BEE552 Biometry Week 9

Maria Feiler

03/29/2022

My Learning Journey

Over the last week, I participated in Biometry in the following ways:

- I asked / answered **1** questions posed in class.
- I asked **6** questions in Slack.
- I answered **0** questions posed by other students on Slack.
- I came to Heather's office hours: **No**
- I came to Jose's office hours: **No**
- I met with Heather or Jose separately from office hours: **No**

Anything not falling into one of the above categories?

> **No**

On a scale of 1 (no knowledge) to 10 (complete expert), how would I rate my comfort with R programming after this week?

> **7**

Any topics from last week that you are still confused about?

> Honestly, this whole problem set. I was able to work through things with others, but my actual understanding of this is very poor.

Problem Set

Part I

For two different combinations of β_0 and β_1 (of your choosing), fill in the tables below (three columns need to be completed, plus the box for the sum of squared residuals). Try to tweak the parameter values to minimize the sum of squared residuals. This exercise is designed to just re-enforce the mechanics of fitting linear models, that each set of parameter values yields a set of predicted Y values, and that the difference between these predictions and the actual data form the basis of calculating the sum of squared errors.

$$\beta_0 = 2$$

$$\beta_1 = 3$$

```
# Store observed x and y variables
x <- c(1, 2, 3, 4, 5, 6)
y <- c(0, 1, 7, 2, 4, 10)

# Store beta variables
beta0 <- 2
beta1 <- 3

# Calculate yhat for each x
yhat <- c()
for (i in 1:length(x)){
        yhat[i] <- 1*(beta0+x[i])*beta1
}

# Calculate epsilon for each x
ep <- c()
for(i in 1:length(x)){
        ep[i] <- yhat[i]-y[i]
}

# Get epsilon squared
ep2 <- ep^2
```

Model matrix	x_i	$\hat{y}$	y_i	ϵ_i	ϵ_i^2
1	1	9	0	9	81
1	2	12	1	11	121
1	3	15	7	8	64
1	4	18	2	16	256
1	5	21	4	17	289
1	6	24	10	14	196

The sum of ϵ^2 is 1007.

$$\beta_0 = 1$$

$$\beta_1 = 1$$

```
# Store beta variables
beta0 <- 1
beta1 <- 1

# Calculate yhat for each x
yhat <- c()
for (i in 1:length(x)){
        yhat[i] <- 1*(beta0+x[i])*beta1
}

# Calculate epsilon for each x
ep <- c()
for(i in 1:length(x)){
        ep[i] <- yhat[i]-y[i]
}

# Get epsilon squared
ep2 <- ep^2
```

Model matrix	x_i	$\hat{y}$	y_i	ϵ_i	ϵ_i^2
1	1	2	0	2	4
1	2	3	1	2	4
1	3	4	7	-3	9
1	4	5	2	3	9
1	5	6	4	2	4
1	6	7	10	-3	9

The sum of ϵ^2 is 39.

Part II

For Part II and III of the problem set, we will analyze a dataset collected by Lucy Donahue and undergraduate Nancy Dong relating to the characteristics of Antarctic passenger vessels over time. This dataset includes several columns you will not need, but we will focus on the columns for vessel length, maximum speed, and engine power.

```
vessels <- read.csv("Vessel_data.csv", header = TRUE)
```

Step 1 *a) Use the 'lm' function to fit the following linear regression model, where MaxSpeed is the response and Length is the covariate: $MaxSpeed \sim Length$. What is the statistical distribution being modeled here?*

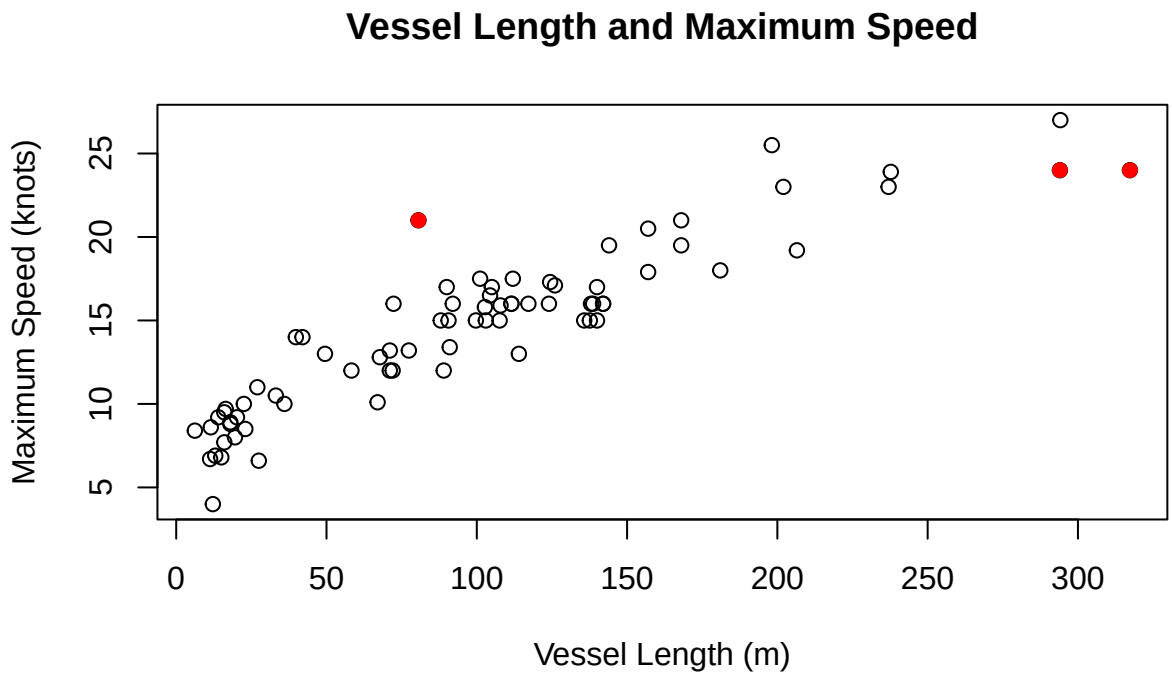
```
fit1 <- lm(vessels$max_speed_knots ~ vessels$length_meters)

fit1$coefficients
```

```
          (Intercept) vessels$length_meters
           8.32687808            0.06324271
```

$$MaxSpeed \sim N(8.33 + 0.063x_i + \epsilon_i, \sigma^2)$$

Are there any data points that may be considered outliers? If so, which one(s)? How does the regression slope change if it/they were to be removed from the dataset?



There may be three outliers((80.6, 21), (317.3, 24), (294, 24)), which are highlighted in red. If they are removed...

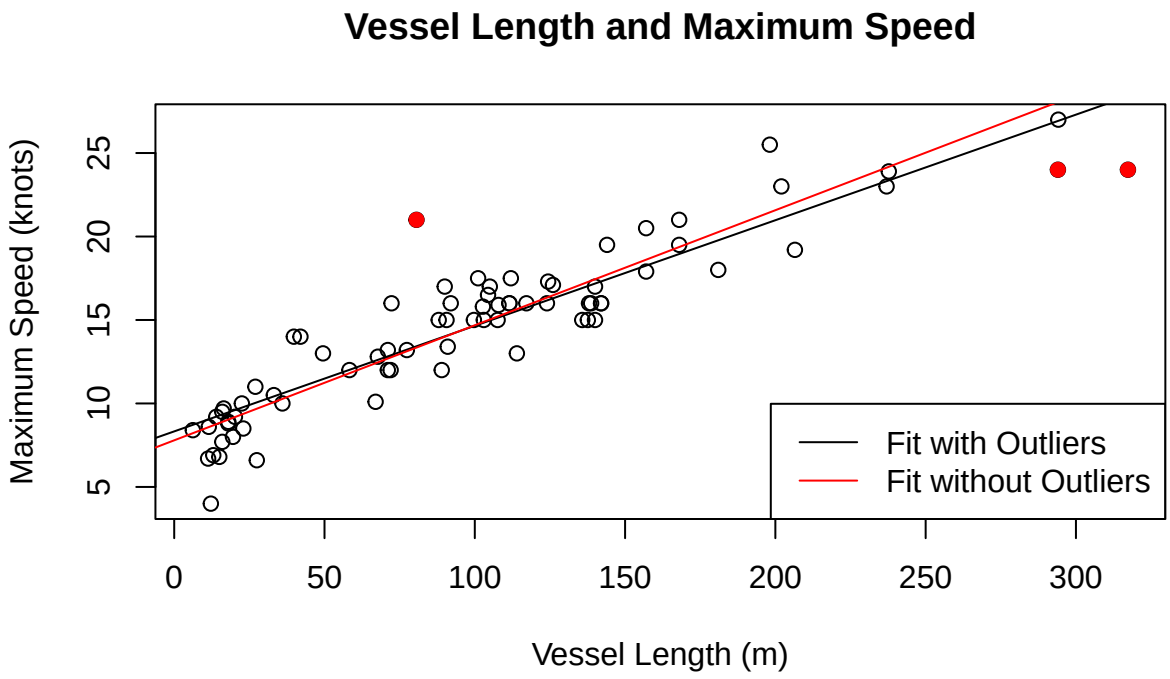
```r
# Remove outliers
vessels2 <- vessels[-c(out1, out2, out3),]

# Refit data
fit2 <- lm(vessels2$max_speed_knots ~ vessels2$length_meters)

fit2$coefficients
```

```
           (Intercept) vessels2$length_meters
            7.79213692             0.06890991
```

. . . the intercept reduces and the slope slightly increases.



b. Use this model (all the data included) to predict the maximum vessel speed for a vessel that is 100 m in length. What is the confidence interval? What is the prediction interval?

```r
est <- unname(fit1$coefficients[1]+100*fit1$coefficients[2])

# Get summary of confidence interval
CI <- summary(predict(fit1,
                      newdata = (data.frame(length_meters = 100)),
                      interval = "confidence"))

# Select the upper and lower mean (4th row, 2nd and 3rd columns of table), split
# it from its label, and assign to numeric to remove floating spaces
llCI <- as.numeric(strsplit(CI[4,2], split = ":")[[1]][2])
ulCI <- as.numeric(strsplit(CI[4,3], split = ":")[[1]][2])

# Get summary of prediction interval
```

```
PI <- summary(predict(fit1,
                      newdata = (data.frame(length_meters = 100)),
                      interval = "prediction"))

# Select the upper and lower mean (4th row, 2nd and 3rd columns of table), split
# it from its label, and assign to numeric to remove floating spaces
llPI <- as.numeric(strsplit(PI[4,2], split = ":")[[1]][2])
ulPI <- as.numeric(strsplit(PI[4,3], split = ":")[[1]][2])
```

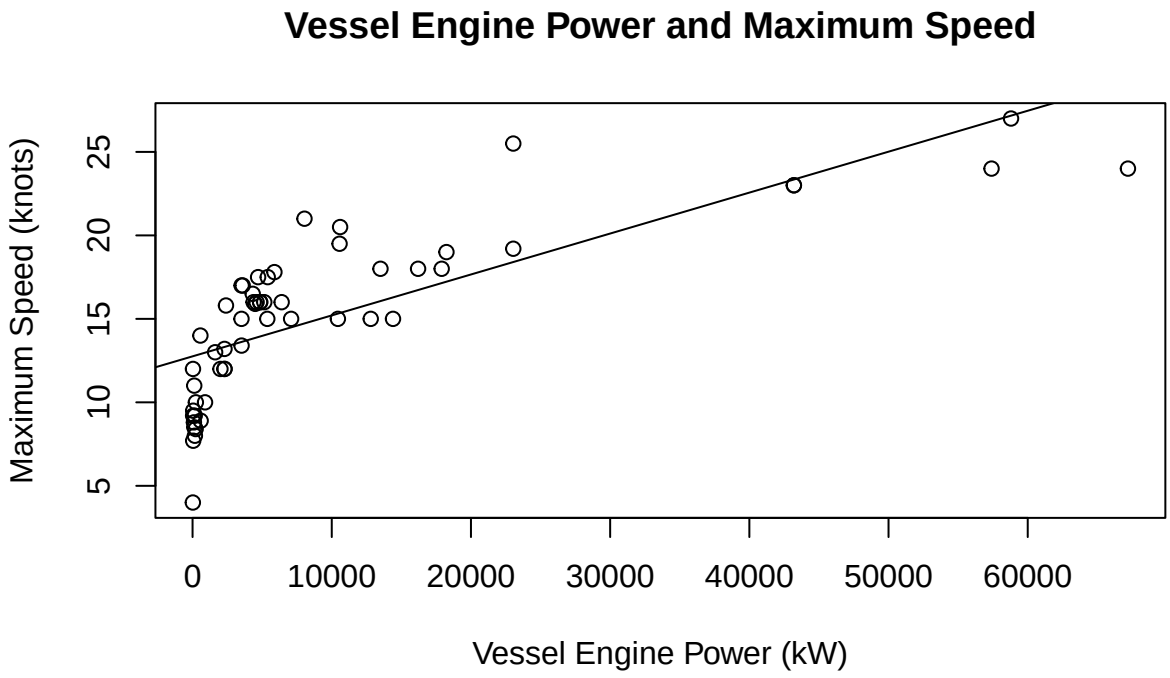
The predicted speed of a vessel 100 meters long is 14.65 knots with a confidence interval of (13.975, 15.22) and a prediction interval of (10.517, 18.67).

c. Use the 'lm' function to fit the following linear regression model: $MaxSpeed \sim EnginePower$. Report the results and plot the data and best fit lines.

```
fit3 <- lm(vessels$max_speed_knots ~ vessels$engine_power_kW)

fit3$coefficients
```

```
          (Intercept) vessels$engine_power_kW
         1.275504e+01            2.451669e-04
```



d. Which covariate "Length" or "EnginePower" explains more of the variation in maximum vessel speed?

Length explains more of of variation in maximum vessel speed because its linear model's mean squared error is smaller than the linear model for EnginePower.

```
# Sum of squares for the EnginePower model
mean(fit3$residuals^2)
```

```
9.906316
```

```
# Sum of squares for the Length model
mean(fit1$residuals^2)
```

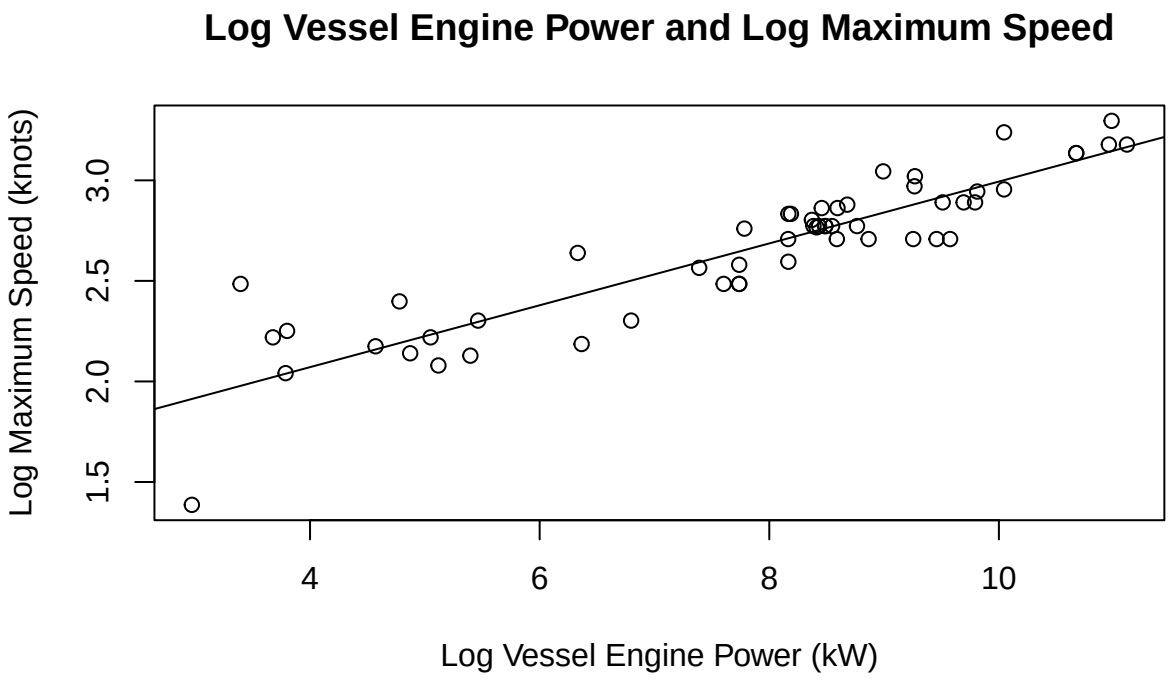
```
3.975558
```

e. Use the 'lm' function to fit the following linear regression model: $ln(MaxSpeed) \sim ln(Length)$. Report the results and plot the data and the best-fit line.

```
fit4 <- lm(log(vessels$max_speed_knots) ~ log(vessels$engine_power_kW))

fit4$coefficients
```

```
                (Intercept) log(vessels$engine_power_kW)
                  1.4558084                   0.1538147
```



Step 2 *Write a function to calculate the negative log likelihood associated with fitting the linear regression model MaxSpeed~Length. (Hint: Your function will need to take as inputs the intercept (β_0), the slope (β_1), and the variance (σ^2).) (For full credit, use the 'dnorm' function.)*

```
# Clean data of NAs for future work
vessels <- na.omit(data.frame("length_meters" = vessels$length_meters,
                              "max_speed_knots" = vessels$max_speed_knots))

# Based on what was said in Slack
# y is the response variable
# x is the covariate
# params is a vector of parameters, beta0, beta1, and sigma
neg.ll.lm <- function(y, x, params){
        # Assign parameters to their labels
        beta0 <- params[1]
        beta1 <- params[2]
        sigma <- params[3]

        # Calculate the predicted y value of an x using the parameters
        yhat <- beta0 + beta1*x

        # Produce the value with some level of randomness using dnorm()
        -sum(dnorm(y,
                   mean = yhat,
                   sd = sigma,
                   log = TRUE)
             )
}
```

Using the 'optim' function in R, minimize the negative log-likelihood to obtain the maximum likelihood regression parameter estimates for β_0, β_1, σ^2. (Hint: You will need starting values for optim; the results of the 'lm' calculation would be reasonable starting points.)

```
# Optimize over the data using the values from the original fit
MLE <- optim(par = c(fit1$coefficients[1],
                     fit1$coefficients[2],
                     sigma(fit1)),
             fn = neg.ll.lm,
             y = vessels$max_speed_knots,
             x = vessels$length_meters)

# Produced object with three values in the par, corresponding to the optimal
# beta0, beta1, and sigma
```

The optimized linear regression is $\hat{y} = 8.33 + 0.063\ x$.