STOR 455/001 - Homework II

September 14, 2024

This homework is due on 11:59 pm (Eastern Time) on Sep 29, 2024. The homework is to be submitted on Gradescope. For Question 2 and Question 3, please use `Rmarkdown` to include the `R`-codes along with the texts in your submission. For Question 1, you are allowed to use pen and paper. You can collaborate with your classmates, but make sure to write down the solutions by your own.

Question 1 (15 points)

Consider the simple linear regression model :

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \;\; i = 1, \ldots, n,$$

where

$$\varepsilon_1, \ldots, \varepsilon_n \overset{i.i.d.}{\sim} N(0, \sigma^2).$$

Let $\hat{\beta}_0$ be the least-square estimator of β_0. In this exercise, we shall derive the distribution of $\hat{\beta}_0$.

(a) (3 points)

Recall that $\hat{\beta}_0 = \bar{Y}_n - \hat{\beta}_1 \bar{x}_n$ and we have derived in the class that

$$\hat{\beta}_1 = \sum_{i=1}^{n} \frac{(x_i - \bar{x}_n)}{(n-1)s_x^2} Y_i.$$

Use the above expression to show that

$$\hat{\beta}_0 = \sum_{i=1}^{n} \left[\frac{1}{n} - \frac{(x_i - \bar{x}_n)\,\bar{x}_n}{(n-1)s_x^2} \right] Y_i. \tag{1}$$

Hint : Write $\bar{Y}_n$ as $\frac{1}{n} \sum_{i=1}^{n} Y_i$.

The definition of $\hat{\beta}_0$ is:

$$\hat{\beta}_0 = \bar{Y}_n - \hat{\beta}_1 \bar{x}_n$$

substitute:

$$\hat{\beta}_0 = \bar{Y}_n - \bar{x}_n \sum_{i=1}^{n} \frac{(x_i - \bar{x}_n)}{(n-1)s_x^2} Y_i$$

Now express $\bar{Y}_n$ as the average of the Y_i's:

$$\bar{Y}_n = \frac{1}{n} \sum_{i=1}^{n} Y_i$$

rewrite as:

$$\hat{\beta}_0 = \frac{1}{n}\sum_{i=1}^{n} Y_i - \bar{x}_n \sum_{i=1}^{n} \frac{(x_i - \bar{x}_n)}{(n-1)s_x^2} Y_i$$

combine into a single summation:

$$\hat{\beta}_0 = \sum_{i=1}^{n} \left[\frac{1}{n} - \frac{(x_i - \bar{x}_n)\bar{x}_n}{(n-1)s_x^2} \right] Y_i$$

this gives the equatoin for $\hat{\beta}_0$.

(b) (3 points)

Recall that $\mathbb{E}(Y_i) = \beta_0 + \beta_1 x_i$, for all $i = 1, \ldots, n$. Use linearity of expectation to show that

$$\mathbb{E}\left(\bar{Y}_n\right) = \beta_0 + \beta_1 \bar{x}_n.$$

Hint : By linearity of expectation we have

$$\mathbb{E}\left(\bar{Y}_n\right) = \mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n} Y_i\right) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\left(Y_i\right).$$

given that:

$$\bar{Y}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i$$

taking the expectation of both sides, apply the linearity of expectation:

$$\mathbb{E}(\bar{Y}_n) = \mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n} Y_i\right) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}(Y_i)$$

from the regression model $Y_i = \beta_0 + \beta_1 x_i + \epsilon_i$, and knowing that $\mathbb{E}(\epsilon_i) = 0$, we get:

$$\mathbb{E}(Y_i) = \beta_0 + \beta_1 x_i$$

Substituting this expression for $\mathbb{E}(Y_i)$ into the expectation of $\bar{Y}_n$:

$$\mathbb{E}(\bar{Y}_n) = \frac{1}{n}\sum_{i=1}^{n} (\beta_0 + \beta_1 x_i)$$

break this summation into two parts:

$$\mathbb{E}(\bar{Y}_n) = \frac{1}{n}\left(\sum_{i=1}^{n} \beta_0 + \sum_{i=1}^{n} \beta_1 x_i\right)$$

This simplifies to:

$$\mathbb{E}(\bar{Y}_n) = \frac{1}{n}\left(n\beta_0 + \beta_1 \sum_{i=1}^{n} x_i\right)$$

$\sum_{i=1}^{n} x_i = n\bar{x}_n$, so we substitute this into the equation:

$$\mathbb{E}(\bar{Y}_n) = \beta_0 + \beta_1 \bar{x}_n$$

thus we show:

$$\mathbb{E}(\bar{Y}_n) = \beta_0 + \beta_1 \bar{x}_n$$

## (c) (2 points)

Recall the formula for $\hat{\beta}_0 = \bar{Y}_n - \hat{\beta}_1 \bar{x}_n$. Use part (b), linearity of expectation and the fact that $\mathbb{E}\left(\hat{\beta}_1\right) = \beta_1$, to conclude that $\mathbb{E}\left(\hat{\beta}_0\right) = \beta_0$.

Hint : By linearity of expectation, we have $\mathbb{E}\left(\hat{\beta}_0\right) = \mathbb{E}\left(\bar{Y}_n\right) - \mathbb{E}\left(\hat{\beta}_1\right) \bar{x}_n$. By linearity of expectation:

$$\mathbb{E}(\hat{\beta}_0) = \mathbb{E}(\bar{Y}_n) - \mathbb{E}(\hat{\beta}_1)\bar{x}_n$$

from part (b):

$$\mathbb{E}(\bar{Y}_n) = \beta_0 + \beta_1 \bar{x}_n$$

Also we know that:

$$\mathbb{E}(\hat{\beta}_1) = \beta_1$$

Substituting these results into the equation:

$$\mathbb{E}(\hat{\beta}_0) = (\beta_0 + \beta_1 \bar{x}_n) - \beta_1 \bar{x}_n$$

Simplifying the right-hand side:

$$\mathbb{E}(\hat{\beta}_0) = \beta_0$$

Thus, we have shown that:

$$\mathbb{E}(\hat{\beta}_0) = \beta_0$$

## (d) (2 points)

Use Equation (1) and linearity of variance (under independence) to conclude that

$$\operatorname{Var}\left(\hat{\beta}_0\right) = \sigma^2 \sum_{i=1}^{n} \left[\frac{1}{n} + \frac{(x_i - \bar{x}_n)\,\bar{x}_n}{(n-1)s_x^2}\right]^2 .$$

Hint : Since Y_i's are independent, we have

$$\operatorname{Var}\left(\hat{\beta}_0\right) = \operatorname{Var}\left(\sum_{i=1}^{n} \left[\frac{1}{n} - \frac{(x_i - \bar{x}_n)\,\bar{x}_n}{(n-1)s_x^2}\right] Y_i\right) = \sum_{i=1}^{n} \operatorname{Var}\left(\left[\frac{1}{n} + \frac{(x_i - \bar{x}_n)\,\bar{x}_n}{(n-1)s_x^2}\right] Y_i\right).$$

Now use the fact that for any random variable Z and any non-random constant a, we have $\operatorname{Var}(aZ) = a^2 \operatorname{Var}(Z)$. Applying the variance operator:

$$\operatorname{Var}(\hat{\beta}_0) = \operatorname{Var}\left(\sum_{i=1}^{n} \left[\frac{1}{n} - \frac{(x_i - \bar{x}_n)\bar{x}_n}{(n-1)s_x^2}\right] Y_i\right)$$

Using the linearity of variance, since Y_i's are independent:

$$\text{Var}(\hat{\beta}_0) = \sum_{i=1}^{n} \text{Var}\left(\left[\frac{1}{n} - \frac{(x_i - \bar{x}_n)\bar{x}_n}{(n-1)s_x^2}\right] Y_i\right)$$

We use the property that for a constant a and a random variable Z, $\text{Var}(aZ) = a^2\text{Var}(Z)$:

$$\text{Var}(\hat{\beta}_0) = \sum_{i=1}^{n} \left[\frac{1}{n} - \frac{(x_i - \bar{x}_n)\bar{x}_n}{(n-1)s_x^2}\right]^2 \text{Var}(Y_i)$$

Since $\text{Var}(Y_i) = \sigma^2$, we get:

$$\text{Var}(\hat{\beta}_0) = \sigma^2 \sum_{i=1}^{n} \left[\frac{1}{n} + \frac{(x_i - \bar{x}_n)\bar{x}_n}{(n-1)s_x^2}\right]^2$$

(e) (5 points)

Simplify the right hand side of the variance expression in part (e) to conclude that

$$\text{Var}\left(\hat{\beta}_0\right) = \sigma^2 \left[\frac{1}{n} + \frac{(\bar{x}_n)^2}{(n-1)s_x^2}\right].$$

Hint : Use the quadratic formula $(a+b)^2 = a^2 + b^2 + 2ab$ to expand

$$\left[\frac{1}{n} + \frac{(x_i - \bar{x}_n)\,\bar{x}_n}{(n-1)s_x^2}\right]^2,$$

and use that

$$\sum_{i=1}^{n}(x_i - \bar{x}_n) = 0, \quad \sum_{i=1}^{n}(x_i - \bar{x}_n)^2 = (n-1)s_x^2.$$

$$\text{Var}(\hat{\beta}_0) = \sigma^2 \sum_{i=1}^{n} \left[\frac{1}{n} + \frac{(x_i - \bar{x}_n)\bar{x}_n}{(n-1)s_x^2}\right]^2$$

use the quadratic expression:

$$\left[\frac{1}{n} + \frac{(x_i - \bar{x}_n)\bar{x}_n}{(n-1)s_x^2}\right]^2 = \frac{1}{n^2} + \left(\frac{(x_i - \bar{x}_n)\bar{x}_n}{(n-1)s_x^2}\right)^2 + 2 \cdot \frac{1}{n} \cdot \frac{(x_i - \bar{x}_n)\bar{x}_n}{(n-1)s_x^2}$$

apply the summation to each term:

- First term:

$$\sum_{i=1}^{n} \frac{1}{n^2} = \frac{1}{n}$$

- Second term:

$$\sum_{i=1}^{n} \left(\frac{(x_i - \bar{x}_n)\bar{x}_n}{(n-1)s_x^2}\right)^2 = \frac{\bar{x}_n^2}{(n-1)^2 s_x^4} \sum_{i=1}^{n}(x_i - \bar{x}_n)^2 = \frac{\bar{x}_n^2}{(n-1)s_x^2}$$

- Third term:

$$2 \cdot \frac{1}{n} \cdot \frac{\bar{x}_n}{(n-1)s_x^2} \sum_{i=1}^{n} (x_i - \bar{x}_n) = 0$$

Summing these, we get:

$$\text{Var}(\hat{\beta}_0) = \sigma^2 \left(\frac{1}{n} + \frac{\bar{x}_n^2}{(n-1)s_x^2} \right)$$

(f) (0 points)

Equation 1 shows that $\hat{\beta}_0$ is a linear combination of $Y_1, \ldots, Y_n$, which are distributed as Normal random variables. Therefore,

$$\hat{\beta}_0 \sim N\left(\beta_0, \text{Var}\left(\hat{\beta}_0\right)\right).$$

Question 2 (70 points)

The Box Turtle Connection is a long-term study anticipating at least 100 years of data collection on box turtles. Their purpose is to learn more about the status and trends in box turtle populations, identify threats, and develop strategies for long-term conservation of the species. Eastern Box Turtle populations are in decline in North Carolina and while they are recognized as a threatened species by the International Union for Conservation of Nature, the turtles have no protection in North Carolina. There are currently more than 30 active research study sites across the state of North Carolina. Turtles are weighed, measured, photographed, and permanently marked. These data, along with voucher photos (photos that document sightings), are then entered into centralized database managed by the NC Wildlife Resources Commission. The `Turtles.csv` dataset contains data collected at The Piedmont Wildlife Center in Durham.

(a) (4 points)

For this assignment let's look at only those turtles in the dataset that have a mass less than 400 grams. Construct a new data-frame named `Turtles_under_400` using the `filter` function that contains only those turtles with a `Mass` below 400 grams. There is also one adult male turtle in the data set that may have been mistakenly entered as having a mass of 6 grams. Remove this turtle from the dataset as well (using `filter` function). You should use this `Turtles_under_400g` data-frame for the remainder of the assignment.

```r
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
```

```
turtles_data <- read.csv("Turtles.csv")
Turtles_under_400 <- turtles_data |>
  filter(Mass < 400)
# remove mistake turtle
Turtles_under_400 <- Turtles_under_400 |>
  filter(Mass != 6)
head(Turtles_under_400)
```

```
##   LifeStage    Sex Annuli Mass StraightlineCL  MaxCW PL_AnteriortoHinge
## 1     Adult   Male     19  340         113.62  93.96              44.87
## 2  Juvenile Female      7  160          89.49  73.51              39.60
## 3     Adult   Male     16  175         127.70 101.16              54.76
## 4  Juvenile Female      7  100          81.00  69.00              35.00
## 5     Adult Female     20  155         122.85  99.38              51.68
## 6     Adult   Male     18  325         115.00  94.00              45.00
##   PL_HingetoPosterior ShellHeightatHinge
## 1               67.61              55.88
## 2               53.65              43.48
## 3               84.72              61.97
## 4               44.00              39.00
## 5               74.73              64.60
## 6               68.00              55.00
```

(b) (4 points)

The annuli rings on a turtle represent growth on the scutes of the carapace and plastron. In the past, it was thought that annuli corresponded to age, but findings suggest that this is not the case. However, the annuli are still counted since it may yield important life history information. Construct a least squares regression line that predicts turtles' `Annuli` by their `Mass`. Include a summary of this model.

```
model <- lm(Annuli ~ Mass, data = Turtles_under_400)
summary(model)
```

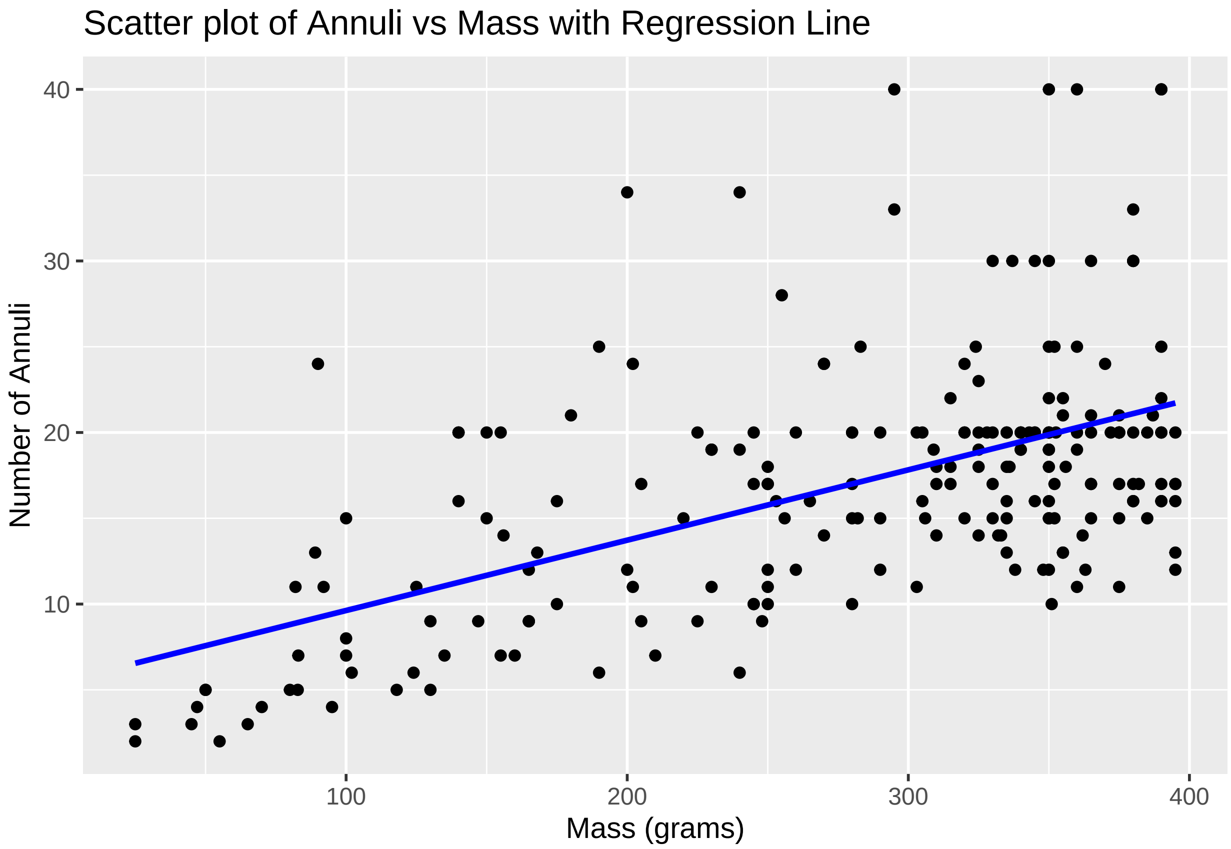
```
##
## Call:
## lm(formula = Annuli ~ Mass, data = Turtles_under_400)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.9146 -4.2587 -0.8985  2.1264 22.3811
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.525504   1.249598   4.422 1.55e-05 ***
## Mass        0.040995   0.004206   9.746  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.147 on 215 degrees of freedom
## Multiple R-squared:  0.3064, Adjusted R-squared:  0.3032
## F-statistic: 94.98 on 1 and 215 DF,  p-value: < 2.2e-16
```

(c) (8 points)

Produce a scatter-plot of this relationship and include the least squares regression line on the plot. State the interpretation of the estimate of the slope parameter of the regression line. Compute the 95% confidence interval for the slope parameter.

```r
library(ggplot2)
ggplot(Turtles_under_400, aes(x = Mass, y = Annuli)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(title = "Scatter plot of Annuli vs Mass with Regression Line",
       x = "Mass (grams)",
       y = "Number of Annuli")
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



```r
confint(model, level = 0.95)
```

```
##                  2.5 %     97.5 %
## (Intercept) 3.06247307 7.98853454
## Mass        0.03270346 0.04928569
```

(d) (6 points)

For a turtle in the Turtles_under_400g dataset with a mass of 200 grams, what does your model predict for this turtle's number of Annuli? Also report the 90% prediction interval. For turtles in the data with a mass of 200 grams (if any), what are the residuals for these cases?

```
mass_200 <- data.frame(Mass = 200)
predicted_annuli <- predict(model, newdata = mass_200)
predicted_annuli
```

```
##        1
## 13.72442
```

```
prediction_interval <- predict(model, newdata = mass_200, interval = "prediction", level = 0.90)
prediction_interval
```

```
##        fit      lwr      upr
## 1 13.72442 3.531087 23.91775
```

```
turtles_mass_200 <- Turtles_under_400 |>
  filter(Mass == 200)
turtles_mass_200 <- turtles_mass_200 |>
  mutate(predicted = predict(model, newdata = turtles_mass_200),
         residual = Annuli - predicted)
turtles_mass_200$residual
```

```
##         1         2
## 20.275581 -1.724419
```

(e) (4 points)

Which turtle (by row number in the Turtles_under_400g dataset) has the largest positive residual? What is the value of that residual?

Hint : Use the function which.max in R.

```
Turtles_under_400 <- Turtles_under_400 |>
  mutate(predicted = predict(model),
         residual = Annuli - predicted)
row_with_max_residual <- which.max(Turtles_under_400$residual)
max_residual_value <- max(Turtles_under_400$residual)
row_with_max_residual
```

```
## 131
## 131
```

```
max_residual_value
```

```
## [1] 22.3811
```

(f) (4 points)

Which turtle (by row number in the `Turtles_under_400g` dataset) has the most negative residual? What is the value of that residual?

Hint : Use the function `which.min` in `R`.

```
row_with_min_residual <- which.min(Turtles_under_400$residual)
min_residual_value <- min(Turtles_under_400$residual)
row_with_min_residual
```

```
## 64
## 64
```

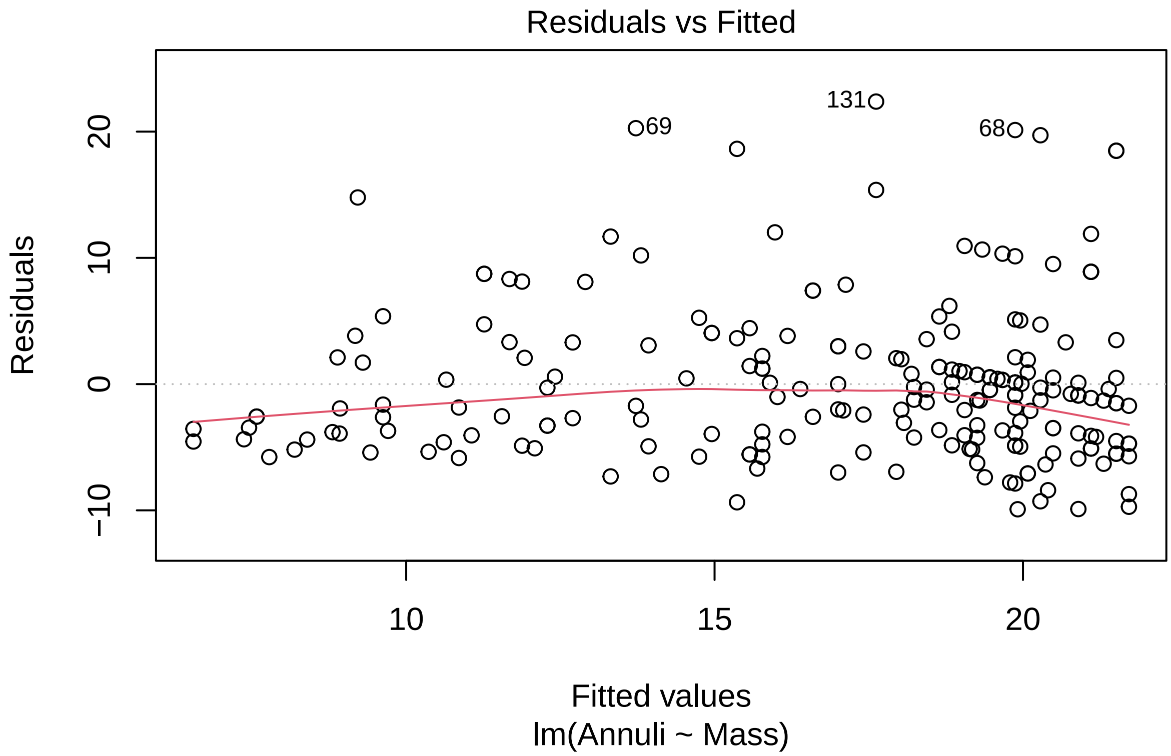
```
min_residual_value
```

```
## [1] -9.9146
```

(g) (8 points)

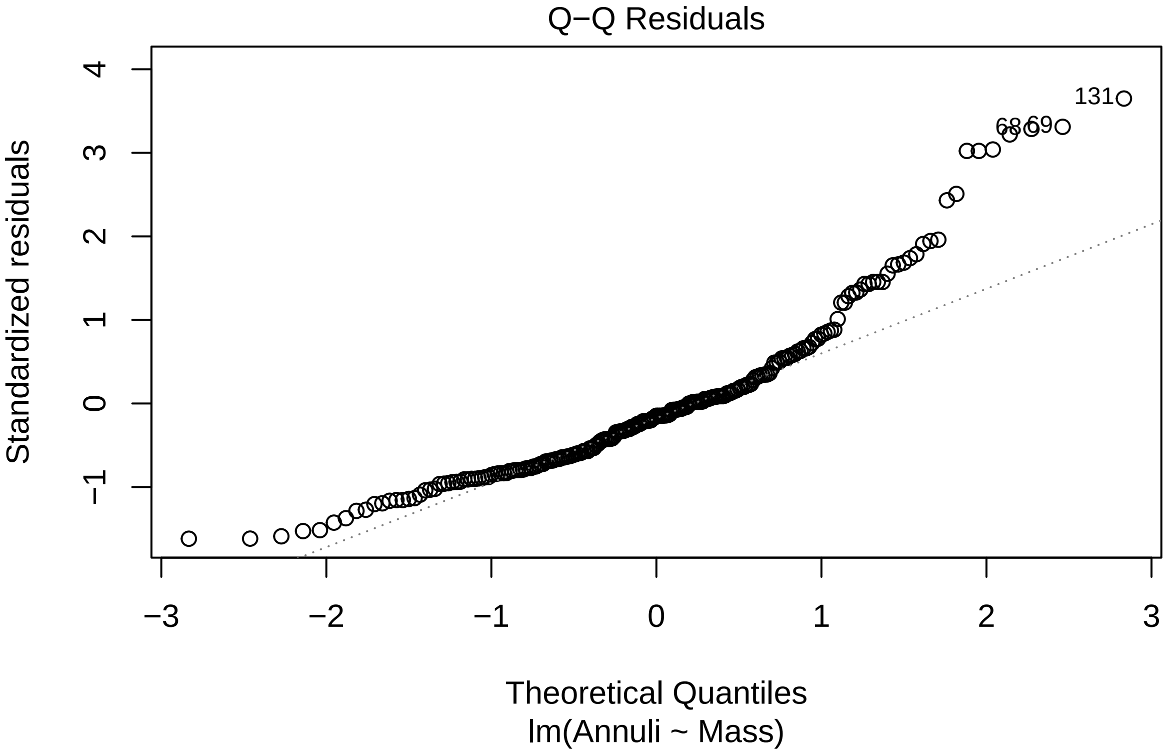
Comment on how each of the conditions for a simple linear model are (or are not) met in this model. Include at least two plots (in addition to the plot in part (c)) - with commentary on what each plot tells you specifically about the appropriateness of the model.

```
plot(model, which = 1)
```



Discussion: there is not a strong pattern which means that the linearity assumption holds. There's some spread in the residuals as the fitted values increase. The spread should ideally be the same for all fitted values. There are some points that are unusual and potential outliers.

```
plot(model, which = 2)
```



Discussion: The Q-Q plot shows some deviation from the straight line, in the high and low ends of distribution. Thus, the residuals are not perfectly normally distributed. The labeled points are significantly deviated. While the central portion of the data aligns wlel, the deviations in the extreme ends could affect the model fit. ## (h) (10 points)

Experiment with two transformations to determine if the models constructed with these transformations appear to do a better job of satisfying each of the simple linear model conditions. For the first transformation, use the natural logarithm of each of the variables to construct the model. For the second transformation, choose another transformation in an attempt to construct a model that better meets the simple linear model conditions. Discuss how these transformations do or do not improve each of the conditions for a simple linear model. Include the summary outputs for fitting these models and scatter-plots of the transformed variable(s) with the least squares lines. Note that there may not be a good transformation for this data based on what we have so far done in class.

```
#first transformation
Turtles_under_400 <- Turtles_under_400 |>
  mutate(log_Mass = log(Mass), log_Annuli = log(Annuli))
model_log <- lm(log_Annuli ~ log_Mass, data = Turtles_under_400)
summary(model_log)
```

```
##
```

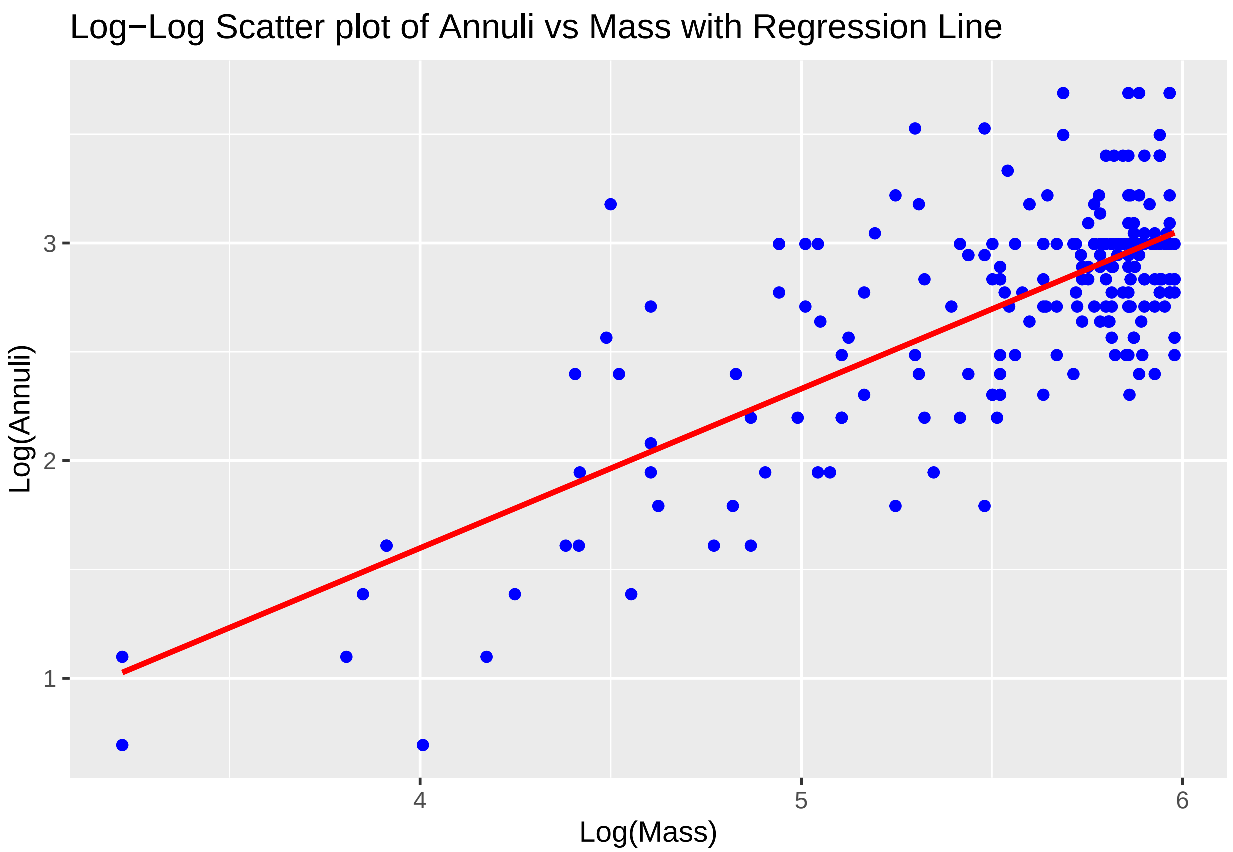
```
## Call:
## lm(formula = log_Annuli ~ log_Mass, data = Turtles_under_400)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.91052 -0.25044 -0.01327  0.18116  1.21384
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.33008    0.25233  -5.271 3.29e-07 ***
## log_Mass     0.73210    0.04542  16.119  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3572 on 215 degrees of freedom
## Multiple R-squared:  0.5472, Adjusted R-squared:  0.5451
## F-statistic: 259.8 on 1 and 215 DF,  p-value: < 2.2e-16
```

```r
#2nd transformation
Turtles_under_400 <- Turtles_under_400 %>%
  mutate(sqrt_Mass = sqrt(Mass), sqrt_Annuli = sqrt(Annuli))
model_sqrt <- lm(sqrt_Annuli ~ sqrt_Mass, data = Turtles_under_400)
summary(model_sqrt)
```

```
##
## Call:
## lm(formula = sqrt_Annuli ~ sqrt_Mass, data = Turtles_under_400)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.41614 -0.50245 -0.06504  0.38426  2.19937
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.17940    0.23196   5.085 8.01e-07 ***
## sqrt_Mass    0.17339    0.01386  12.509  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7105 on 215 degrees of freedom
## Multiple R-squared:  0.4212, Adjusted R-squared:  0.4185
## F-statistic: 156.5 on 1 and 215 DF,  p-value: < 2.2e-16
```

```r
ggplot(Turtles_under_400, aes(x = log_Mass, y = log_Annuli)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "Log-Log Scatter plot of Annuli vs Mass with Regression Line",
       x = "Log(Mass)",
       y = "Log(Annuli)")
```

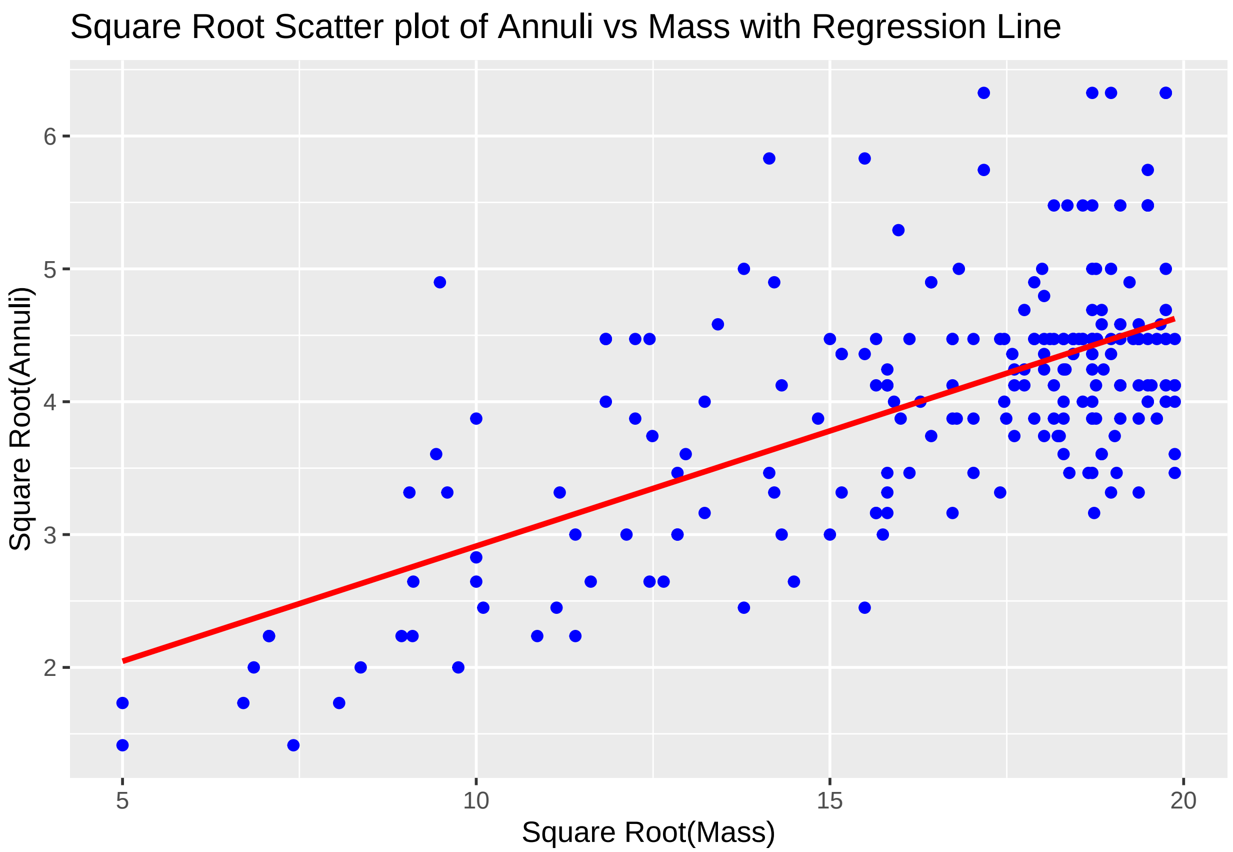
```
## `geom_smooth()` using formula = 'y ~ x'
```



Discussion: produced a pretty linear relationship between the transformed variables, as shown in the plot. The spread of data points is more uniform along the line compared to the original untransformed data.

```
ggplot(Turtles_under_400, aes(x = sqrt_Mass, y = sqrt_Annuli)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "Square Root Scatter plot of Annuli vs Mass with Regression Line",
       x = "Square Root(Mass)",
       y = "Square Root(Annuli)")
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



discussion: The square root transformation also shows a reasonably linear relationship, but the scatter of points is less uniform compared to the log-log transformation. Some clusters of data points appear around certain regions of the plot, which might indicate issues with the fit.

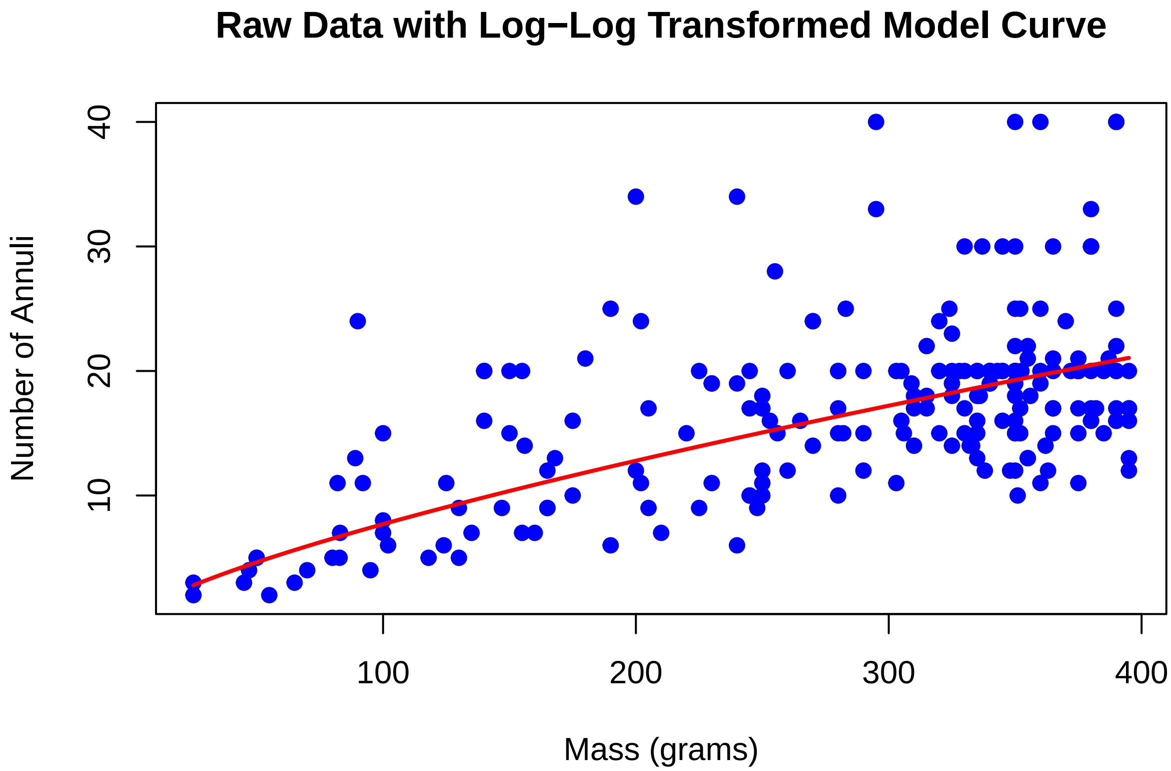
Conclusion: Linearity is affected because the log-log transformation helps in linearizing multiplicative relationships. by applying natural log to predictor and response vars, the data will show a more linear pattern. For the square root of both variables, the relationship is again linearized. Residuals are more normally distributed for the log-log transformation but for the square root transformation, the normality of residuals is dependent on the data distribution. Both transofrmations assume that the observations are independent.

(i) (4 points)

For your model with the first transformation from part (h) (it still may not be an ideal model), plot the raw data (not the transformed data) with the model (now a curve) on the same axes.

Hint : Use the `curve` function in `R` to plot a non-linear function.

```
beta_0 <- coef(model_log)[1]
beta_1 <- coef(model_log)[2]
plot(Turtles_under_400$Mass, Turtles_under_400$Annuli,
     main = "Raw Data with Log-Log Transformed Model Curve",
     xlab = "Mass (grams)", ylab = "Number of Annuli",
     pch = 19, col = "blue")
curve(exp(beta_0) * x^beta_1, add = TRUE, col = "red", lwd = 2)
```



## (j) (6 points)

For a turtle in the `Turtles_under_400g` dataset with a mass of 200 grams, what does the model with the first transformation from part (h) predict for this turtle's number of `Annuli`? In terms of `Annuli`, how different is this prediction from the observed values in the data for turtles with a mass of 200 grams (if any)? Explain how this is different than the value of the residual.

```
log_mass_200 <- log(200)
predicted_log_annuli <- beta_0 + beta_1 * log_mass_200
predicted_annuli <- exp(predicted_log_annuli)
predicted_annuli
```

```
## (Intercept)
##    12.79171
```

```
turtles_mass_200 <- Turtles_under_400 |>
  filter(Mass == 200)
observed_annuli <- turtles_mass_200$Annuli
observed_annuli
```

```
## [1] 34 12
```

```
difference <- predicted_annuli - observed_annuli
difference
```

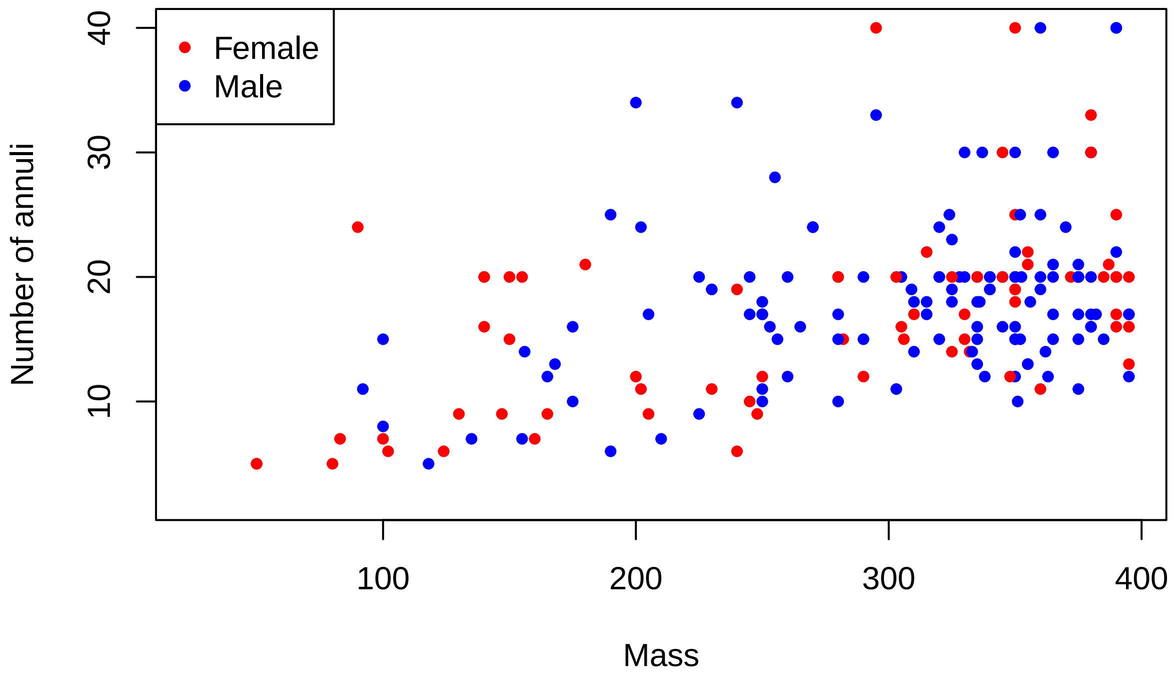
```
## [1] -21.2082931   0.7917069
```

The predicted value for a turtle with a mass of 200 grams is based on the log-log transformation model and represents what the model expects for such a turtle. The difference between this predicted value and the observed values for turtles with a mass of 200 grams tells us how well the model performs for these specific cases. ## (k) (12 points)

For your model with the first transformation from part (h), could the relationship between `Mass` and `Annuli` be different depending on the `Sex` of the turtle? There are some turtles in the dataset whose `Sex` recorded as `Unknown`. Remove these observations from the dataset using `filter` function. Construct two new data-frames (beginning with the `Turtles_under_400g` data-frame), one with only male turtles, and one with only female turtles. Using your model with the first transformation from part (h), construct two new models to predict `Annuli` with `Mass` for male and female turtles separately. Plot the raw data for `Anulli` and `Mass` for all turtles as well as each of these new models on the same plot. You should use different colors for each model (which are now curves). What does this plot tell you about the relationship between `Mass` and `Annuli` depending on the `Sex` of the turtles? Compare the goodness-of-fit of the model on these two subsets of the data-set.

Hint : To create the scatter-plot of the raw data with different colors for each sexes, use the following `R`-code.

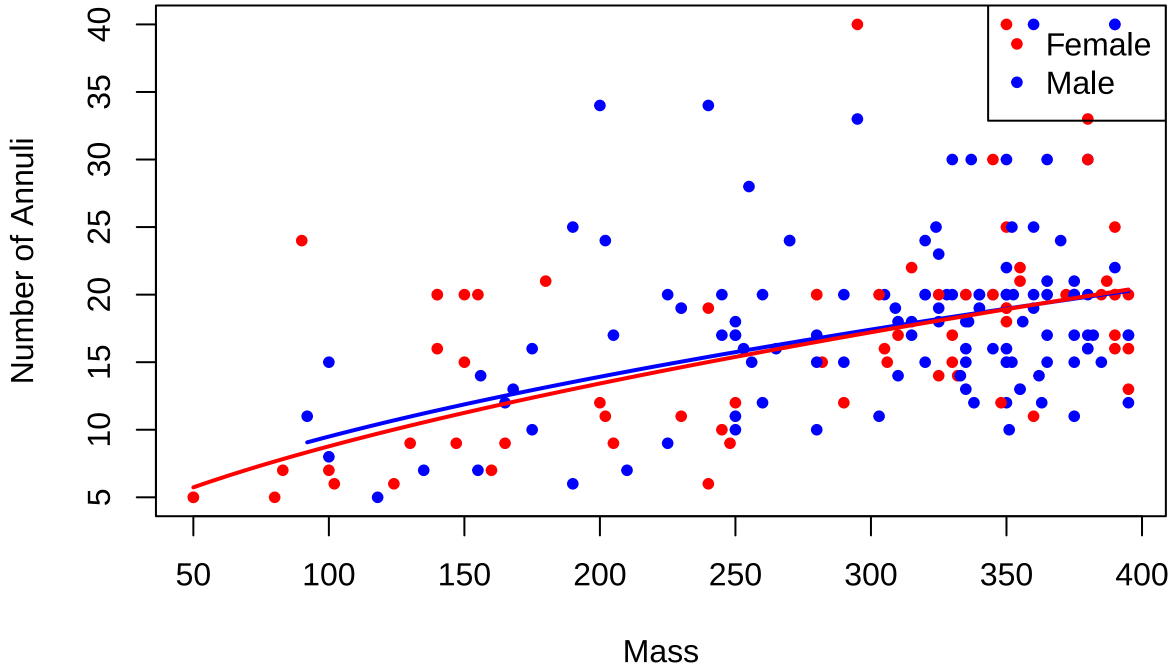
```
plot(Turtles_under_400$Mass,Turtles_under_400$Annuli,
     xlab="Mass",ylab="Number of annuli",
     col=c("red","blue")[as.factor(Turtles_under_400$Sex)],pch=20)

legend("topleft",legend=c("Female","Male"),col=c("red","blue"),pch=c(20,20))
```



Make sure you understand the role of each input / argument in the above code chunk. The `legend` function in `R` creates the labeling that appears on the topleft corner of the plot.

```
Turtles_under_400 <- Turtles_under_400 |>
  filter(Sex != "Unknown")
male_turtles <- Turtles_under_400 |>
  filter(Sex == "Male")
female_turtles <- Turtles_under_400 |>
  filter(Sex == "Female")
model_log_male <- lm(log(Annuli) ~ log(Mass), data = male_turtles)
model_log_female <- lm(log(Annuli) ~ log(Mass), data = female_turtles)
```

```
plot(Turtles_under_400$Mass, Turtles_under_400$Annuli,
     xlab = "Mass", ylab = "Number of Annuli",
     col = c("red", "blue")[as.factor(Turtles_under_400$Sex)], pch = 20)
legend("topright", legend = c("Female", "Male"), col = c("red", "blue"), pch = 20)
curve(exp(coef(model_log_male)[1]) * x^coef(model_log_male)[2], from = min(male_turtles$Mass), to = max
curve(exp(coef(model_log_female)[1]) * x^coef(model_log_female)[2], from = min(female_turtles$Mass), to
```



```
summary(model_log_male)
```

```
##
## Call:
## lm(formula = log(Annuli) ~ log(Mass), data = male_turtles)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.81331 -0.21519  0.01527  0.15719  0.89297
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.29202    0.55889  -0.522    0.602
## log(Mass)    0.55214    0.09812   5.627 1.17e-07 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.3332 on 123 degrees of freedom
## Multiple R-squared:  0.2047, Adjusted R-squared:  0.1983
## F-statistic: 31.66 on 1 and 123 DF,  p-value: 1.175e-07
```

```r
summary(model_log_female)
```

```
##
## Call:
## lm(formula = log(Annuli) ~ log(Mass), data = female_turtles)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.91706 -0.23989 -0.02565  0.20728  1.07085
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.65286    0.47535  -1.373    0.174
## log(Mass)    0.61337    0.08624   7.113 8.88e-10 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.3718 on 68 degrees of freedom
## Multiple R-squared:  0.4266, Adjusted R-squared:  0.4182
## F-statistic: 50.59 on 1 and 68 DF,  p-value: 8.883e-10
```

Question 3 (40 points)

This exercise will introduce you to the technique of using simulated datasets to evaluate and compare the performances of different statistical inference procedures. Towards that goal, we start by recalling the frequentist interpretation of $(100-\alpha)\%$ confidence intervals.

“If we perform the same random sampling procedure from the same population, independently from each other, N times (where N is very large, generally in the thousands) and compute (using the algebraic formula) the $(100-\alpha)\%$ confidence interval for each of those N samples, then approximately $(100-\alpha)\%$ of those N intervals will contain the true value of the parameter.”

For a real-life dataset, this interpretation remains a thought experiment since we do not know the true value of the parameter and we also don’t have the capacity to perform the sampling procedure a large number of times. Nevertheless, having access to large computing power enables us to simulate large number of datasets from some assumed model and evaluate the veracity of the above interpretation.

(a) (20 points)

We shall simulate our dataset from the following model :

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \ \ i = 1, \ldots, n,$$

where

$$x_1, \ldots, x_n \overset{i.i.d.}{\sim} N(0,1), \ \ \varepsilon_1, \ldots, \varepsilon_n \overset{i.i.d.}{\sim} N(0, \sigma^2),$$

with $\beta_0 = 2, \beta_1 = 5, \sigma = 5$. x_i's and ε_i's are independent of each other.

In R, we use the command `rnorm(n,mu,sigma)` to generate an i.i.d. (independent and identically distributed) sample of size `n` from Normal distribution with mean `mu` and standard deviation `sigma` (Warning : `rnorm` takes standard deviation of the Normal distribution as input, not the variance). We can use this command to generate $(x_1, \ldots, x_n)$ and $(\varepsilon_1, \ldots, \varepsilon_n)$; and thus generate $(y_1, \ldots, y_n)$. Then we fit the simple linear regression model and compute the $100(1-\alpha)\%$ confidence interval for β_1. We record whether the true value of β_1 (i.e., $\beta_1 = 5$) lies in this interval or not. If we perform this whole procedure independently N times (using `for` loop of otherwise), we can compute what proportion of those intervals contain the true value of β_1. This proportion is called the "coverage probability" of the confidence inetrval.

Perform the above experiment with $\alpha = 0.05$, $n = 20, 50, 100, 200$ and $N = 100000$. Report the coverage probabilities. Are they close to the theoretical value of 0.95?

```r
beta_0 <- 2
beta_1 <- 5
sigma <- 5
alpha <- 0.05
N <- 100000
sample_sizes <- c(20, 50, 100, 200)

simulate_CI_coverage <- function(n, N, beta_0, beta_1, sigma, alpha) {
  coverage_count <- 0
  for (i in 1:N) {
    x <- rnorm(n, mean = 0, sd = 1)
    epsilon <- rnorm(n, mean = 0, sd = sigma)

    y <- beta_0 + beta_1 * x + epsilon

    model <- lm(y ~ x)

    CI <- confint(model, level = 1 - alpha)[2, ]

    if (CI[1] <= beta_1 && CI[2] >= beta_1) {
      coverage_count <- coverage_count + 1
    }
  }

  #find coverage probability
  coverage_prob <- coverage_count / N
  return(coverage_prob)
}
```

```r
coverage_results <- sapply(sample_sizes, function(n) {
  simulate_CI_coverage(n, N, beta_0, beta_1, sigma, alpha)
})
```

```
# Display the results
coverage_results
```

```
## [1] 0.95118 0.95038 0.95018 0.95001
```

these values are close to 0.95. ## (b) (20 points)

In part (a), the modelling assumptions are satisfied clearly (since we have simulated the dataset using that model). Now we shall try to evaluate the performance of the confidence inetrval if some modelling assumptions are not met with. In this exercise, we focus on constant variance (or homoscedasticity) assumption. We shall simulate our dataset from the following model :

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \;\; i = 1, \ldots, n,$$

where

$$x_1, \ldots, x_n \overset{i.i.d.}{\sim} N(0,1), \;\; \varepsilon_i \overset{ind}{\sim} N(0, \sigma_i^2),$$

with $\beta_0 = 2, \beta_1 = 5, \sigma_i = 5\sqrt{|x_i|}$. x_i's and ε_i's are independent of each other. This model is clearly heteroscedastic, i.e., has non-constant variance.

Simulate from this dataset and report coverage probabilities for $\alpha = 0.05$, $n = 20, 50, 100, 200$ and $N = 100000$. How do these compare with those reported in part (a)?

Hint : The command `rnorm(n,mu,sigma)` allows the input `mu` and `sigma` to be arrays, where the i-th element of the sample will be Normally distributed with mean and standard deviation being the i-th entries of `mu` and `sigma` respectively. If `mu` (or `sigma`) is not an array but a nnmber, then all entries of the sample has the same mean (or standard deviation). After generating $(x_1, \ldots, x_n)$, create an array of size n whose i-th entry is $5\sqrt{|x_i|}$ and pass this array as input to `rnorm` in order to generate $(\varepsilon_1, \ldots, \varepsilon_n)$.

```
beta_0 <- 2
beta_1 <- 5
alpha <- 0.05
N <- 10000
sample_sizes <- c(20, 50, 100, 200)
simulate_heteroscedastic_CI_coverage <- function(n, N, beta_0, beta_1, alpha) {
  coverage_count <- 0

  for (i in 1:N) {

    x <- rnorm(n, mean = 0, sd = 1)

    epsilon <- rnorm(n, mean = 0, sd = 5 * sqrt(abs(x)))

    y <- beta_0 + beta_1 * x + epsilon

    model <- lm(y ~ x)

    CI <- confint(model, level = 1 - alpha)[2, ]

    if (CI[1] <= beta_1 && CI[2] >= beta_1) {
      coverage_count <- coverage_count + 1
    }
  }
  coverage_prob <- coverage_count / N
  return(coverage_prob)
}

coverage_results_heteroscedastic <- sapply(sample_sizes, function(n) {
  simulate_heteroscedastic_CI_coverage(n, N, beta_0, beta_1, alpha)
})

coverage_results_heteroscedastic
```

```
## [1] 0.8466 0.8407 0.8363 0.8333
```

These results are lower because of the smaller sample sizes. The assumption of constant variance is violated, which affects the performance of the confidence intervals