Speed&Distance Predictive_Analysis

Akolade Sofiyyah Iwalewa,22201441

2022-11-28

Introduction to data set

The data gives the speed of cars (mph) and the distances (ft) taken to stop. The cars data set that comes with R by default

```
# Having an overview of the data set
head(cars)
```

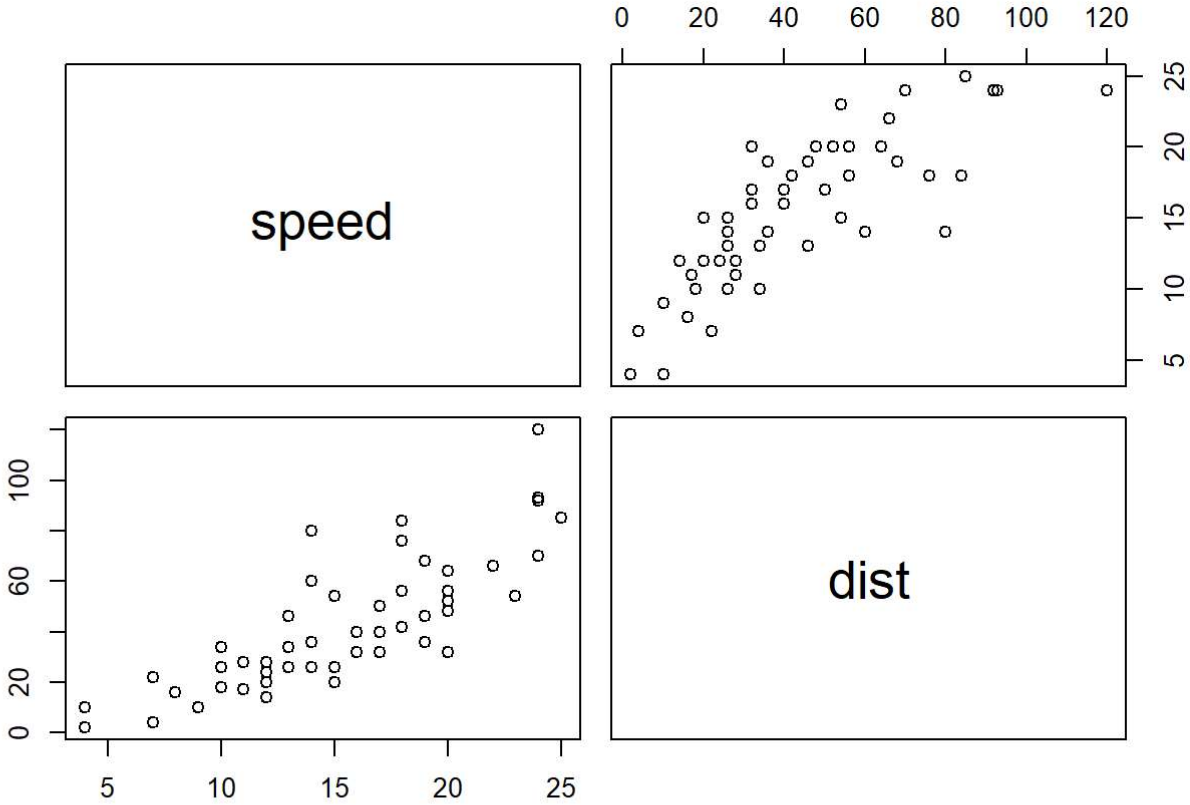
```
##   speed dist
## 1     4    2
## 2     4   10
## 3     7    4
## 4     7   22
## 5     8   16
## 6     9   10
```

```
# Checking the dimension of the data
dim(cars)
```

```
## [1] 50  2
```

This data set consists of 50 observations (rows) and 2 variables(columns) – distance and speed.

```
# creating a scatter plot to Visualise the linear relationship between the predictor and the response variable
pairs(cars)
```



There exist a positive relationship between the distance and speed.

```
# Calculating the correlation co-efficient that measures the strength of the linear relationship between the predictor and response variable.
cor(cars)
```

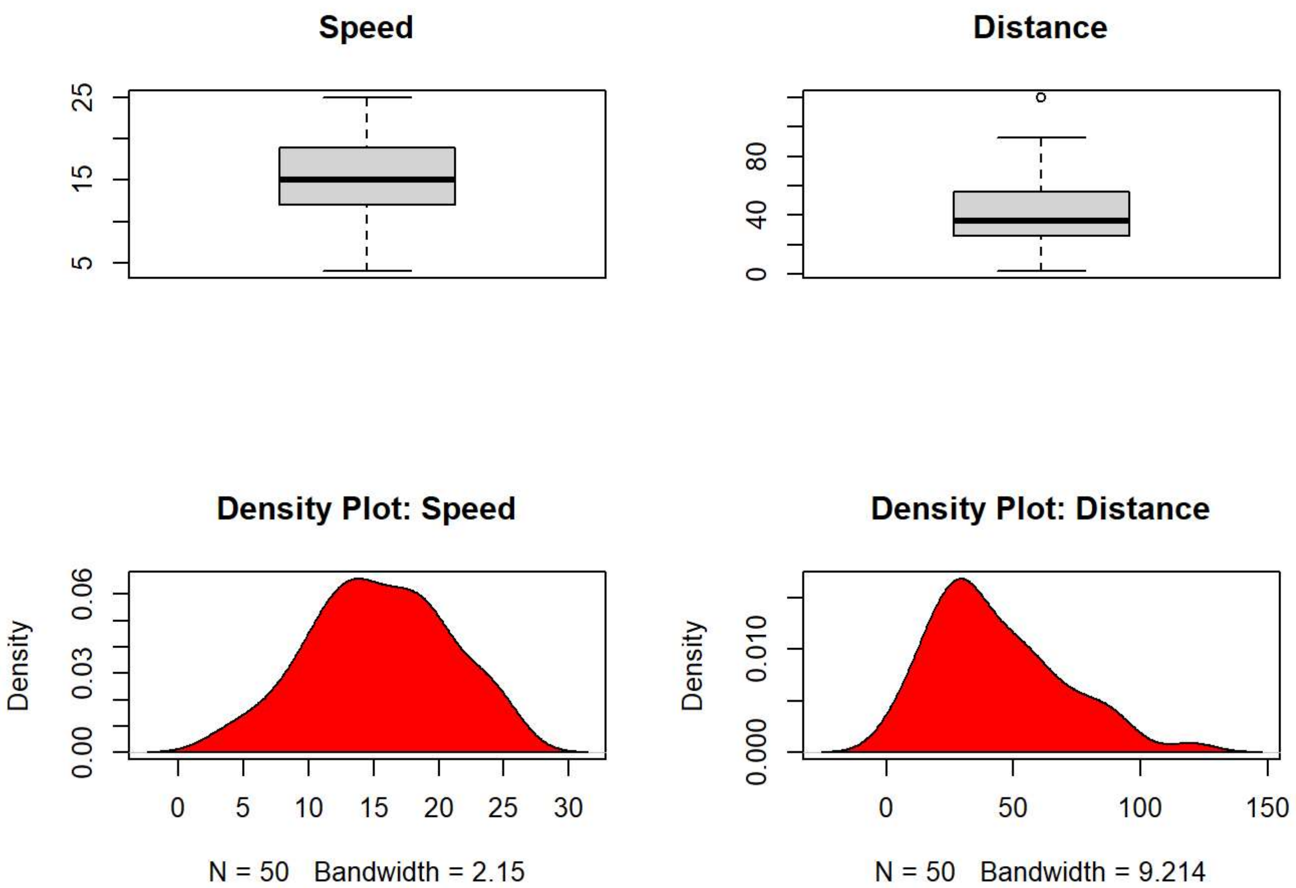
```
##           speed      dist
## speed 1.0000000 0.8068949
## dist  0.8068949 1.0000000
```

0.80 indicates a strong linear relationship between distance and speed.

```
# Getting the distribution of the variables by numerical values.
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00  
```

```
# Getting the distribution of the variables by plotting graphs.
par(mfrow=c(2, 2))
boxplot(cars$speed, main="Speed")
boxplot(cars$dist, main="Distance")
plot(density(cars$speed), main="Density Plot: Speed")
polygon(density(cars$speed), col="red")
plot(density(cars$dist), main="Density Plot: Distance")
polygon(density(cars$dist), col="red")
```



The minimum speed recorded was 4 km/h and the maximum speed is 25 km/h. 25% of the car speed fall bellow 12 km/h, 50% of the car speed fall below 15 km/h,75% of the car speed fall below 19 km/h. The minimum distance recorded was 2 km and the maximum distance was 120 km. 25% of the car distance fall below 26 km , 50% of the car distance fall below 36km, 75% of the car distance fall below 56 km. From the box plot, a particular seem to an outlier.

```
# Checking the distance that is greater 100
which(cars$dist>100)
```

```
## [1] 49
```

```
cars[49,]
```

```
##    speed dist
## 49    24  120
```

Car 49 has a distance of 120 km which is greater than 100

I will be fitting the linear model below: distancei = β0 + β1speedi + i

```
# Fitting the linear model.
SXX = sum((cars$speed-mean(cars$speed))^2)
SXY = sum((cars$speed-mean(cars$speed))
*(cars$dist-mean(cars$dist)))
beta1 <- SXY / SXX
beta0 <- mean(cars$dist) - beta1 * mean(cars$speed)
c(beta0,beta1)
```

```
## [1] -17.579095   3.932409
```

```
linearMod <- lm(dist ~speed, data=cars)
linearMod
```

```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
## Coefficients:
## (Intercept)        speed
##     -17.579        3.932
```

THe model becomes: distance ˆi = −17.58 + 3.93 × speed

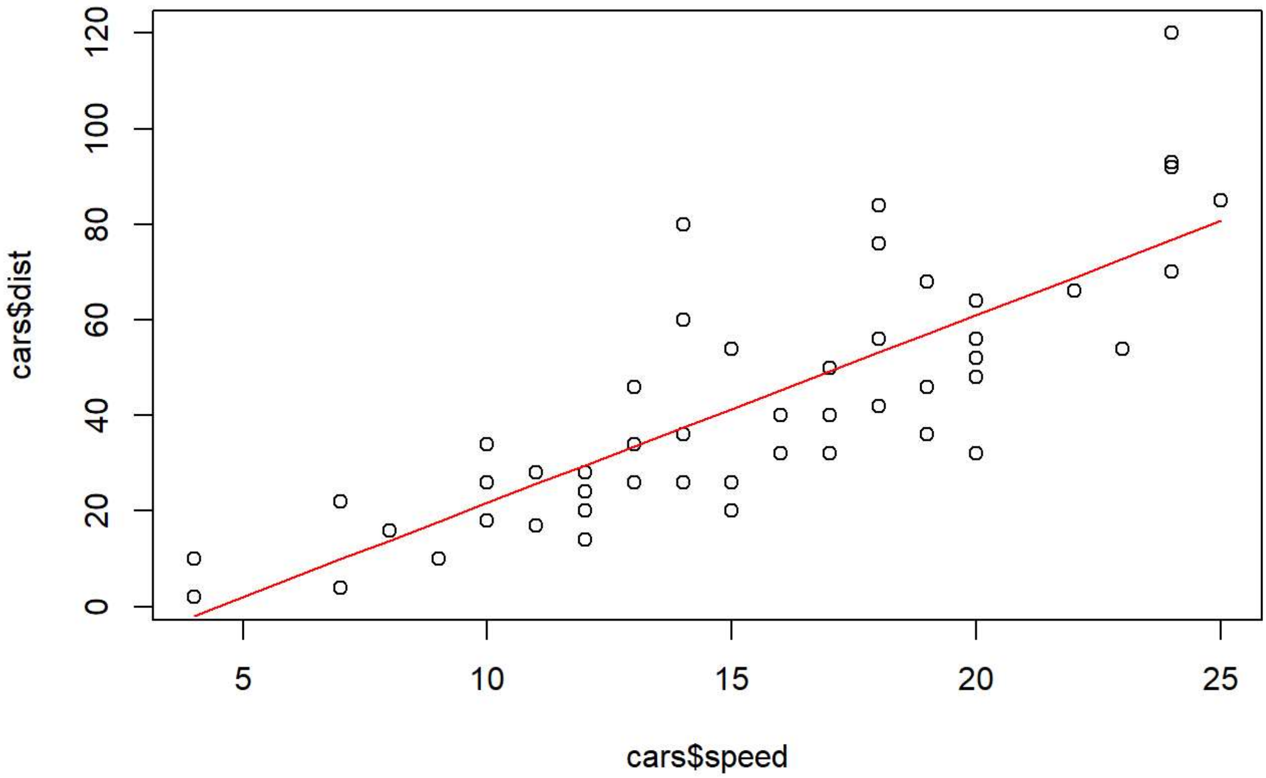
The baseline of 0 mph actually has a negative stopping distance −17.58. Thus we restrict the interpretability of the model to speeds 4.48 (≈ 17.58/3.93) or more mph. This model estimates that increasing the speed by 1 mph will result in an extra 3.93 feet of stopping distance. to aid interpretation of the intercept we can minus the mean from the predictor variable speed. This will be 0 when the speedi is at the average speed which is X¯ = 15.4

```
# Substracting the mean from the predictor variable speed
cars$cen_speed <-cars$speed-mean(cars$speed)
linearMod <- lm(dist ~cen_speed, data=cars)
linearMod
```

```
##
## Call:
## lm(formula = dist ~ cen_speed, data = cars)
##
## Coefficients:
## (Intercept)    cen_speed
##      42.980        3.932
```

Thus I have my new model to be: distance ˆi = 42.98 + 3.93 × (speedi − average(speed))

```
# Creating a line chart for speed and distance
plot(cars$speed,cars$dist)
lines(cars$speed,fitted(linearMod),col="red")
```



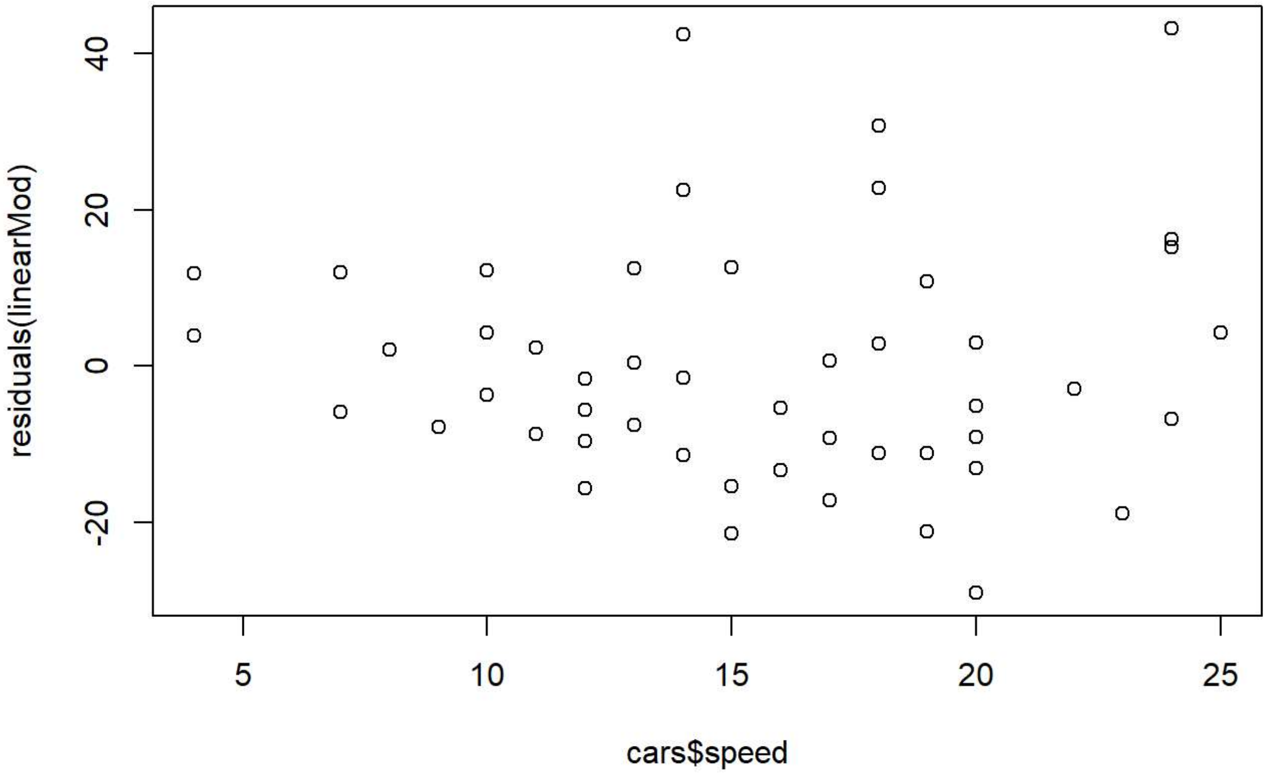
Assumptions of the LS estimators: I will be testing the LS estimators assumptions.

1. It is required there is zero conditional mean and constant variance (constant variability about the zero mean

```
# Getting the summary of the residuals
summary(residuals(linearMod))
```

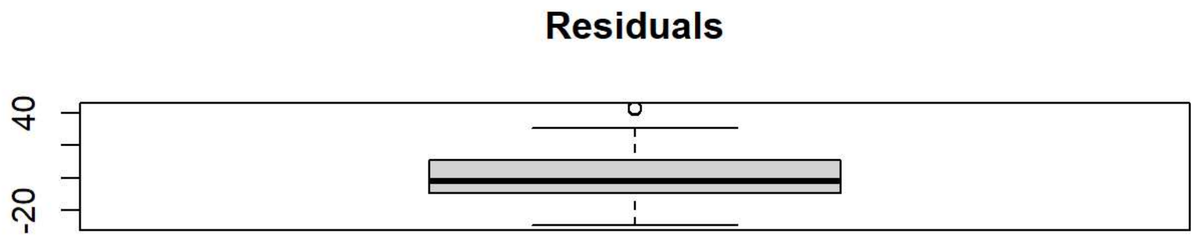
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -29.069  -9.525  -2.272   0.000   9.215  43.201
```

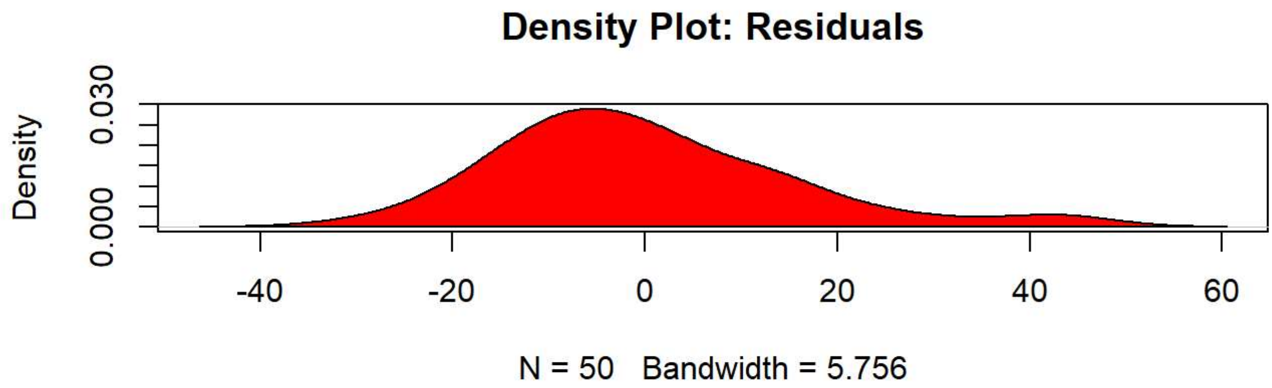
```
#Creating a scatter plot for the residuals
plot(cars$speed,residuals(linearMod))
```



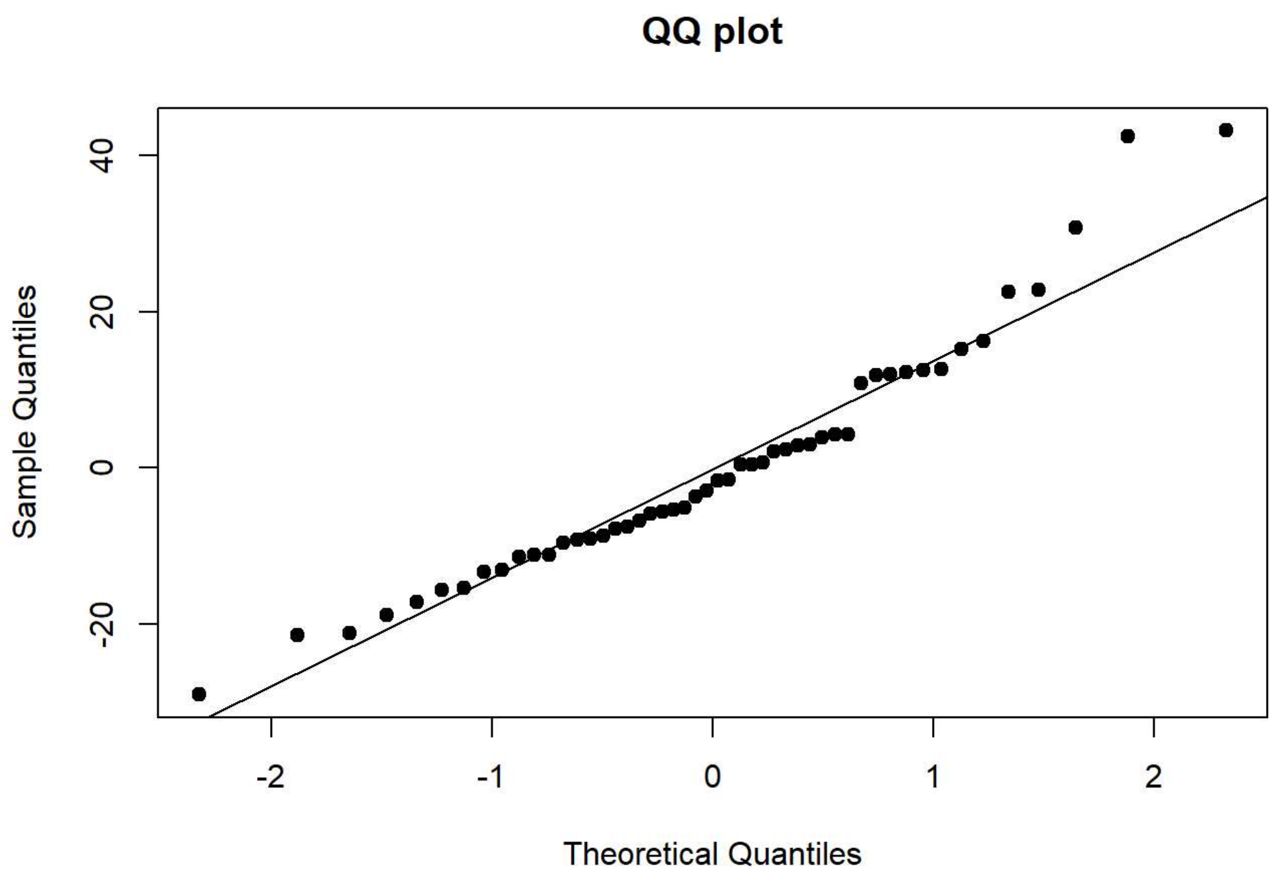
Note: To perform model inference (T-test, F-test, CI, PI) the errors must be normally distributed or at least approximately normally distributed

```
# Creating a box-plot and density plot for the residuals
par(mfrow=c(2, 1))
boxplot(residuals(linearMod), main="Residuals")
plot(density(residuals(linearMod)),
main="Density Plot: Residuals")
polygon(density(residuals(linearMod)), col="red")
```






```
# Creating a QQ plot
qqnorm(residuals(linearMod),main="QQ plot",pch=19)
qqline(residuals(linearMod))
```



I will be using the Shapiro-Wilk normality test to test for nomality.

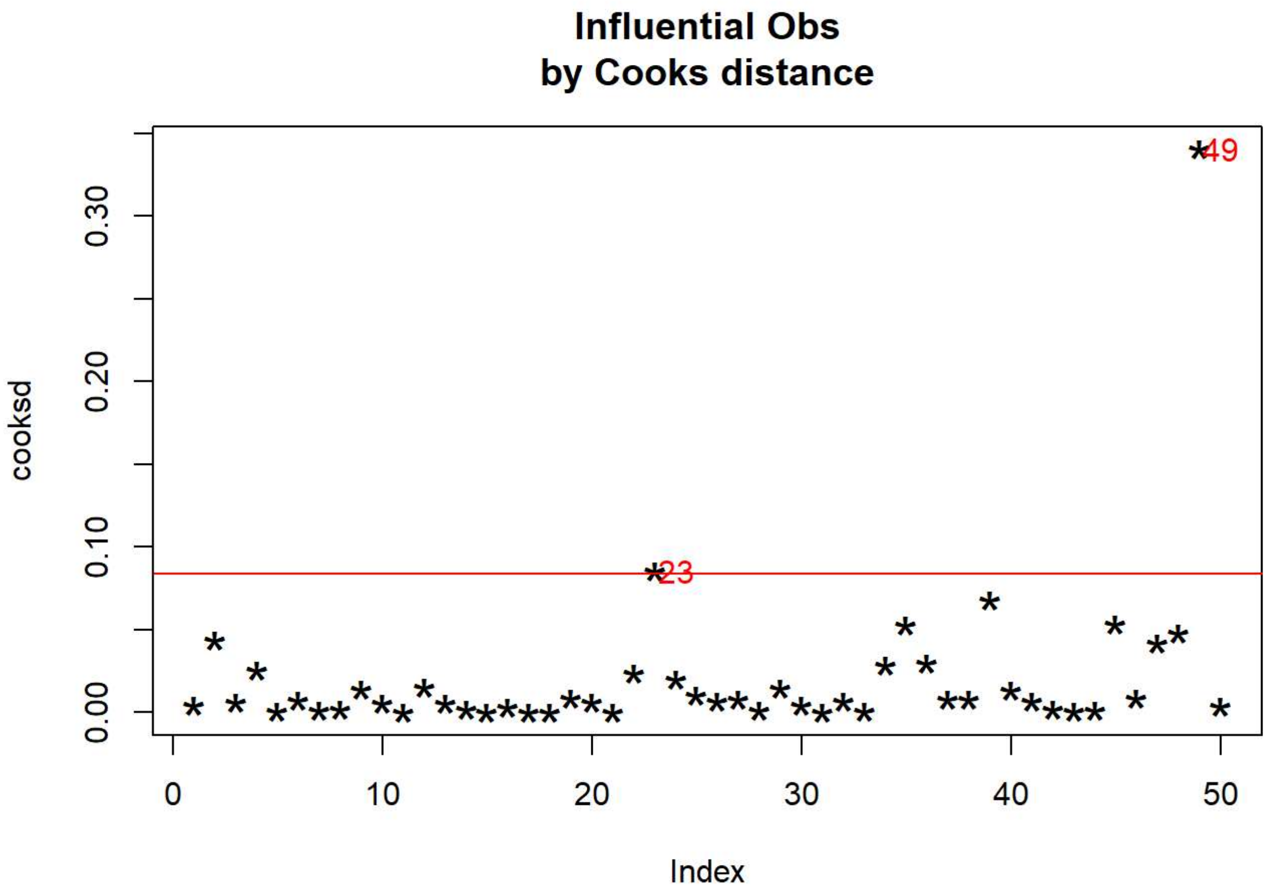
```
residuals(linearMod)
```

```
##          1          2          3          4          5          6          7 
##   3.849460  11.849460  -5.947766  12.052234   2.119825  -7.812584  -3.744993 
##          8          9         10         11         12         13         14 
##   4.255007  12.255007  -8.677401   2.322599 -15.609810  -9.609810  -5.609810 
##         15         16         17         18         19         20         21 
##  -1.609810  -7.542219   0.457781   0.457781  12.457781 -11.474628  -1.474628 
##         22         23         24         25         26         27         28 
##  22.525372  42.525372 -21.407036 -15.407036  12.592964 -13.339445  -5.339445 
##         29         30         31         32         33         34         35 
## -17.271854  -9.271854   0.728146 -11.204263   2.795737  22.795737  30.795737 
##         36         37         38         39         40         41         42 
## -21.136672 -11.136672  10.863328 -29.069080 -13.069080  -9.069080  -5.069080 
##         43         44         45         46         47         48         49 
##   2.930920  -2.933898 -18.866307  -6.798715  15.201285  16.201285  43.201285 
##         50 
##   4.268876
```

The p-value 0.02 < 0.05 implying that the distribution of the data is significantly different from a normal distribution. Hence, we cannot assume normality.

Cooks distance: I will be using the Cooks distance to compute the influence exerted by each data point on the predicted outcome.

```
cooksd <- cooks.distance(linearMod)
plot(cooksd, pch="*", cex=2, main="Influential Obs
by Cooks distance")
# add cutoff line
abline(h = 4*mean(cooksd, na.rm=T), col="red")
# add labels
text(x=1:length(cooksd)+1, y=cooksd,
labels=ifelse(cooksd>4*mean(cooksd,
na.rm=T),names(cooksd),""), col="red")
```



I will remove the Influential observation (it is likely that it is a typo) a speed of 24 has a typical stopping distance of 40 feet and a distance of 120 feet typically corresponds to a speed of 40

```
cars[49,]
```

```
##    speed dist cen_speed
## 49    24  120       8.6
```

```
cars = cars[-49,]
linearMod <- lm(dist ~cen_speed, data=cars)
linearMod
```

```
##
## Call:
## lm(formula = dist ~ cen_speed, data = cars)
##
## Coefficients:
## (Intercept)    cen_speed
##       42.05         3.64
```

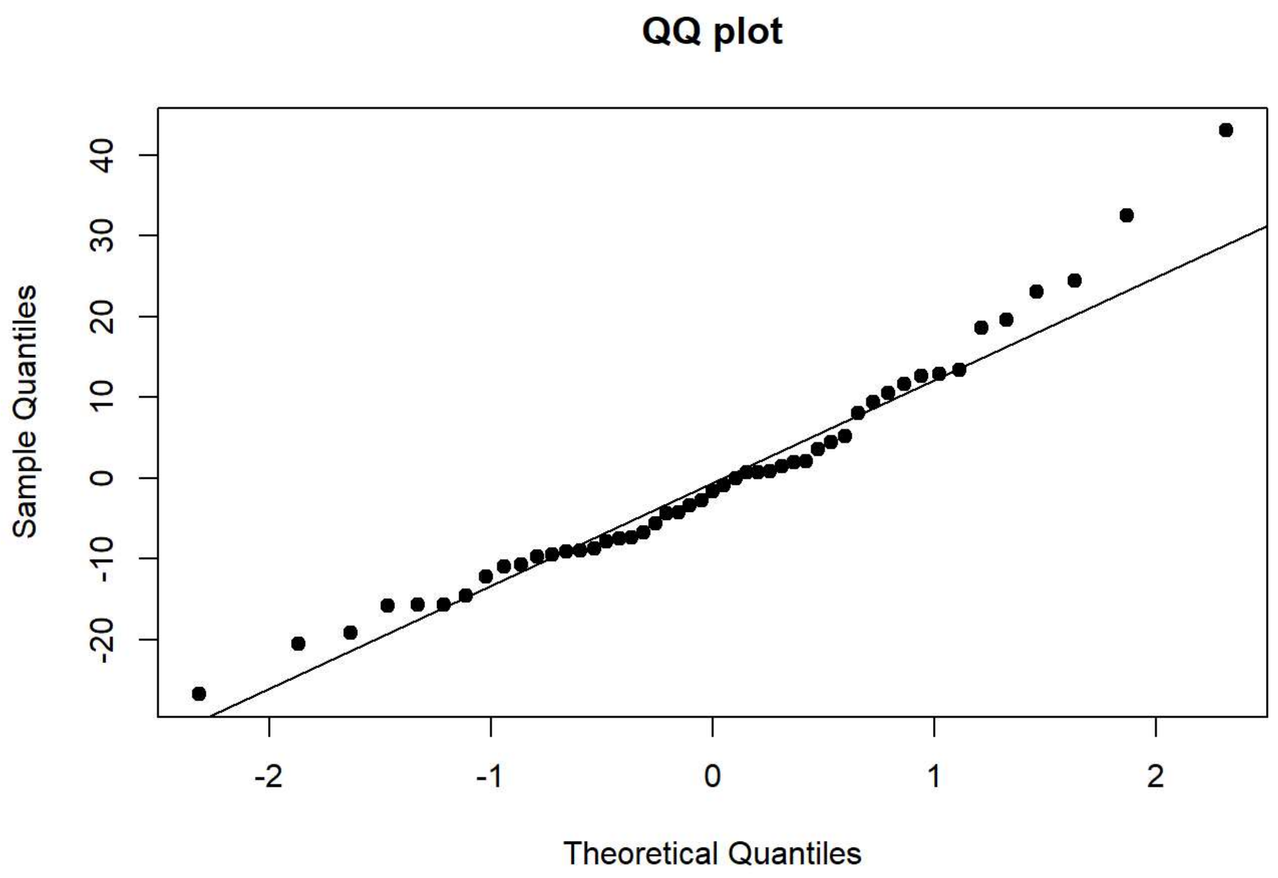
```
# Check for normaility
shapiro.test(residuals(linearMod))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(linearMod)
## W = 0.95814, p-value = 0.07941
```

```
residuals(linearMod)
```

```
##           1           2           3           4           5           6
##   1.44393997   9.44393997  -7.47471308  10.52528692   0.88573591  -8.75381511
##           7           8           9          10          11          12
##  -4.39336612   3.60663388  11.60663388  -9.03291714   1.96708286 -15.67246815
##          13          14          15          16          17          18
##  -9.67246815  -5.67246815  -1.67246815  -7.31201917   0.68798083   0.68798083
##          19          20          21          22          23          24
##  12.68798083 -10.95157019  -0.95157019  23.04842981  43.04842981 -20.59112120
##          25          26          27          28          29          30
## -14.59112120  13.40887880 -12.23067222  -4.23067222 -15.87022323  -7.87022323
##          31          32          33          34          35          36
##   2.12977677  -9.50977425   4.49022575  24.49022575  32.49022575 -19.14932526
##          37          38          39          40          41          42
##  -9.14932526  12.85067474 -26.78887628 -10.78887628  -6.78887628  -2.78887628
##          43          44          45          46          47          48
##   5.21112372  -0.06797831 -15.70752932  -3.34708034  18.65291966  19.65291966
##          50
##   8.01336865
```

```
qqnorm(residuals(linearMod),main="QQ plot",pch=19)
qqline(residuals(linearMod))
```



The p-value 0.08 > 0.05 implying that the distribution of the data is not significantly different from a normal distribution. In other words, we can assume normality.

Hypothesis testing

```
N = length(cars$cen_speed)
MSE = sum(linearMod$residuals^2/(N-2))
SXX = sum((cars$cen_speed-mean(cars$cen_speed))^2)
VARB0 = MSE*(1/ N + (mean(cars$cen_speed)^{2}/SXX))
T = (linearMod$coefficients[1]-0)/sqrt(VARB0)
```

```
alpha = 0.05
TDIST = qt(1-alpha/2, N-2)
PVALUE = 2 *( 1- pt(T, df = N- 2))
```

At the 5% level of significance, the evidence is not strong enough to indicate that β0 = 0. Indicating that when the speed is at 15.4 (the mean) the stopping distance is non-zero.

```
N = length(cars$cen_speed)
MSE = sum(linearMod$residuals^2/(N-2))
SXX = sum((cars$cen_speed-mean(cars$cen_speed))^2)
VARB1 = MSE/SXX
T = (linearMod$coefficients[2]-0)/sqrt(VARB1)
```

```
alpha = 0.05
TDIST = qt(1-alpha/2, N-2)
PVALUE = 2*(1-pt(T, df = N - 2))
```

At the 5% level of significance, the evidence is not strong enough to indicate that β1 = 0. Indicating that a relation exists between speed and stopping distance.

```
MSR = sum((fitted(linearMod) - mean(cars$dist))^2) / 1
MSE = sum(linearMod$residuals^2/(N-2))
F = MSR/MSE
alpha = 0.05
FDIST = qf(1-alpha,1,N-2)
PVALUE = pf(1-F, 1, N - 2)
```

At the 5% level of significance, the evidence is not strong enough to indicate that β1 = 0. Indicating that a relation exists between speed and stopping distance.

```
N = length(cars$cen_speed)
MSE = sum(linearMod$residuals^2/(N-2))
SXX = sum((cars$cen_speed-mean(cars$cen_speed))^2)
VARB0 = MSE*(1/ N + (mean(cars$cen_speed)^{2}/SXX))
alpha=0.05
beta0 = linearMod$coefficients[1]
c(beta0 - qt(1-alpha/2,N-2)*sqrt(VARB0),
beta0 + qt(1-alpha/2,N-2)*sqrt(VARB0))
```

```
## (Intercept) (Intercept)
##    37.99366    46.10022
```

We are 95% confident that β0 lies between 38.0 < β0 < 46.1

```
N = length(cars$cen_speed)
SSE = sum(linearMod$residuals^2)
MSE = SSE/(N-2)
SXX = sum((cars$cen_speed - mean(cars$cen_speed))^2)
VARB1 = MSE/SXX
beta1= linearMod$coefficients[2]
alpha=0.05
c(beta1 - qt(1-alpha/2,N-2)*sqrt( VARB1),
beta1 + qt(1-alpha/2,N-2)*sqrt( VARB1))
```

```
## cen_speed cen_speed
##  2.851426  4.427676
```

We are 95% confident that β1 representing the average increase in stopping distance given a one unit increase in speed is between 2.85 and 4.43 feet.

```
SST = sum((cars$dist-mean(cars$dist))^2)
SSE = sum(linearMod$residuals^2)
R2 <- (SST - SSE) /SST
R2
```

```
## [1] 0.6474321
```

Approximately 64% of the observed variation in stopping distances can be explained by the cars speed.

```
N = length(cars$cen_speed)
RMSE = sqrt(SSE/(N-2))
RMSE
```

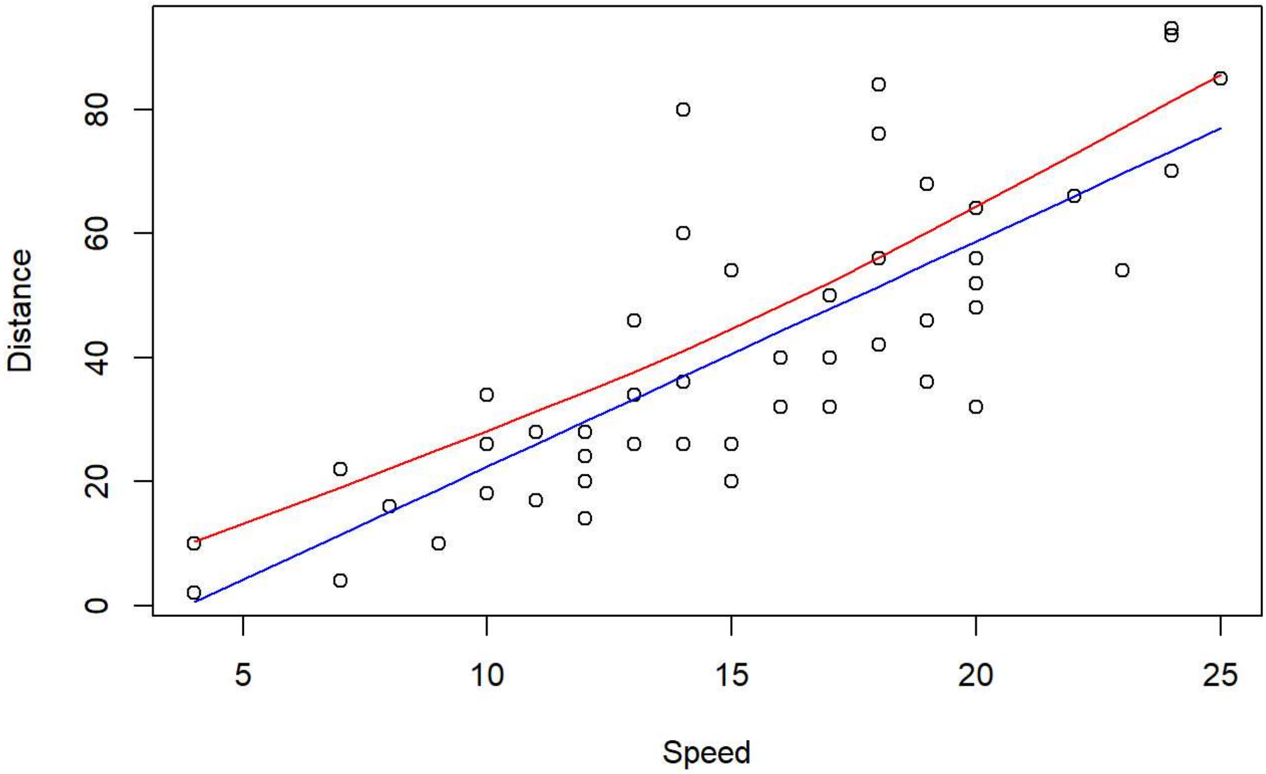
```
## [1] 14.09546
```

So we can say that the speed accurately predicts distance with about 14 feet error on average.

```
N = length(cars$cen_speed)
SXX = sum((cars$cen_speed - mean(cars$cen_speed))^2)
MSE = SSE/(N-2)
VAR_Y = MSE*(1/N+(cars$cen_speed-mean(cars$cen_speed))^2/SXX)
Yhat = fitted(linearMod)
cbind(Yhat- qt(1-alpha/2,N-2)*sqrt(VAR_Y),
Yhat + qt(1-alpha/2,N-2)*sqrt(VAR_Y))
```

```
##         [,1]     [,2]
## 1  -9.173634 10.28575
## 2  -9.173634 10.28575
## 3   3.831070 19.11836
## 4   3.831070 19.11836
## 5   8.126469 22.10206
## 6  12.391774 25.11586
## 7  16.617196 28.16954
## 8  16.617196 28.16954
## 9  16.617196 28.16954
## 10 20.789350 31.27648
## 11 20.789350 31.27648
## 12 24.890405 34.45453
## 13 24.890405 34.45453
## 14 24.890405 34.45453
## 15 24.890405 34.45453
## 16 28.898002 37.72604
## 17 28.898002 37.72604
## 18 28.898002 37.72604
## 19 28.898002 37.72604
## 20 32.787289 41.11585
## 21 32.787289 41.11585
## 22 32.787289 41.11585
## 23 32.787289 41.11585
## 24 36.536344 44.64590
## 25 36.536344 44.64590
## 26 36.536344 44.64590
## 27 40.133907 48.32744
## 28 40.133907 48.32744
## 29 43.584430 52.15602
## 30 43.584430 52.15602
## 31 43.584430 52.15602
## 32 46.905988 56.11356
## 33 46.905988 56.11356
## 34 46.905988 56.11356
## 35 46.905988 56.11356
## 36 50.122999 60.17565
## 37 50.122999 60.17565
## 38 50.122999 60.17565
## 39 53.259378 64.31837
## 40 53.259378 64.31837
## 41 53.259378 64.31837
## 42 53.259378 64.31837
## 43 53.259378 64.31837
## 44 59.365369 72.77059
## 45 62.361564 77.05349
## 46 65.331862 81.36230
## 47 65.331862 81.36230
## 48 65.331862 81.36230
## 50 68.282235 85.69103
```

```
plot(cars$speed,cars$dist,xlab="Speed",ylab="Distance")
lines(cars$speed,Yhat,col="blue")
lines(cars$speed,Yhat+qt(1-alpha/2,N-2)*sqrt(VAR_Y),col="red")
```




```
N = length(cars$cen_speed)
SXX = sum((cars$cen_speed - mean(cars$cen_speed))^2)
MSE = SSE/(N-2)
Var_E = MSE*(1 + 1/N + (cars$cen_speed-mean(cars$cen_speed))^2/SXX)
Yhat = fitted(linearMod)
cbind(Yhat- qt(1-alpha/2,N-2)*sqrt( Var_E),
Yhat + qt(1-alpha/2,N-2)*sqrt( Var_E))
```

```
##           [,1]      [,2]
## 1  -29.4231484  30.53527
## 2  -29.4231484  30.53527
## 3  -17.8938290  40.84326
## 4  -17.8938290  40.84326
## 5  -14.0904491  44.31898
## 6  -10.3075262  47.81516
## 7   -6.5453645  51.33210
## 8   -6.5453645  51.33210
## 9   -6.5453645  51.33210
## 10  -2.8042287  54.87006
## 11  -2.8042287  54.87006
## 12   0.9156583  58.42928
## 13   0.9156583  58.42928
## 14   0.9156583  58.42928
## 15   0.9156583  58.42928
## 16   4.6141180  62.00992
## 17   4.6141180  62.00992
## 18   4.6141180  62.00992
## 19   4.6141180  62.00992
## 20   8.2910184  65.61212
## 21   8.2910184  65.61212
## 22   8.2910184  65.61212
## 23   8.2910184  65.61212
## 24  11.9462751  69.23597
## 25  11.9462751  69.23597
## 26  11.9462751  69.23597
## 27  15.5798525  72.88149
## 28  15.5798525  72.88149
## 29  19.1917642  76.54868
## 30  19.1917642  76.54868
## 31  19.1917642  76.54868
## 32  22.7820727  80.23748
## 33  22.7820727  80.23748
## 34  22.7820727  80.23748
## 35  22.7820727  80.23748
## 36  26.3508888  83.94776
## 37  26.3508888  83.94776
## 38  26.3508888  83.94776
## 39  29.8983703  87.67938
## 40  29.8983703  87.67938
## 41  29.8983703  87.67938
## 42  29.8983703  87.67938
## 43  29.8983703  87.67938
## 44  36.9301857  95.20577
## 45  40.4150526  99.00001
## 46  43.8796456 102.81452
## 47  43.8796456 102.81452
## 48  43.8796456 102.81452
## 50  47.3243235 106.64894
```

```
plot(cars$speed,cars$dist,xlab="Speed",ylab="Distance")
lines(cars$speed,Yhat,col="blue")
lines(cars$speed,Yhat +qt(1-alpha/2,N-2)*sqrt(Var_E),col="red")
lines(cars$speed,Yhat -qt(1-alpha/2,N-2)*sqrt(Var_E),col="red")
lines(cars$speed,Yhat-qt(1-alpha/2,N)*sqrt(VAR_Y),col="red")
```