HW 1 - SS 9159

Ravin Lathigra - 250620601

September 17, 2018

import data

```
## Parsed with column specification:
## cols(
##   x1 = col_double(),
##   x2 = col_character()
## )
```

HW1 Data

x1	x2
2.8284356	L
2.8649852	H
8.1586113	H
4.9411552	L
9.9168868	H

Question 1

A - x1 observations greater than 6

```
one_a <- hw1_data %>%
  filter(x1>6) %>%
  dplyr::summarise('Number of Observations where x1 > 6' = n())
```

```
## Warning: package 'bindrcpp' was built under R version 3.3.3
```

```
kable(one_a, caption = "Question 1a", align = rep("c", ncol(one_a))) %>%
  kable_styling()
```

Question 1a

Number of Observations where x1 > 6

26

The table above illustrates that within the data, there are 26 observations where x1 >6.

B - x1 observations greater than 6 and x2 = H

```
one_b <- hw1_data %>%
  filter(x1>6) %>%
  filter(x2=="H") %>%
  dplyr::summarise('Number of observations such that x1 > 6 and X2 = H'= n())


kable(one_b,
      caption = "Question 1b",
      align = rep("c", ncol(one_b))) %>%
  kable_styling()
```

Question 1b

Number of observations such that x1 > 6 and X2 = H
23

The table above illustrates that within the data, there are 23 observations where x1 >6 and x2 = H.

C - Summary statistics of x1 conditioned on x2 = H

```
A <- hw1_data %>%
  filter(x2=="H") %>%
  dplyr::summarise('Mean' = round(mean(x1),2), 'Median' = round(median(x1),2), 'Std. Deviation
' = round(sd(x1),2))

kable(A,
      caption = "Summary Statistics for x1 given x2 = H",
      align = rep("c", ncol(A))) %>%
  kable_styling()
```

Summary Statistics for x1 given x2 = H

Mean	Median	Std. Deviation
5.83	5.68	1.79

D - t-Test of the true mean of x1

$$H_0 : \mu = 4$$

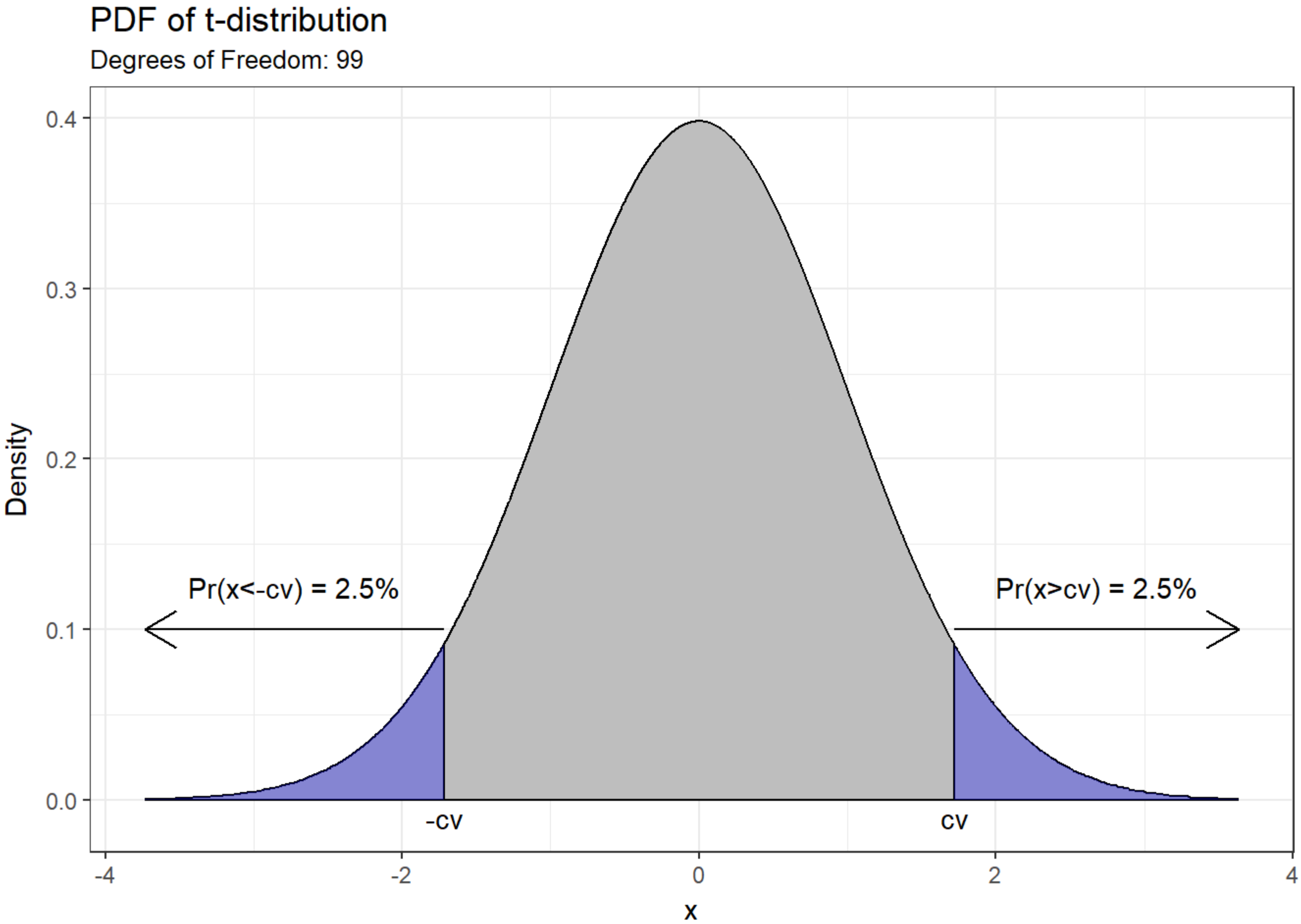
$$H_a : \mu \neq 4$$

α = 0.05

```
x1 <- hw1_data$x1

x1_t_test <- t.test(x = x1,                                     #Perform two sided t-test.
                    alternative = "two.sided",
                    mu = 4,
                    conf.level = 0.95)

cv <- x1_t_test$statistic                                       #Store critical value from t-t
est.

x <- rt(10000, 99)                                              #generate samples from a t-dis
tribution for visualization

y <- data.frame(t= x, t_density = dt(x, 99))


y<- y %>%                                                       #Add identifier to data to ide
ntify if observations are more extreme than critical values
  mutate(cutoffs = ifelse(abs(t)>cv,"True","False"))

y_false <- y%>%
  filter(abs(t)<cv)
y_true_less<- y %>%
  filter(-t>=cv)
y_true_greater<- y %>%
  filter(t>=cv)

ggplot() +                                                      #Plot t-distributions
  geom_ribbon(data = y,aes(x = t,
                  y = t_density, ymin = 0,
                  ymax = t_density),
                  fill = "grey",
                  show.legend = FALSE) +
  geom_line(data = y,aes(x = t,
                         y = t_density)) +
  geom_segment(data = y,aes(x =cv,                              #show critical value on plot
                            y = 0,
                            xend = cv,
                            yend = dt(-cv,99)))  +
  geom_segment(data = y,aes(x =-cv,                             #show critical values on plot
                            y = 0,
                            xend = -cv,
                            yend = dt(cv,99))) +
  geom_segment(data = y,aes(x = min(x),                         #frame the plot with horizonta
l line
                            y = 0,
                            xend = max(x),
                            yend = 0 )) +

  geom_segment(data = y,aes(x = min(x),                         #Add arrow to show area less t
han -cv
```

```
                             y = .1,
                             xend = -cv,
                             yend = .1 ),
                arrow = arrow(length = unit(0.5, "cm"),
                              ends= "first")) +
  geom_segment(data = y,aes(x = cv,                            #Add arrow to show area less t
han -cv
                             y = .1,
                             xend = max(x),
                             yend = .1 ),
                arrow = arrow(length = unit(0.5, "cm"),
                             ends= "last")) +


  geom_area(data = y_true_less,                                #Fill in area under the t-dist
ribution less than critical value
            aes(x=y_true_less$t,
                y=y_true_less$t_density),
            fill="blue", alpha= .3) +
  geom_area(data = y_true_greater,                             #Fill in area under the t-dist
ribution greater than critical value
            aes(x=y_true_greater$t,
                y=y_true_greater$t_density),
            fill="blue", alpha= .3) +

  xlab("x") +
  ylab("Density") +
  ggtitle("PDF of t-distribution") +
  labs(subtitle = "Degrees of Freedom: 99") +

  annotate("text", x =(min(x)+-cv)/2, y = .125, label = "Pr(x<-cv) = 2.5%") +
  annotate("text", x =(cv+max(x))/2, y = .125, label = "Pr(x>cv) = 2.5%") +
  annotate("text", x = cv, y = -0.01, label = "cv") +
  annotate("text", x = -cv, y = -0.01, label = "-cv") +
  theme_bw()
```

```
## Warning: Ignoring unknown aesthetics: y
```




```
print(x1_t_test)
```

```
##
##  One Sample t-test
##
## data:  x1
## t = 1.7192, df = 99, p-value = 0.08871
## alternative hypothesis: true mean is not equal to 4
## 95 percent confidence interval:
##  3.932958 4.936674
## sample estimates:
## mean of x
##  4.434816
```

Using a two-sided t test, the p-value was larger than the significance level which indicates that there is no significant evidence against the Null hypothesis, therefore we fail to reject the Null.

E - T-Test of the true mean of x1 where x2 = H

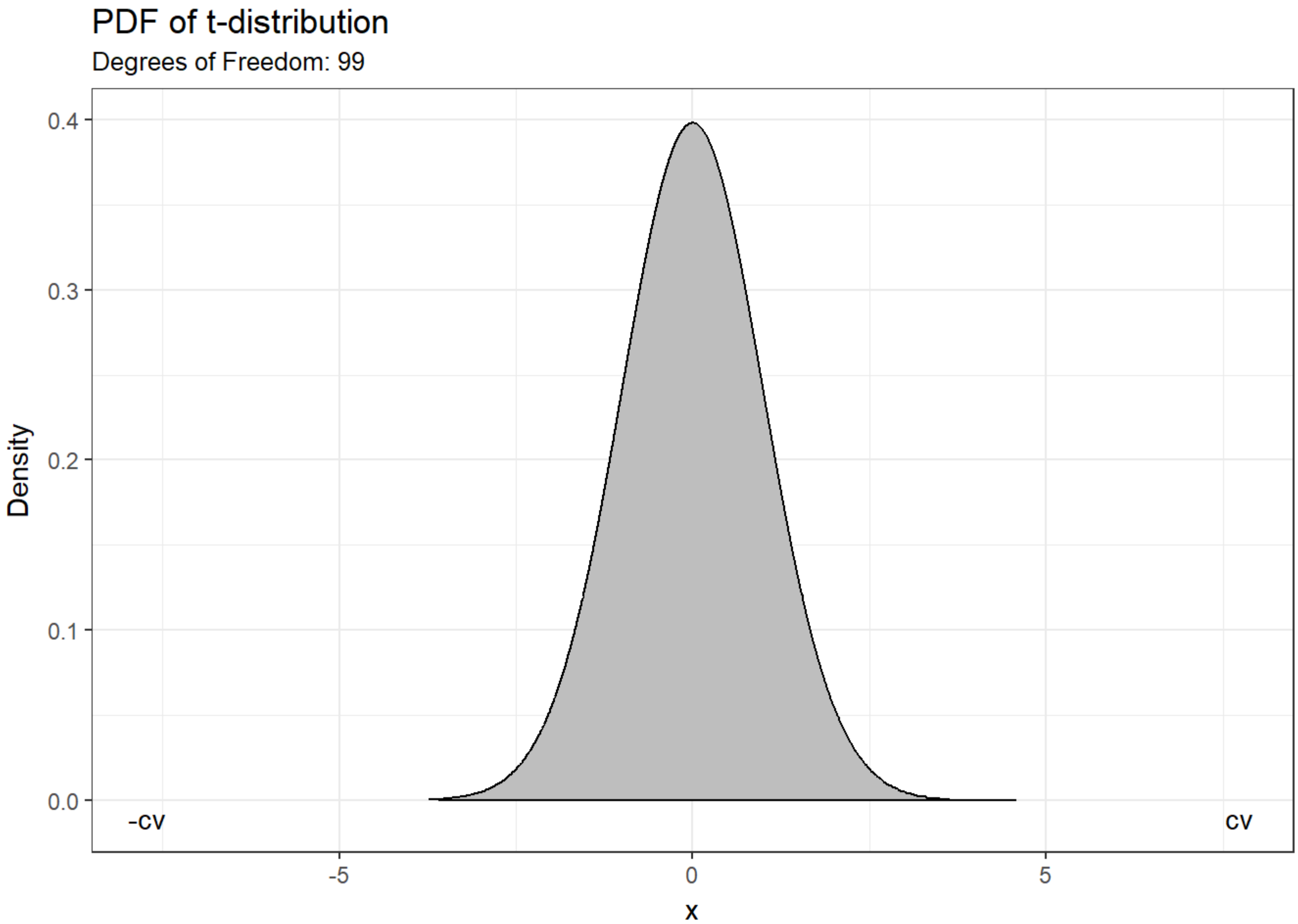
$$H_0 : \mu > 4$$

$$H_a : \mu \leq 4$$

$$\alpha = 0.05$$

```
one_e <- data.frame(hw1_data %>%                              #Gather data needed for t-test
.
  filter(x2 == "H") %>%
  select(x1))


  x1_t_test <- t.test(x = one_e,                              #Perform t-test
                      alternative = "less",
                      mu = 4,
                      conf.level = 0.95)

cv <- x1_t_test$statistic

x <- rt(10000, 99)

y<- y %>%                                                      #Add identifier to data to sh
ow when values exceeed the critical values
  mutate(cutoffs = ifelse(abs(t)>cv,"True","False"))

y_false <- y%>%
  filter(abs(t)<cv)
y_true_less<- y %>%
  filter(-t>=cv)
y_true_greater<- y %>%
  filter(t>=cv)

ggplot() +                                                     #plot t-distribution and add
labels
  geom_ribbon(data = y,aes(x = t,
                y = t_density, ymin = 0,
                ymax = t_density),
                fill = "grey",
                show.legend = FALSE) +
  geom_line(data = y,aes(x = t, y = t_density)) +
  geom_segment(data = y,aes(x =cv, y = 0, xend = cv, yend = dt(-cv,99)))  +
  geom_segment(data = y,aes(x =-cv, y = 0, xend = -cv, yend = dt(cv,99))) +
  geom_segment(data = y,aes(x = min(x), y = 0, xend = max(x), yend = 0 )) +

  geom_area(data = y_true_less,aes(x=y_true_less$t, y=y_true_less$t_density), fill="blue", alp
ha= .3) +
  geom_area(data = y_true_greater,aes(x=y_true_greater$t, y=y_true_greater$t_density), fill="b
lue", alpha= .3) +

  xlab("x") +
  ylab("Density") +
  ggtitle("PDF of t-distribution") +
  labs(subtitle = "Degrees of Freedom: 99") +

  annotate("text", x = cv, y = -0.01, label = "cv") +
  annotate("text", x = -cv, y = -0.01, label = "-cv") +
  theme_bw()
```

```
## Warning: Ignoring unknown aesthetics: y
```




```
print(x1_t_test)
```

```
##
##  One Sample t-test
##
## data:  one_e
## t = 7.7278, df = 56, p-value = 1
## alternative hypothesis: true mean is less than 4
## 95 percent confidence interval:
##      -Inf 6.229616
## sample estimates:
## mean of x
##  5.832919
```

Considering a one-sided t-test, the p-value exceeds the significance level of 5% which suggest that we fail to reject the Null hypothesis that the mean of x1 given x2 = H is greater than 4.

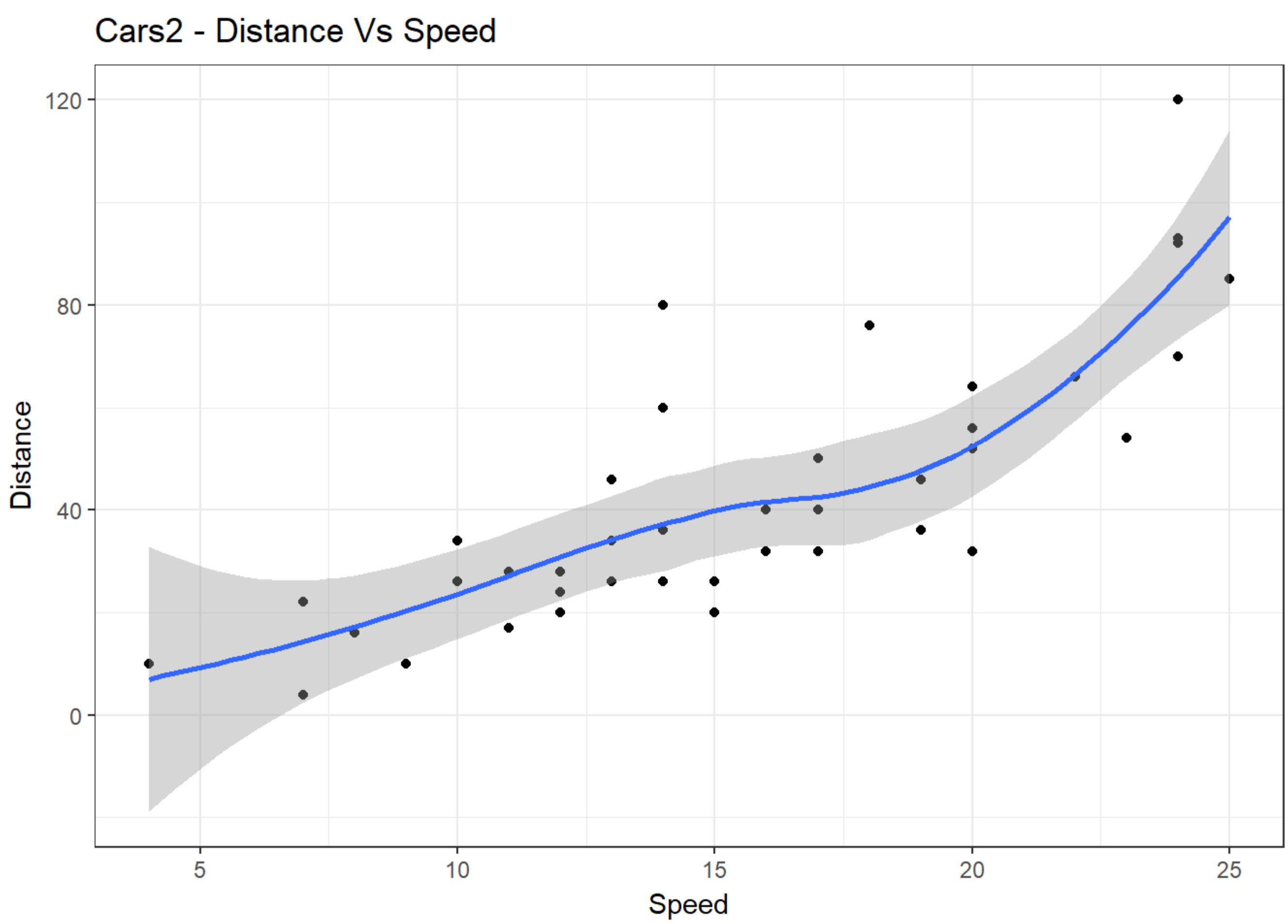
Question 2

```
set.seed(50)
idx <- sample(nrow(cars), 40, replace = FALSE)
cars2 <- cars[idx,]
```

A - Relationship between Speed and Distance

```
spd_dist_plot <- ggplot(cars2)+
                    geom_point(aes(x = speed, y = dist)) +      #Scatterplot of Cars Data
                    geom_smooth(aes(x = speed, y = dist)) +     #Apply 95% C.I using lm
                    ggtitle("Cars2 - Distance Vs Speed") +
                    xlab("Speed") +
                    ylab("Distance") +
                    theme_bw()

print(spd_dist_plot)
```

```
## `geom_smooth()` using method = 'loess'
```



From the scatter plot “Cars2 - Distance Vs Speed” we can see that there is a positive correlation between speed and distance. Superimposing a smooth plot onto the existing plot with a 95% confidence interval aids in ilustrating there is a relationship between speed and distance and that modelling this relationship with a linear model may be appropriate.

B - Least Squares Estimates

```
two_b_lm <- lm(dist ~ speed, data= cars2)                #Create Linear Model

ls_sigma <- summary(two_b_lm)$sigma^2                    #Extract LS Estimate for Sigma
ls_betas <- summary(two_b_lm)$coefficients[,1]           #Extract LS Estimate for Beta0, Beta1
```

```
beta <- intToUtf8(946)                                   #Convert Unicode for greek beta into s
ymbol
sigma_sq <- intToUtf8(963)                               #Convert Unicode for greek sigma into
symbol

ls_estimates <- data.frame(                              #Create Data.Frame for LS estimates
                  cbind(ls_betas[1],
                        ls_betas[2],
                        ls_sigma)
                        )

colnames(ls_estimates) = c(paste0(beta,c(0:1)),         #Apply greek symbols to column Headers
                           paste0(sigma_sq,"^2"))
rownames(ls_estimates) = "Estimates"                     #Rename row

kable(ls_estimates,                                      #Create table smmarizing LS esimates f
or output
      caption = "Least Squares Estimates: Distance ~ Speed",
      align = rep("c", ncol(ls_estimates))) %>%
  kable_styling()
```

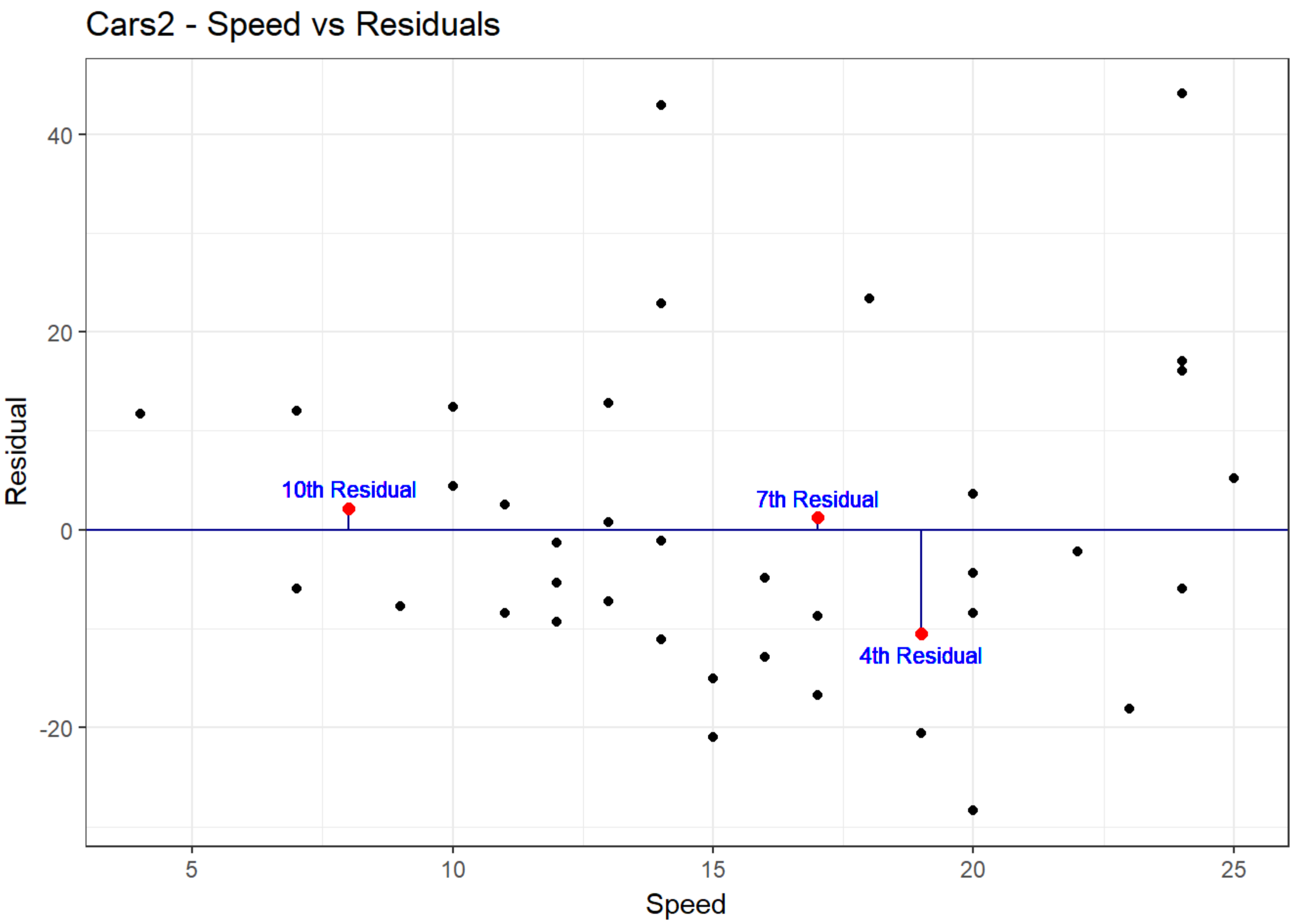
Least Squares Estimates: Distance ~ Speed

	ß0	ß1	s^2
Estimates	-17.23691	3.881985	251.0771

C - Calculating 4th, 7th and 10th residuals

```
resid <-two_b_lm$residuals                               #Extract residuals from linear model


cars2 <- cars2 %>%                                       #Append Residuals to Cars2 data
  mutate(residual = resid)


ggplot(cars2) +
  geom_point(aes(x = cars2$speed, two_b_lm$residuals))+ #Plot Speed vs residuals

  geom_segment(aes(x = cars2$speed[4],                   #Draw line segment showing 4th residua
l
                   y = 0,
                   xend = cars2$speed[4],
                   yend = cars2$residual[4]),
               colour = "darkblue") +

  geom_segment(aes(x = cars2$speed[7],                   #Draw line segment showing 7th residua
l
                   y = 0,
                   xend = cars2$speed[7],
                   yend = cars2$residual[7]),
```

```
               colour = "darkblue") +

  geom_segment(aes(x = cars2$speed[10],                  #Draw line segment showing 10th residu
al
                   y = 0,
                   xend = cars2$speed[10],
                   yend = cars2$residual[10]),
               colour = "darkblue") +

  geom_point(data = cars2[c(4,7,10),],                   #Recolour 4th, 7th, and 10th residuals
             aes(x = cars2$speed[c(4,7,10)],
                 y= cars2$residual[c(4,7,10)]),
                 colour = "red", size = 2,
             show.legend = FALSE) +

  geom_text(aes(x = cars2$speed[c(4)],                   #Label 4th residuals
                y= cars2$residual[c(4)],
                label = "4th Residual",
                alpha = 1),
            show.legend = FALSE ,
            size = 3,
            nudge_y = -2,
            colour = "Blue") +

  geom_text(aes(x = cars2$speed[c(7)],                   #Label 7th residuals
                y= cars2$residual[c(7)],
                label = "7th Residual",
                alpha = 1),
            show.legend = FALSE ,
            size = 3,
            nudge_y = 2,
            colour = "Blue") +

  geom_text(aes(x = cars2$speed[c(10)],                  #Label 10th residuals
                y= cars2$residual[c(10)],
                label = "10th Residual",
                alpha = 1),
            show.legend = FALSE ,
            size = 3,
            nudge_y = 2,
            colour = "Blue") +

  geom_hline(yintercept = 0, colour= "darkblue") +      #Add horizontal line at y=0

  ggtitle("Cars2 - Speed vs Residuals") +

  labs(x  = "Speed",
       y  = "Residual") +

  theme_bw()
```



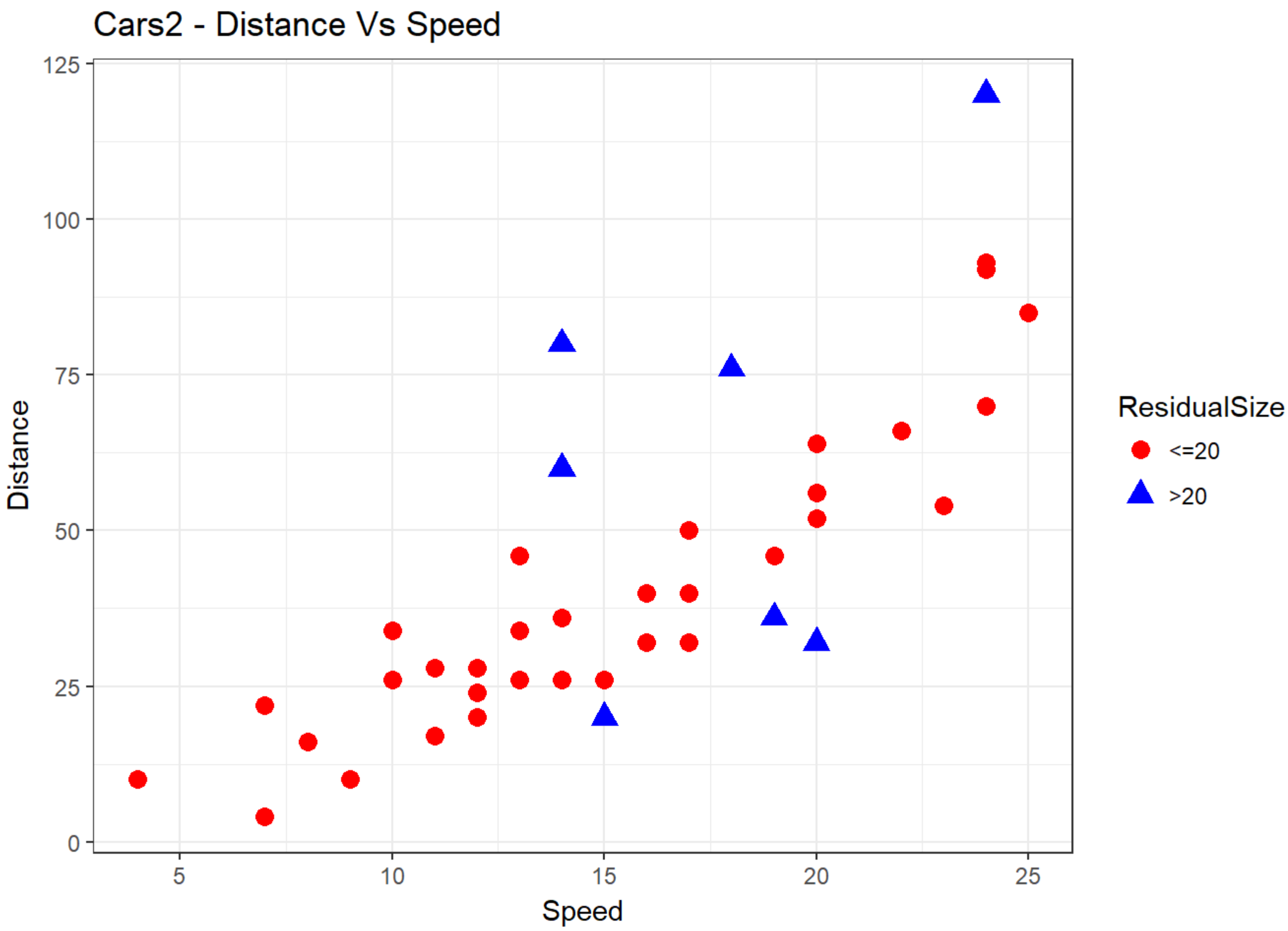

```
cars_residuals <- data.frame(cars2$residual[c(4,7,10)])

rownames(cars_residuals) = c("4th Residual",
                             "7th Residual",
                             "10th Residual")
colnames(cars_residuals) = c("Residual")


kable(cars_residuals,                                 #Create table smmarizing LS esimates
 for output
      caption = "4th, 7th and 10th Residuals",
      align = rep("c", ncol(cars_residuals))) %>%
  kable_styling()
```

4th, 7th and 10th Residuals

	Residual
4th Residual	-10.520797
7th Residual	1.243172
10th Residual	2.181033

D - Indicating residuals >20.

```
cars_w_residual_size <- cars2 %>%                          #add an identifier to Cars2 data in
idcating the size of residuals i.e >20 or <20.
                mutate(ResidualSize = ifelse(abs(residual)>20,">20","<=20"))

cars2_plot <-                                              #plot data, applying breaks to allo
w aesthetics to be applied based on the size of residuals
  ggplot(cars_w_residual_size,
         aes(x = speed,
             y = dist,
             group = ResidualSize,
             size = ResidualSize,
             shape = ResidualSize,
             colour = ResidualSize))+
  geom_point() +
  scale_colour_manual(breaks = c("<=20",">20"),
                      values = c("red", "blue"))+          #Apply different colours based on r
esidual size
  scale_size_manual(breaks = c("<=20",">20"),
                    values = c(3, 3.5))+                   #Scale the plotted points

  ggtitle("Cars2 - Distance Vs Speed") +                   #Change Plot titles & theme
  xlab("Speed") +
  ylab("Distance") +
  theme_bw()

cars2_plot
```

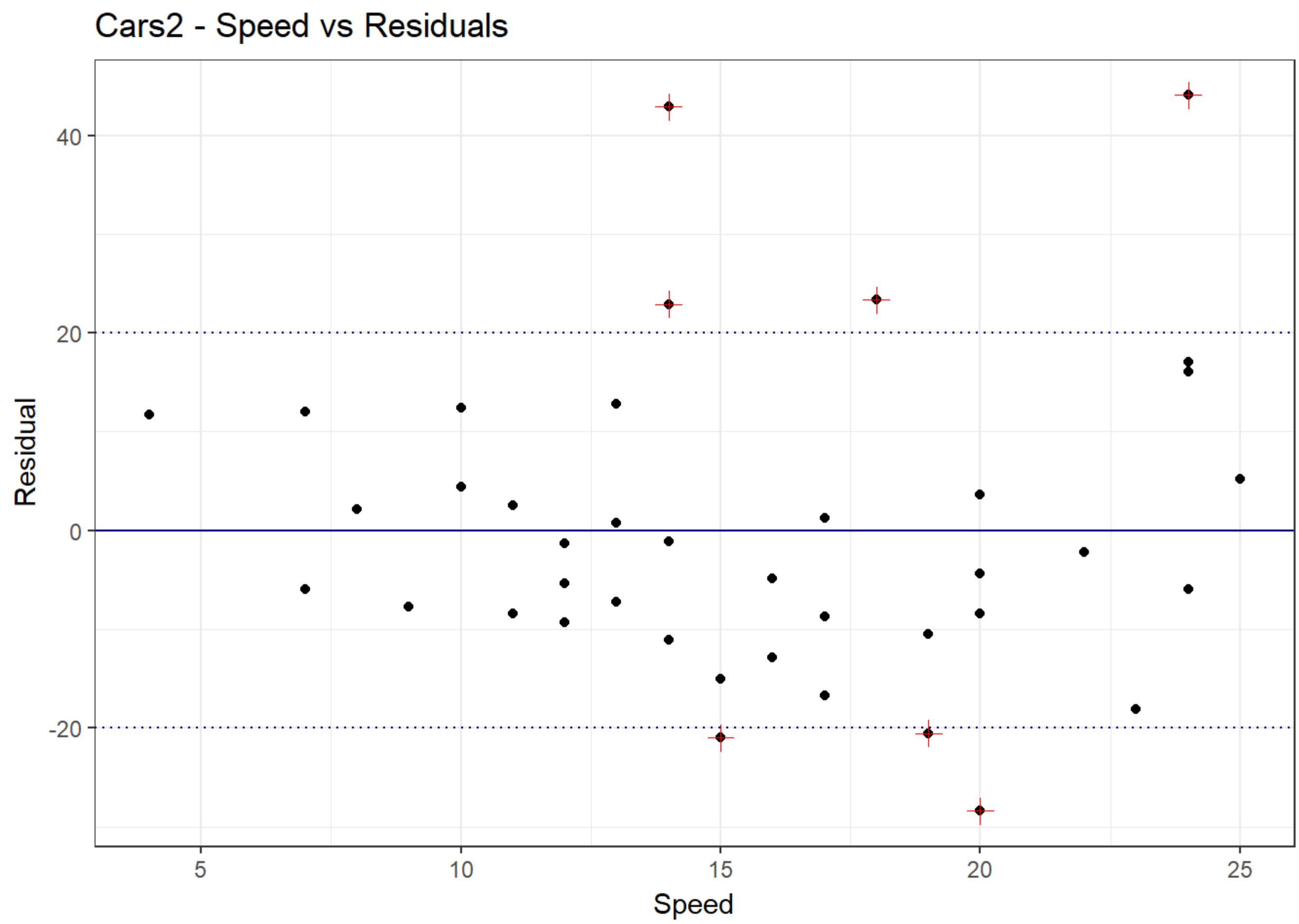


The above plot shows speed vs distance for the Cars2 data. Observations that had residuals larger than 20 were highlighted in blue.

We can better understand the residuals of the regression line better if we plot the observations against their residuals. The following plot shows the relationship between residuals and speed.

```
large_resid <- cars2 %>%                                    #filter to Cars2 data by size of re
siduals i.e >20.
                filter( abs(residual)>20)

cars2_plot_residual <- ggplot(cars2) +                      #Create residual plot of data

  geom_point(aes(x = cars2$speed,
                 y=two_b_lm$residuals))+

  geom_point(data = large_resid,
             aes(x = speed,
                 y= residual),
                 colour = "red",
                 shape = 3,
                 size = 3,
             show.legend = TRUE) +

  geom_hline(yintercept = 0,                                #Insert horizontal line at y=0
             colour= "darkblue") +
```

```r
  geom_hline(yintercept = 20,                         #Insert horizontal line at y=20
             linetype = "dotted",
             colour = "darkblue") +

  geom_hline(yintercept = -20,                        #Insert horizontal line at y=-20
             linetype = "dotted",
             colour = "darkblue") +

  ggtitle("Cars2 - Speed vs Residuals") +             #Add titles and themes

  labs(x  = "Speed",
       y  = "Residual") +

  theme_bw()

cars2_plot_residual
```



It is worth noting that apart from the 7 observations that had large residuals i.e above 20, the remaining 82.5% of the data was within 20 and showed no general trend amongst the residuals.

E - Sum of Residuals

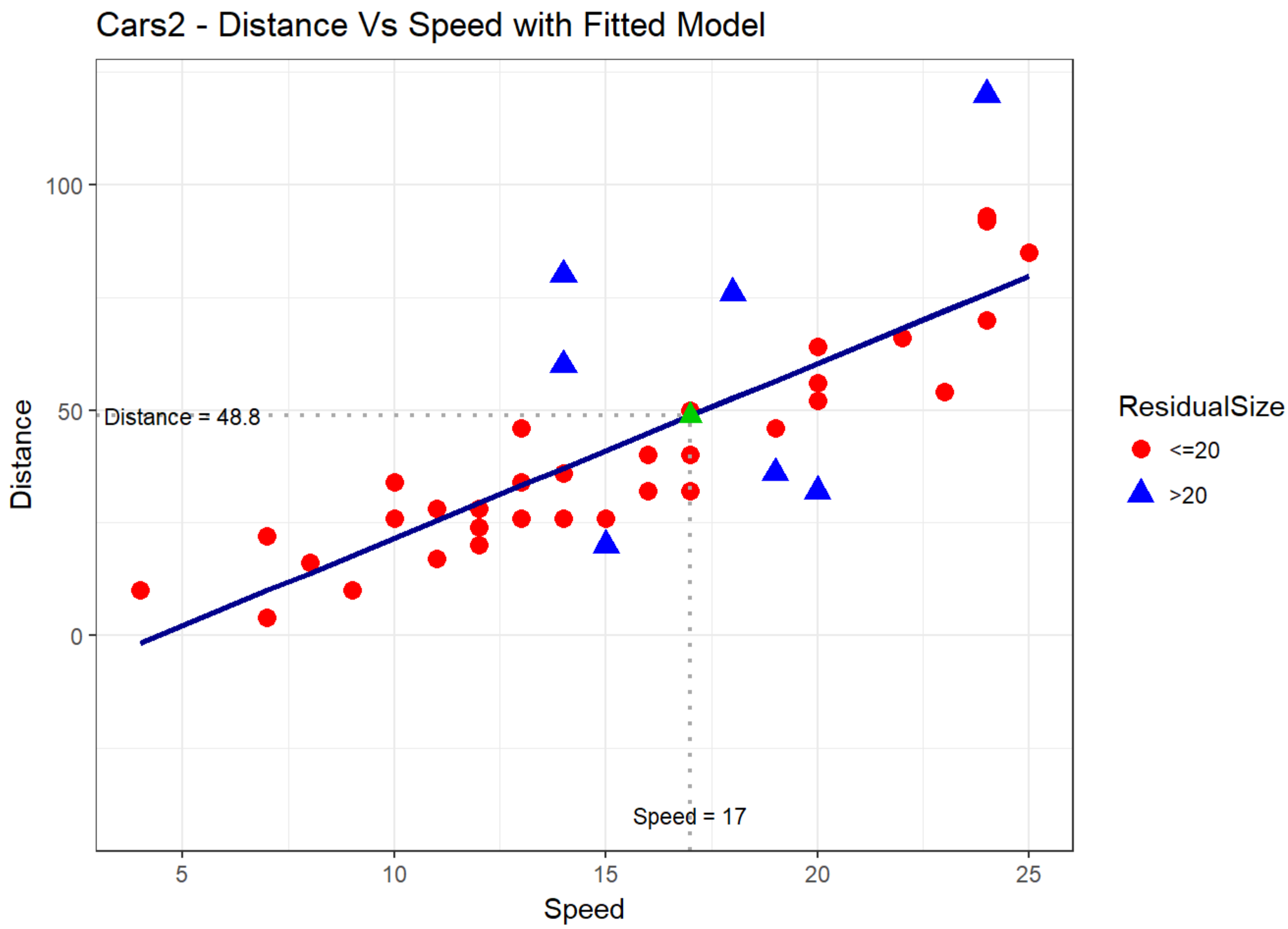
```r
sum_of_residuals <- sum(cars2$residual)               #Calculate sum of residuals

cat("Sum of residuals =", round(sum_of_residuals,5))
```

```
## Sum of residuals = 0
```

F - Plotting fitted Line & Predicting values

```
beta0 <- two_b_lm$coefficients[1]                          #Extract coefficients from lm.
beta1 <- two_b_lm$coefficients[2]

cars2_plot +

  geom_line(aes(x = speed, y = beta0 + beta1*speed),       #Add the fitted model to the existi
ng cars2_plot
            linetype = "solid",
            colour = "darkblue",
            size = 1,
            show.legend = FALSE)+

  scale_fill_discrete("")+                                 #Remove legend title

  geom_point(aes(x = 17,                                   #Add dotted lines showing predicted
 value when speed =17
                 y = predict(two_b_lm,
                             newdata = data.frame(speed = 17))),
             colour = "green3", size = 3, shape = 17) +

  geom_segment(aes(x = 17,                                 #Add dotted lines showing predicted
 value when speed =17
                   xend = 17,
                   y = -Inf,
                   yend = predict(two_b_lm, newdata = data.frame(speed = 17))),
               colour= "darkgrey",
               linetype = "dotted",
               size = .75) +

  geom_segment(aes(x = -Inf,                               #Add dotted lines showing predicted
 value when speed =17
                   xend = 17,
                   y = predict(two_b_lm, newdata = data.frame(speed = 17)),
                   yend = predict(two_b_lm, newdata = data.frame(speed = 17))),
               colour= "darkgrey",
               linetype = "dotted",
               size = .75) +

  annotate("text", x = 17,                                 #Add text to plot
           y = -40,
           label = "Speed = 17",
           size = 3)  +

  annotate("text",                                         #Add text to plot
           x = min(cars2$speed)+1,
           y = predict(two_b_lm,
                       newdata = data.frame(speed = 17)),
           label = "Distance = 48.8", size = 3)  +
```

```
ggtitle("Cars2 - Distance Vs Speed with Fitted Model")   #Add title
```




```
fit_model <- predict(two_b_lm, newdata = data.frame(speed = 17)) #Predict using the fittded mo
del distance to stop given a speed of 17 mph.


cat("Considering the fitted model, the predicted distance taken to stop when the speed of the
car is 17 mph is", round(fit_model,2))
```

```
## Considering the fitted model, the predicted distance taken to stop when the speed of the ca
r is 17 mph is 48.76
```

The LS estimate for the the regression line that models distance as a function of speed for the car2 dataset is:

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X$$

where;

$$\hat{\beta}_0 = -17.24$$

$$\hat{\beta}_1 = 3.88$$

G - Goodness of fit

```
SSE<- sum((cars2$residual)^2)

SST <- sum((cars2[,c("dist")]-mean(cars2$dist))^2)

r_2 <- 1-SSE/SST

paste("Proportion of Variance explained by the regression model",round(r_2,2), sep = "-> ")
```

```
## [1] "Proportion of Variance explained by the regression model-> 0.64"
```

H - Extrapolating Data

```
cars2_speed_summary <- rbind(data.frame(min(cars2$speed),     #Generate key summary statistics
 of the domain of the cars2 speed data.
                                        mean(cars2$speed),
                                        max(cars2$speed),
                                        sd(cars2$speed)))



fit_model_100 <- predict(two_b_lm,                                #USing the fitted model, predic
t the outcome considering a speed of 100mph.
                         newdata = data.frame(speed = 100))



rownames(cars2_speed_summary) = c("Statistics")
colnames(cars2_speed_summary) = c("Min", "Mean", "Max", "Standard Dev.")



paste("Considering the regression model for distance, at 100km/h we predict that the response
i.e the distance to be: ", round(fit_model_100,2), "Km", sep = " ")
```

```
## [1] "Considering the regression model for distance, at 100km/h we predict that the response
 i.e the distance to be:  370.96 Km"
```

```
kable(cars2_speed_summary, caption = "Domain statistics of Speed for Cars2 data",
      align = rep("c", ncol(cars2_speed_summary))) %>%
  kable_styling(position = "center")
```

Domain statistics of Speed for Cars2 data

	Min	Mean	Max	Standard Dev.
Statistics	4	15.575	25	5.40127

The summary provied above displays the domain of the sample data for the Cars2 dataset. Generally, infering data from the regression line is suitable however, exprapolating data from the regression line can lead to poor predictions. In this case, making predictions for the distance at 100mph is an unreliable prediction as the support for the model has a maximum value of 25 mph, and a minimum of 4mph. 100mph is several standard deviations from the mean and outside the max speed in the training data. Therefore, making decisions, or drawing conclusions from this data is irreponsible.

I - Relationship between x and Y - Hypothesis Testing

```
test_b1 <- summary(two_b_lm)

print(test_b1)
```

```
##
## Call:
## lm(formula = dist ~ speed, data = cars2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.403  -8.904  -3.285   6.818  44.069
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.2369     7.7336  -2.229   0.0318 *
## speed         3.8820     0.4698   8.264 5.15e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.85 on 38 degrees of freedom
## Multiple R-squared:  0.6425, Adjusted R-squared:  0.6331
## F-statistic: 68.29 on 1 and 38 DF,  p-value: 5.152e-10
```

Considering the following Null and Alternative Hypothesis:

$$H_0 : \beta_1 = 0$$

;

$$H_A : \beta_1 \neq 0$$

and significance level:

$$\alpha = 0.05$$

The above summary shows that there is strong evidence against the Null hypothesis, therefore we reject the Null hypothesis.

J - Hypothesis Testing - One sided

```
beta1_true <- 4

se_beta1 <- summary(two_b_lm)$coefficients[2,2]

t_stat <- (beta1-beta1_true)/(se_beta1)

pvalue <- pt(t_stat, 38)
```

```r
t_test_df <- data.frame('Beta 1 hat' = beta1, 'Beta 1 null' = beta1_true, 'Standard Error' = s
e_beta1, 't-stat' = t_stat, 'Pr(>|t|)' = pvalue)

colnames(t_test_df) = c("Beta Hat", "Beta-Null", "Standard Error", "t Stat", "Pr(>|t|)")
rownames(t_test_df) = c("Key Values")

kable(t_test_df,
      caption = "t-Test Statistics",
      align = rep("c", ncol(t_test_df))) %>%
  kable_styling()
```

t-Test Statistics

	Beta Hat	Beta-Null	Standard Error	t Stat	Pr(>|t|)
Key Values	3.881985	4	0.4697592	-0.2512253	0.4014967

Considering the following Null and Alternative Hypothesis:

$$H_0 : \beta_1 = 4$$

;

$$H_A : \beta_1 \leq 4$$

and significance level:

$$\alpha = 0.05$$

The above table, shows the p-value is significantly greater than significance level which indicates that there is no evidence against the Null Hypothesis. Therefore, we fail to reject the Null.