  )
portraitSales <- data.frame(Sales, Under16, Income)
```

a) Use lm() to fit a multiple regression model with portrait Sales as the response (Y) and number of persons Under16 (X1) and per capita Income (X2) as explanatory variables. Then use summary() to obtain the results. Write out the equation of the fitted plane.
Code:

```
my.reg3 <- lm(Sales ~ Under16 + Income, data = portraitSales)
summary(my.reg3)
```

```
##
## Call:
## lm(formula = Sales ~ Under16 + Income, data = portraitSales)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.4239  -6.2161   0.7449   9.4356  20.2151
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -68.8571    60.0170  -1.147   0.2663
## Under16       1.4546     0.2118   6.868    2e-06 ***
## Income        9.3655     4.0640   2.305   0.0333 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.01 on 18 degrees of freedom
## Multiple R-squared:  0.9167, Adjusted R-squared:  0.9075
## F-statistic:  99.1 on 2 and 18 DF,  p-value: 1.921e-10
```

Equation: Y = -68.8571 + 1.4546 * X1 + 9.3655 * X2.

b) Obtain the predicted sales for a city whose number of persons under 16 is 45.0 (thousand) and whoseper capita income is 17.0 (thousand dollars) in two ways:

1. By plugging 45.0 and 17.0 into the equation for X1 and X2.
2. By using predict(). Report both sets of your R commands for obtaining the predicted sales.
   Code:

```
-68.8471 + 1.4546 * 45 + 9.3655 * 17
```

```
## [1] 155.8234
```

```
newUndInc <- data.frame(Under16 = 45.0, Income = 17.0)
predict(my.reg3, newdata = newUndInc)
```

```
##        1
## 155.8116
```

c) By how much do we expect sales to increase for each additional 1.0 thousand people under 16 (holding income constant)?
We can expect sales to increase by approximately 1.4546 (thousand) for each increase in people under 16 holding income constant.

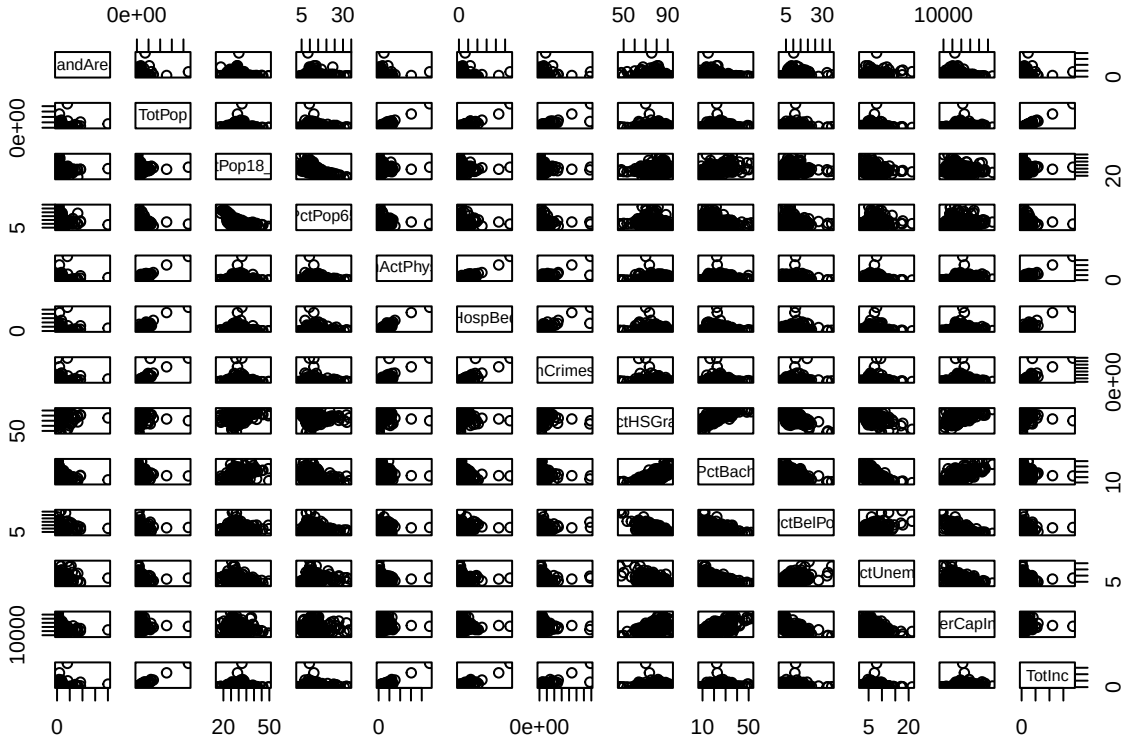
d) By how much do we expect sales to increase for each additional 1.0 thousand dollars in per capita income (holding number of people under 16 constant)?
We can expect an approximate gain of 9.3655 gain of sales based on the income holding the number of people under 16 constant.

Exercise 10: Do the following using the cdi data set.

Load in Library:

a) Use pairs() to make a scatterplot matrix of these variables. Report your R command.
Code:

```
pairs(select(cdi, -c(ID, County, State, Region)))
```



b) Use cor() to make the correlation matrix of the data. Report your R command.
Code:

```
cor_cdi <- cor(select(cdi, LandArea:TotInc))
head(cor_cdi, 3)
```

```
##                LandArea     TotPop PctPop18_34     PctPop65   nActPhys
## LandArea     1.00000000 0.17308335 -0.05487812  0.005770871 0.07807466
## TotPop       0.17308335 1.00000000  0.07837212 -0.029037393 0.94024859
## PctPop18_34 -0.05487812 0.07837212  1.00000000 -0.616309639 0.11969924
##              nHospBeds    nCrimes   PctHSGrad    PctBach  PctBelPov
## LandArea    0.07304727 0.12947537 -0.09859811 -0.1372377 0.17134335
## TotPop      0.92373836 0.88633185 -0.01742690  0.1468138 0.03801951
## PctPop18_34 0.07453191 0.08994063  0.25058429  0.4560970 0.03397551
##                PctUnemp   PerCapInc     TotInc
## LandArea     0.199209277 -0.18771513 0.12707426
## TotPop       0.005351703  0.23561019 0.98674763
## PctPop18_34 -0.278527058 -0.03164843 0.07116151
```

```
cdi_nPhys <- lm(nActPhys ~ TotPop + LandArea + TotInc, data = cdi)
summary(cdi_nPhys)
```

```
##
## Call:
## lm(formula = nActPhys ~ TotPop + LandArea + TotInc, data = cdi)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1855.6  -215.2   -74.6    79.0  3689.0
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.332e+01  3.537e+01  -0.377 0.706719
## TotPop       8.366e-04  2.867e-04   2.918 0.003701 **
## LandArea    -6.552e-02  1.821e-02  -3.597 0.000358 ***
## TotInc       9.413e-02  1.330e-02   7.078 5.89e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 560.4 on 436 degrees of freedom
## Multiple R-squared:  0.9026, Adjusted R-squared:  0.902
## F-statistic:  1347 on 3 and 436 DF,  p-value: < 2.2e-16
```

The Equation: Y = -1.332e+01 + 8.366e-04 * X1 - 6.552e-02 * X2 + 9.413e-02 * X3

d) Obtain the predicted number of active physicians for a county with a total population of 400,000, a land area of 1,000 square miles, and total personal income of 8,000 million dollars in two ways:

1. By plugging 400,000, 1,000, and 8,000 into the equation for X1, X2, and X3.
2. By using predict().

```
-13.32 + 0.0008366 * 400000 - 0.06552 * 1000 + 0.09413 * 8000
```

```
## [1] 1008.84
```

```
newCdi <- data.frame(TotPop = 400000,
                     LandArea = 1000,
                     TotInc = 8000)
predict(cdi_nPhys, newdata = newCdi)
```

```
##        1
## 1008.864
```

e) How much does number of active physicians increase for each additional 1-person increase in total population (holding land area and total personal income constant)?
For each additional person, there is an increase of 0.0008366 active physicians.

f) How much does number of active physicians increase for each additional 1.0 million dollars in totalpersonal income (holding land area and total population constant)?
The increase in active physicians increases by 0.09413 for each additional 1.0 million dollars in total personal income.

Exercise 11: Do the following.

Data frame:

```
cdi <- mutate(cdi, PopDens = TotPop/LandArea)
```

a) Use lm() to fit a multiple regression model with number of active physicians as the response (Y) and population density (X1), percent of population 65 or older (X2), and per capita income (X3) as explanatory variables. Then use summary() to obtain the results. Report the equation of the fitted model.
Code:

```
cdi.reg <- lm(nActPhys ~ PopDens + PctPop65 + PerCapInc, data = cdi)
summary(cdi.reg)
```

```
##
## Call:
## lm(formula = nActPhys ~ PopDens + PctPop65 + PerCapInc, data = cdi)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4998.0  -522.8  -293.8    64.2 22067.6
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.088e+03  4.240e+02  -2.566   0.0106 *
## PopDens      2.873e-01  3.554e-02   8.083 6.27e-15 ***
## PctPop65    -7.964e+00  1.900e+01  -0.419   0.6753
## PerCapInc    1.033e-01  1.921e-02   5.377 1.24e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1589 on 436 degrees of freedom
## Multiple R-squared:  0.2173, Adjusted R-squared:  0.2119
## F-statistic: 40.35 on 3 and 436 DF,  p-value: < 2.2e-16
```

Equation: Y = -1.088e+03 + 2.873e-01 * x1 - 7.964e+00 * X2 + 1.033e-01 * X3

b) Obtain the predicted number of active physicians for a county with a population density of 900 per square mile, 15 percent of its population over 65, and per capita income of 20,000 dollars in two ways:

1. By plugging 900, 15, and 20,000 into the equation for X1, X2, and X3.
2. By using predict(). Report both sets of your R commands for obtaining the predicted number of active physicians.
   Code:

```
-1.088e+03 + 2.873e-01 * 900 -7.964e+00 * 15 + 1.033e-01 * 20000
```

```
## [1] 1117.11
```

```
newCdi_PopDens <- data.frame(PopDens = 900,
                             PctPop65 = 15,
                             PerCapInc = 20000)
predict(cdi.reg, newdata = newCdi_PopDens)
```

```
##        1
## 1117.385
```

Exercise 12: Do the following.

Data frame:

```
my.reg <- lm(nActPhys ~ TotPop + LandArea + TotInc, data = cdi)
summary(my.reg)
```

```
##
## Call:
## lm(formula = nActPhys ~ TotPop + LandArea + TotInc, data = cdi)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1855.6  -215.2   -74.6    79.0  3689.0
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.332e+01  3.537e+01  -0.377 0.706719
## TotPop       8.366e-04  2.867e-04   2.918 0.003701 **
## LandArea    -6.552e-02  1.821e-02  -3.597 0.000358 ***
## TotInc       9.413e-02  1.330e-02   7.078 5.89e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 560.4 on 436 degrees of freedom
## Multiple R-squared:  0.9026, Adjusted R-squared:  0.902
## F-statistic:  1347 on 3 and 436 DF,  p-value: < 2.2e-16
```

```
cdi <- mutate(cdi, PopDens = TotPop / LandArea)
my.reg <- lm(nActPhys ~ PopDens + PctPop65 + PerCapInc, data = cdi)
summary(my.reg)
```

```
##
## Call:
## lm(formula = nActPhys ~ PopDens + PctPop65 + PerCapInc, data = cdi)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4998.0  -522.8  -293.8    64.2 22067.6
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.088e+03  4.240e+02  -2.566   0.0106 *
## PopDens      2.873e-01  3.554e-02   8.083 6.27e-15 ***
## PctPop65    -7.964e+00  1.900e+01  -0.419   0.6753
## PerCapInc    1.033e-01  1.921e-02   5.377 1.24e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1589 on 436 degrees of freedom
## Multiple R-squared:  0.2173, Adjusted R-squared:  0.2119
## F-statistic: 40.35 on 3 and 436 DF,  p-value: < 2.2e-16
```

a) Using the residual standard error in the output from summary(), which model fits the data better? Hint: A smaller residual standard error indicates a better fitting model.
The better fitting model is the one containing the total population, land area, and total income based on the residual standard error.

b) Using the R2 (labeled Multiple R-squared) in the output from summary(), which model fits the data better? Hint: A larger R2 indicates a better fitting model.
The first still indicates a better fitting model based on R2.

c) Based on your answers to Parts a and b, which model would you expect to give better predictions of the number of active physicians in a county?
I would expect the first model to provide a better prediction based on the two errors seen in parts *a* and *b*.

```
dues.file <- file.choose()

dues <- read.csv(dues.file, header = TRUE, sep = "", stringsAsFactors = FALSE)

head(dues)
```

```
##   NotRenew DuesIncr
## 1        0       25
## 2        0       27
## 3        0       30
## 4        0       30
## 5        0       31
## 6        0       32
```

Section 9.7 Exercises

Exercise 13: Do the following.

```
my.logreg <-
  glm(NotRenew ~ DuesIncr, family = "binomial", data = dues)
summary(my.logreg)
```

```
##
## Call:
## glm(formula = NotRenew ~ DuesIncr, family = "binomial", data = dues)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.12290  -0.37290   0.08522   0.47113   1.78651
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -15.4157     5.6780  -2.715  0.00663 **
## DuesIncr      0.3902     0.1421   2.745  0.00605 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 41.455  on 29  degrees of freedom
## Residual deviance: 20.083  on 28  degrees of freedom
## AIC: 24.083
##
## Number of Fisher Scoring iterations: 6
```

a) Obtain the (estimated) probability that a person won't renew their membership if the dues increase is 45 dollars in two ways:

1. By plugging 45 into the equation for X.
2. By using predict().
   Code:

```
exp(-15.42 + 0.39 * 45)/(1 + exp(-15.42 + 0.39 * 45))
```

```
## [1] 0.893785
```

```
newDues <- data.frame(DuesIncr = 45)
predict(my.logreg, newDues, type = "response")
```

```
##        1
## 0.894954
```

b) Now obtain the (estimated) probability that a person won't renew their membership if the dues increase is only 35 dollars dollars in two ways:

1. By plugging 35 into the equation for X.
2. By using predict(). Report both sets of your R commands for obtaining the (estimated) probability of not renewing.
   Code:

```
exp(-15.42 + 0.39 * 35)/(1 + exp(-15.42 + 0.39 * 35))
```

```
## [1] 0.1455423
```

```
newDues <- data.frame(DuesIncr = 35)
predict(my.logreg, newDues, type = "response")
```

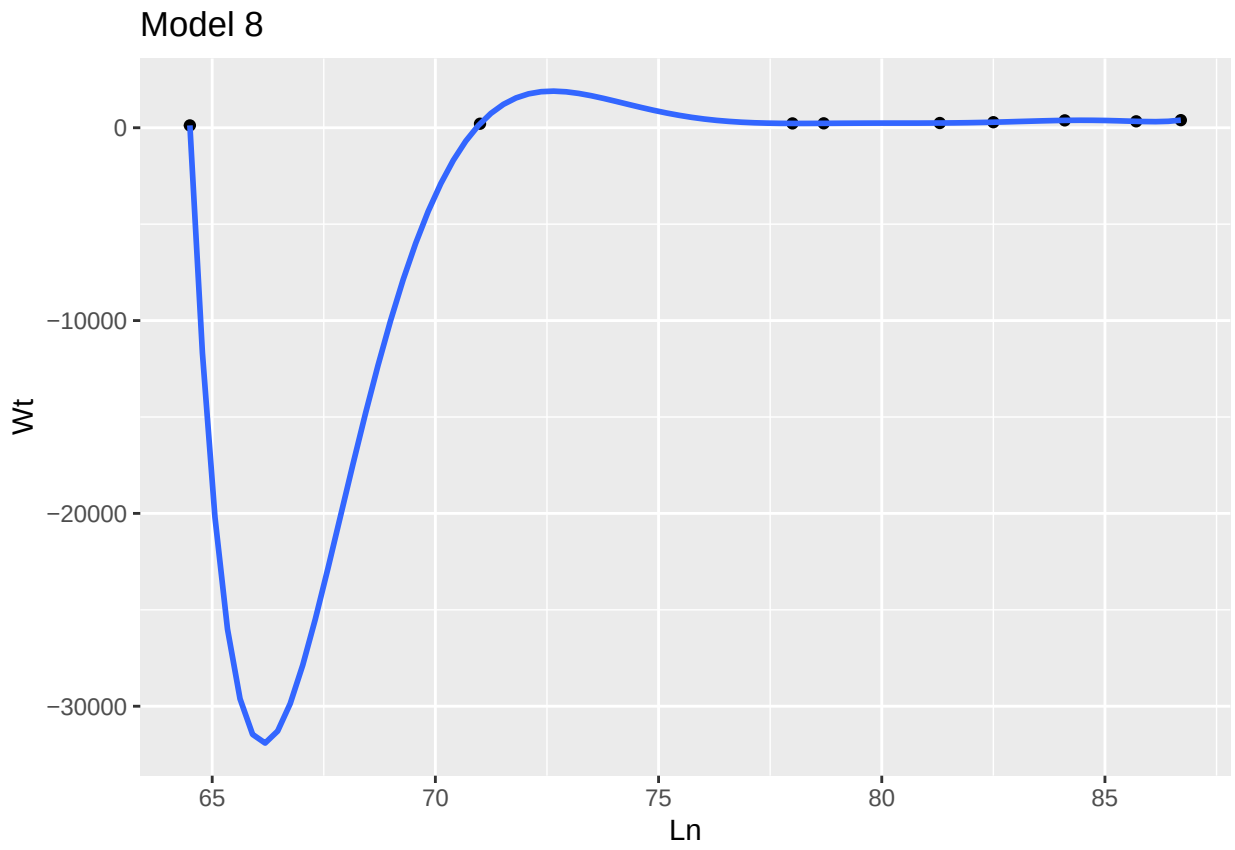
```
##         1
## 0.1468632
```

i ## Section 9.8 Exercises

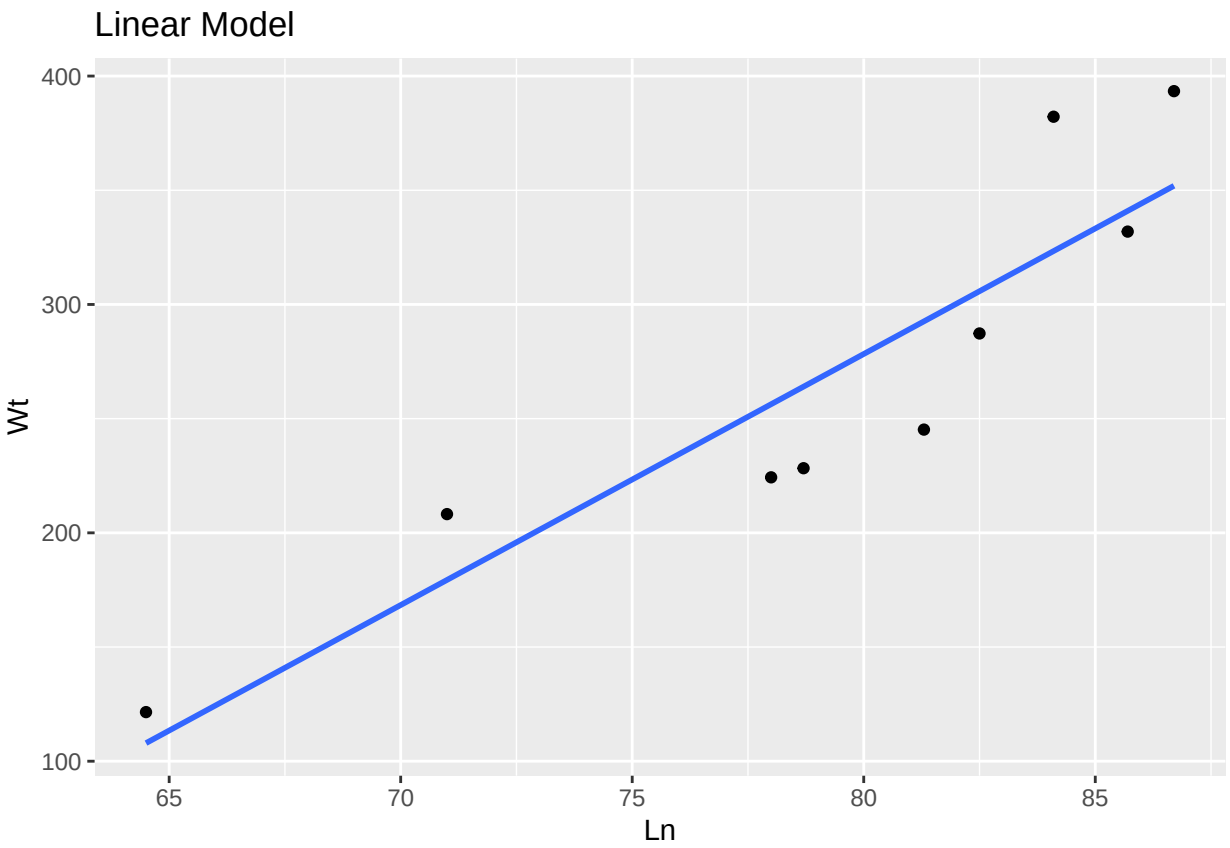
Exercise 14: Do the following.

Data:

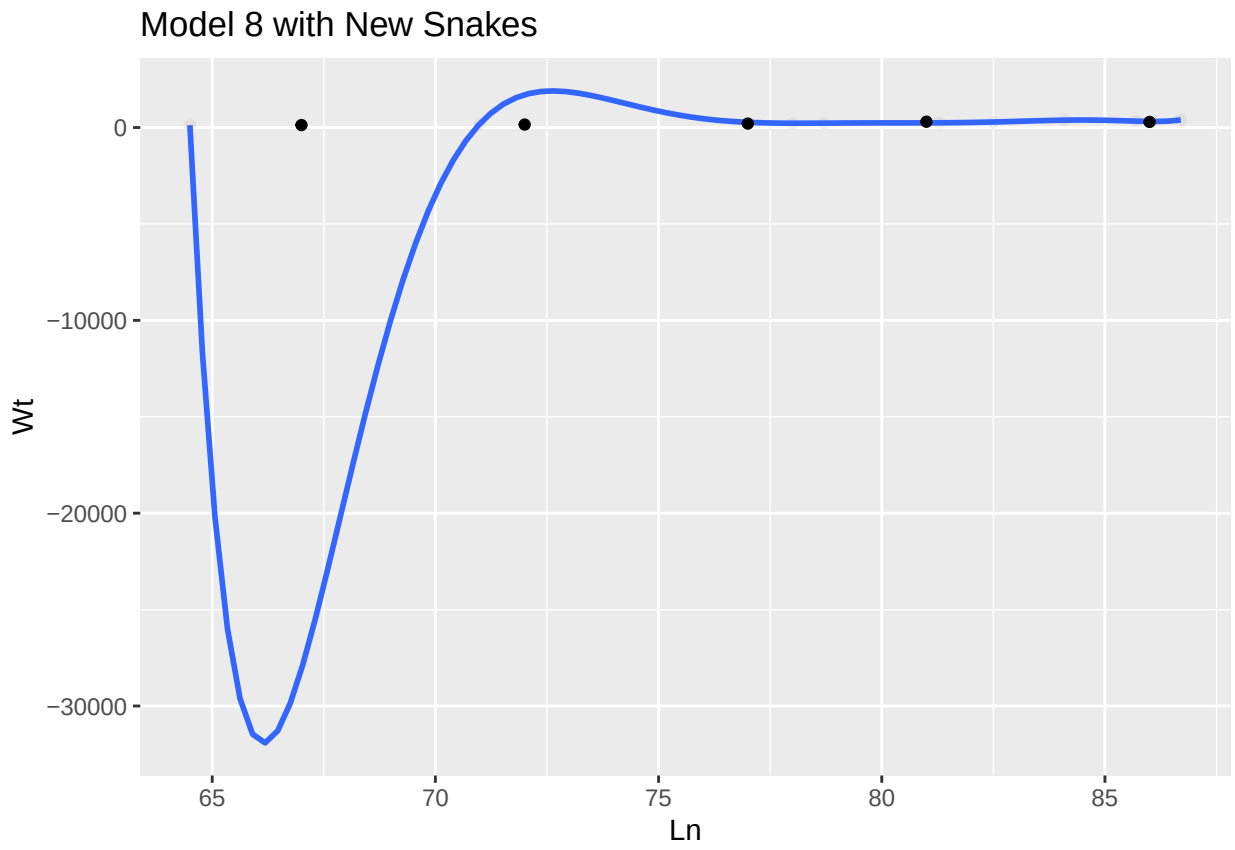
```
Ln <- c(85.7, 64.5, 84.1, 82.5, 78.0, 81.3, 71.0, 86.7, 78.7)
Wt <-
  c(331.9, 121.5, 382.2, 287.3, 224.3, 245.2, 208.2, 393.4, 228.3)
snakes <- data.frame(Length = Ln, Weight = Wt)

g <- ggplot(Snakes, aes(x = Ln, y = Wt)) + geom_point()
# Eighth degree polynomial model (Model 8):
g + stat_smooth(method = "lm",
                formula = y ~ poly(x, 8),
                se = F) +
  ggtitle(label = "Model 8")
```



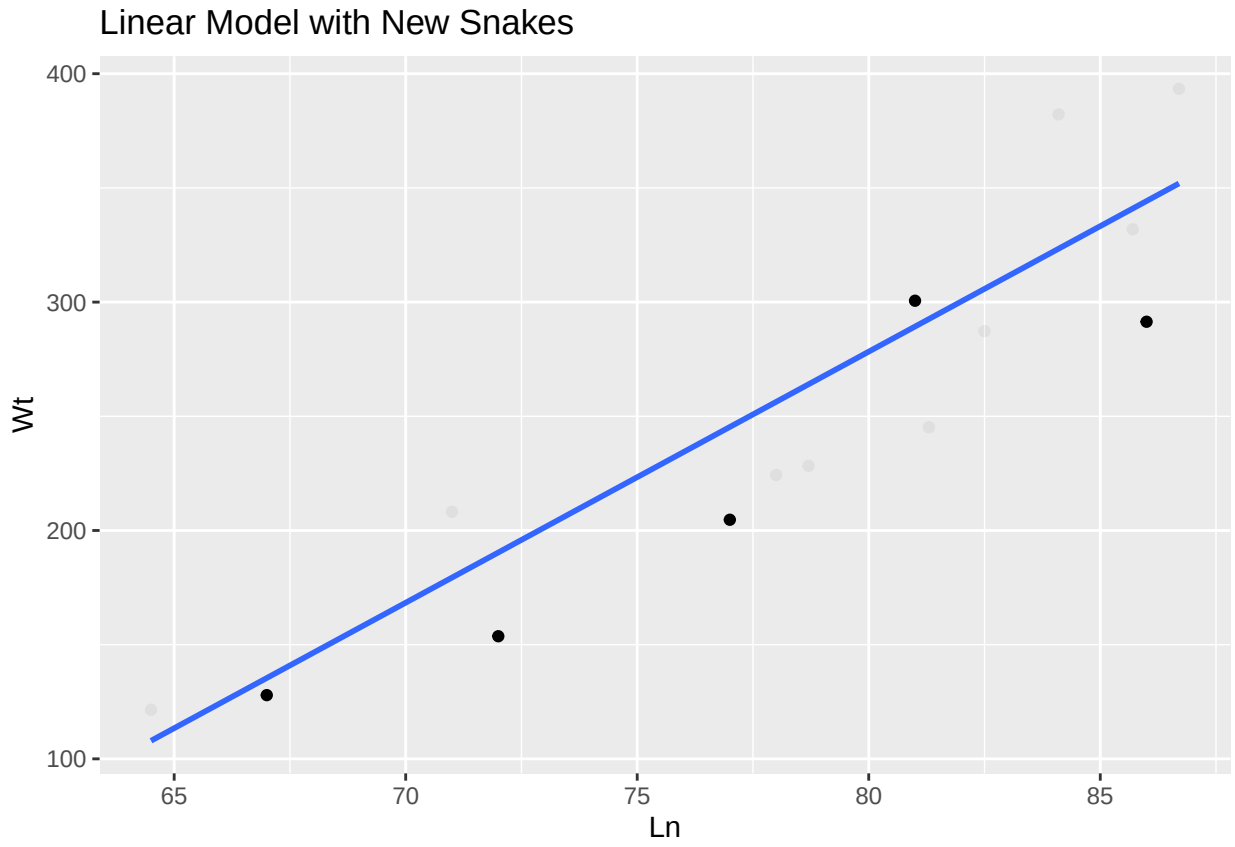
```
g + stat_smooth(method = "lm", formula = y ~ x, se = F) +
  ggtitle(label = "Linear Model")
```



```r
# Models with five new snakes
newSnakes <- data.frame(Ln = c(67, 72, 77, 81, 86),
                        Wt = c(127.9, 153.7, 204.7, 300.6,   291.4))

g_new <- ggplot(Snakes, aes(x = Ln, y = Wt)) +
  geom_point(alpha = 0.05)

g_new + stat_smooth(method = "lm",
                    formula = y ~ poly(x, 8),
                    se = F) +
  ggtitle(label = "Model 8 with New Snakes") +
  geom_point(data = newSnakes)
```



```
g_new + stat_smooth(method = "lm",
                    formula = y ~ x,
                    se = F) +
  ggtitle(label = "Linear Model with New Snakes") +
  geom_point(data = newSnakes)
```



a) How well does the fitted model predict the weights of the original nine snakes?
The model 8 graph seems to fit the points better than the linear model.

b) How well do you think the fitted model would predict the weights of five new snakes?
The model 8 had a huge amount of error in one of the points for a sample of new snakes. Overall, the new snakes fit better with the linear model even though the linear model has the appearance of more error for more of the points.