UE20CS312 - Data Analytics - Worksheet 2a - Simple Linear Regression

PES University

'SUNDEEP A, Dept. of CSE - PES1UG20CS445'

2022-09-06

Simple Linear Regression

Simple linear regression is a statistical technique for finding the existence of an association relationship between a dependent variable and an independent variable. Simple linear regression implies that there is only one independent variable in the model. Regression is one of the most important techniques in predictive analytics since many prediction problems are modeled using regression.Action Potentials in Dragons

Brain cells, called neurons (diagram shown below), send information throughout the brain and body. The information is sent via electro-chemical signals known as action potentials that travel down the length of the neuron. These neurons are then triggered to release chemical messengers at synapses, called neurotransmitters, which help trigger action potentials in nearby cells, and so help spread the signal all over. An action potential travels down a neuron's axon in an ion cascade. (Source: Khan Academy).

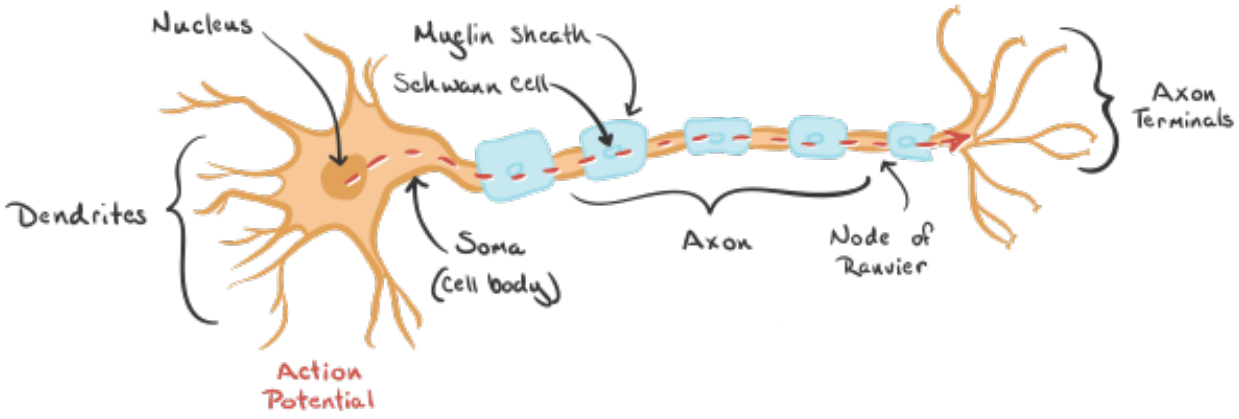


Figure 1: Diagram of a neuron - Source: Khan Academy

In the imaginary land of Westeros, the once extinct dragons were spotted again. The maesters of the capital, King's Landing, were summoned to study the nervous systems of these dragons. They were curious about how such large beings were able to move around so quickly. They studied 67 nerve bundles of two dragons and measured the **maximal conduction velocity** across fibers and the **axon diameter** of the largest fiber (Similar to the study conducted by Hursh in 1939). What they observed is stored on the GitHub repository.

Data Dictionary

```
axon_diameter: diameter of the axon in micrometers
conduction_velocity: conduction velocity of action potentials in meters per second
```

Points

The problems in this worksheet are for a total of 10 points with each problem having a different weightage.

- *Problem 1*: 1 point
- *Problem 2*: 3 points
- *Problem 3*: 3 points
- *Problem 4*: 1 point
- *Problem 5*: 2 points

Data reading

```
dragon_neurons <- read.csv('dragon_neurons.csv')
head(dragon_neurons)
```

```
##   X axon_diameter conduction_velocity X.1
## 1 0            72            4.541130  NA
## 2 1            66            4.275300  NA
## 3 2            74            4.912093  NA
## 4 3             9            2.872806  NA
## 5 4             9            2.395194  NA
## 6 5            65            5.120160  NA
```

Problem 1 (1 point)

Find if a linear model is appropriate for representing the relationship between the conduction velocity (response variable) and axon diameter (explanatory variable) by finding the OLS solution. Print out the slope and the coefficient. Plot the OLS best-fit line of the model (Hint: use the `ggplot` library).

```
#lm(dragon_neurons$conduction_velocity~dragon_neurons$axon_diameter)
#plot(dragon_neurons$conduction_velocity,dragon_neurons$conduction_velocity)
library(ggplot2)
ggp <- ggplot(dragon_neurons, aes(dragon_neurons$conduction_velocity,dragon_neurons$axon_diameter)) +
  geom_point(color='red') +
  ggtitle("Conduction Velocity vs Axon Diameter") + labs(y = "Conduction Velocity", x = "Axon Diameter")
ggp +stat_smooth(method = "lm",se= FALSE)
```

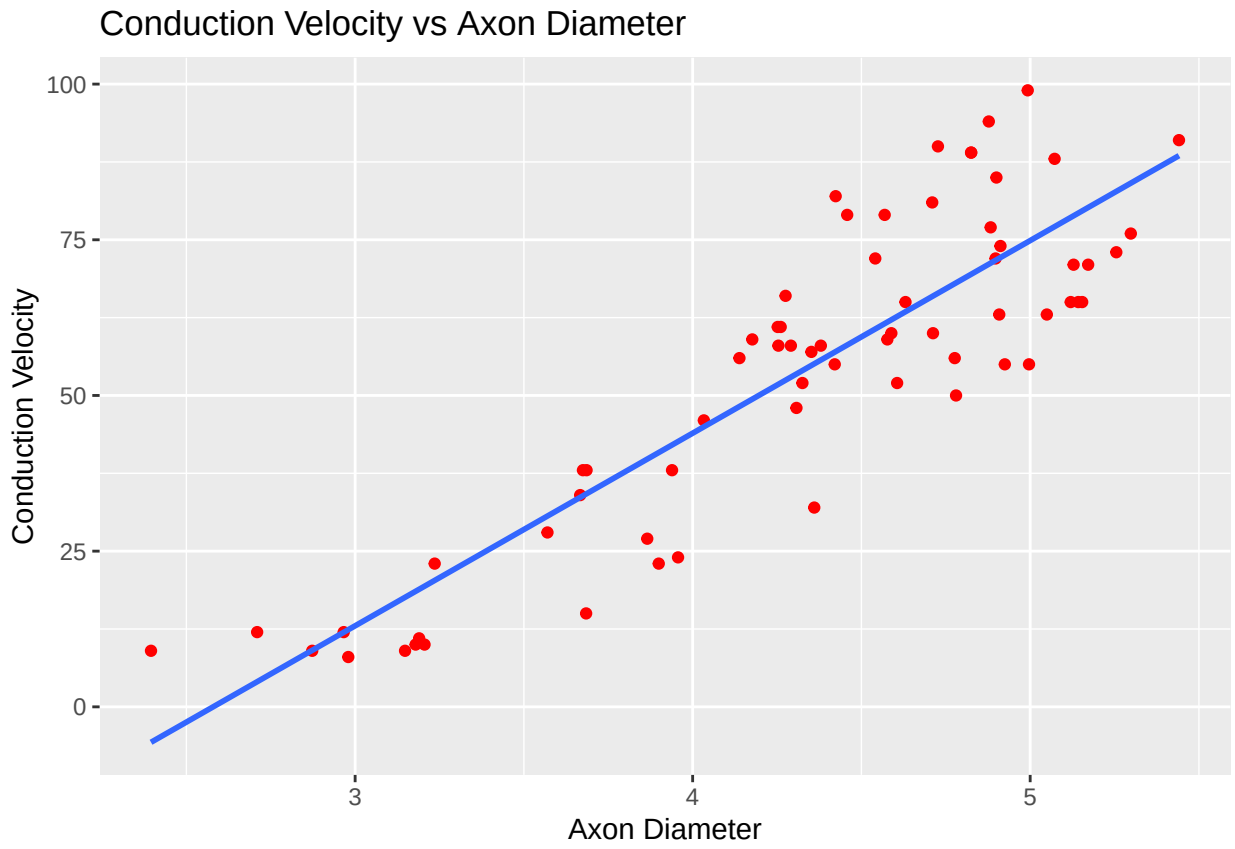
```
## Warning: Use of `dragon_neurons$conduction_velocity` is discouraged. Use
## `conduction_velocity` instead.
```

```
## Warning: Use of `dragon_neurons$axon_diameter` is discouraged. Use
## `axon_diameter` instead.
```

```
## Warning: Use of `dragon_neurons$conduction_velocity` is discouraged. Use
## `conduction_velocity` instead.
```

```
## Warning: Use of `dragon_neurons$axon_diameter` is discouraged. Use
## `axon_diameter` instead.
```

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




```
model <- lm(dragon_neurons$conduction_velocity ~ dragon_neurons$axon_diameter, data=dragon_neurons)

print("The co-efficients are : ")
```

```
## [1] "The co-efficients are : "
```

```
print(model)
```

```
##
## Call:
## lm(formula = dragon_neurons$conduction_velocity ~ dragon_neurons$axon_diameter,
##     data = dragon_neurons)
##
## Coefficients:
##                   (Intercept)  dragon_neurons$axon_diameter
##                       2.98761                       0.02475
```

```
summary(model)
```

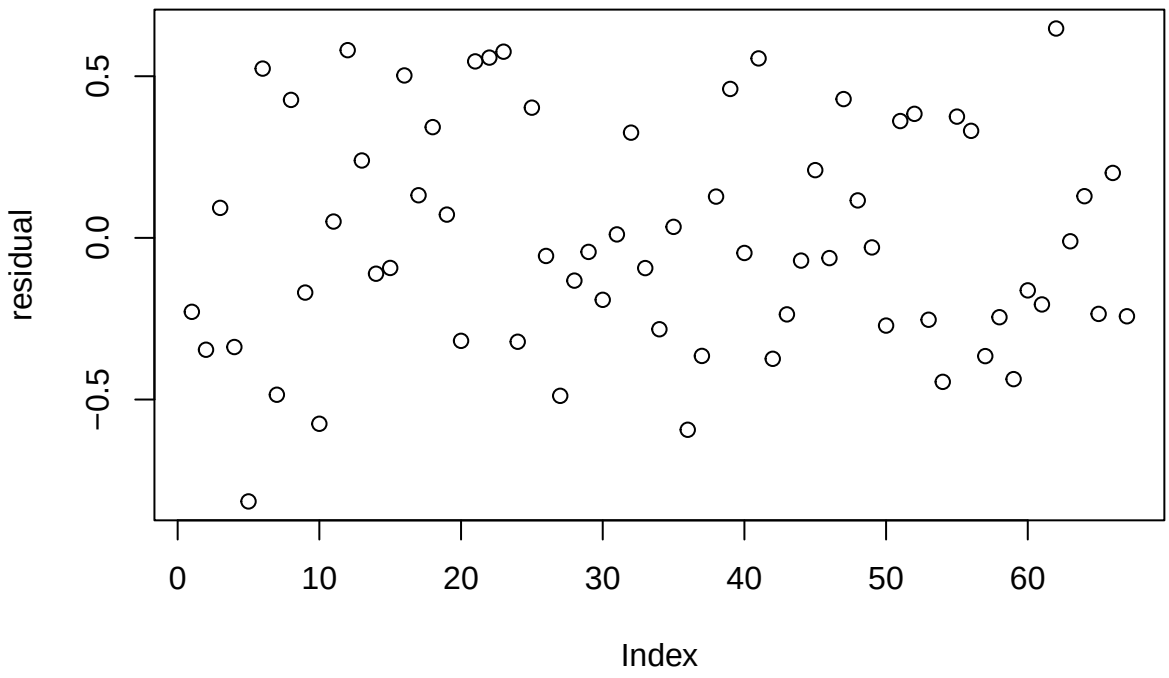
```
##
## Call:
## lm(formula = dragon_neurons$conduction_velocity ~ dragon_neurons$axon_diameter,
##     data = dragon_neurons)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.81519 -0.24935 -0.04665  0.32827  0.64757
##
```

```
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   2.987611   0.101069   29.56   <2e-16 ***
## dragon_neurons$axon_diameter 0.024753   0.001699   14.57   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3509 on 65 degrees of freedom
## Multiple R-squared:  0.7656, Adjusted R-squared:  0.762
## F-statistic: 212.3 on 1 and 65 DF,  p-value: < 2.2e-16
```

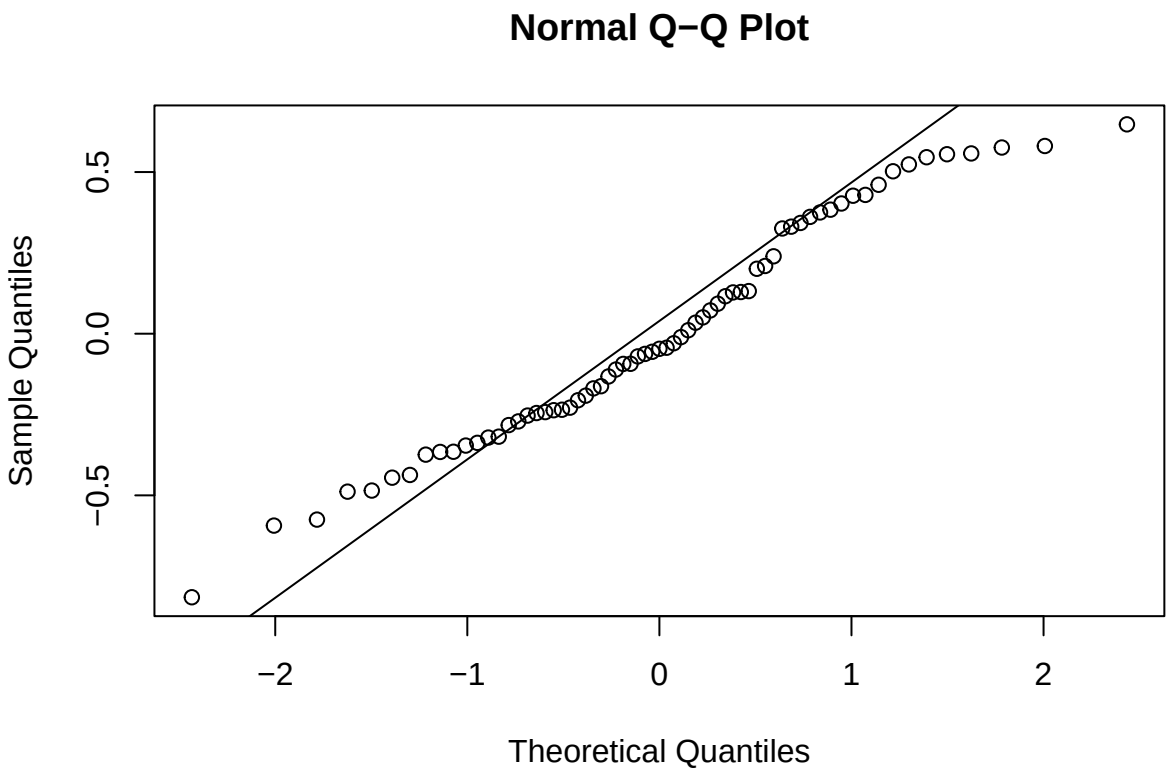
Problem 2 (3 points)

Plot the residuals of the model. Do the residuals look like white noise? If they do not, try to find a suitable functional form (hint: try transforming either x or y using natural-log or squares).

```
residual= resid(model)
plot(residual)
```



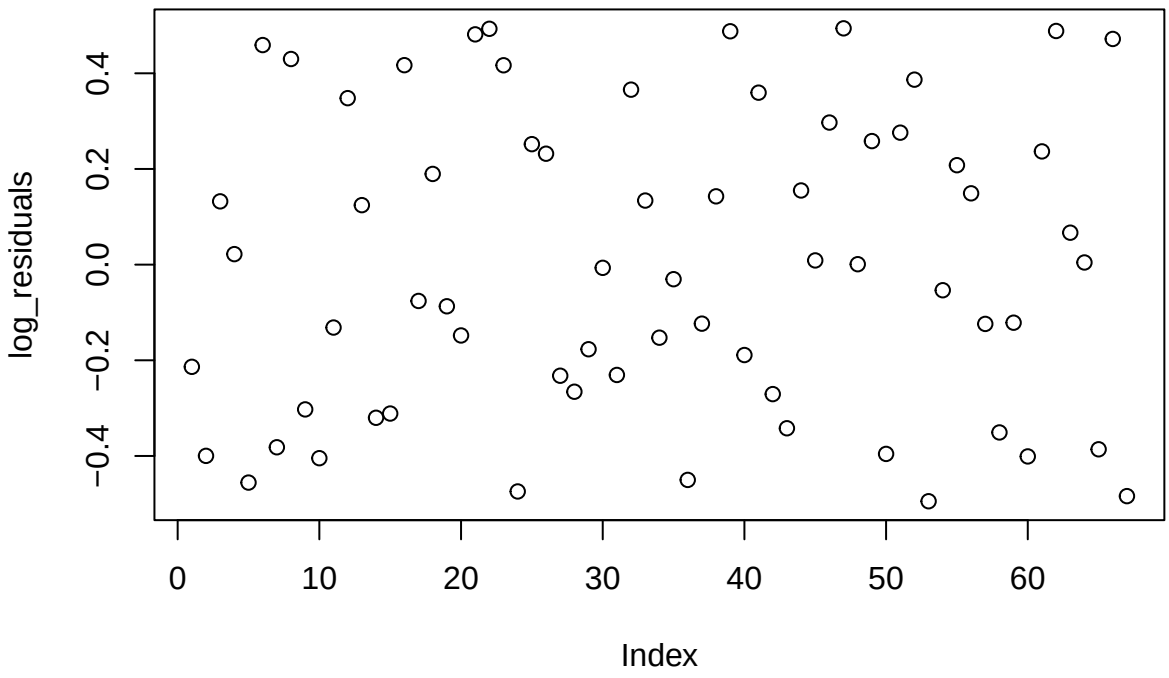

```
qqnorm(residual)
qqline(residual)
```



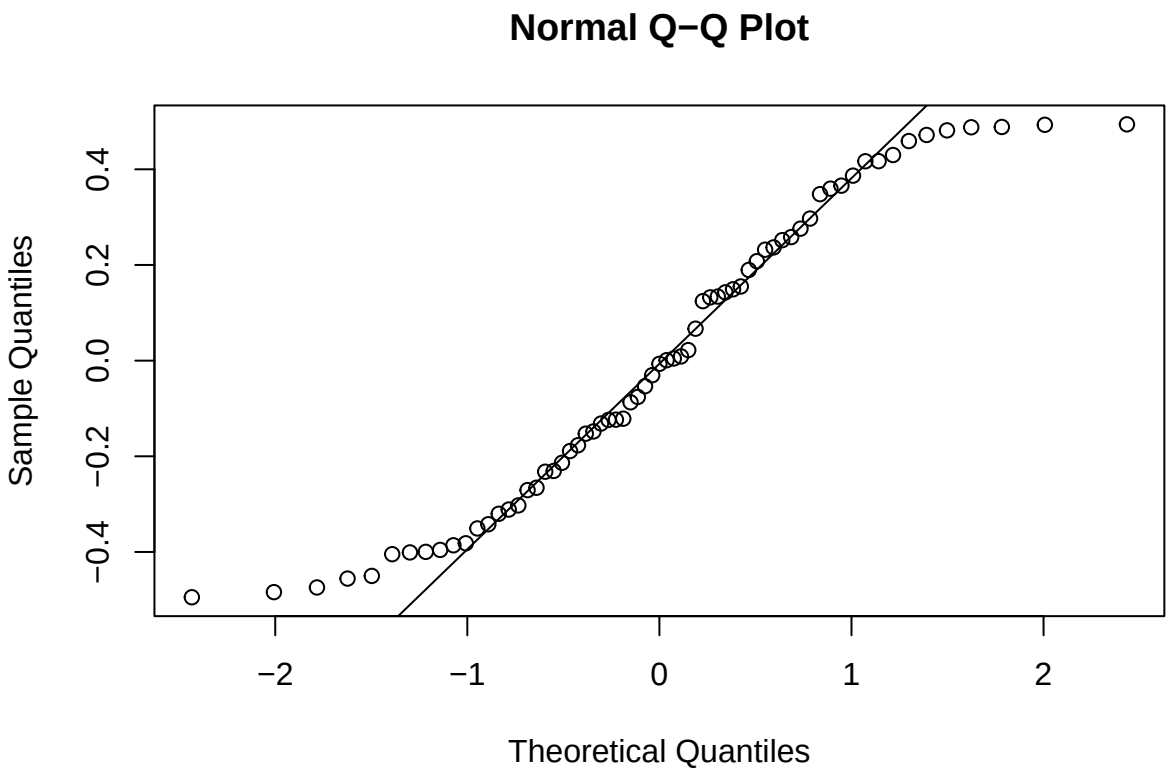

```
print("The residuals don't look like White noise.")
```

```
## [1] "The residuals don't look like White noise."
```

```
dragon_neurons$log_axon = log(dragon_neurons$axon_diameter)
log_model = lm(dragon_neurons$conduction_velocity ~ log_axon, data = dragon_neurons)
log_residuals =resid(log_model)
plot(log_residuals)
```




```
qqnorm(log_residuals)
qqline(log_residuals)
```



Reasoning:

From the initial Residuals data we can say that they don't look like White noise, We can infer this from the qqplot, since most of the data don't lie on the 45 degree line. So , we try to apply log function on axon_diameter parameter. After applying, we can see from the new results that most of the points lie on the 45 degree line in a qqplot.

Problem 3 (3 points)

Using Mahalanobis distance as a metric, are there any potential outliers you notice? What are their Mahalanobis distances? Use the model that you decided on in the previous problem (Problem 2) as your regression model. Ensure that you plot the ellipse with a radius equal to the square root of the Chi-square value with 2 degrees of freedom and 0.95 probability.

```
library(car)
```

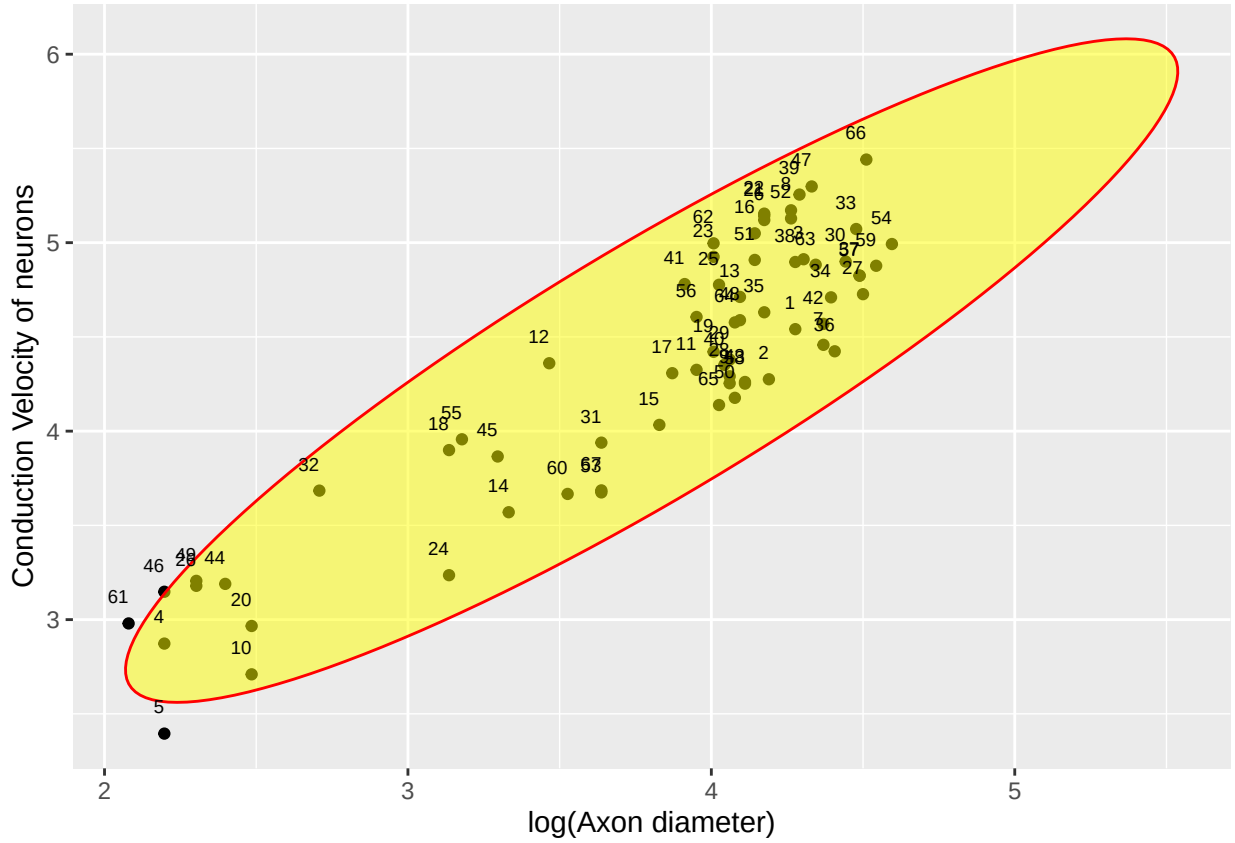
```
## Loading required package: carData
```

```
#here the data is conduction_velocity and log(axon_diameter) because we used that data in previous ques
data=dragon_neurons[,c("log_axon","conduction_velocity")]
data$mahalanobis_distance<-mahalanobis(data,colMeans(data),cov(data))
#Finding the outliers
outliers <- qchisq(p = 0.95 , df = 2)
print(data[data$mahalanobis_distance > outliers, ])
```

```
##    log_axon conduction_velocity mahalanobis_distance
## 5  2.197225           2.395194             7.289960
## 46 2.197225           3.147883             6.052955
```

```
## 61 2.079442            2.979719             6.500782
```

```
Chi_sqr  = qchisq(p = 0.95 , df = 2)
# Square root of Chi-Square value
ellipse_radius  = sqrt(Chi_sqr)
ellipse <- car::ellipse(center = colMeans(data[, c("log_axon", "conduction_velocity")]), shape = cov(da
ellipse <- as.data.frame(ellipse)
colnames(ellipse) <- colnames(data[, c("log_axon", "conduction_velocity")])
ggplot(data , aes(x = log_axon , y = conduction_velocity)) +
       geom_point(size = 1.5) +
       geom_polygon(data = ellipse , fill = "yellow" , color = "red" ,alpha = 0.5)+geom_text( aes(label
```



From the graph, we can see that there are 3 outliers. This is also verified from the mahalanobis distance

Problem 4 (1 point)

What are the R-squared values of the initial linear model and the functional form chosen in Problem 2? What do you infer from this? (hint: use the `summary` function on the created linear models)

```
print("The Summary of the original data")
```

```
## [1] "The Summary of the original data"
```

```
summary(model)
```

```
##
## Call:
## lm(formula = dragon_neurons$conduction_velocity ~ dragon_neurons$axon_diameter,
##     data = dragon_neurons)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.81519 -0.24935 -0.04665  0.32827  0.64757
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  2.987611   0.101069   29.56   <2e-16 ***
## dragon_neurons$axon_diameter 0.024753   0.001699   14.57   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3509 on 65 degrees of freedom
## Multiple R-squared:  0.7656, Adjusted R-squared:  0.762
## F-statistic: 212.3 on 1 and 65 DF,  p-value: < 2.2e-16
```

```
print("The summary of the modified data(log of axon_diameter)")
```

```
## [1] "The summary of the modified data(log of axon_diameter)"
```

```
summary(log_model)
```

```
##
## Call:
## lm(formula = dragon_neurons$conduction_velocity ~ log_axon, data = dragon_neurons)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.49467 -0.26822 -0.00671  0.25506  0.49396
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.83911    0.21037   3.989 0.000171 ***
## log_axon     0.91559    0.05439  16.833  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3131 on 65 degrees of freedom
## Multiple R-squared:  0.8134, Adjusted R-squared:  0.8105
## F-statistic: 283.3 on 1 and 65 DF,  p-value: < 2.2e-16
```

Inference:

The R-squared value of the Original Data Model is 0.765. The R-squared value of the Modified Data Model is 0.813. In general, the higher the R-squared, the better the model fits your data. So , we can say that The Modified Data Model is a better fit for the Data.

Problem 5 (2 points)

Using the same `summary` function as Problem 4, determine if there is a statistically significant linear relationship at a significance value of 0.05 of the **overall model** chosen in Problem 2. What do you understand about the relationship between dragons' axon diameters and conduction velocity? (Hint: understand the values displayed in `summary` and search for the right data).

```
summary(log_model)
```

```
##
```

```
## Call:
## lm(formula = dragon_neurons$conduction_velocity ~ log_axon, data = dragon_neurons)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.49467 -0.26822 -0.00671  0.25506  0.49396
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.83911    0.21037   3.989 0.000171 ***
## log_axon     0.91559    0.05439  16.833  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3131 on 65 degrees of freedom
## Multiple R-squared:  0.8134, Adjusted R-squared:  0.8105
## F-statistic: 283.3 on 1 and 65 DF,  p-value: < 2.2e-16
```

Analysis:

NULL Hypothesis: There is no significant Linear Relationship between axon diameters and conduction Velocity.

Alternate Hypothesis: There is a significant Linear Relationship between axon diameters and conduction Velocity.

Since the p value[2.2e-16] is very less than the significant value (0.05), Null hypothesis is rejected.Therefore there is a significant linear relationship between the axon diameters and conduction velocity