Fitting a Multiple Linear Model in F1 data to predict Fastest Lap Speed.

Alex Gichuki

2023-04-08

```
# Setup

library(tidyverse)
library(tidymodels)
library(kableExtra)
library(psych)
library(GGally)
library(ggfortify)
library(car)
```

Reading the clean F1 data. Working with data from race 841 to race 1046 for all drivers who completed the race.

```
# Reading the full data
f1_full_df <- read.csv("Formula_One.csv")

# Creating a subset data set: removing driverId from the f1_full_df
f1_df <- f1_full_df %>%
  select(-driverId)

# Glance at the dataset
kable(head(f1_df))
```

raceId	circuitId	grid	finishingPosition	points	laps	raceDuration	rank	fastestLapSpeed	altitude	averagePitstopDuration	averageLapDuration	fastestLapDuration
841	1	1	1	25	58	5370259	4	212.488	10	23319.50	92590.67	89844
841	1	2	2	18	58	5392556	8	211.382	10	23213.00	92975.10	90314
841	1	6	3	15	58	5400819	7	211.969	10	25109.00	93117.57	90064
841	1	5	4	12	58	5402031	2	213.336	10	24055.00	93138.47	89487
841	1	3	5	10	58	5408430	3	213.066	10	24058.67	93248.79	89600
841	1	4	6	8	58	5424563	5	212.396	10	20950.33	93526.95	89883

Specifying the categorical variables

```
# Convert the following variables to categorical variables using as.factor()
f1_df$raceId <- as.factor(f1_df$raceId)
f1_df$circuitId <- as.factor(f1_df$circuitId)
f1_df$grid <- as.factor(f1_df$grid)
f1_df$finishingPosition <- as.factor(f1_df$finishingPosition)
f1_df$rank <- as.factor(f1_df$rank)
```

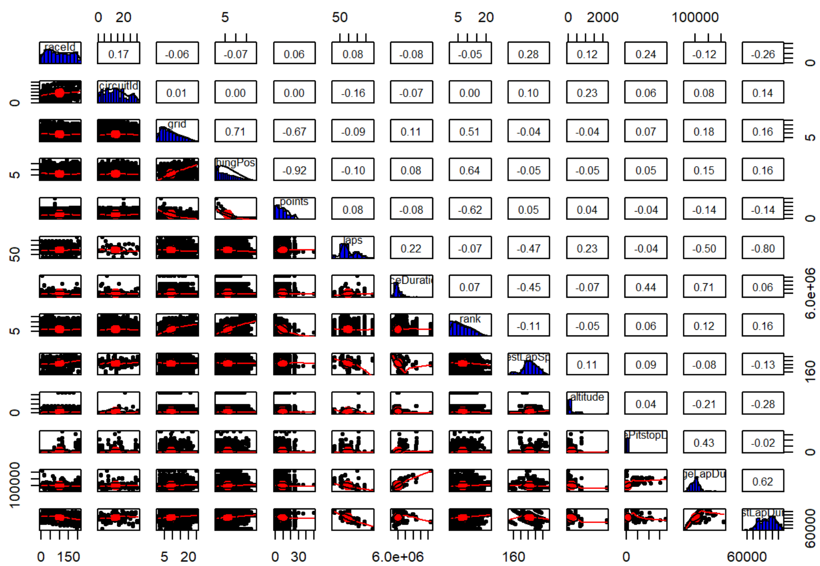
Evaluating the data before fitting the model

```
kable(summary(f1_df))
```

raceId	circuitId	grid	finishingPosition	points	laps	raceDuration	rank	fastestLapSpeed	altitude	averagePitstopDuration	average
855 : 19	9 : 145	1 : 190	1 :208	Min. : 0.00	Min. :43.00	Min. : 4526665	2 : 195	Min. :148.6	Min. : -7.0	Min. : 17349	Min. : 62
991 : 19	13 : 133	2 : 183	2 :208	1st Qu.: 2.00	1st Qu.:53.00	5426327	1 : 193	1st Qu.:193.3	1st Qu.: 7.0	1st Qu.: 21919	1st Qu.:
887 : 18	3 : 121	3 : 182	3 :208	Median : 8.00	Median :57.00	Median : 5767019	3 : 184	Median :205.1	Median : 45.0	Median : 23624	Median
862 : 17	15 : 109	4 : 168	4 :204	Mean : 9.47	Mean :59.35	Mean : 6005680	4 : 178	Mean :204.4	Mean : 191.8	Mean : 68089	Mean :1
877 : 17	17 : 108	5 : 147	5 :195	3rd Qu.:15.00	3rd Qu.:66.00	3rd Qu.: 6185797	5 : 164	3rd Qu.:219.0	3rd Qu.: 162.0	3rd Qu.: 26934	3rd Qu.:
894 : 17	24 : 106	6 : 141	6 :186	Max. :50.00	Max. :87.00	Max. :14743144	6 : 150	Max. :255.0	Max. :2227.0	Max. :1747072	Max. :21
(Other):2010	(Other):1395	(Other):1106	(Other):908	NA	NA	NA	(Other):1053	NA	NA	NA	NA

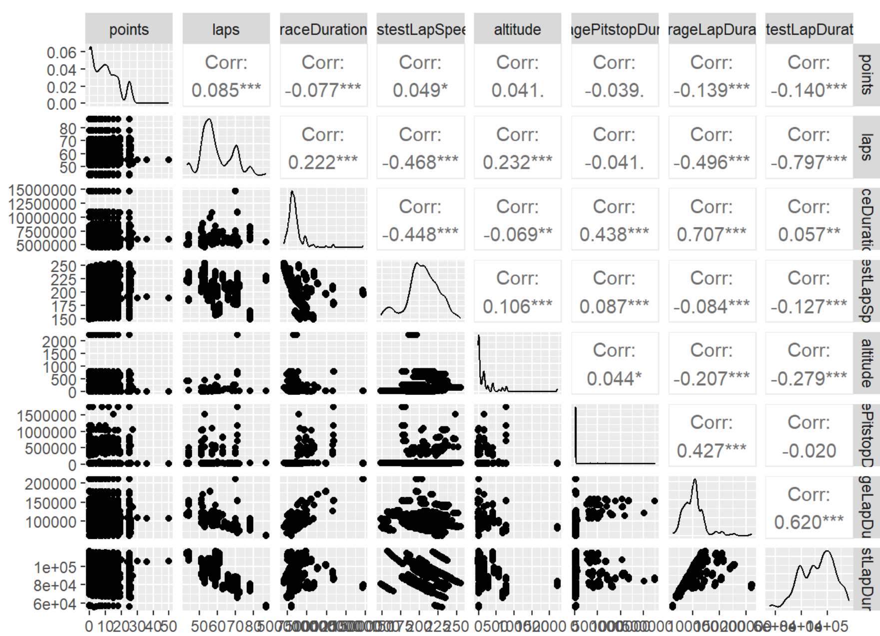
Creating a Scatter plot Matrix to evaluate correlations

```
pairs.panels(
  f1_df,
  hist.col = "blue",
  method= "pearson",
  density = TRUE,
  ellipses = TRUE
)
```




```
# Creating a list with continous variables to evaluate correlations
cont_vars <- c("points", "laps", "raceDuration", "fastestLapSpeed", "altitude",
               "averagePitstopDuration", "averageLapDuration", "fastestLapDuration")

# Create scatterplot matrix using ggpairs for continuous variables
ggpairs(data = f1_df[, cont_vars])
```



#Fitting a Multiple Linear Regression Model

model1 = Using all variables

```
# Fitting a linear model in all variables to predict Fastest Lap Speed
model1 <- lm(fastestLapSpeed ~ raceId + circuitId + grid + finishingPosition +
                 points + laps + raceDuration + rank + altitude + fastestLapDuration +
              averageLapDuration + averagePitstopDuration, data = f1_df)

tidy(model1)
```

term <chr>	estimate <dbl>	std.error <dbl>	statistic <dbl>	p.value <dbl>
(Intercept)	3.723314e+02	3.676941e+00	101.26117888	0.000000e+00
raceId842	4.509409e+00	5.026297e-01	8.97163310	7.016861e-19
raceId843	1.495968e+00	4.759307e-01	3.14324633	1.697481e-03
raceId844	1.065721e+00	1.641964e-01	6.49052490	1.096766e-10
raceId845	-2.365035e+01	1.124630e+00	-21.02944385	3.635466e-88
raceId846	-7.997381e+01	3.252617e+00	-24.58752578	1.132657e-115
raceId847	-5.809546e+01	6.727294e+00	-8.63578451	1.241174e-17
raceId848	2.855135e-01	5.022888e-01	0.56842496	5.698158e-01
raceId849	1.732362e+01	9.202477e-01	18.82495503	1.935014e-72
raceId850	-7.793233e+00	4.576138e-01	-17.03015204	1.648168e-60

1-10 of 308 rows Previous **1** 2 3 4 5 6 ... 31 Next

Model 1 Summary statistics

```
glance(model1)
```

r.squared <dbl>	adj.r.squared <dbl>	sigma <dbl>	statistic <dbl>	p.value <dbl>	df <dbl>	logLik <dbl>	AIC <dbl>	BIC <dbl>	deviance <dbl>
0.9997739	0.9997401	0.3388739	29602.58	0	275	-565.1665	1684.333	3251.531	211.4122

1 row | 1-10 of 12 columns

The r squared is very high which could suggest over fitting. Additionally the resulting model is very complex due to the categorical variables in the data with many levels. A simpler model would be better.

Fitting model 2: Removing one variable from each variable pair with high correlation. Here I remove averagePitstopDuration and averageLapDuration

```
model2 <- lm(fastestLapSpeed ~ raceId + circuitId + grid + finishingPosition +
                 points + laps + raceDuration + rank + altitude + fastestLapDuration,
               data = f1_df)

tidy(model2)
```

term <chr>	estimate <dbl>	std.error <dbl>	statistic <dbl>	p.value <dbl>
(Intercept)	3.721949e+02	3.691063e+00	1.008368e+02	0.000000e+00
raceId842	5.728528e+00	4.087811e-01	1.401368e+01	1.792793e-42
raceId843	2.717478e+00	3.747219e-01	7.251986e+00	6.018542e-13
raceId844	1.064661e+00	1.648713e-01	6.457527e+00	1.357638e-10
raceId845	-2.794940e+01	4.213689e-01	-6.633002e+01	0.000000e+00
raceId846	-9.177788e+01	1.544249e+00	-5.943205e+01	0.000000e+00
raceId847	-7.263865e+01	5.741895e+00	-1.265064e+01	3.090636e-35
raceId848	8.904665e-01	4.826403e-01	1.844990e+00	6.519935e-02
raceId849	2.101010e+01	2.194115e-01	9.575657e+01	0.000000e+00
raceId850	-8.969758e+00	3.584500e-01	-2.502373e+01	3.231651e-119

1-10 of 306 rows Previous **1** 2 3 4 5 6 ... 31 Next

```
glance(model2)
```

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
0.9997718	0.999738	0.3402734	29574.6	0	273	-575.0408	1700.082	3255.964	213.3936

1 row | 1-10 of 12 columns

This model reduces the r squared by a little bit. But the model is still complex.

Fitting Model 3: How about removing race id

```
# Fitting a linear model in all variables except race id to predict Fastest Lap Speed
model3 <- lm(fastestLapSpeed ~ circuitId + grid + finishingPosition +
                points + laps + raceDuration + rank + altitude + fastestLapDuration +
             averageLapDuration + averagePitstopDuration, data = f1_df)

tidy(model3)
```

term	estimate	std.error	statistic	p.value
<chr>	<dbl>	<dbl>	<dbl>	<dbl>
(Intercept)	5.342308e+02	2.521893e+00	2.118372e+02	0.000000e+00
circuitId2	6.860154e+00	2.635776e-01	2.602707e+01	5.274919e-129
circuitId3	1.730827e+00	2.201560e-01	7.861822e+00	6.112316e-15
circuitId4	-8.673097e+00	2.619902e-01	-3.310466e+01	2.582753e-192
circuitId5	2.158893e+00	3.494692e-01	6.177633e+00	7.849273e-10
circuitId6	-4.321983e+01	3.762589e-01	-1.148673e+02	0.000000e+00
circuitId7	-1.197246e+01	2.819337e-01	-4.246552e+01	4.577153e-282
circuitId9	1.117709e+01	2.180482e-01	5.125973e+01	0.000000e+00
circuitId10	-1.039156e+01	2.694709e-01	-3.856282e+01	3.365050e-244
circuitId11	-1.298076e+01	2.939708e-01	-4.415664e+01	1.483901e-298

1-10 of 101 rows Previous **1** 2 3 4 5 6 ... 11 Next

```
glance(model3)
```

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
0.9956394	0.9954253	1.421807	4651.811	0	99	-3697.707	7597.414	8168.847	4077.438

1 row | 1-10 of 12 columns

Has high R squared but still the model is complex to interpret.

Fitting model 4: Fitting a model without specifying race Id and circuit Id

```
# Fitting a linear model without specifying race Id and Circuit ID to predict Fastest Lap Speed
model4 <- lm(fastestLapSpeed ~ grid + finishingPosition +
                points + laps + raceDuration + rank + altitude + fastestLapDuration +
             averageLapDuration + averagePitstopDuration, data = f1_df)

tidy(model4)
```

term	estimate	std.error	statistic	p.value
<chr>	<dbl>	<dbl>	<dbl>	<dbl>
(Intercept)	6.142154e+02	6.573291e+00	93.44108787	0.000000e+00
grid1	4.480685e+00	3.219906e+00	1.39155771	1.642076e-01
grid2	4.112473e+00	3.207848e+00	1.28200357	1.999866e-01
grid3	4.189320e+00	3.200443e+00	1.30898149	1.906876e-01
grid4	4.181141e+00	3.200935e+00	1.30622475	1.916230e-01
grid5	4.253645e+00	3.201427e+00	1.32867143	1.841046e-01
grid6	4.596418e+00	3.203581e+00	1.43477503	1.515040e-01
grid7	4.811214e+00	3.205054e+00	1.50113382	1.334753e-01
grid8	4.475252e+00	3.215268e+00	1.39187514	1.641115e-01

term	estimate	std.error	statistic	p.value
<chr>	<dbl>	<dbl>	<dbl>	<dbl>
grid9	4.078153e+00	3.212590e+00	1.26942841	2.044327e-01

1-10 of 71 rows Previous 1 2 3 4 5 6 ... 8 Next

```
glance(model4)
```

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
0.9146567	0.9117369	6.245255	313.2538	0	70	-6845.751	13835.5	14242.86	79800.58

1 row | 1-10 of 12 columns

The resulting model has a significantly lower R squared than the previous 3 models, and is lesser complex 3 models. But we could do better.

Fitting model 5: Fitting a model that does not include any of the categorical variables: That is a model that does not include raceId, circuit Id, grid, finishingPosition, and rank.

```
# Fitting a linear model without any of the categorical variables predict Fastest Lap Speed
model5 <- lm(fastestLapSpeed ~ points + laps + raceDuration +  altitude + fastestLapDuration +
              averageLapDuration + averagePitstopDuration, data = f1_df)

tidy(model5)
```

term	estimate	std.error	statistic	p.value
<chr>	<dbl>	<dbl>	<dbl>	<dbl>
(Intercept)	6.197824e+02	4.111012e+00	150.7615266	0.000000e+00
points	-2.670302e-02	1.743921e-02	-1.5312059	1.258685e-01
laps	-3.400933e+00	5.098050e-02	-66.7104699	0.000000e+00
raceDuration	-3.276077e-06	4.506326e-07	-7.2699510	5.042177e-13
altitude	5.248524e-03	4.023939e-04	13.0432477	1.874199e-37
fastestLapDuration	-2.322844e-03	2.191553e-05	-105.9908019	0.000000e+00
averageLapDuration	1.999731e-04	3.037322e-05	6.5838618	5.768063e-11
averagePitstopDuration	-4.186296e-07	9.062293e-07	-0.4619467	6.441672e-01

8 rows

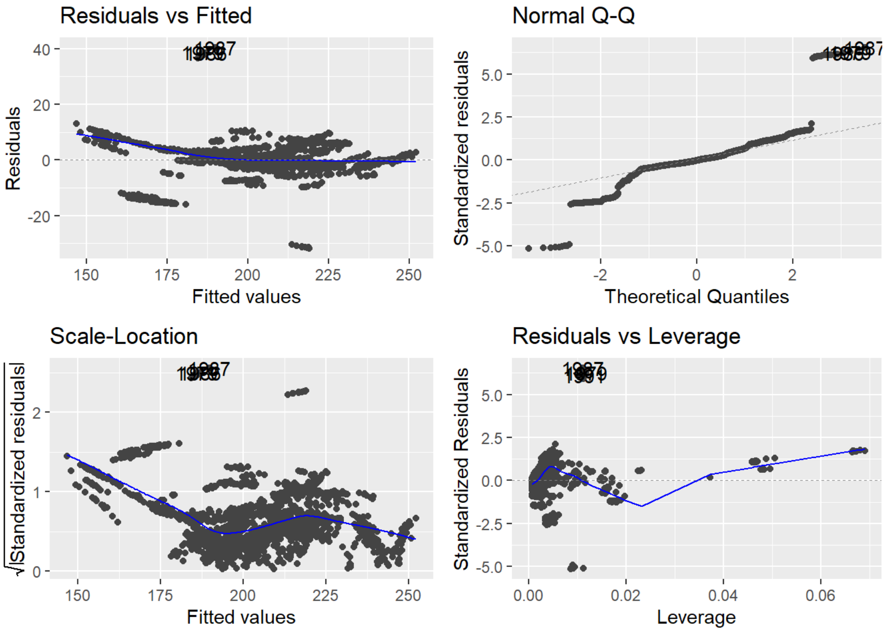
```
glance(model5)
```

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
0.9129958	0.9127071	6.210836	3161.602	0	7	-6866.153	13750.31	13801.23	81353.59

1 row | 1-10 of 12 columns

Model 5 does not include any categorical variables. The resulting structure is simple and easy to interpret and the R squared is high too at 0.9129958. This could be a good candidate for a final model. It explains 91.30% variation in fastestLapSpeed.

```
autoplot(model5)
```



No vedidence of Normality assumption being violated. ## Testing Multicolinearity

Checking multicolinearlity

```
vif(model5)
```

```
##                 points                   laps           raceDuration
##               1.027706              11.100297              15.220442
##               altitude     fastestLapDuration     averageLapDuration
##               1.103511               3.928247              20.455395
## averagePitstopDuration
##               1.486094
```

averageLapDuration, laps, and raceDuration have very high multi-colinearity since the VIF values are higher than 10. Creating a model without this variables. This indicates that the assumption of multicolinearlity is violated by these three variables,

Fitting model 6: A model that does not include variables with high mutlicolinearity from model 5.

```
# Fitting a linear model without any of the categorical variables predict Fastest Lap Speed
model6 <- lm(fastestLapSpeed ~ points + altitude + fastestLapDuration + averagePitstopDuration,
             data = f1_df)

tidy(model6)
```

term <chr>	estimate <dbl>	std.error <dbl>	statistic <dbl>	p.value <dbl>
(Intercept)	2.179206e+02	3.814285e+00	57.132758	0.000000e+00
points	9.421390e-02	5.803440e-02	1.623415	1.046500e-01
altitude	4.365306e-03	1.332458e-03	3.276130	1.069436e-03
fastestLapDuration	-1.717898e-04	3.878366e-05	-4.429437	9.930629e-06
averagePitstopDuration	9.677020e-06	2.485803e-06	3.892915	1.021202e-04

5 rows

```
glance(model6)
```

r.squared <dbl>	adj.r.squared <dbl>	sigma <dbl>	statistic <dbl>	p.value <dbl>	df <dbl>	logLik <dbl>	AIC <dbl>	BIC <dbl>	deviance <dbl>
0.0294426	0.02760442	20.72917	16.01728	6.32124e-13	4	-9419.164	18850.33	18884.28	907523.5

1 row | 1-10 of 12 columns

```
vif(model6)
```

```
##                points              altitude     fastestLapDuration
##              1.021698              1.086218               1.104406
## averagePitstopDuration
##              1.003783
```

The resulting model has a very very low R squared, which suggests that the removed variables make significant contributions to the model. I now consider re-introducing the removed variables with high VIF, one by one.

Fitting Model 7: Since the model has a very low r squared, I return "laps" variable and evaluate the resulting model

```
# Fitting a linear model
model7 <- lm(fastestLapSpeed ~ points + laps + altitude + fastestLapDuration + averagePitstopDuration,
             data = f1_df)

tidy(model7)
```

term <chr>	estimate <dbl>	std.error <dbl>	statistic <dbl>	p.value <dbl>
(Intercept)	6.394469e+02	3.139615e+00	203.670486	0.000000e+00
points	-2.995163e-02	1.761945e-02	-1.699918	8.929361e-02
laps	-3.722877e+00	2.577846e-02	-144.418115	0.000000e+00
altitude	5.543227e-03	4.041394e-04	13.716126	4.441421e-41
fastestLapDuration	-2.320132e-03	1.896333e-05	-122.348309	0.000000e+00
averagePitstopDuration	-8.517595e-07	7.573171e-07	-1.124706	2.608413e-01

6 rows

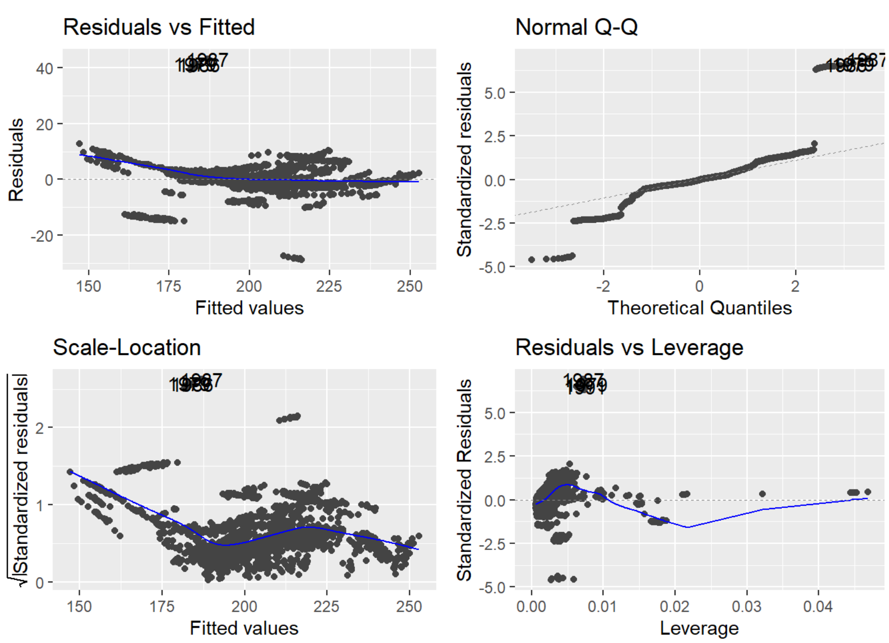
```
glance(model7)
```

r.squared <dbl>	adj.r.squared <dbl>	sigma <dbl>	statistic <dbl>	p.value <dbl>	df <dbl>	logLik <dbl>	AIC <dbl>	BIC <dbl>	deviance <dbl>
0.910794	0.9105827	6.285954	4310.666	0	5	-6892.607	13799.21	13838.82	83412.41

1 row | 1-10 of 12 columns

The r squared is high again, suggesting that laps is indeed an important variable.

```
autoplot(model7)
```



The residual plots look good and QQ plots look fairly good and there is no strong evidence that the assumptions of normality and linearlity are violated.

```
vif(model7)
```

```
##                points                  laps              altitude 
##              1.024137              2.770755              1.086661 
##     fastestLapDuration averagePitstopDuration 
##              2.871323              1.013172
```

The VIF values are less than 10, so this looks good.

Fitting Model 8: How about re-introducing averageLapDuration or raceDuration to see if the resulting model fits well, and if the VIF values are acceptable?

```
# Fitting a linear model
# reintroducing averageLapDuration
model8 <- lm(fastestLapSpeed ~ points + laps + altitude + fastestLapDuration + averagePitstopDuration + averageLapDuration,
             data = f1_df)

tidy(model8)
```

term <chr>	estimate <dbl>	std.error <dbl>	statistic <dbl>	p.value <dbl>
(Intercept)	6.393842e+02	3.141214e+00	203.5468581	0.000000e+00
points	-3.058501e-02	1.764391e-02	-1.7334598	8.316006e-02
laps	-3.721806e+00	2.582524e-02	-144.1150678	0.000000e+00
altitude	5.519486e-03	4.055573e-04	13.6096339	1.705723e-40
fastestLapDuration	-2.312003e-03	2.213176e-05	-104.4653977	0.000000e+00
averagePitstopDuration	-4.830822e-07	9.172529e-07	-0.5266620	5.984837e-01
averageLapDuration	-7.499093e-06	1.052394e-05	-0.7125746	4.761878e-01

7 rows

```
glance(model8)
```

r.squared <dbl>	adj.r.squared <dbl>	sigma <dbl>	statistic <dbl>	p.value <dbl>	df <dbl>	logLik <dbl>	AIC <dbl>	BIC <dbl>	deviance <dbl>
0.9108155	0.9105619	6.286687	3591.469	0	6	-6892.352	13800.7	13845.97	83392.34

1 row | 1-10 of 12 columns

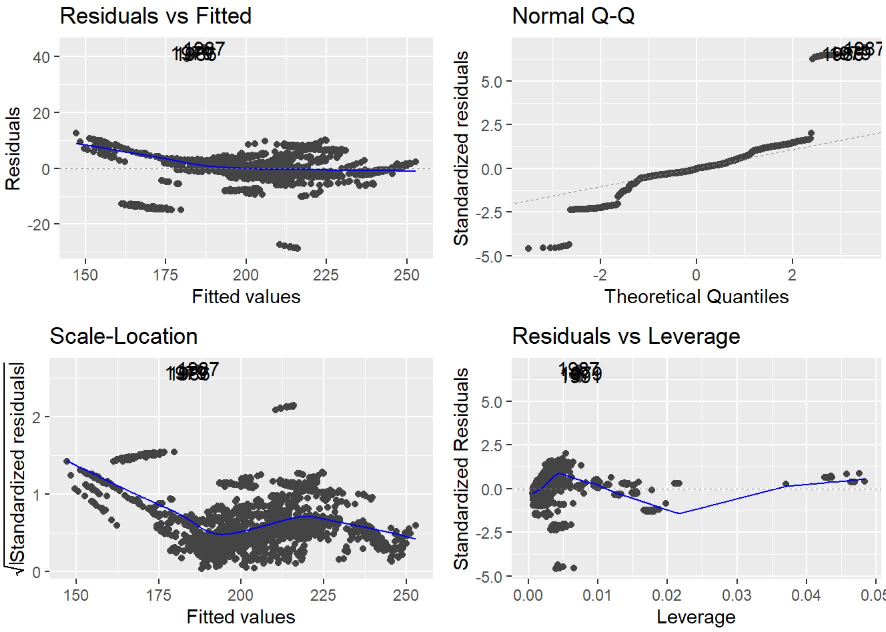
The r squared is significantly high at 0.9108155.

```
vif(model8)
```

```
##                points                  laps              altitude 
##              1.026742              2.780171              1.094044 
##     fastestLapDuration averagePitstopDuration     averageLapDuration 
##              3.910059              1.485952              2.396842
```

VIF values are all less than 10 which is strong evidence that multicolinearity assumption is not violated.

```
autoplot(model8)
```



From the plot above, linearity seems to hold reasonably well in the Residual vs Fitted plot, since the blue solid line closely follows the dashes line. Also, from the QQ plot, the data is normal distributed since most of the data points lie close the dashed line.

I settle on model 8 as the best model to predict “fastestLapSpeed”. The final model from this analysis shows that “fastestLapSpeed” can be predicted using “points”, “lap”, “altitude”, “fastestLapDuration”, “averagePitstopDuration” and “averageLapDuration”. The resuylting model has an r squared value of 0.9108155 which is significantly highy, showing that there is a strong relationship between the response variable with the predictor variables. The model can explain 91.08% of the variation in “fastestLapSpeed” using the 6 predictor variables.