StoppingDistance

Jillian Maw

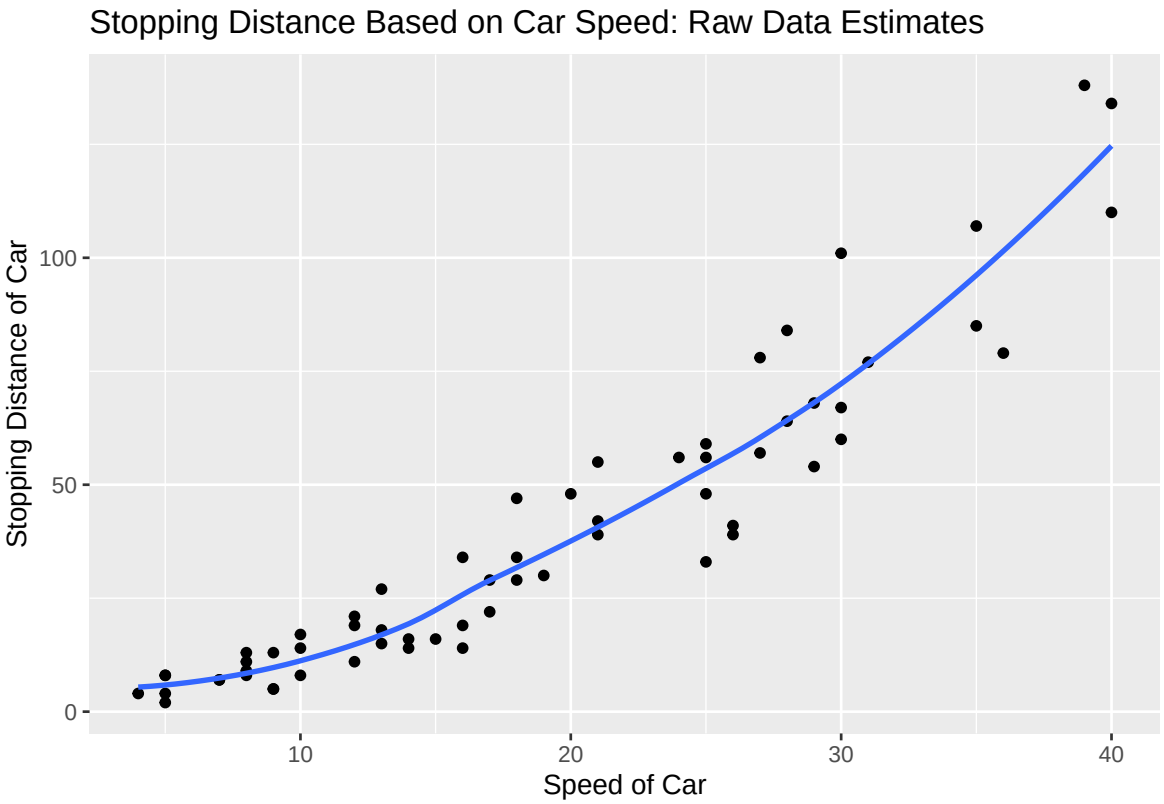
1/19/2022

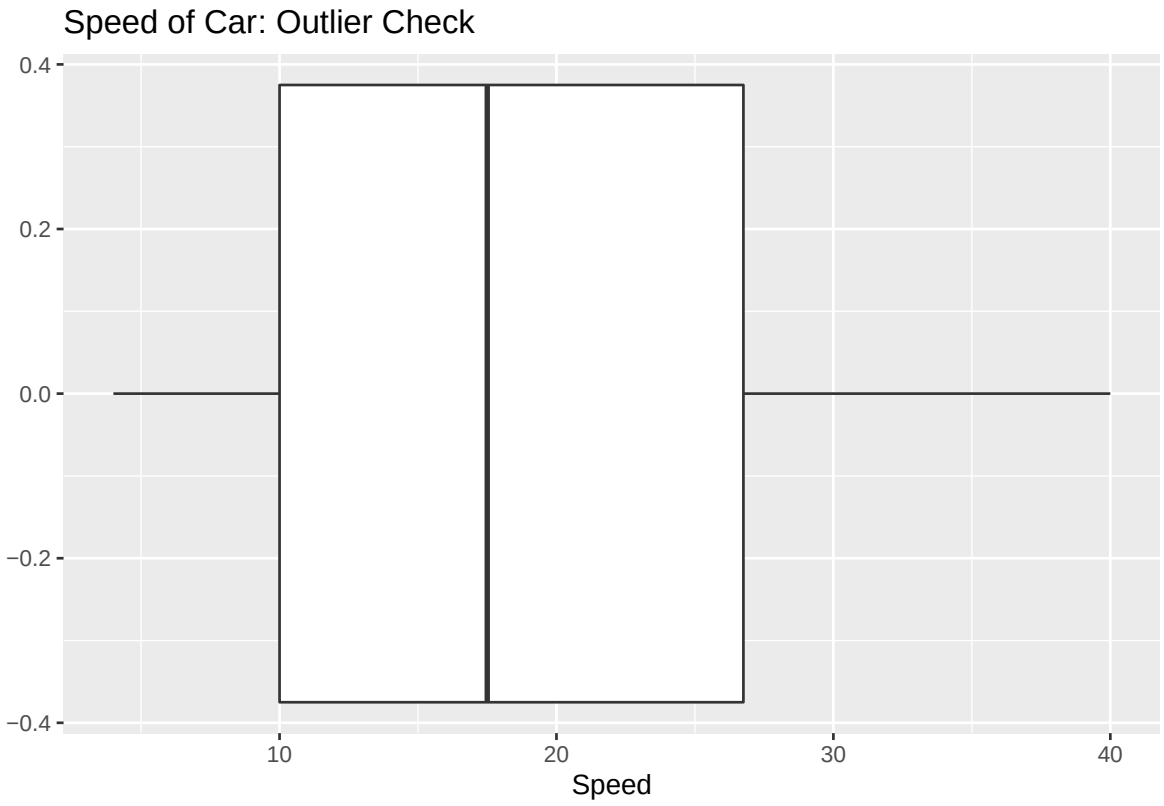
HOMEWORK ANALYSIS #2 - STOPPING DISTANCE

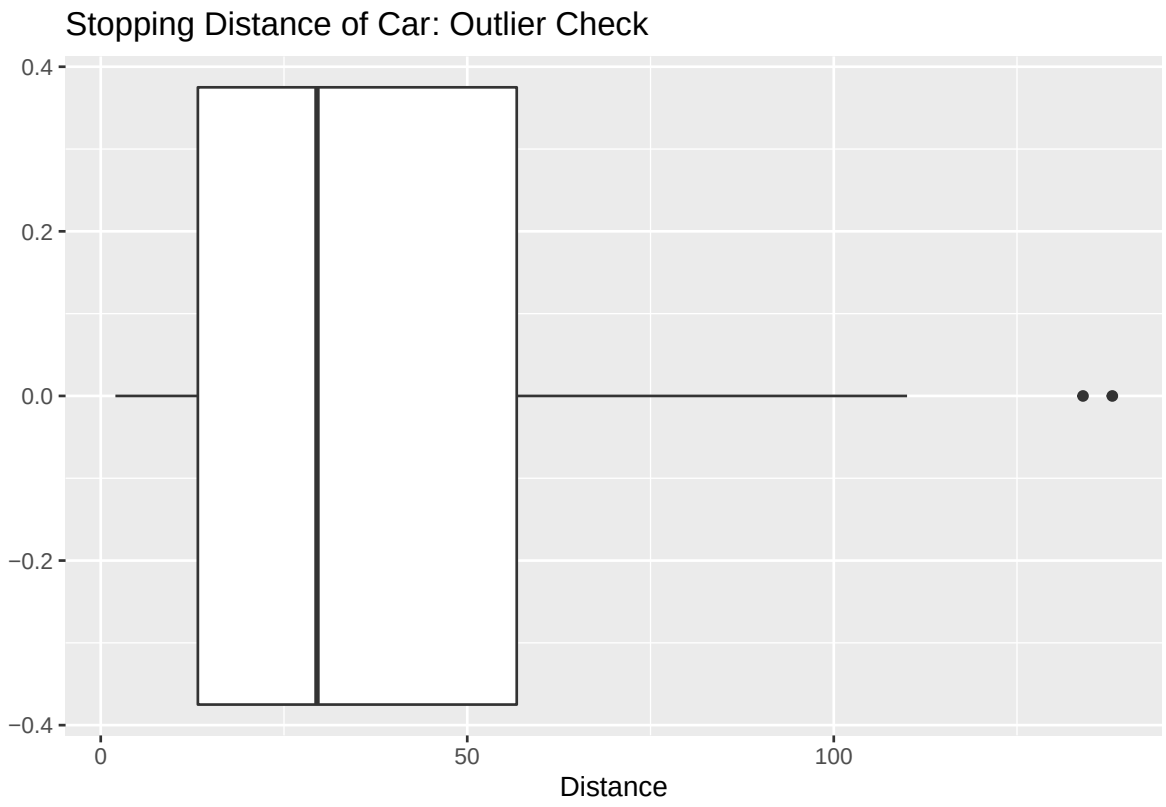
One key component of determining appropriate speed limits is the amount of distance that is required to stop at a given speed. For example, in residential neighborhoods, when pedestrians are commonly in the roadways, it is important to be able to stop in a very short distance to ensure pedestrian safety.
The dataset `StoppingDistance.txt` compares the distance (in feet) required for a car to stop on a certain rural road against the speed of the car. In each of the following questions, assume that your audience (the people you are writing your answer to) are law enforcement officials who have a weak statistical and mathematical background (be sure to explain things quite simply). Please attach your clearly commented code (R or Python) to the back of your answers as an appendix.

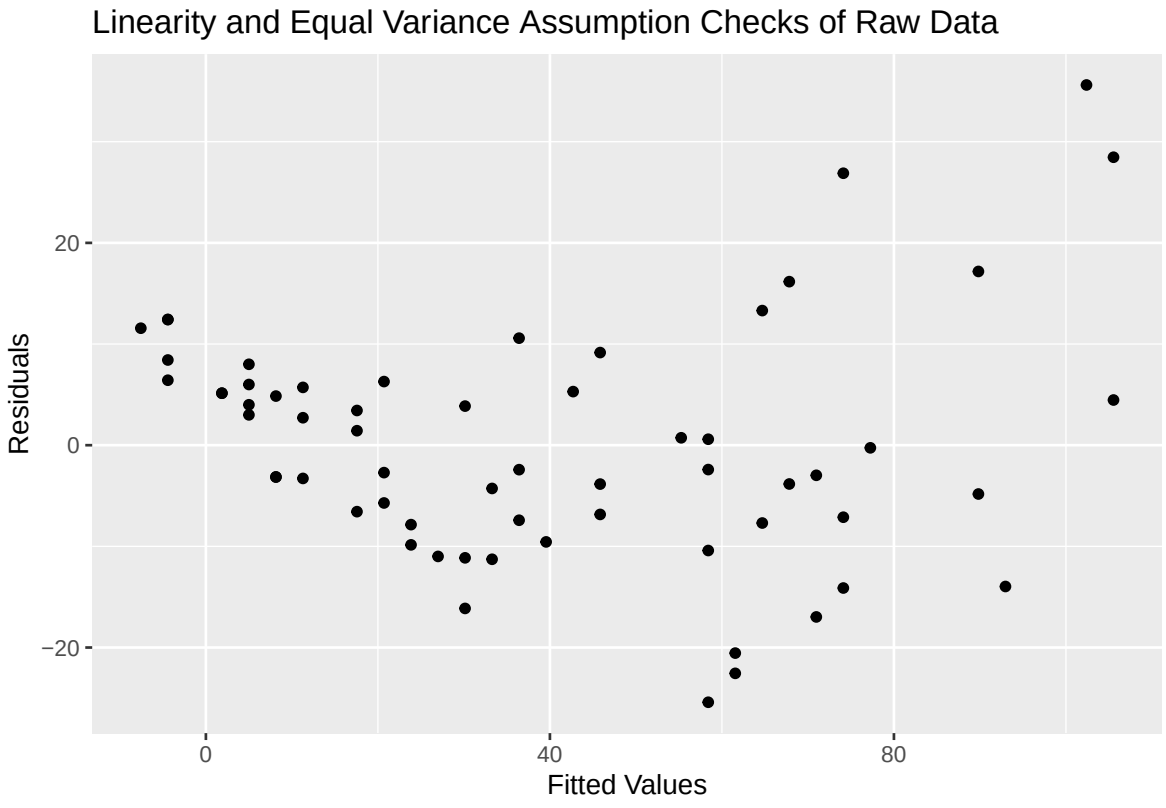
1. In your own words, summarize the overarching problem and any specific questions that need to be answered using the stopping distance data. Discuss how statistical modeling will be able to answer the posed questions.
   The overarching problem for the collected data is to determine how far a car travels when breaking at different speeds. This is important to determine so as to prevent serious injury or death from a car and pedestrian accident; cars need to be able to stop before a maximum distance to avoid hitting street-crossing pedestrians. Statistical modeling will allow us to quantify the strength of the relationship between speed and distance required for breaking, and help us make predictions about the best speed limits to post for certain required distances, such as intersections of roads. This will increase public safety.

2. Use the data to assess if a simple linear regression model (without doing any transformations) is suitable to analyze the stopping distance data. Justify your answer using any necessary graphics and relevant summary statistics. Provide discussion on *why* an SLR model on the raw data (not transformed) is or is not appropriate.


Stopping Distance Based on Car Speed: Raw Data Estimates
Stopping Distance of Car
100
50
0
10
20
30
40
Speed of Car


Speed of Car: Outlier Check
0.4
0.2
0.0
−0.2
−0.4
10
20
30
40
Speed

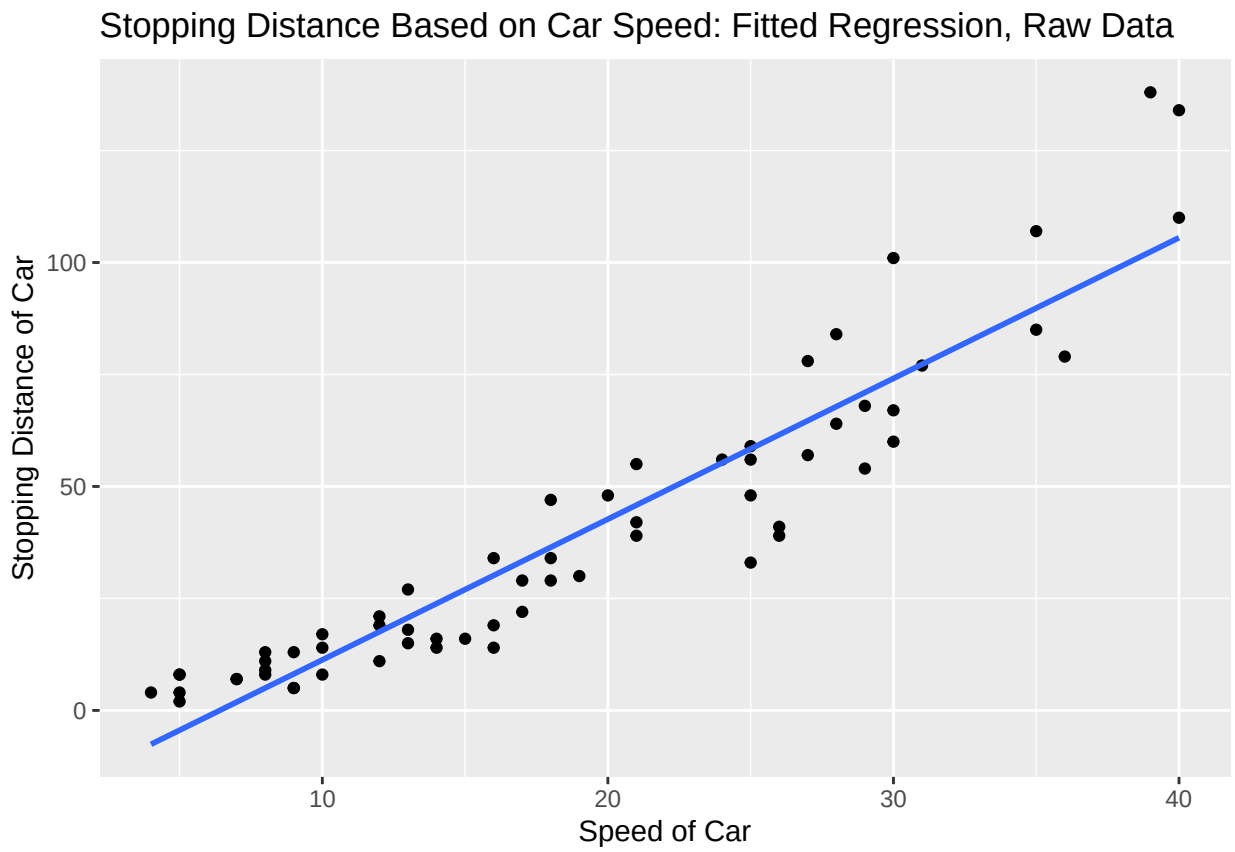





```
## Warning in ks.test(std.resids, "pnorm"): ties should not be present for the
```

```
## Kolmogorov-Smirnov test

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  std.resids
## D = 0.082017, p-value = 0.7985
## alternative hypothesis: two-sided
```



A simple linear model is not appropriate to analyze the stopping distance data, because the data is not following a linear relationship, especially in the ranges of data of 30 to 40 mph. For one, the first graph above shows that the estimated relationship for the data is a positively increasing curve instead of a positively increasing straight line. For two, the boxplot of Distance data shows two possible outliers that affect the strength of the relationship. Three, a more robust, mathematical check of the linearity of the data via a fitted values vs. residuals plot shows the curving pattern and widening spread in the data. Four, a goodness of fit test on the data shows it is not as normal a distribution as it could be. It would be better to transform the data to account for the relationship between speed and breaking distance increasing over speed.

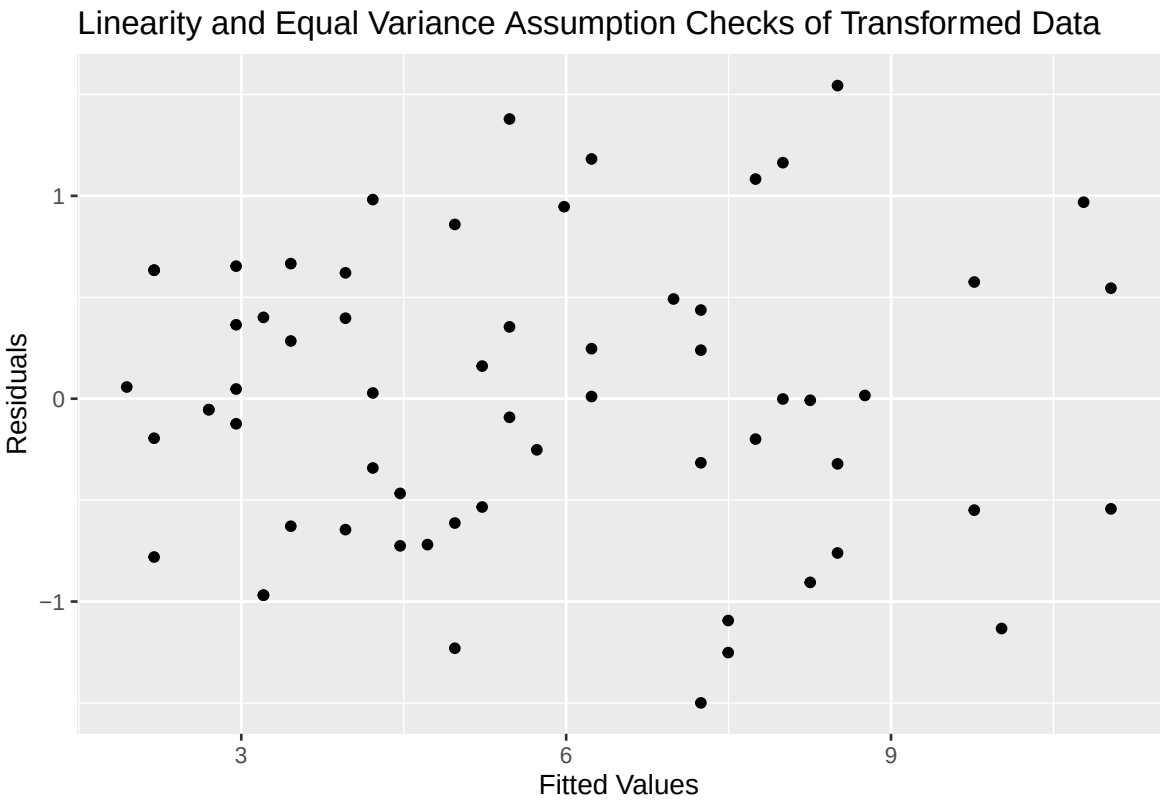
3. Write out (in mathematical form with Greek letters) a *justifiable* (perhaps after a transformation) SLR model that would help answer the questions in problem. Provide an interpretation of each mathematical term (variable or parameter) included in your model. Using the mathematical form, discuss how your model, after fitting it to the data, will be able to answer the questions in this problem. (symbolic interpretations) We will be using the model

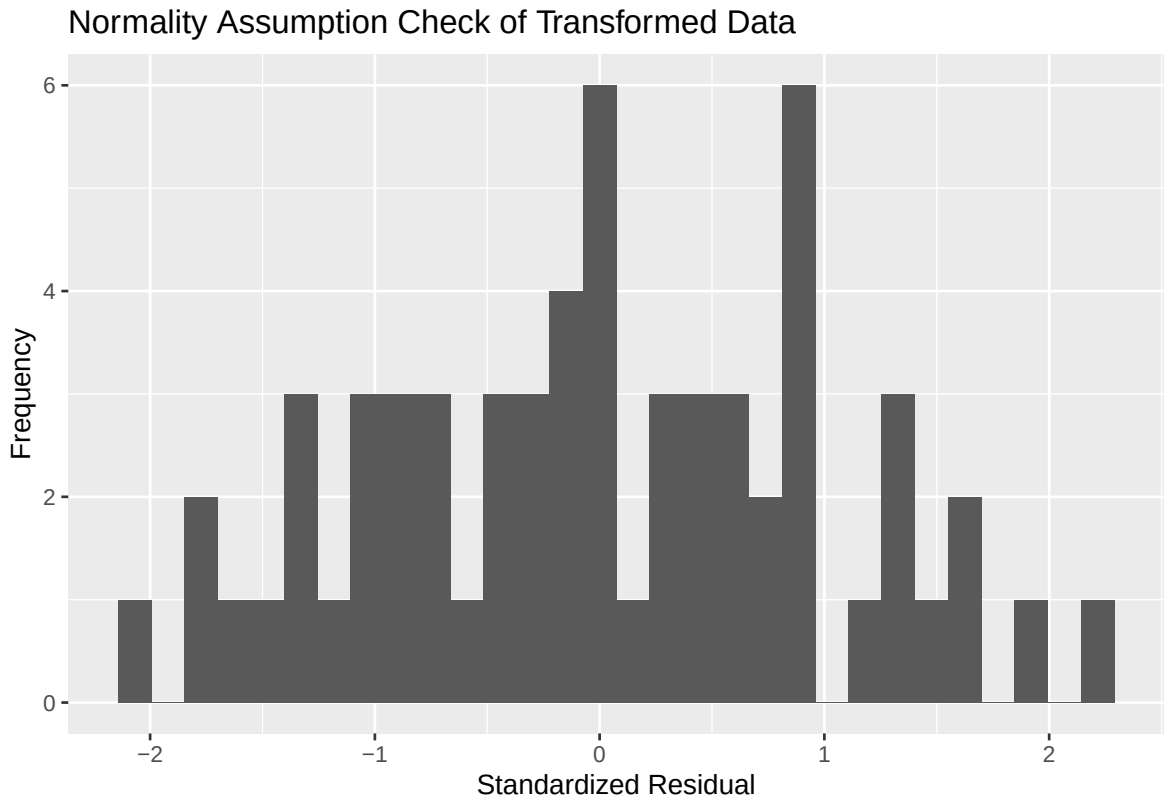
$$y_i \overset{iid}{\sim} N(\sqrt{\beta_0 + \beta_1 x_i},\ \sigma^2)$$

.

In this model, y_i represents the distance required for a car to come to a full stop, at a given observation i. It also represents the Distance the car requires to come to a full stop. The response variable of this report's data is the distance required by a car to slow down to a full stop. The x_i represents the explanatory variable at a given observation i, which we are using to explain the response variable (using statistical modeling) In this model, it represents the speed the car was traveling at. The symbol $\overset{iid}{\sim}$ means "independent and identically distributed", which means it meets two of the assumptions needed for simple linear regression. The N is short for Normal distribution, meaning the model's data follows that distribution's shape and behaviors. β_0 represents the intercept coefficient, which says when x_i is 0, the average y_i is the intercept coefficient. $beta_0$ also represents the average distance required by a car to come to a full stop, if the speed of the car was 0 mph. β_1 is the slope coefficient, which says as x_i increases, the average y_i increases by the slope coefficient. In this data's case, it means that as the speed of the car increases by 1 mph, the average distance required to come to a complete stop also increases. The symbol σ^2 represents the variance of the data around the regression line fitted to the data by this model. Another way to think of the variance is that it is the square of the standard deviation. The standard deviation shows that for any x_i, 99.7% of the response variable will be within 3 standard deviations of the regression line made by $\sqrt{(\beta_0 + \beta_1 x_i)}$, the intercept coefficient plus the product of the slope intercept and explanatory variable. I will be transforming the data set by taking the square root of the distance, as shown in the model above.

4. List, then discuss and justify the assumptions from your model in #3 using appropriate graphics or summary statistics.






```
## Warning in ks.test(standardized.residuals, "pnorm"): ties should not be present
## for the Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  standardized.residuals
## D = 0.062753, p-value = 0.9676
## alternative hypothesis: two-sided
```

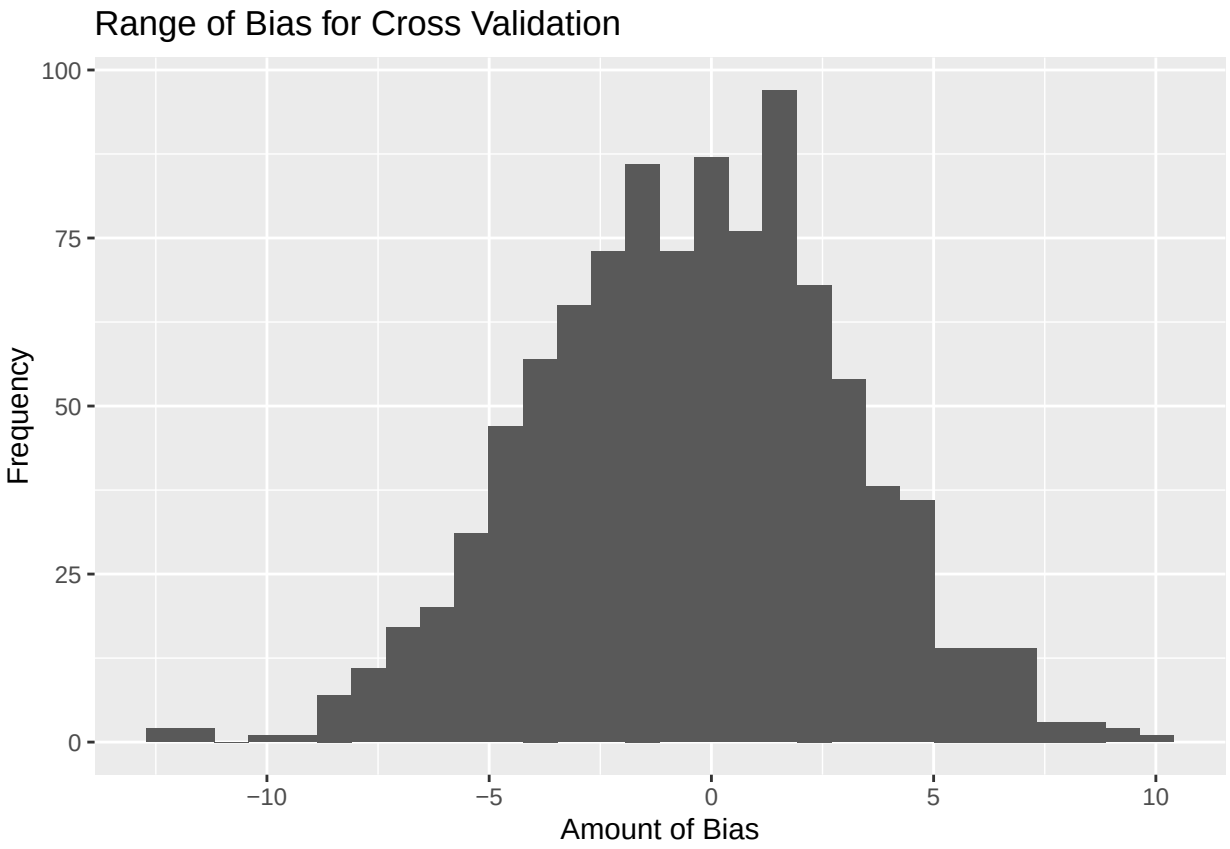
```
##
##  Jarque-Bera test for normality
##
## data:  standardized.residuals
## JB = 1.2723, p-value = 0.431
```

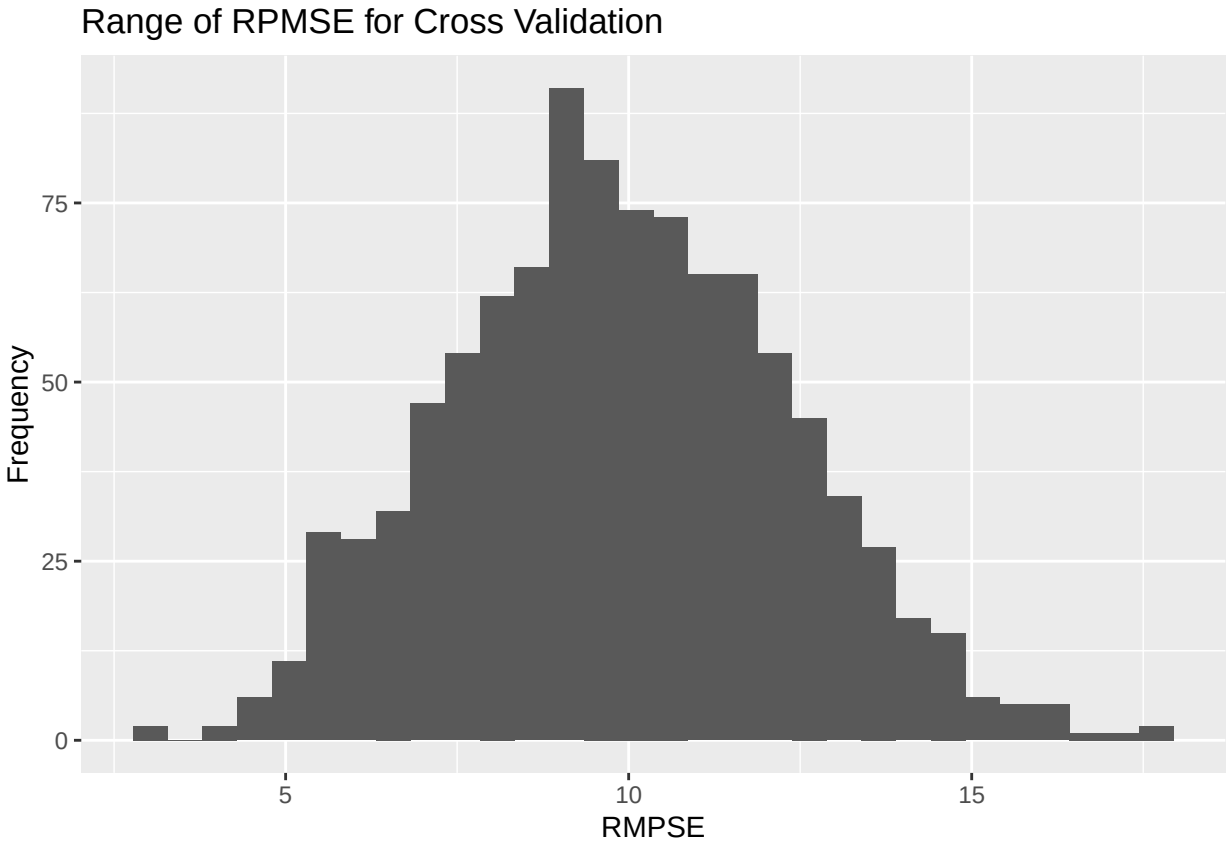
```
##
##  studentized Breusch-Pagan test
##
## data:  stopdist.trans
## BP = 3.5597, df = 1, p-value = 0.0592
```

The first assumption is linearity, which we proved is improved for creating a linear relationship. The fitted values vs. standardized residuals plot shows the transformed data shows no patterns or curves. The second assumption is independence. Because the data was collected on the same road with the same car, it is independently collected. The third assumption is for Normality. The histogram of standardized residuals

visually shows that there is a roughly Normal distribution. The KS test and JB tests also prove this mathematically. Finally, the last assumption is equal variance. The BP test and fitted values vs. standardized residuals plot show that there is an equal spread of the data.

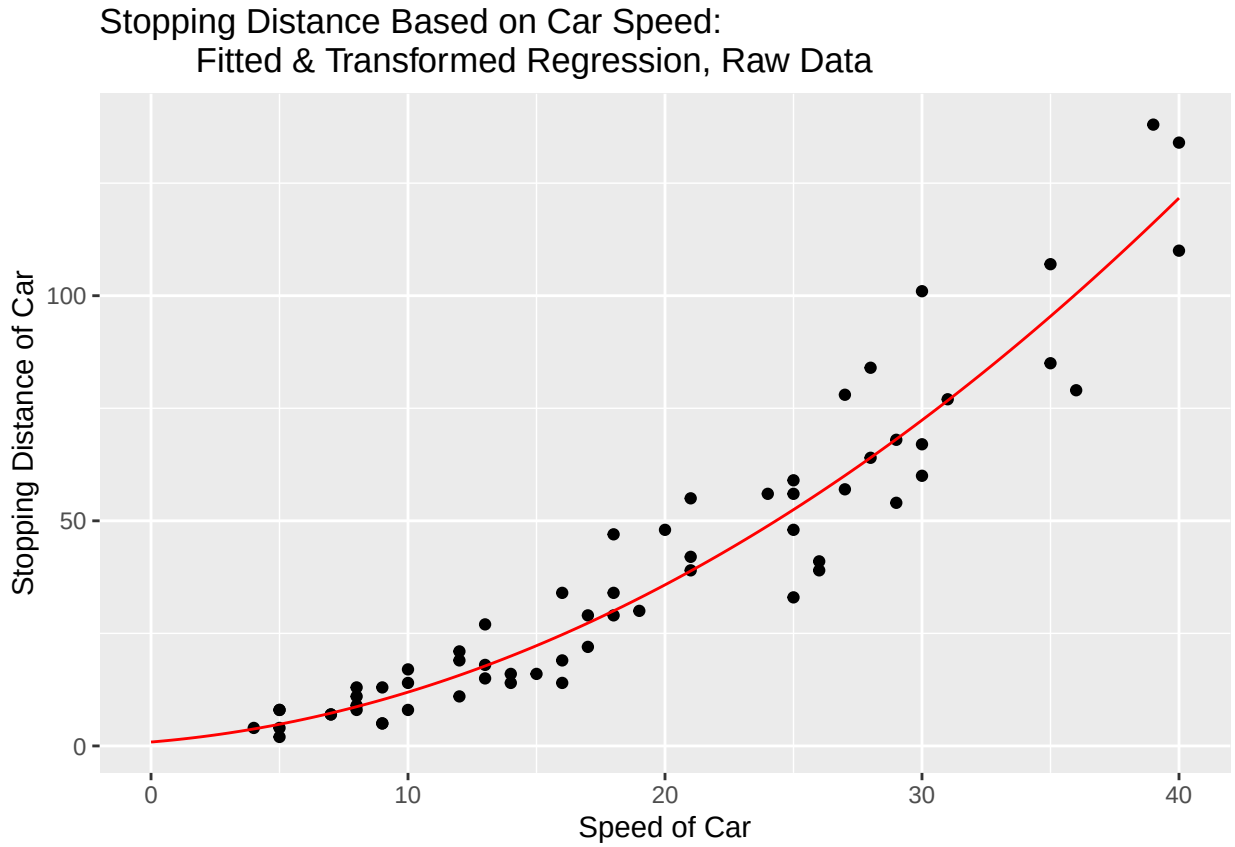
5. Assess and interpret the fit and predictive accuracy of your model on the level of your target audience. Make conclusions regarding if your predictive accuracy is "good" relative to the original spread of the response variable.





The average spread of the data is 11.7687. The Root Predictive Mean Square Error is 9.8758125, which is a close approximation. Therefore, the predictive model is a good predictive accuracy. The bias is -0.4214923.

6. Summarize the results of fitting your model in #3 by writing out the fitted model in equation form (do NOT just provide a screen shot of the R or Python output). Interpret each of the estimated parameters in the context of the problem. Provide a plot of the data with a fitted regression line **on the original scale of the data**.



In this model, y_i represents the distance required for a car to come to a full stop, at a given observation i. It also represents the Distance the car requires to come to a full stop. The response variable of this report's data is the distance required by a car to slow down to a full stop. The x_i represents the explanatory variable at a given observation i, which we are using to explain the response variable (using statistical modeling) In this model, it represents the speed the car was traveling at. The symbol $\overset{iid}{\sim}$ means "independent and identically distributed", which means it meets two of the assumptions needed for simple linear regression. The N is short for Normal distribution, meaning the model's data follows that distribution's shape and behaviors. β_0 represents the intercept coefficient, which says when x_i is 0, the average y_i is the intercept coefficient. $beta_0$ also represents the average distance required by a car to come to a full stop, if the speed of the car was 0 mph. β_1 is the slope coefficient, which says as x_i increases, the average y_i increases by the slope coefficient. In this data's case, it means that as the speed of the car increases by 1 mph, the average distance required to come to a complete stop also increases. The symbol σ^2 represents the variance of the data around the regression line fitted to the data by this model. Another way to think of the variance is that it is the square of the standard deviation. The standard deviation shows that for any x_i, 99.7% of the response variable will be within 3 standard deviations of the regression line made by $\sqrt{(\beta_0 + \beta_1 x_i)}$, the intercept coefficient plus the product of the slope intercept and explanatory variable. I will be transforming the data set by taking the square root of the distance, as shown in the model above.

7. The local law enforcement is considering implementing a speed limit of 35 MPH. Use your model to obtain a prediction of the distance required by a vehicle to stop when traveling at 35 MPH. How much of a reduction in stopping distance would be achieved by making it a 30 MPH speed limit instead? Given that the road is a *suburb* road with many homes, provide an argument for or against the use of 35 MPH.

A more reasonable speed limit for an area of medium-density pedestrian traffic would be 30 mph, due to the more reasonable average stopping distance of 72.4 feet, as opposed to an average stopping distance of 95.4

feet. Not everyone will travel at exactly 30 mph, they be one to four miles above the speed limit, so having a more minimal speed when driving will serve the community well.

Appendix of Code

```
knitr::opts_chunk$set(echo = FALSE, include = TRUE)
library(tinytex)
library(ggplot2)
library(MASS)
library(normtest)
library(lmtest)
stopdist = read.table("~/R programming/STAT_330/StoppingDistance.txt",
                      sep = ' ', header = TRUE)
#x=Speed is explanatory variable; y=Distance is response variable
stopdist.scatter <- ggplot(data = stopdist, mapping=aes(x=Speed, y=Distance)) +
  geom_point() +
  geom_smooth(se=FALSE) +
  xlab('Speed of Car') +
  ylab('Stopping Distance of Car') +
  ggtitle('Stopping Distance Based on Car Speed: Raw Data Estimates')
suppressMessages(print(stopdist.scatter))
#Calculate the correlation and covariance between Speed and Distance
stopdist.cov <- cov(stopdist$Speed, stopdist$Distance)
stopdist.cor <- cor(stopdist$Speed, stopdist$Distance)
#Provide a summary of the main features of the data
stopdist.Speed.box <- ggplot(data = stopdist, mapping=aes(Speed)) +
  geom_boxplot() +
  ggtitle("Speed of Car: Outlier Check")
stopdist.Distance.box <- ggplot(data = stopdist, mapping=aes(Distance)) +
  geom_boxplot() +
  ggtitle("Stopping Distance of Car: Outlier Check")
suppressMessages(print(stopdist.Speed.box))
suppressMessages(print(stopdist.Distance.box))
#Fit a simple linear model to the stopdist data where Distance is the response variable
#and Speed is the explanatory variable.
stopdist.regress <- lm(formula=Distance~Speed, data=stopdist)
#Identify the estimates beta sub zero, beta sub one, and sigma squared.
stopdist.beta.0 <- round(as.numeric(coef(stopdist.regress)["(Intercept)"]), digits = 4)
stopdist.beta.1 <- round(as.numeric(coef(stopdist.regress)["Speed"]), digits = 4)
stopdist.var <- round(sigma(stopdist.regress)^2, digits = 4)
stopdist.r2 <- summary(stopdist.regress)$r.squared
# Draw a fitted values vs. residuals plot to check the L and E assumption.
fitted.vs.resids.plot1 <- ggplot(stopdist.regress, aes(x=stopdist.regress$fitted.values,
                                                       y=stopdist.regress$residuals)) +
  geom_point() +
  xlab('Fitted Values') +
  ylab('Residuals') +
  ggtitle('Linearity and Equal Variance Assumption Checks of Raw Data')
suppressMessages(print(fitted.vs.resids.plot1))
std.resids <- stdres(stopdist.regress)
ks.test(std.resids, "pnorm")
#Add your estimated regression line to the scatterplot you created above.
```

```
stopdist.est.reg <- ggplot(stopdist, aes(x=Speed,y=Distance)) +
  geom_point() +
  geom_smooth(method="lm",se=FALSE) +
  xlab('Speed of Car') +
  ylab('Stopping Distance of Car') +
  ggtitle('Stopping Distance Based on Car Speed: Fitted Regression, Raw Data')
suppressMessages(print(stopdist.est.reg))
# Summary statistics of new transformation
stopdist.trans <- lm(sqrt(Distance)~Speed, data=stopdist)
trans.cov <- cov(stopdist$Speed, sqrt(stopdist$Distance))
trans.cor <- cor(stopdist$Speed, sqrt(stopdist$Distance))
trans.beta.0 <- round(as.numeric(coef(stopdist.trans)["(Intercept)"]), digits = 4)
trans.beta.1 <- round(as.numeric(coef(stopdist.trans)["Speed"]), digits = 4)
trans.var <- round(sigma(stopdist.trans)^2, digits = 4)

# Draw a fitted values vs. residuals plot to check the L and E assumption.
fitted.vs.resids.plot <- ggplot(stopdist.trans, aes(x=stopdist.trans$fitted.values,
                                                    y=stopdist.trans$residuals)) +
  geom_point() +
  xlab('Fitted Values') +
  ylab('Residuals') +
  ggtitle('Linearity and Equal Variance Assumption Checks of Transformed Data')
suppressMessages(print(fitted.vs.resids.plot))

# Draw a histogram (or density plot) of standardized residuals to check the N assumption.
standardized.residuals <- stdres(stopdist.trans)
stopdist.trans.reg <- ggplot() +
  geom_histogram(mapping=aes(x=standardized.residuals)) +
  xlab('Standardized Residual') +
  ylab('Frequency') +
  ggtitle('Normality Assumption Check of Transformed Data')
suppressMessages(print(stopdist.trans.reg))

# Conduct a KS and JB test for normality.
ks.test(standardized.residuals, "pnorm")
jb.norm.test(standardized.residuals)

# Conduct a BP test for equal variance.
bptest(stopdist.trans)

# Identify any outlying observations using Cook's distance.
cooks <- cooks.distance(stopdist.trans)
outlier.where <- 4/length(stopdist$Speed)
outliers <- stopdist[which(cooks>outlier.where),]
#outliers <- round(cooks[which(cooks>outlier.where)], 4) to get cook's distance of outliers
n.cv <- 1000 #Number of CV studies we'll run
bias <- rep(NA, n.cv) #n.cv empty biases (one for each CV)
RPMSE <- rep(NA, n.cv) #n.cv empty RPMSE (one for each CV)
n.test <- 10 #How big my test set is
for(i in 1:n.cv){
  # Choose which obs. to put in test set
  test.obs <- sample(1:nrow(stopdist), n.test)
```

```
  # Split data into test and training sets
  test.set <- stopdist[test.obs,]
  train.set <- stopdist[-test.obs,]

  # Using training data to fit a (possibly transformed) model
  train.lm <- lm(sqrt(Distance)~Speed,data=train.set)

  # Predict test set
  test.preds <- (predict.lm(train.lm, newdata = test.set))^2 #how to untransform
  #If needed, untransform here

  # Calculate bias
  bias[i] <- mean(test.preds-test.set$Distance)

  # Calculate RPMSE, this is left for you to figure out on your own
  RPMSE[i] <- sqrt(mean((test.preds - test.set$Distance)^2))
}

stopdist.CV.bias <- ggplot() +
  geom_histogram(mapping=aes(x=bias)) +
  xlab('Amount of Bias') +
  ylab('Frequency') +
  ggtitle('Range of Bias for Cross Validation')
suppressMessages(print(stopdist.CV.bias))

stopdist.CV.RPMSE <- ggplot() +
  geom_histogram(mapping=aes(x=RPMSE)) +
  xlab('RMPSE') +
  ylab('Frequency') +
  ggtitle('Range of RPMSE for Cross Validation')
suppressMessages(print(stopdist.CV.RPMSE))

mean.bias <- mean(bias)
mean.RPMSE <- mean(RPMSE)

#compare to original std dev

stopdist.stddev <- round(sqrt(sigma(stopdist.regress)^2), digits = 4)
speed.seq <- seq(0, 40, length=1000)
preds <- data.frame(Speed=speed.seq)
preds$Distance <- (predict.lm(stopdist.trans, newdata=preds))^2
#gives prediction on transformed scale

#ggplot
stopdist.est.reg <- ggplot() +
  geom_point(data=stopdist, mapping=aes(x=Speed,y=Distance)) +
  geom_line(data=preds, mapping=aes(x=Speed, y= Distance), color="red") +
  xlab('Speed of Car') +
  ylab('Stopping Distance of Car') +
  ggtitle('Stopping Distance Based on Car Speed:
          Fitted & Transformed Regression, Raw Data')
suppressMessages(print(stopdist.est.reg))
#Prediction
```

```r
stopdist.predict <- data.frame(Speed = c(30, 35))
stopdist.points <- (predict.lm(stopdist.trans, newdata=stopdist.predict))^2
#End of homework's code
```