CS 422 Home Work 3

Rajesh Mavi

- Part 2.1 Install package ISLR. This package contains a dataset called Auto.
- Part 2.1 (a) From the training dataset, create a model using all the predictors except name to predict mpg.
- Part 2.1 (a) (i) Why is using name as a predictor not a reasonable thing to do?
- Part 2.1 (a) (ii) Print the summary of the regression model, and comment on how well the model fits the data by studying the R 2 , RSE and RMSE. (Print out the values of R2 , RSE and RMSE.)
- Part 2.1 (a) (iii) Plot the residuals of the model.
- Part 2.1 (a) (iv) Plot a histogram of the residuals of the model. Does the histogram follow a Gaussian distribution? What can you say about the distribution of the residuals?
- Part 2.1 (b) Using the regression model you have created in (a), your aim is to narrow down the features to the 3 attributes will act as the best predictors for regressing on mpg.
- Part 2.1 (b) (i) Determine which predictors are statistically significant and which are not. Eliminate those that are not statistically significant and create a new model using only those 3 predictors that you believe are statistically significant.
- Part 2.1 (b)(ii) Print the summary of the regression model created in (b)(i) and comment on how well the model fits the data by studying the R2 , RSE and RMSE. (Print out the values of R2 , RSE, and RMSE.)
- Part 2.1 (b)(iii) Plot the residuals of the model.
- Part 2.1 (b)(iv) Plot a histogram of the residuals of the model. Does the histogram follow a Gaussian distribution? What can you say about the distribution of the residuals?
- Part 2.1 (b)(v) Comparing the summaries of the model produced in (a) and in (b), including residual analysis of each model. Which model do you think is better, and why?
- Part 2.1 (c) Using the predict() method, fit the test dataset to the model you created in (b) and perform the analysis below.
- Part 2.1 (d) Count how many of the fitted values matched the mpg in the test dataset at a 95% confidence level by creating confidence intervals.
- Part 2.1 (e) Follow the same instructions in (d) except this time, you will be using a prediction interval
- Part 2.1 (f) Comment on the results of (d) and (e):
- Part 2.1 (f)(i) Which of (d) or (e) results in more matches?
- Part 2.1 (f)(ii) Why?

Part 2.1 Install package ISLR. This package contains a dataset called Auto.

```
library(ISLR)
head(Auto)
```

	…	cylinders	displacement	horsepower	wei…	acceleration	y…	origin	name
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<fct>
1	18	8	307	130	3504	12.0	70	1	chevrolet chevelle m
2	15	8	350	165	3693	11.5	70	1	buick skylark 320
3	18	8	318	150	3436	11.0	70	1	plymouth satellite

	…	cylinders	displacement	horsepower	wei…	acceleration	y…	origin	name
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<fct>
4	16	8	304	150	3433	12.0	70	1	amc rebel sst
5	17	8	302	140	3449	10.5	70	1	ford torino
6	15	8	429	198	4341	10.0	70	1	ford galaxie 500

6 rows

```
set.seed(1122)
index <- sample(1:nrow(Auto),0.95*dim(Auto)[1])
train.df <- Auto[index,]
test.df <- Auto[-index,]
```

Part 2.1 (a) From the training dataset, create a model using all the predictors except name to predict mpg.

```
model <- lm(mpg ~ . -name, data = train.df)
```

Part 2.1 (a) (i) Why is using name as a predictor not a reasonable thing to do?

1. Mileage of a vehicle does not depends on a name of the vehicle.
2. Attribute ‘name’ is not correlated with mileage and does not significantly help to predict mileage. That’s why it is not reasonable to take ‘name’ as a predictor.

Part 2.1 (a) (ii) Print the summary of the regression model, and comment on how well the model fits the data by studying the R 2 , RSE and RMSE. (Print out the values of R2 , RSE and RMSE.)

```
summ <- summary(model)
summ
```

```
##
## Call:
## lm(formula = mpg ~ . - name, data = train.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.6805 -2.1786 -0.0977  1.9180 13.0364
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.660e+01  4.780e+00  -3.472 0.000578 ***
## cylinders    -5.235e-01  3.340e-01  -1.567 0.117947
## displacement  2.042e-02  7.760e-03   2.632 0.008857 **
## horsepower   -1.750e-02  1.424e-02  -1.229 0.219908
## weight       -6.416e-03  6.785e-04  -9.457  < 2e-16 ***
## acceleration  8.742e-02  1.031e-01   0.848 0.396859
## year          7.383e-01  5.259e-02  14.039  < 2e-16 ***
## origin        1.516e+00  2.893e-01   5.240 2.73e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.367 on 364 degrees of freedom
## Multiple R-squared:  0.817,  Adjusted R-squared:  0.8135
## F-statistic: 232.2 on 7 and 364 DF,  p-value: < 2.2e-16
```

```
cat("R square : ",summ$r.squared,"\n")
```

```
## R square :  0.8170336
```

```
cat("Adjusted R square : ",summ$adj.r.squared,"\n")
```

```
## Adjusted R square :  0.813515
```

```
cat("RSE : ",summ$sigma,"\n")
```

```
## RSE :  3.366918
```

```
cat("RMSE : ",sqrt(mean((model$residuals)^2)))
```

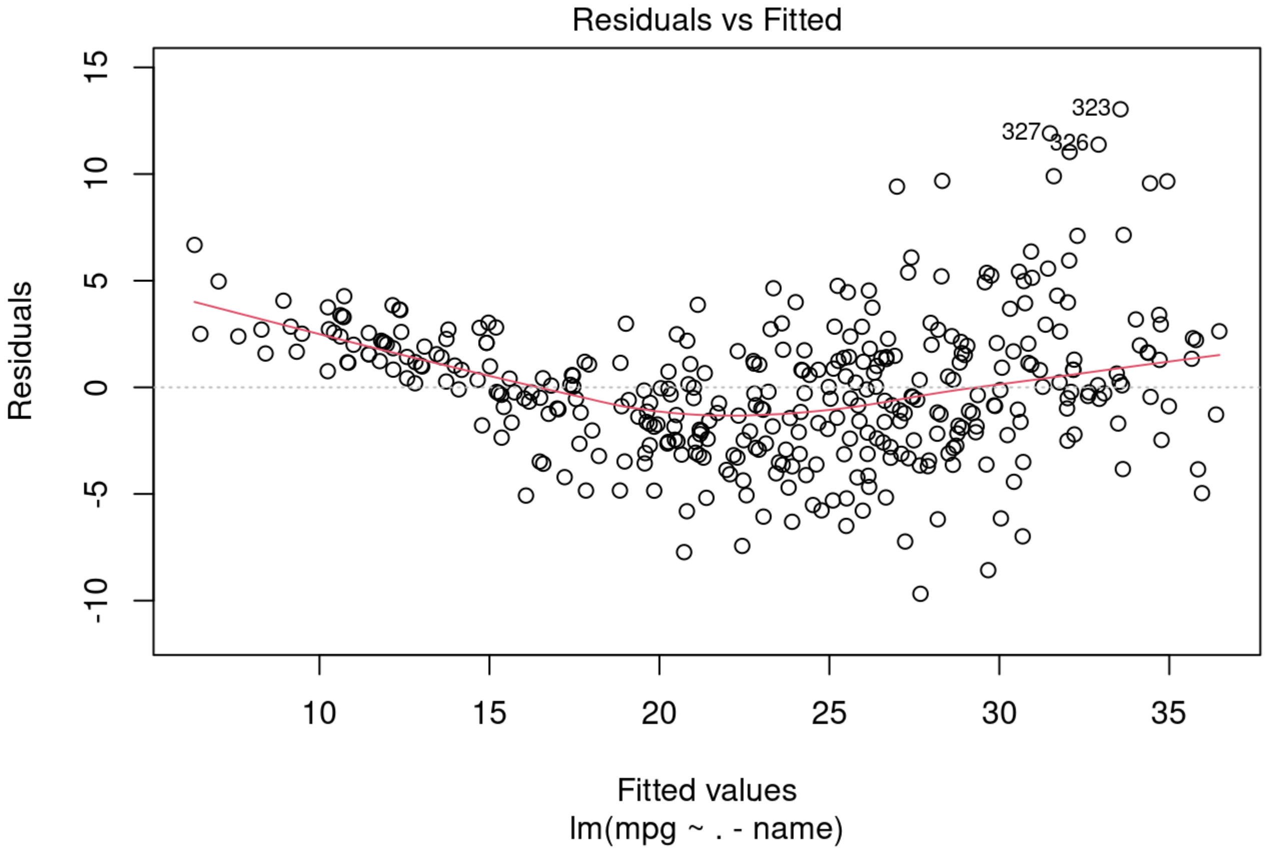
```
## RMSE :  3.330518
```

1. R square means the correlation between predicted and actual values. It should be nearer to 1. For this particular model the value of R square is 0.817 hence, model is good fitted to data.
2. The residual standard error (RSE) is a way to measure the standard deviation of the residuals.
3. Root Mean Square Error (RMSE) is also the standard deviation of the residuals. RSE and RMSE both should be as low as possible.

4. For this particular model, by studying R square, RSE and RMSE, It seems that model has fitted to data quite well.

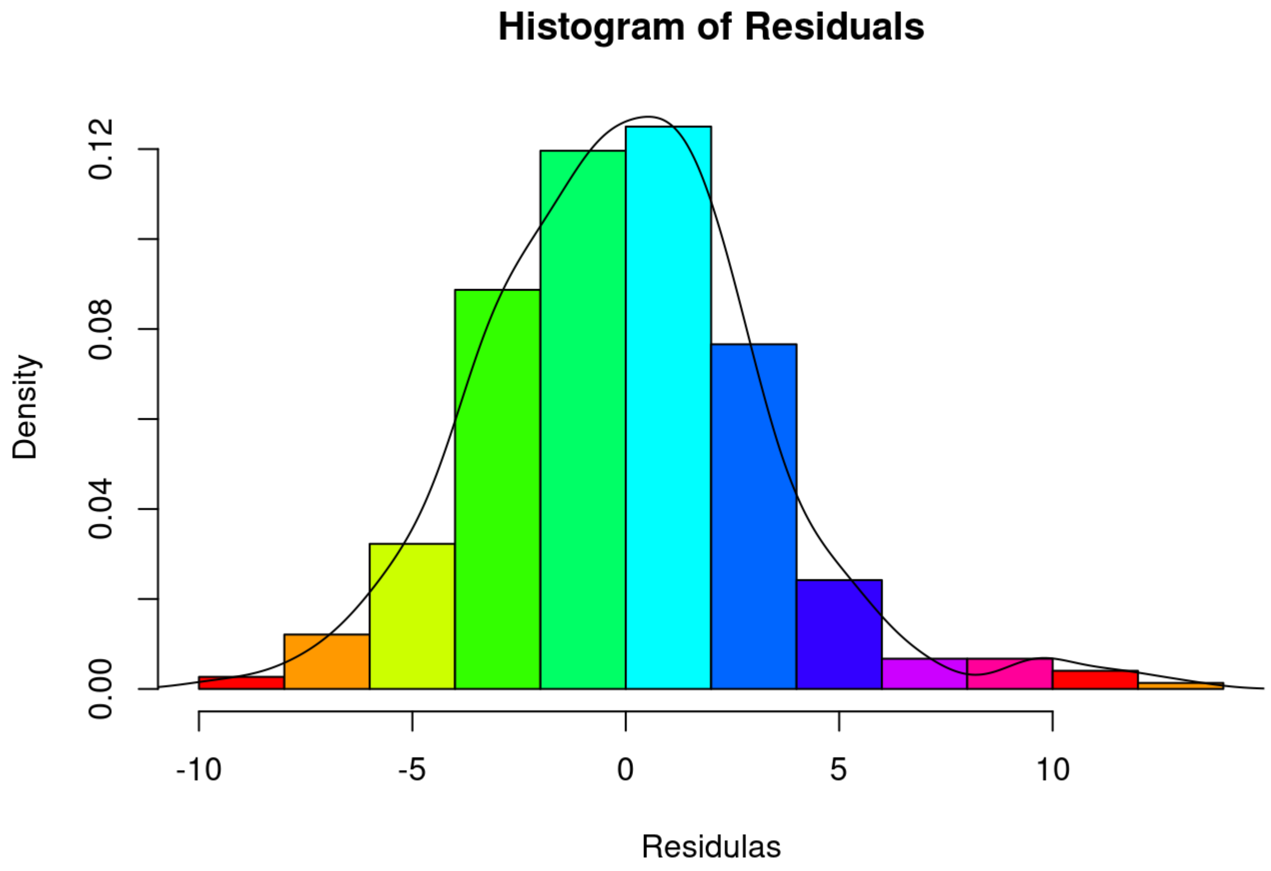
Part 2.1 (a) (iii) Plot the residuals of the model.

```
plot(model,1)
```



Part 2.1 (a) (iv) Plot a histogram of the residuals of the model. Does the histogram follow a Gaussian distribution? What can you say about the distribution of the residuals?

```
hist(model$residuals,main = "Histogram of Residuals", xlab = "Residulas", col = rainbow(
10), freq = F)
lines(density(model$residuals))
```



As shown in histogram, Residuals follow Gaussian distribution. Density is high nearby zero, it means residuals are nicely clustered around zero line.It indicates our model is good fitted to data.

Part 2.1 (b) Using the regression model you have created in (a), your aim is to narrow down the features to the 3 attributes will act as the best predictors for regressing on mpg.

```
summ
```

```
##
## Call:
## lm(formula = mpg ~ . - name, data = train.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.6805 -2.1786 -0.0977  1.9180 13.0364
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.660e+01  4.780e+00  -3.472 0.000578 ***
## cylinders    -5.235e-01  3.340e-01  -1.567 0.117947
## displacement  2.042e-02  7.760e-03   2.632 0.008857 **
## horsepower   -1.750e-02  1.424e-02  -1.229 0.219908
## weight       -6.416e-03  6.785e-04  -9.457  < 2e-16 ***
## acceleration  8.742e-02  1.031e-01   0.848 0.396859
## year          7.383e-01  5.259e-02  14.039  < 2e-16 ***
## origin        1.516e+00  2.893e-01   5.240 2.73e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.367 on 364 degrees of freedom
## Multiple R-squared:  0.817,  Adjusted R-squared:  0.8135
## F-statistic: 232.2 on 7 and 364 DF,  p-value: < 2.2e-16
```

Part 2.1 (b) (i) Determine which predictors are statistically significant and which are not. Eliminate those that are not statistically significant and create a new model using only those 3 predictors that you believe are statistically significant.

1. As per the summary of previous model it appears that, the p value of predictor “cylinders” is 0.12, “horsepower” is 0.22 and “acceleration” is 0.39 which is greater than 0.1.
2. It means predictors “cylinder”, “horsepower”, “acceleration” are not statistically significant enough to predict response variable.
3. So, we can eliminate these three to narrow down the features.
4. Predictor “displacement” is also a significant but not as statistically significant as predictors “weight”, “year”, “origin” are.
5. So, we can eliminate “displacement” too.
6. Now, We can form a new model based on predictors which are statistically significant ie. “weight”, “year”, “origin”.

```
new.model <- lm(mpg ~ weight+year+origin,data = train.df)
```

Part 2.1 (b)(ii) Print the summary of the regression model created in (b)(i) and comment on how well the model fits the data by studying the R2 , RSE and RMSE. (Print out the values of R2 , RSE, and RMSE.)

```
new.summ <- summary(new.model)
new.summ
```

```
##
## Call:
## lm(formula = mpg ~ weight + year + origin, data = train.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.0433  -2.1120  -0.0448   1.6867  13.2596
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.731e+01  4.123e+00  -4.197 3.39e-05 ***
## weight      -5.973e-03  2.657e-04 -22.481  < 2e-16 ***
## year         7.448e-01  4.983e-02  14.946  < 2e-16 ***
## origin       1.223e+00  2.701e-01   4.525 8.15e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.389 on 368 degrees of freedom
## Multiple R-squared:  0.8126, Adjusted R-squared:  0.8111
## F-statistic: 531.8 on 3 and 368 DF,  p-value: < 2.2e-16
```

```
cat("R square : ",new.summ$r.squared,"\n")
```

```
## R square :  0.8125806
```

```
cat("Adjusted R square : ",new.summ$adj.r.squared,"\n")
```

```
## Adjusted R square :  0.8110527
```

```
cat("RSE : ",new.summ$sigma,"\n")
```

```
## RSE :  3.389074
```

```
cat("RMSE : ",sqrt(mean((new.model$residuals)^2)))
```

```
## RMSE :  3.370804
```

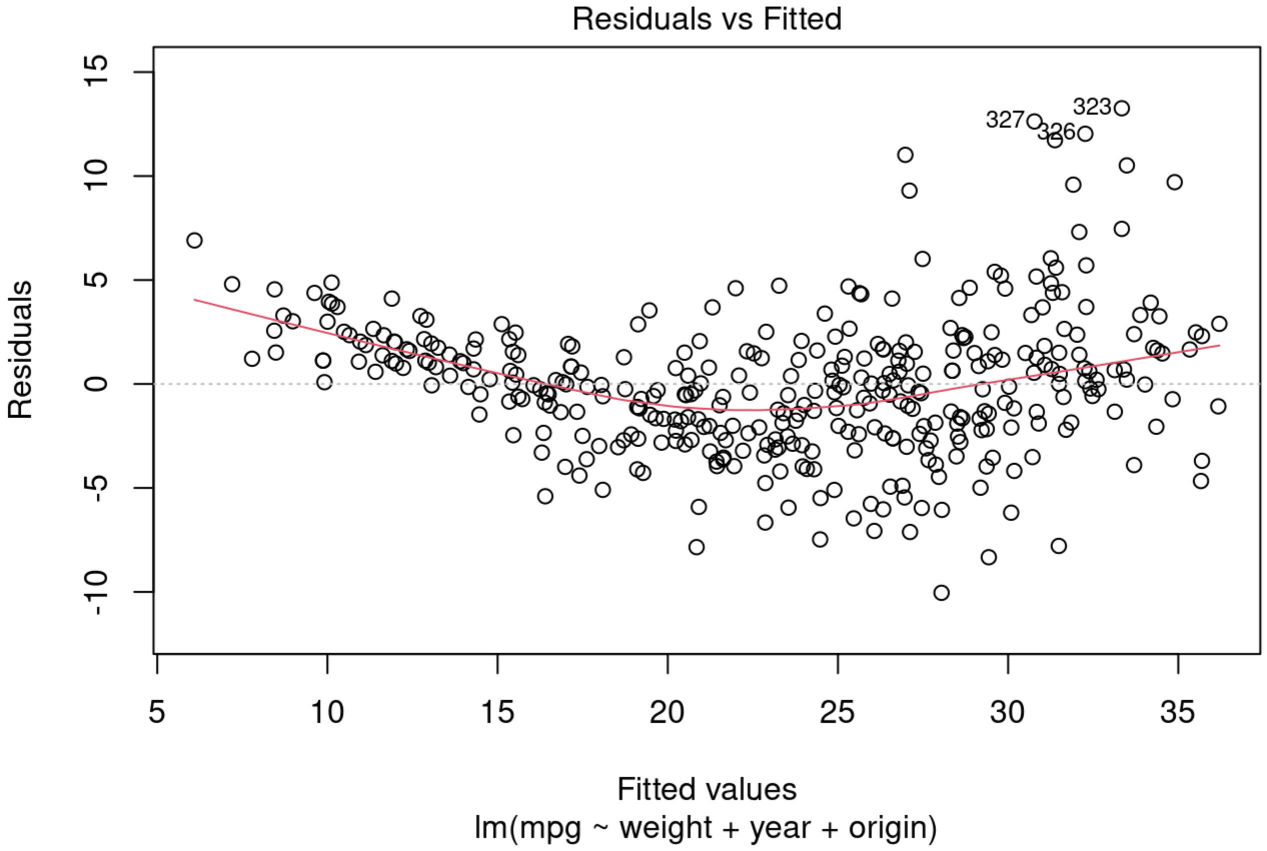
1. R square value is 0.813 which is greater than 1.
2. RSE and RMSE are nearby zero.
3. For this particular model, by studying R square, RSE and RMSE, It seems that model has fitted to data quite well.

Part 2.1 (b)(iii) Plot the residuals of the model.

Best fit line of residuals is little curvy.

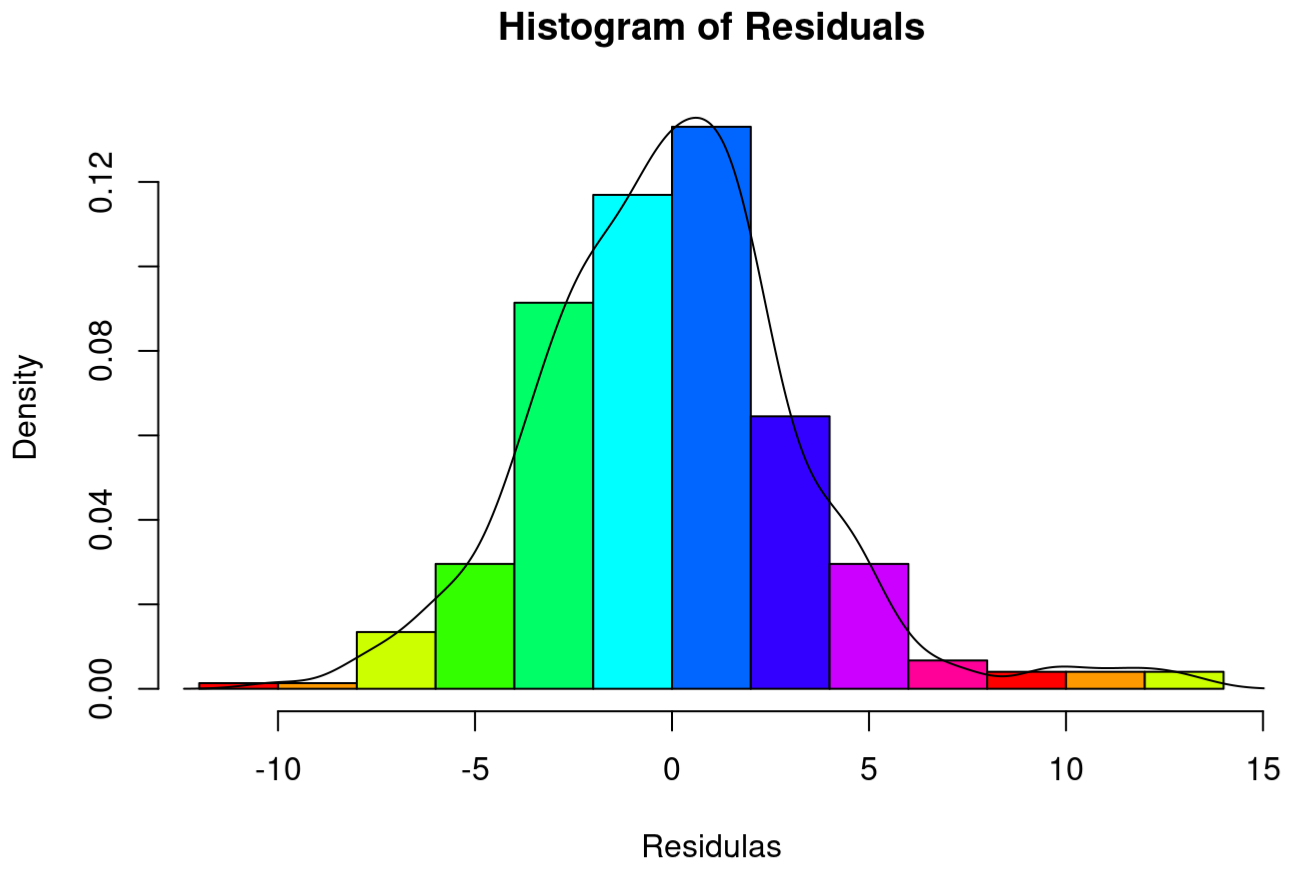
Residuals are clustered around zero line.

```
plot(new.model,1)
```



Part 2.1 (b)(iv) Plot a histogram of the residuals of the model. Does the histogram follow a Gaussian distribution? What can you say about the distribution of the residuals?

```
hist(new.model$residuals,main = "Histogram of Residuals", xlab = "Residulas", col = rain
bow(10), freq = F)
lines(density(new.model$residuals))
```



As shown in histogram, Residuals follow Gaussian distribution.
Density is high nearby zero, it means residuals are nicely clustered around zero line.
It indicates our model is good fitted to data.

Part 2.1 (b)(v) Comparing the summaries of the model produced in (a) and in (b), including residual analysis of each model. Which model do you think is better, and why?

Comparing summary of first model and second model,
1. The F-statistic of model 1 is 232.2 and The F-statistic of model 2 is 531.8.
F statistic of model 2 is greater than model 1 which means all the predictors are statistically significant of model 2 to predict response variable.
2. the p value for both model is nearly zero.
3. But there is no improvement on RSE and adjusted R squared values of both the models.
4. As F Statistics of model 2 have been improved from 232.2 to 531.8.
5. Both the residual plot of model 1 and model 2 are same. There is no significant difference between them.
6. It appears that density curve of residuals of model 2 is little narrower as compare to model 1.

As per the comparison, model 2 is better than model 1.

Part 2.1 (c) Using the predict() method, fit the test dataset to the model you created in (b) and perform the analysis below.

Predicted values of test sample

```
pred <- predict.lm(new.model, test.df[,c("weight","year","origin")], interval = "confide
nce")
pred
```

```
##           fit       lwr      upr
## 23  23.087261 22.298650 23.87587
## 86  13.796155 13.202787 14.38952
## 96   8.713373  7.789197  9.63755
## 111 26.520256 25.711209 27.32930
## 121 22.377070 21.874010 22.88013
## 140 11.327620 10.541322 12.11392
## 153 20.278919 19.850586 20.70725
## 161 16.438462 15.923494 16.95343
## 176 29.427242 28.827257 30.02723
## 178 24.905895 24.492442 25.31935
## 179 23.335070 22.912596 23.75755
## 189 15.492969 14.864803 16.12114
## 259 21.820442 21.326734 22.31415
## 279 31.345046 30.804392 31.88570
## 302 29.613034 28.873699 30.35237
## 319 29.750817 28.901381 30.60025
## 343 29.997654 29.233435 30.76187
## 345 33.043740 32.123167 33.96431
## 348 34.891522 34.097803 35.68524
## 359 30.949529 30.072125 31.82693
```

New Dataframe is created

```
df <- data.frame("Prediction"=round(pred[,"fit"],2),"Response"=test.df$mpg)
df
```

	Prediction	Response
	<dbl>	<dbl>
23	23.09	25.0
86	13.80	13.0
96	8.71	12.0
111	26.52	22.0
121	22.38	19.0
140	11.33	14.0
153	20.28	19.0
161	16.44	17.0
176	29.43	29.0
178	24.91	23.0

1-10 of 20 rows Previous **1** 2 Next

Add confidence intervals in new dataframe

```
df <- cbind(df, round(pred[,c(2,3)],2))
df
```

	Prediction <dbl>	Response <dbl>	lwr <dbl>	upr <dbl>
23	23.09	25.0	22.30	23.88
86	13.80	13.0	13.20	14.39
96	8.71	12.0	7.79	9.64
111	26.52	22.0	25.71	27.33
121	22.38	19.0	21.87	22.88
140	11.33	14.0	10.54	12.11
153	20.28	19.0	19.85	20.71
161	16.44	17.0	15.92	16.95
176	29.43	29.0	28.83	30.03
178	24.91	23.0	24.49	25.32

1-10 of 20 rows Previous **1** 2 Next

Part 2.1 (d) Count how many of the fitted values matched the mpg in the test dataset at a 95% confidence level by creating confidence intervals.

```
f <- function(x) {
  if(x[2]>x[3] && x[2]<x[4])
  {
    return(1)
  }
  else{
    return(0)
  }}

res <- apply(df,1,f)
df$Matches <- res
df
```

	Prediction <dbl>	Response <dbl>	lwr <dbl>	upr <dbl>	Matches <dbl>
23	23.09	25.0	22.30	23.88	0

	Prediction	Response	lwr	upr	Matches
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
86	13.80	13.0	13.20	14.39	0
96	8.71	12.0	7.79	9.64	0
111	26.52	22.0	25.71	27.33	0
121	22.38	19.0	21.87	22.88	0
140	11.33	14.0	10.54	12.11	0
153	20.28	19.0	19.85	20.71	0
161	16.44	17.0	15.92	16.95	0
176	29.43	29.0	28.83	30.03	1
178	24.91	23.0	24.49	25.32	0

1-10 of 20 rows Previous **1** 2 Next

```
cat("Total observations correctly predicted: ",sum(df$Matches))
```

```
## Total observations correctly predicted:  7
```

Part 2.1 (e) Follow the same instructions in (d) except this time, you will be using a prediction interval

Predicted values of test sample of second model

```
pred2 <- predict.lm(new.model, test.df[,c("weight","year","origin")], interval = "predic
tion")
pred2
```

```
##           fit       lwr      upr
## 23  23.087261 16.376384 29.79814
## 86  13.796155  7.105412 20.48690
## 96   8.713373  1.985219 15.44153
## 111 26.520256 19.806946 33.23357
## 121 22.377070 15.693730 29.06041
## 140 11.327620  4.617014 18.03823
## 153 20.278919 13.600788 26.95705
## 161 16.438462  9.754215 23.12271
## 176 29.427242 22.735908 36.11858
## 178 24.905895 18.228702 31.58309
## 179 23.335070 16.657313 30.01283
## 189 15.492969  8.799050 22.18689
## 259 21.820442 15.137800 28.50308
## 279 31.345046 24.658771 38.03132
## 302 29.613034 22.907769 36.31830
## 319 29.750817 23.032520 36.46911
## 343 29.997654 23.289599 36.70571
## 345 33.043740 26.316079 39.77140
## 348 34.891522 28.180043 41.60300
## 359 30.949529 24.227639 37.67142
```

Create new Dataframe

```
df2 <- data.frame("Prediction"=round(pred2[,"fit"],2),"Response"=test.df$mpg)
df2
```

	Prediction <dbl>	Response <dbl>
23	23.09	25.0
86	13.80	13.0
96	8.71	12.0
111	26.52	22.0
121	22.38	19.0
140	11.33	14.0
153	20.28	19.0
161	16.44	17.0
176	29.43	29.0
178	24.91	23.0

1-10 of 20 rows Previous **1** 2 Next

Add prediction intervals in new dataframe

```
df2 <- cbind(df2, round(pred2[,c(2,3)],2))
df2
```

	Prediction <dbl>	Response <dbl>	lwr <dbl>	upr <dbl>
23	23.09	25.0	16.38	29.80
86	13.80	13.0	7.11	20.49
96	8.71	12.0	1.99	15.44
111	26.52	22.0	19.81	33.23
121	22.38	19.0	15.69	29.06
140	11.33	14.0	4.62	18.04
153	20.28	19.0	13.60	26.96
161	16.44	17.0	9.75	23.12
176	29.43	29.0	22.74	36.12
178	24.91	23.0	18.23	31.58

1-10 of 20 rows Previous **1** 2 Next

Check matching

```
res2 <- apply(df2,1,f)
df2$Matches <- res2
df2
```

	Prediction <dbl>	Response <dbl>	lwr <dbl>	upr <dbl>	Matches <dbl>
23	23.09	25.0	16.38	29.80	1
86	13.80	13.0	7.11	20.49	1
96	8.71	12.0	1.99	15.44	1
111	26.52	22.0	19.81	33.23	1
121	22.38	19.0	15.69	29.06	1
140	11.33	14.0	4.62	18.04	1
153	20.28	19.0	13.60	26.96	1
161	16.44	17.0	9.75	23.12	1
176	29.43	29.0	22.74	36.12	1
178	24.91	23.0	18.23	31.58	1

1-10 of 20 rows Previous **1** 2 Next

```
cat("Total observations correctly predicted: ",sum(df2$Matches))
```

```
## Total observations correctly predicted:  20
```

Part 2.1 (f) Comment on the results of (d) and (e):

Part 2.1 (f)(i) Which of (d) or (e) results in more matches?

Result (E) has more matches which is 20, than result (D) which is 7.

Part 2.1 (f)(ii) Why?

1.In (D), we calculate matches based on confidence interval and in (E) based on prediction interval
2. Confidence intervals tell us how well we have determined a parameter of interest, such as a mean or regression coefficient Whereas, Prediction intervals tell us where we can expect to see the next data point sampled.
3. So, the range of prediction interval is always more than confidence interval and hence, actual values are more likely to fall under prediction interval.