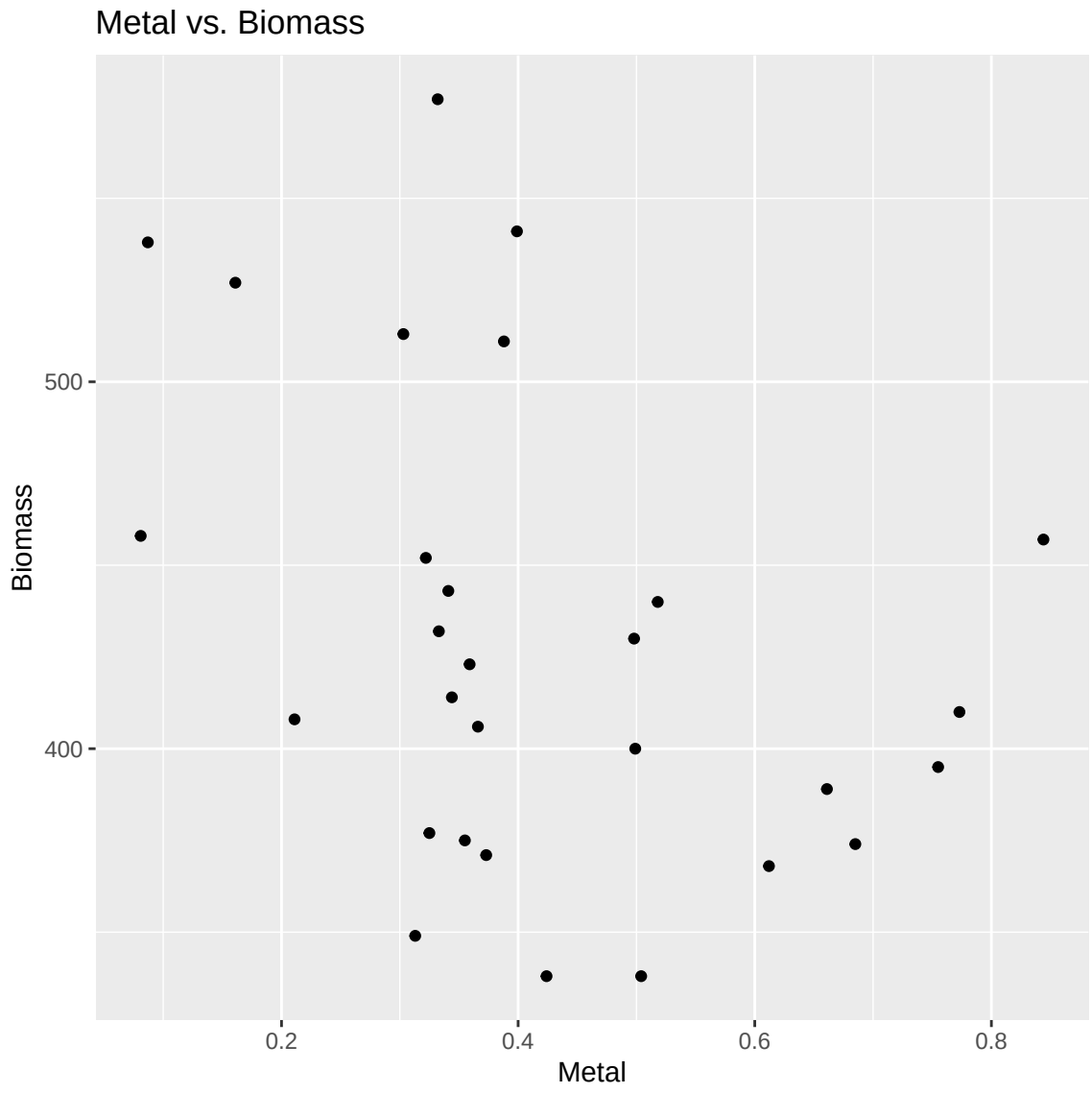
Setup CHUNK

```
suppressMessages(library(magrittr))
suppressMessages(library(ggplot2))
suppressMessages(library(tidyr))
suppressMessages(library(dplyr))
suppressMessages(library(modelr))
suppressMessages(library(broom))
suppressMessages(library(infer))

Fisheries <- read.csv("Fisheries.csv")
Civic <- read.csv("Civics.csv")
```

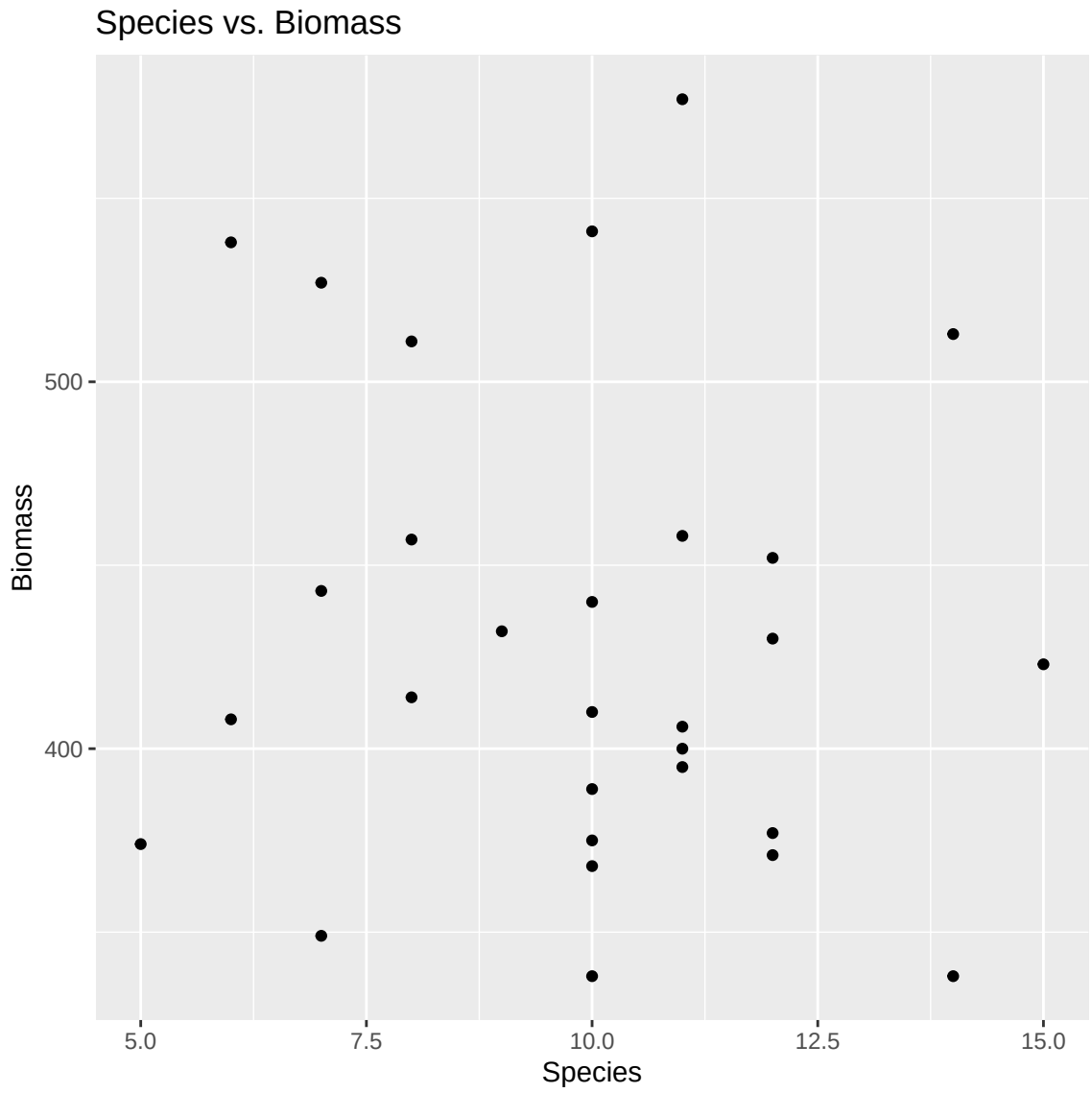
EX 1

a)

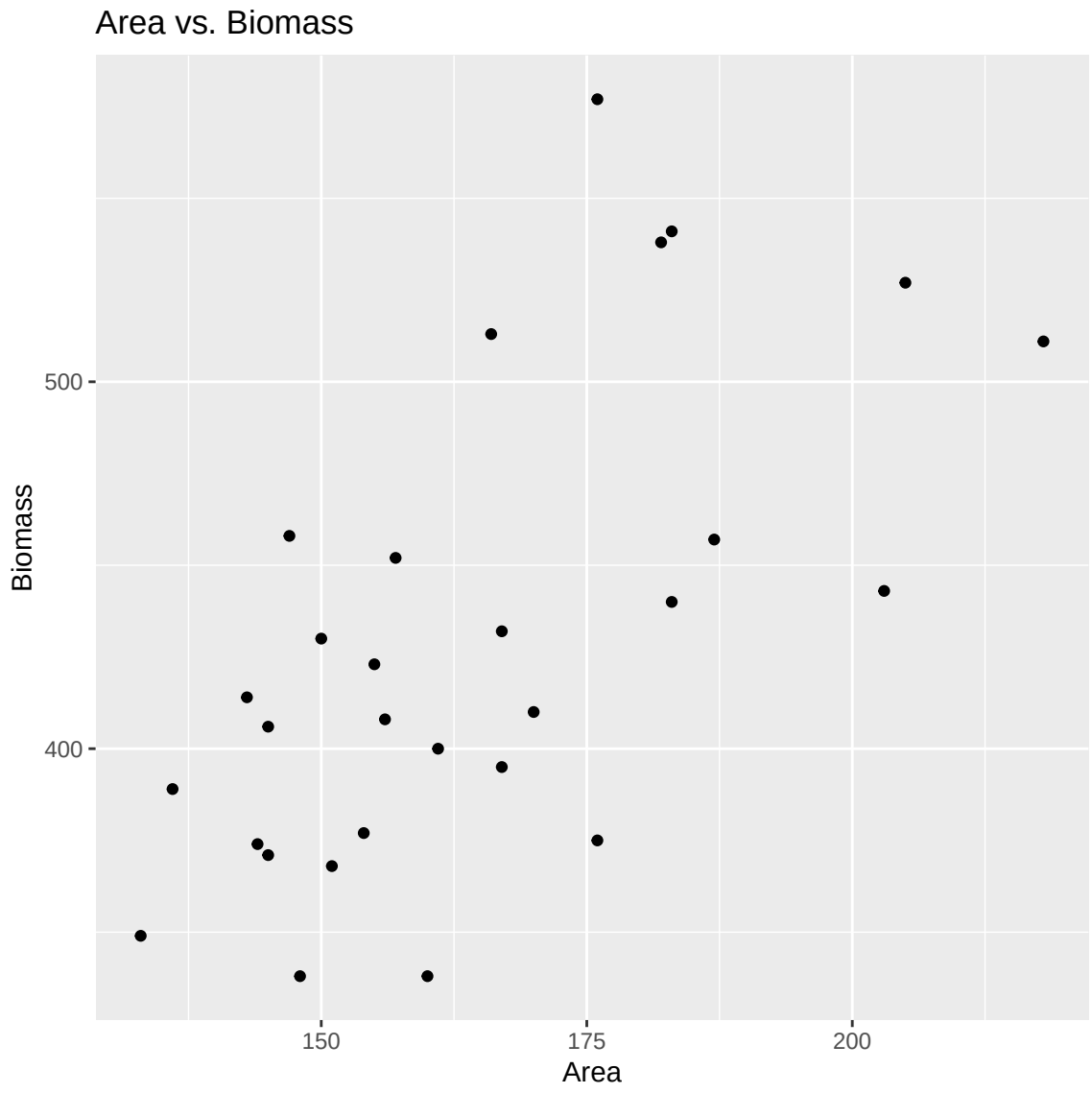
```
# Scatterplot of Metal vs. Biomass
ggplot(Fisheries, aes(x = Metal, y = Biomass)) +
  geom_point() +
  labs(title = "Metal vs. Biomass", x = "Metal", y = "Biomass")
```




```
# Scatterplot of Species vs. Biomass
ggplot(Fisheries, aes(x = Species, y = Biomass)) +
  geom_point() +
  labs(title = "Species vs. Biomass", x = "Species", y = "Biomass")
```




```
# Scatterplot of Area vs. Biomass
ggplot(Fisheries, aes(x = Area, y = Biomass)) +
  geom_point() +
  labs(title = "Area vs. Biomass", x = "Area", y = "Biomass")
```



b) ***There seems to be a moderately strong, positive, and linear correlation between area and biomass. However, the same cannot be said about Either the species or the metal levels as there is no clear shape, strength, or trend for these distributions.***

c) ***The coefficient of correlation for Area v Biomass is 0.63. The coefficient of correlation for Metal v Biomass is -0.36. The coefficient of correlation for Species v Biomass is 0.11***

```
cor(Fisheries$Area, Fisheries$Biomass)  # Correlation between Area and Biomass
```

```
## [1] 0.6373746
```

```r
cor(Fisheries$Metal, Fisheries$Biomass) # Correlation between Depth and Biomass
```

```
## [1] -0.367857
```

```r
cor(Fisheries$Species, Fisheries$Biomass) # Correlation between Temperature and Biomass
```

```
## [1] -0.1118514
```

d) ***Based on the calculated correlation coefficients (area has the largest), it appears that "Area" has the strongest relationship with "Biomass" as such it seems as if her claim is relatively true.***

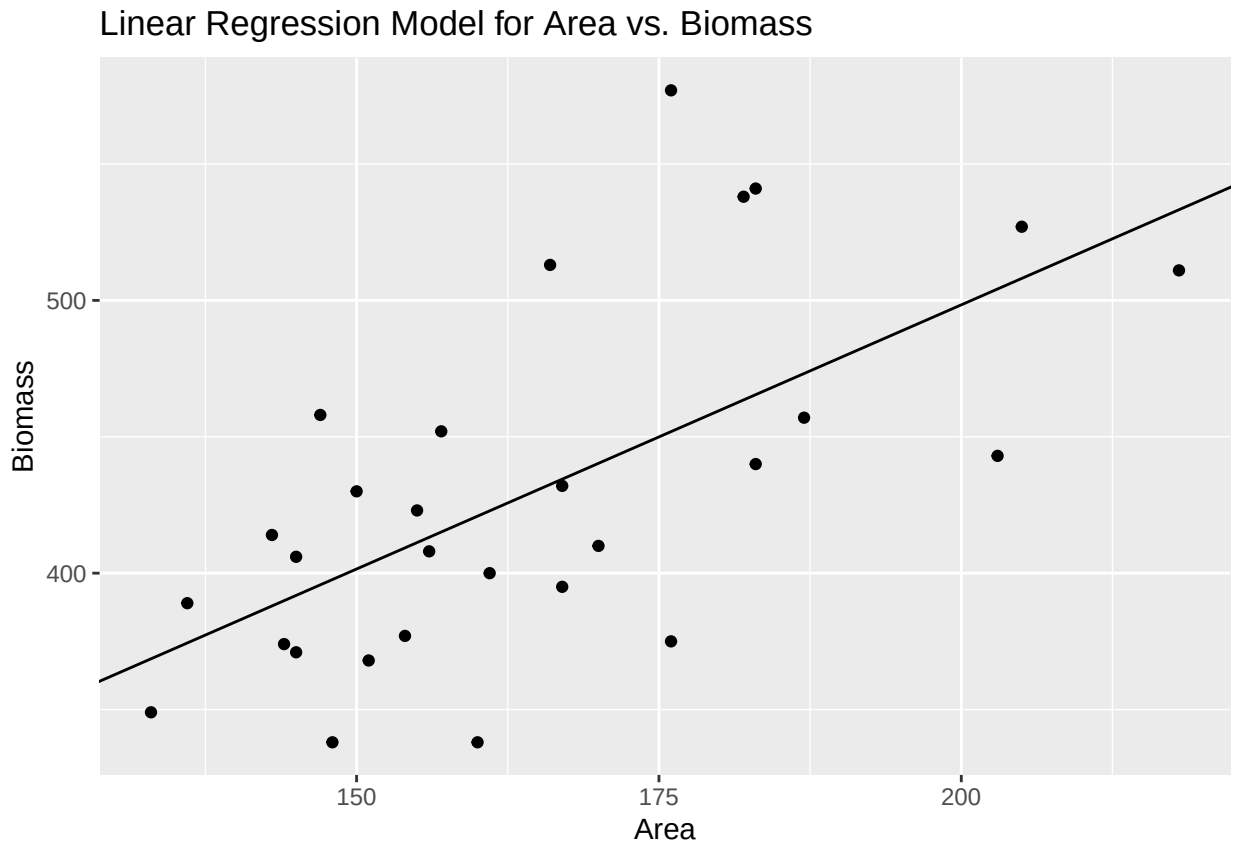
e)

```r
# Making the Regression Model
area_model <- lm(Biomass ~ Area, data = Fisheries)

# Outputting the Results from the Model
area_model %>%
  tidy()
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   111.      74.7        1.49 0.149
## 2 Area            1.94     0.451      4.30 0.000200
```

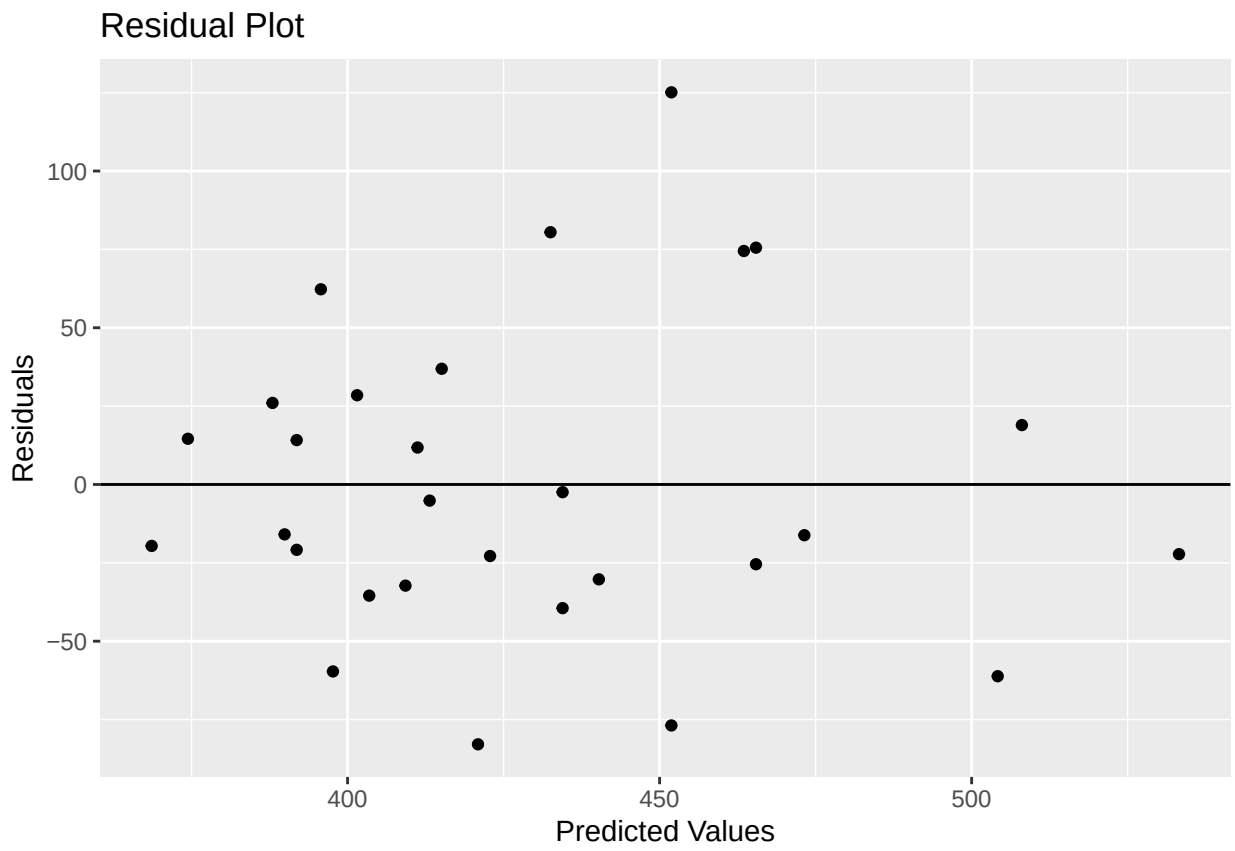
f)

```r
# Scatterplot of Area vs. Biomass
ggplot(Fisheries, aes(x = Area, y = Biomass)) +
  geom_abline(slope = area_model$coefficients[2],
              intercept = area_model$coefficients[1]) +
  geom_point() +
  labs(title = "Linear Regression Model for Area vs. Biomass", x = "Area", y = "Biomass")
```

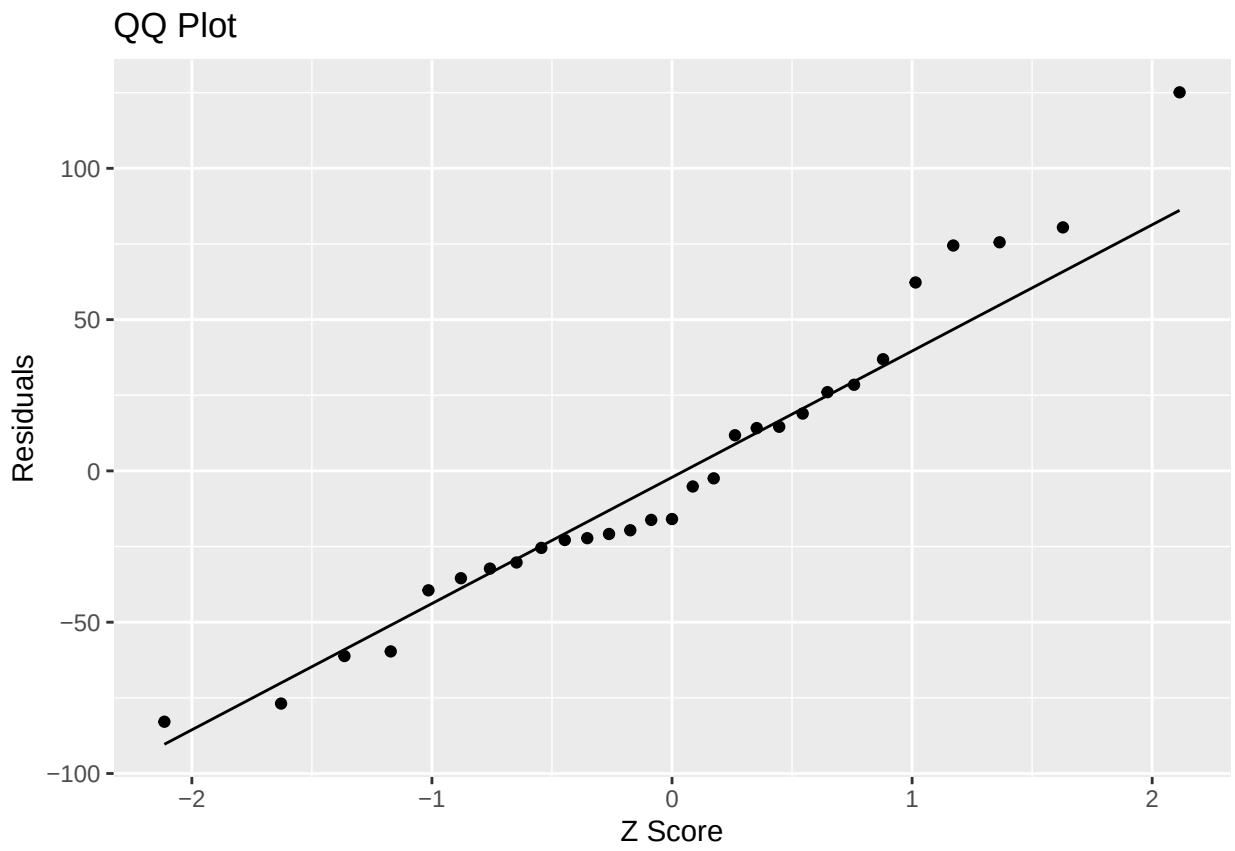


g)

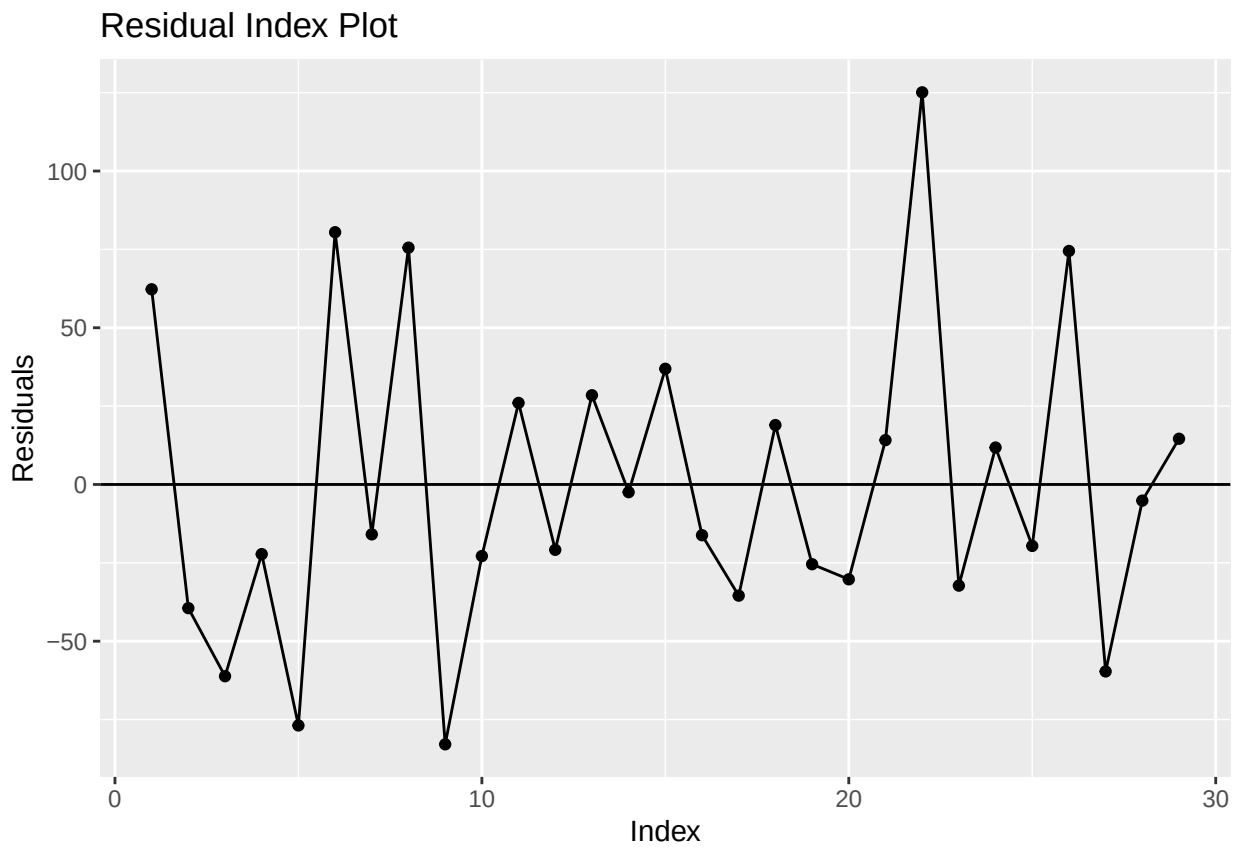
```
# Finding the Residuals and Predicted Values
Fisheries <- Fisheries %>%
  mutate(add_residuals(Fisheries,
                       area_model,
                       var = "resid"),
         add_predictions(Fisheries,
                         area_model,
                         var = "pred"))
Fisheries <- Fisheries %>%
  mutate(index = 1:29)
# Creating the Residual Plot
Fisheries %>%
  ggplot() +
  geom_point(mapping = aes(x = pred, y = resid)) +
  geom_hline(yintercept = 0) +
  labs(x = "Predicted Values",
       title = "Residual Plot",
       y = "Residuals")
```




```
# Creating the QQ plots
Fisheries %>%
  ggplot() +
  geom_qq(aes(sample = resid)) +
  geom_qq_line(aes(sample = resid)) +
  labs(x = "Z Score",
       title = "QQ Plot",
       y = "Residuals")
```




```
Fisheries %>%
  ggplot() +
  geom_line(mapping = aes(x = index, y = resid)) +
  geom_point(mapping = aes(x = index, y = resid)) +
  geom_hline(yintercept = 0) +
  labs(x = "Index",
       title = "Residual Index Plot",
       y = "Residuals")
```



h) ***The model seems to pass all of the assumptions we do not observe any grouping of positive or negative residuals within the index plot, a relatively normal distribution of residuals (as seen in the QQ plot), and relative homoscedasticity (as seen in the residual plot).***

i) ***Within the context of this model, we observe a 1.93 increase in body mass for every unit increase in the area. When the area is equal to 0, the body mass is equal to 111.02.*** **Our equation: E(y) or Body Mass = (1.93 or Beta1)(Area) + (111.02 or Beta2) + error term**

j) ***Our 95 percent confidence interval for the slope is 1.01 to 2.86***

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

```
##                   2.5 %     97.5 %
## (Intercept) -42.200413 264.248778
## Area          1.012162   2.861285
```

k) ***HYPOTHESES: NULL: Beta1 = 0 ALT: Beta1 =/= 0***

- *To calculate our test statistic, we must divide the difference between the observed statistic (which is 1.93) and our null parameter (which in this case is 0) by our standard error (which is 0.45) to get our*

test stat of 4.29. We then use this test stat on a t distribution with 27 (29-(1+1)) degrees of freedom to get our p value of 0.0002.

```
area_model %>%
  tidy() %>%
    select(term:p.value)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   111.      74.7        1.49 0.149
## 2 Area            1.94     0.451      4.30 0.000200
```

- *Our p value of 0.0002 tells us that we have significant evidence at a 0.01 significance level to suggest that our slope differs from 0 (our slope is significant).*

l) ***Within the context of this model, we observe a 1.93 increase in body mass for every unit increase in the area.***

m) ***It is not meaningful to interpret the y-intercept of this regression because it is impossible or very impractical for a Fishery to have an area of 0 acres.***

n) ***To calculate our R^2 value we need to calculate divide the Sum of the Squares for the Error 68916.49 by the Sum of the Squares for the Total 116069.17 and subtract that value from 1 to get 0.406 for our R^2. The work/code for this shown below***

```
# Getting our SSTotal
SStot <- sum((Fisheries$Biomass - mean(Fisheries$Biomass))^2)

# Getting our SSError
SSres <- sum(summary(area_model)$residuals^2)

# Calculating our R^2
rsqd <- 1 - SSres/SStot

print(paste("Our R^2 value is:", round(rsqd, 3)))
```

```
## [1] "Our R^2 value is: 0.406"
```

```
# Verifying our Results
area_model %>%
  glance %>%
    select(r.squared)
```

```
## # A tibble: 1 x 1
##   r.squared
##       <dbl>
## 1     0.406
```

- *This value of 0.4062 tells us that the ratio between the variation explained by our model and variation caused by random error is 0.4062. In other words we are told that roughly 40 percent of the variation is explained by the model.*

o) *To calculate our predicted values using our model, we must use the following equation 111.024182576256 + 1.93672372174676 * (area). Our results are 430.58, 498.36, and 566.15 each respective to 165, 200, and 235 acres.*

```
# Verification:
predict(area_model, newdata = data.frame(Area = c(165, 200, 235)))
```

```
##        1        2        3
## 430.5836 498.3689 566.1543
```

p)

```
# Generate predictions and intervals for specific values
new_data <- data.frame(Area = c(165, 200, 235))
pred_intervals <- predict(area_model, newdata = new_data, interval = "confidence")
pred_intervals2 <- predict(area_model, newdata = new_data, interval = "prediction")

# Create data frame with predictions and intervals
pred_df <- data.frame(AREA = new_data$Area,
                      LOWER = pred_intervals[,2],
                      UPPER = pred_intervals[,3])

pred_df2 <- data.frame(AREA = new_data$Area,
                      LOWER = pred_intervals2[,2],
                      UPPER = pred_intervals2[,3])

print(pred_df %>% mutate(TYPE = "confidence"))
```

```
##   AREA    LOWER    UPPER       TYPE
## 1  165 411.3263 449.8408 confidence
## 2  200 460.2498 536.4880 confidence
## 3  235 498.1132 634.1953 confidence
```

```
print(pred_df2 %>% mutate(TYPE = "prediction"))
```

```
##   AREA    LOWER    UPPER       TYPE
## 1  165 325.1477 536.0195 prediction
## 2  200 387.9200 608.8178 prediction
## 3  235 442.1564 690.1521 prediction
```

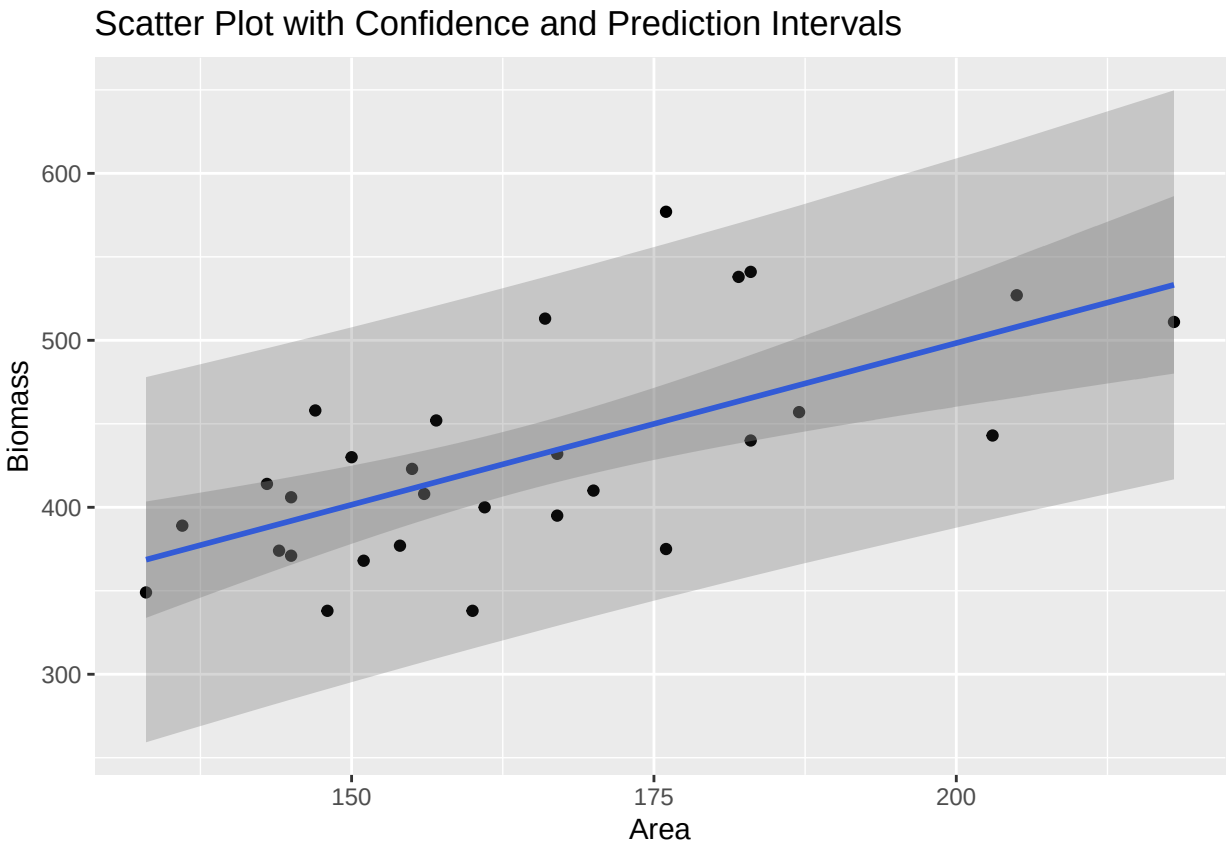
q) ***The 165 acres interval was the smallest. As the predictions move away from the mean "Area" value, there is more uncertainty and variability in the data, leading to wider confidence intervals and therefore less accuracy in the predictions.***

r) ***Yes, the value of 235 was extrapolation because that value is not within the range of our observed data (our observed value range is 133 to 218).***

s)

```
ggplot(Fisheries, aes(x = Area, y = Biomass)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE, level = 0.95) +
  geom_ribbon(aes(ymin = predict(area_model,
                                 interval = "predict",
                                 level = 0.95)[, "lwr"],
                  ymax = predict(area_model,
                                 interval = "predict",
                                 level = 0.95)[, "upr"]),
              alpha = 0.2) +
  labs(x = "Area",
       y = "Biomass",
       title = "Scatter Plot with Confidence and Prediction Intervals")
```

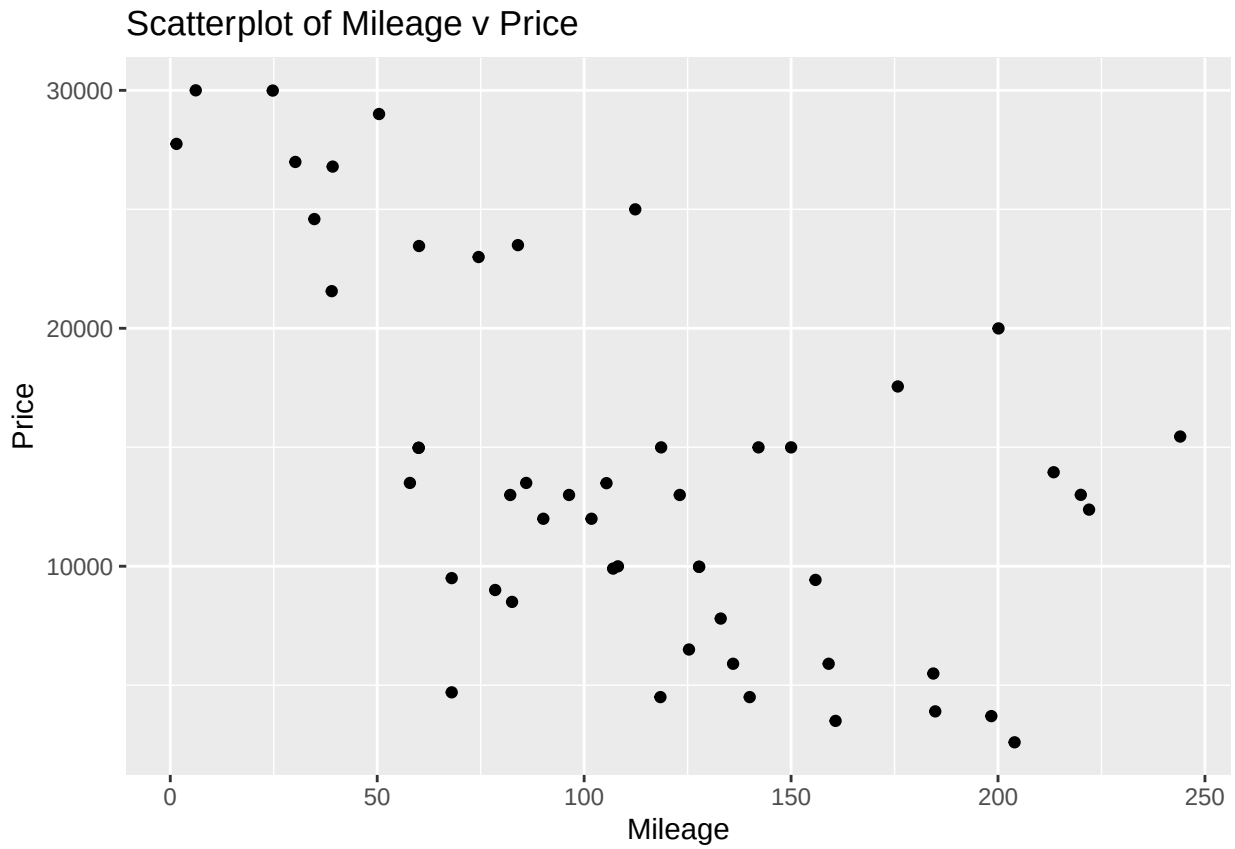
```
## Warning in predict.lm(area_model, interval = "predict", level = 0.95): predictions on current data r

## Warning in predict.lm(area_model, interval = "predict", level = 0.95): predictions on current data r

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



Exercise 2

a)

```
ggplot(Civic, aes(x = Mileage, y = Price)) +
  geom_point() +
  labs(title = "Scatterplot of Mileage v Price",
       x = "Mileage",
       y = "Price")
```



b) ***There seems to be a weak, negative, linear correlation between the Mileage and the Price of a car.***

c) ***To calculate our first fitted value and residual we must multiply the value we want to predict for by our slope which is -74.6748. We will then add this to our intercept of 22525.2285 to get a predicted value of 22413.2163. We will then subtract this from the observed value to get a residual value of 5336.784***

```
# Creating the linear regression model:
civic_model <- lm(Price ~ Mileage, data = Civic)

# Displaying the output
civic_model %>%
  tidy()
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)  22525.     1944.      11.6  1.21e-15
## 2 Mileage        -74.7      15.2     -4.91 1.04e- 5
```

```r
# Getting the residuals and predictions:
Civic <- Civic %>%
  mutate(add_residuals(Civic, civic_model,
                       var = "resid"),

         add_predictions(Civic, civic_model,
                       var = "pred"))

# Get the first mileage value
Civic <- Civic %>%
  arrange(Mileage)

# Getting first prediction and residual
print("The predicted value is:")
```

```
## [1] "The predicted value is:"
```

```r
Civic$pred[1]
```

```
## [1] 22413.22
```

```r
print("The residual is:")
```

```
## [1] "The residual is:"
```

```r
Civic$resid[1]
```

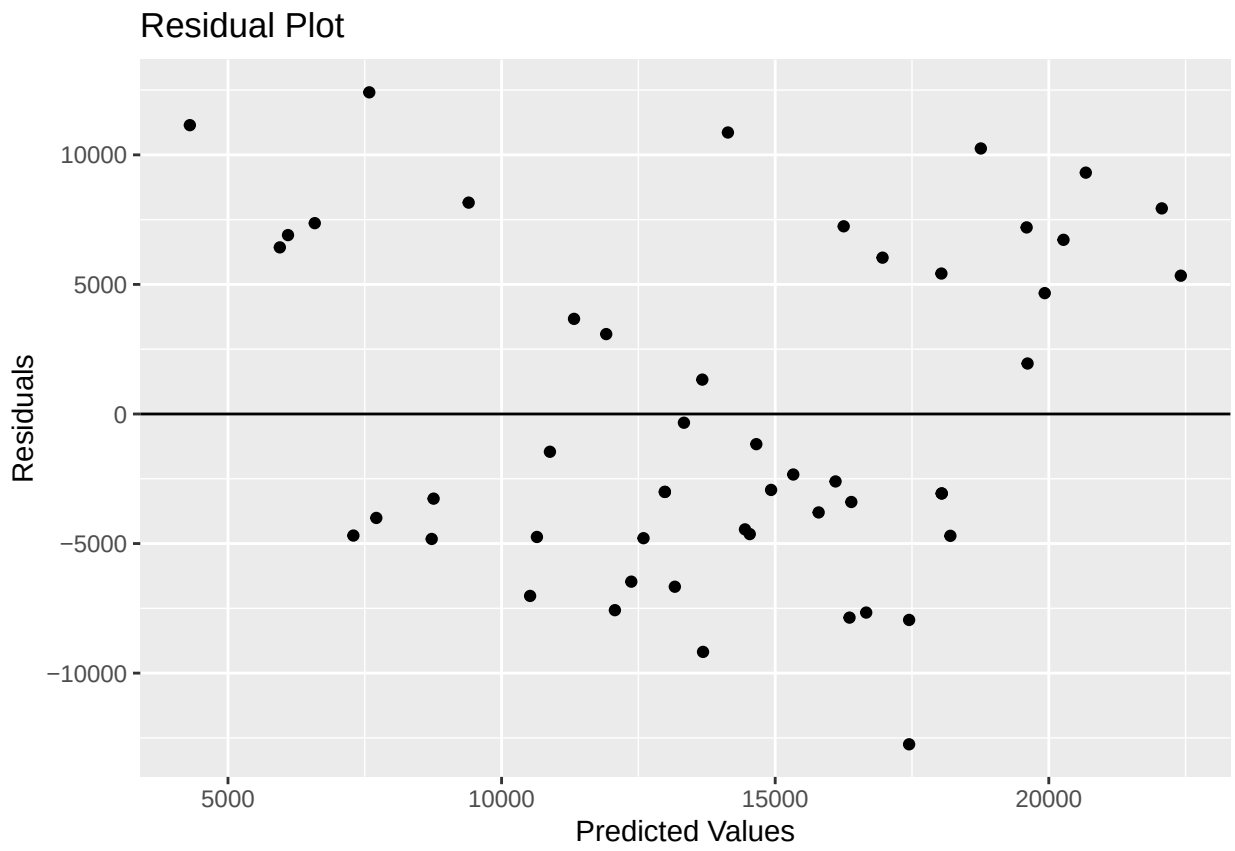
```
## [1] 5336.784
```

d)

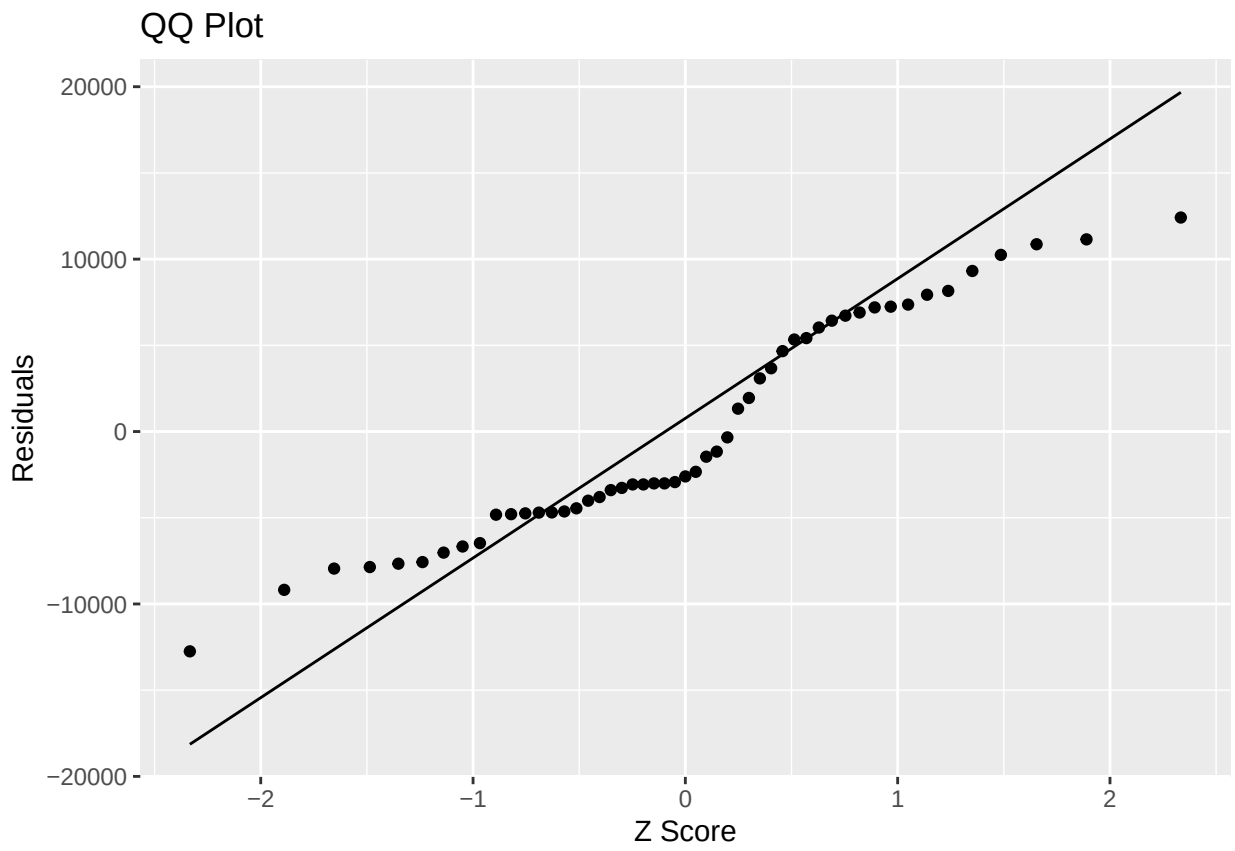
```r
Civic %>%
  ggplot() +
  geom_point(mapping = aes(x = Mileage, y = Price)) +
  geom_abline(slope = civic_model$coefficients[2],
              intercept = civic_model$coefficients[1]) +
  labs(title = "Linear Regression Model for Mileage v Price")
```

e) ***The model seems to sort of pass of one assumption as we do not observe a normal distribution of residuals (as seen by the non linear shape the QQ plot), and relative homoscedasticity (as seen in the residual plot)***

```
# Residual Plots
Civic %>%
  ggplot() +
  geom_point(mapping = aes(x = pred, y = resid)) +
  geom_hline(yintercept = 0) +
  labs(title = "Residual Plot",
       x = "Predicted Values",
       y = "Residuals")
```

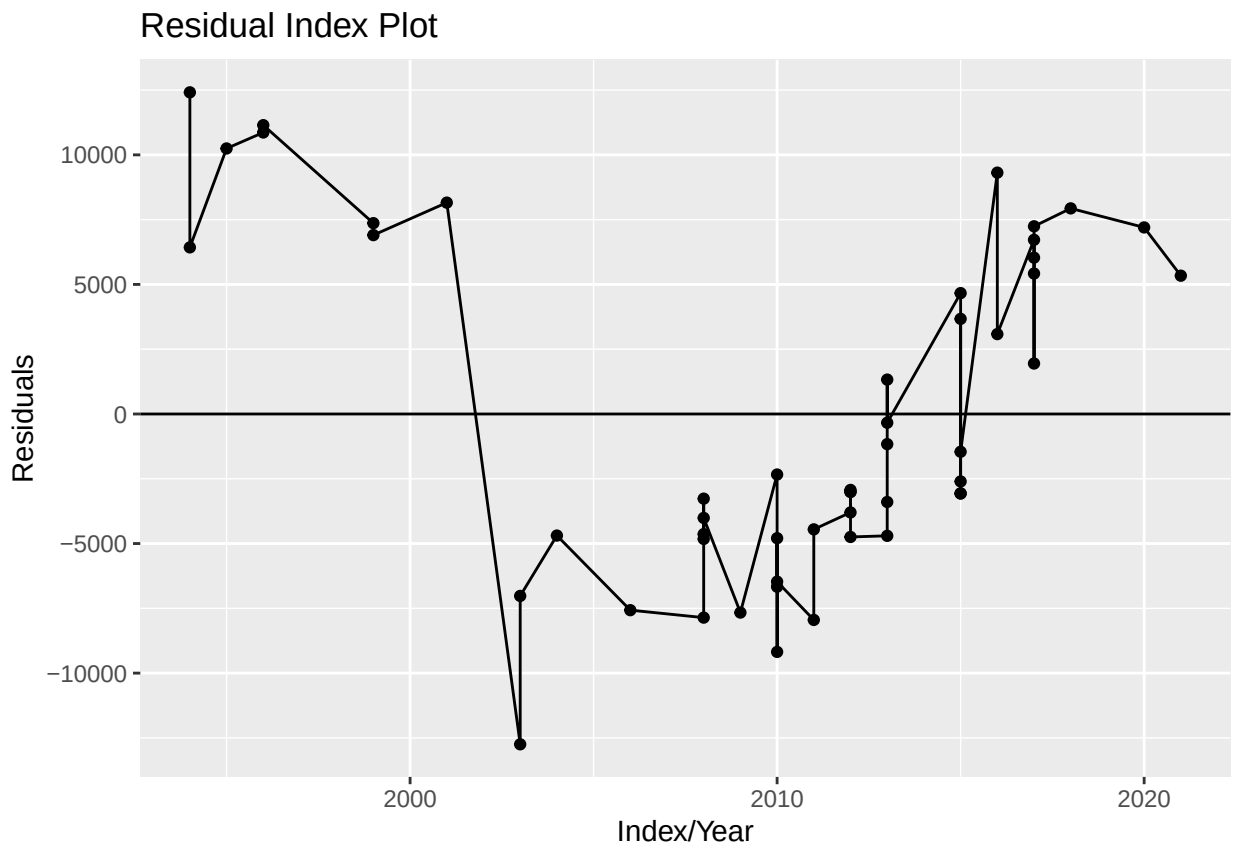



```
Civic %>%
  ggplot() +
  geom_qq(aes(sample = resid)) +
  geom_qq_line(aes(sample = resid)) +
  labs(x = "Z Score",
       title = "QQ Plot",
       y = "Residuals")
```



f) ***The model fails as we see grouping of positive or negative residuals within the index plot.***

```
Civic %>%
  ggplot() +
  geom_point(mapping = aes(x = Year, y = resid)) +
  geom_line(mapping = aes(x = Year, y = resid)) +
  geom_hline(yintercept = 0) +
  labs(title = "Residual Index Plot",
       x = "Index/Year",
       y = "Residuals")
```



g) _Our equation is E(y) or Price = (-74.6748 or Beta1)*(Miles) + (22525.2285 or Beta0). Our slope within the context of this is that with every 1k mile increase in the mileage we observe a 74.64 dollar decrease in the price._

h) _Our standard error for the slope is 15.1940303331733. If we were to repeat our sampling and re-fit the model many times, using different random samples from the population, about 68% of the time the true value of the Mileage coefficient would be within one standard error of the estimated coefficient (i.e., between estimate - 15.194 and estimate + 15.194). Similarly, about 95% of the time the true value of the Mileage coefficient would be within two standard errors of the estimated coefficient (i.e., between estimate - 2*15.194 and estimate + 2*15.194). 21 of the residuals are greater than 2*SE which probably means our model does not do such a great job._

```
# Extract the standard error for the Mileage coefficient
se_mileage <- summary(civic_model)$coefficients[2, "Std. Error"]

# View the standard error for the Mileage coefficient
print(paste("Standard error for the Mileage coefficient:", se_mileage))
```

```
## [1] "Standard error for the Mileage coefficient: 15.1940303331733"
```

```r
# See how many residuals are greater than 2*SE
counter = 0
for (i in Civic$resid){
  if (i > 2*se_mileage){
    counter <- counter + 1
  }
}

print(counter)
```

```
## [1] 21
```

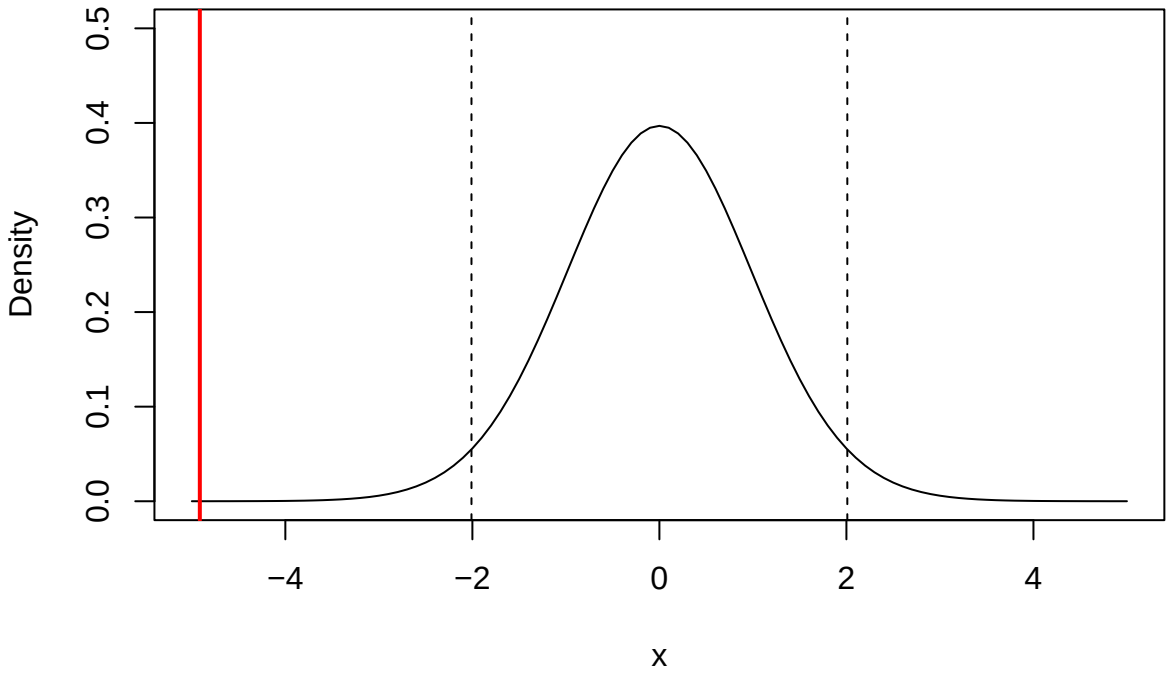
i) ***To get our test statistic for our slope, we must divide the difference between the slope (-74.6748) and the null parameter (which in this case would be 0) by the standard error of the slope (15.1940303331733). After we do this we get a test statistic of -4.9147. Our p value is 0.00001035873. This provides significance evidence to suggest that the slope is significant at a 0.05 significance level.***

- *Hypothesis tests: Null Hypothesis: The true slope coefficient of the regression model is equal to zero. Alternative Hypothesis: The true slope coefficient of the regression model is not equal to zero.*

```r
# Set the degrees of freedom and test statistic
df <- 49
t_stat <- -4.9147

# Calculate the p-value (two-tailed test)
p_val <- 2 * pt(abs(t_stat), df, lower.tail = FALSE)

# Print the p-value
cat("p-value =", format(p_val, scientific = FALSE), "\n")
```

```
## p-value = 0.00001035873
```

```r
# Visualize the t-distribution with the critical region shaded
curve(dt(x, df), xlim = c(-5, 5), ylim = c(0, 0.5), ylab = "Density")
abline(v = qt(0.025, df), lty = 2)
abline(v = -qt(0.025, df), lty = 2)
abline(v = t_stat, lwd = 2, col = "red")
```



j) ***An R-squared value of 0.3301869 for a linear model with mileage as the explanatory variable and price as the response variable indicates that 33.02% of the variation in the price of the car can be explained by the variation in its mileage.***

```
civic_model %>%
  glance() %>%
    select(r.squared)
```

```
## # A tibble: 1 x 1
##   r.squared
##       <dbl>
## 1     0.330
```

k) ***The change is not really discernible but we can notice a very slightly steeper downward slope and a slight upwards shift.***

```
Civic2 <- Civic[Civic$Year != 2001, ]
```

```
civic2_model <- lm(Price ~ Mileage, data = Civic2)

Civic2 %>%
  ggplot() +
  geom_point(mapping = aes(x = Mileage, y = Price)) +
  geom_abline(slope = civic2_model$coefficients[2], intercept = civic2_model$coefficients[1])
```