Simple Regression Analysis VMT vs HGF

Ernesto Garcia

3/31/2021

FDOT VMT DVMT vs HFG. EDA and Regression Analysis

Explanation of terms:

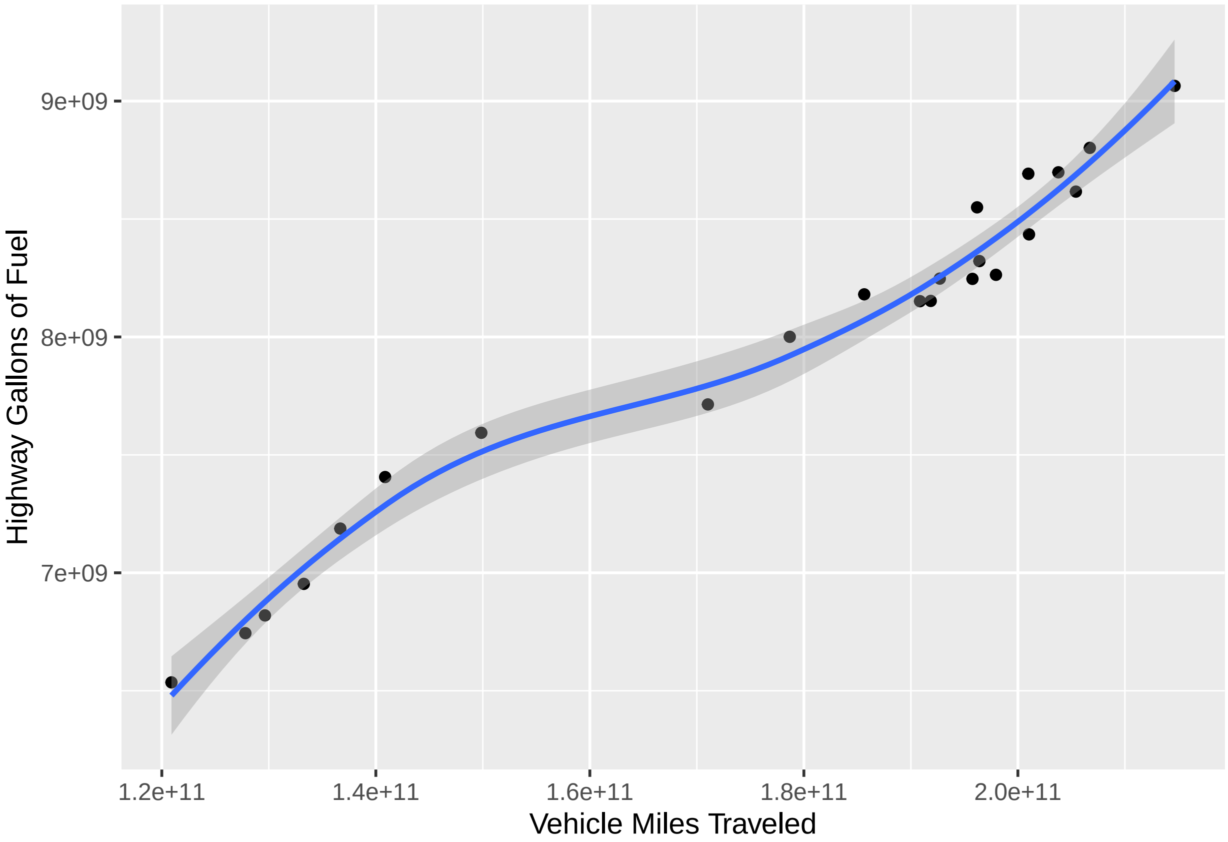
- VMT=Vehicle Miles Traveled
- DVMT= Daily Vehicle Miles Traveled
- HGF= Highway Gallons of Fuel comsumed

```
dataset=read.csv('VMTvsHGF.csv')
```

Creating Scatter Plot between VMT (x) and HGF (y)

```
library(ggplot2)
scatter<-ggplot(dataset, aes(VMT, HGF))
scatter+geom_point()+labs(x="Vehicle Miles Traveled",
                          y="Highway Gallons of Fuel")+geom_smooth()
```

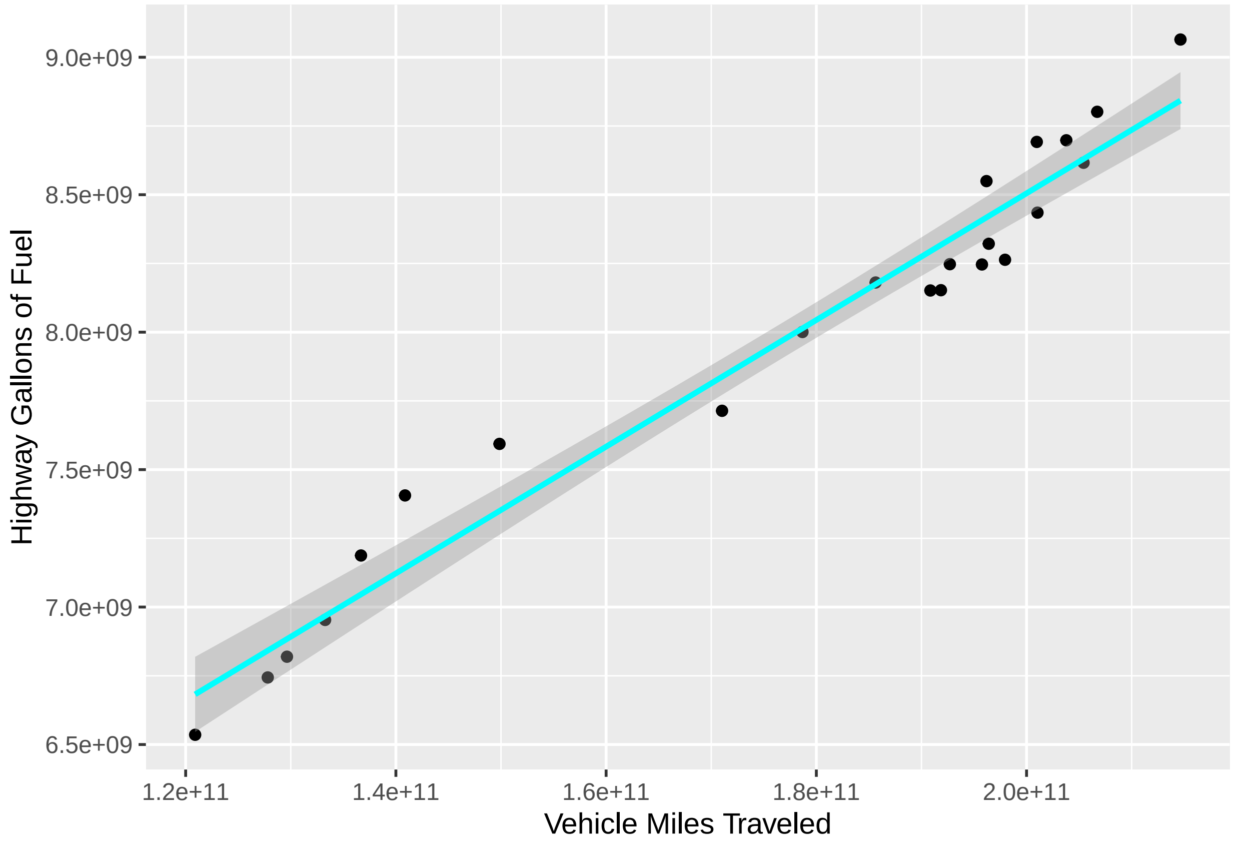
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Changing Method to create a "straight line" with 95% confidence interval

```
scatter+geom_point()+labs(x="Vehicle Miles Traveled", y="Highway Gallons of Fuel")+
  geom_smooth(method="lm", colour="Cyan")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



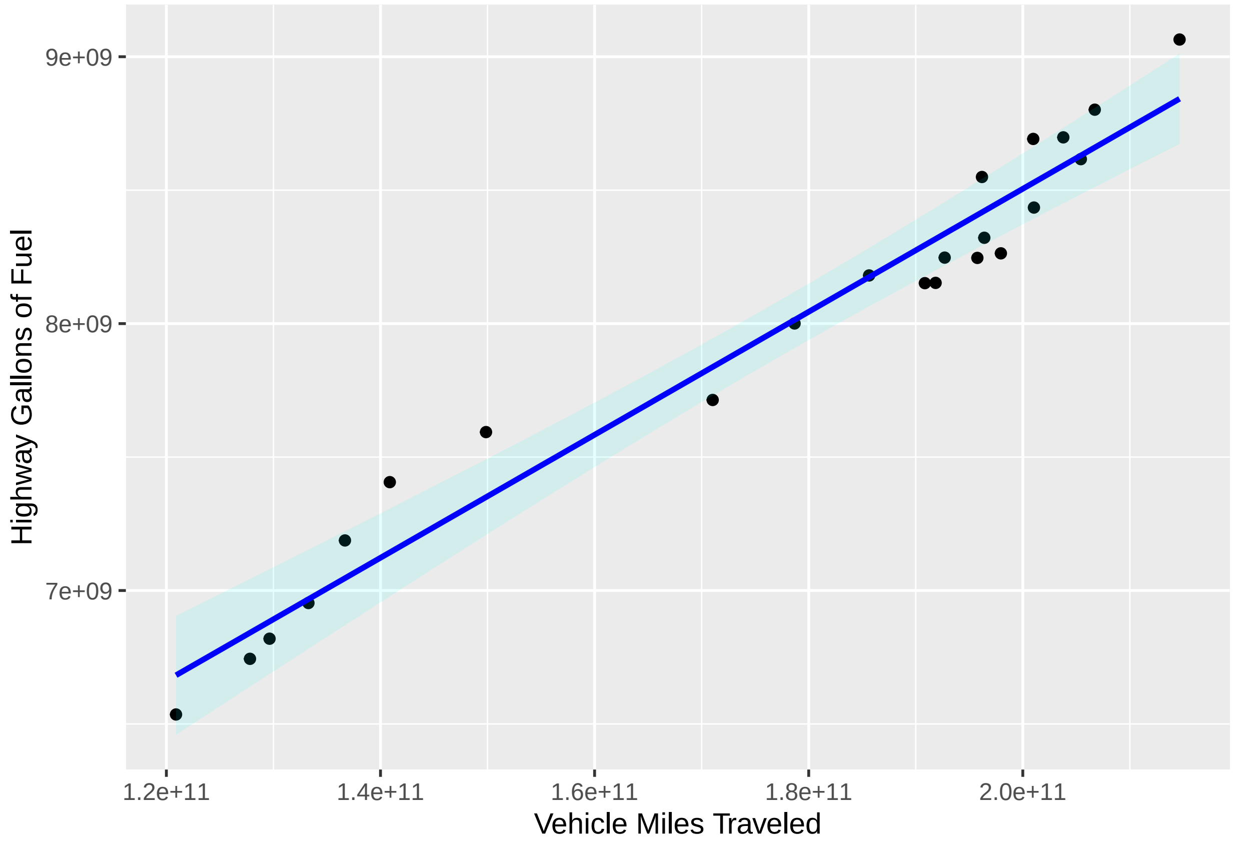
Robust linear model used for regression line:

Robust regression is an alternative to least squares regression when data are contaminated with outliers or influential observations, and it can also be used for the purpose of detecting influential observations.

Note: The confidence interval is set to 99%

```
library(MASS)
#Robust regression with 99% confidence interval
scatter+geom_point()+labs(x="Vehicle Miles Traveled", y="Highway Gallons of Fuel")+
  geom_smooth(method="rlm", colour="Blue", level=0.99, alpha=0.1, fill="Cyan")
```

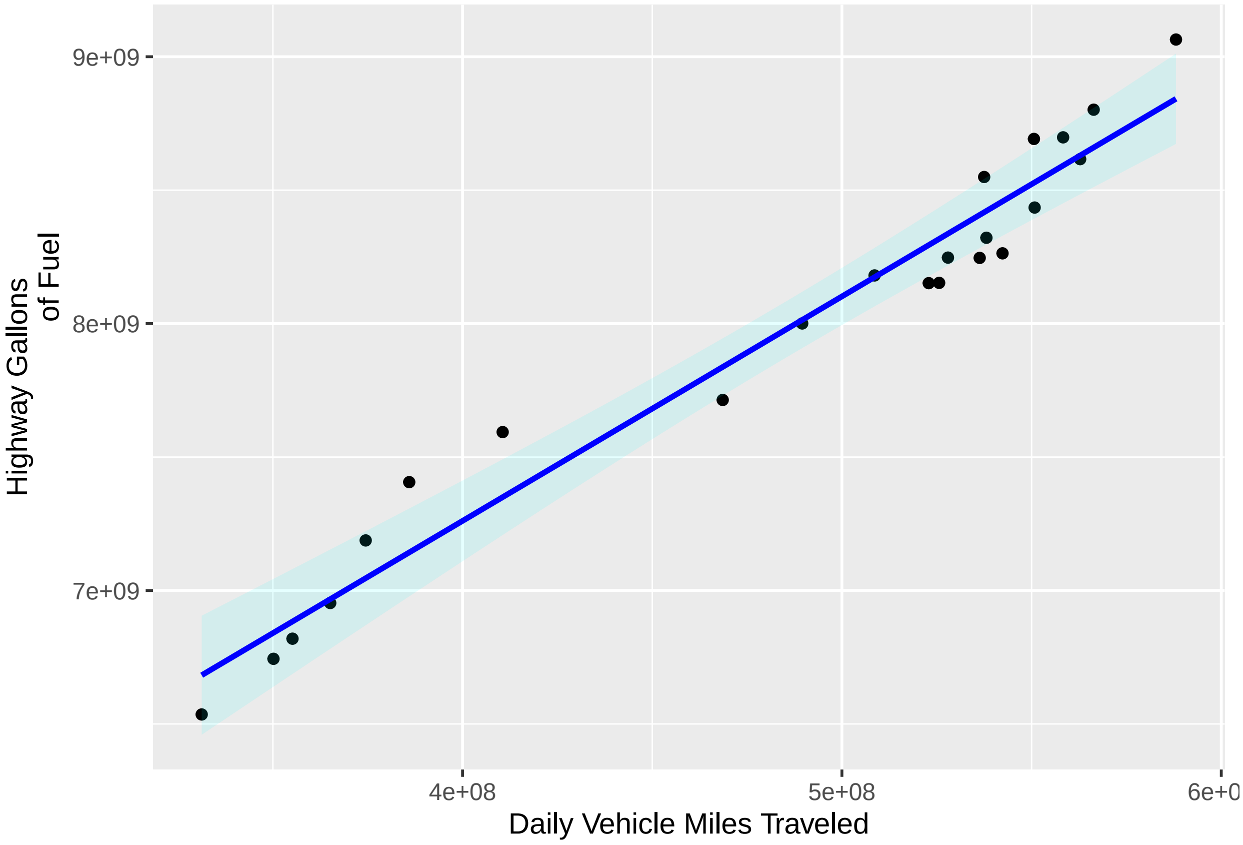
```
## `geom_smooth()` using formula 'y ~ x'
```



Scatter plot of DVMT vs HGF

```
#Robust regression with 99% confidence interval
scatter2<-ggplot(dataset, aes(DVMT, HGF))
scatter2+geom_point()+labs(x="Daily Vehicle Miles Traveled", y="Highway Gallons
                           of Fuel")+geom_smooth(method="rlm", colour="Blue",
                           level=0.99, alpha=0.1, fill="Cyan")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Fitting linear Regression for VMT vs HGF

```
regressor<-lm(formula=HGF~ VMT, data<-dataset)
#Summary table
summary(regressor)
```

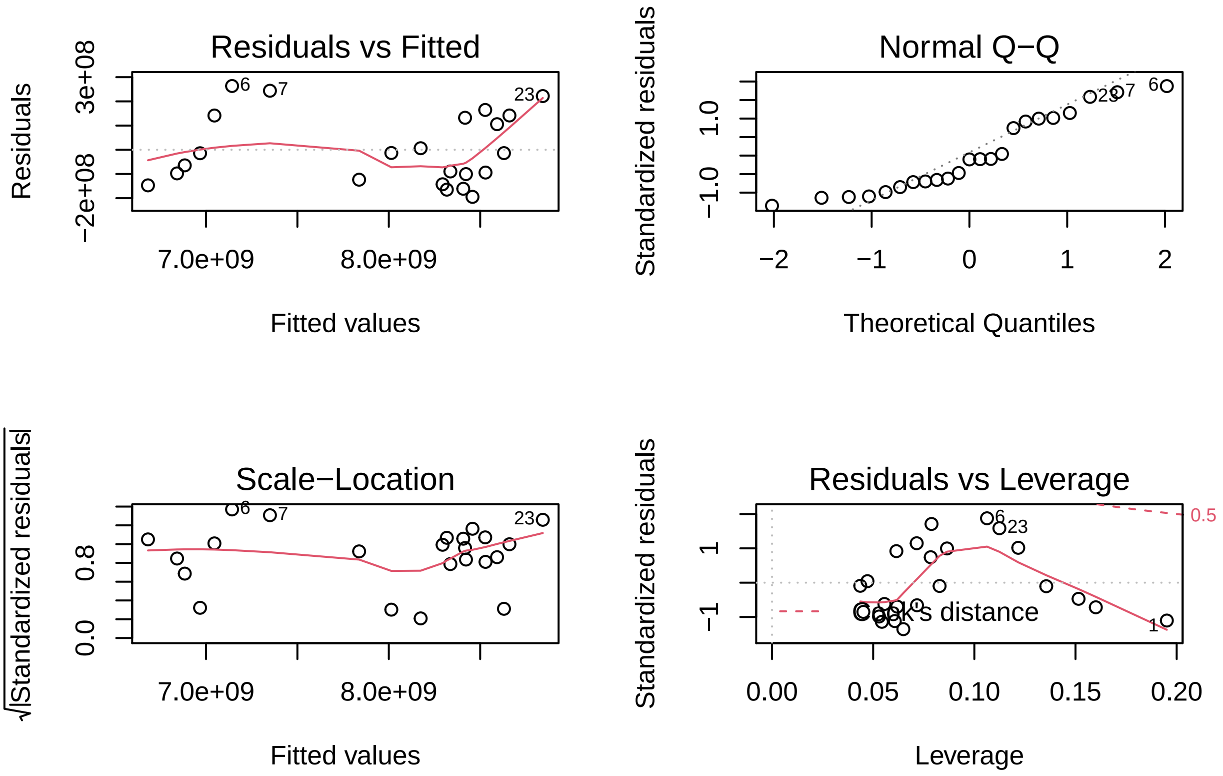
```
##
## Call:
## lm(formula = HGF ~ VMT, data = data <- dataset)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -194551021 -112049952  -14316203  136858930  263447125
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.897e+09  1.854e+08   21.02 1.38e-15 ***
## VMT         2.304e-02  1.033e-03   22.30 4.20e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 148600000 on 21 degrees of freedom
## Multiple R-squared:  0.9595, Adjusted R-squared:  0.9576
## F-statistic: 497.4 on 1 and 21 DF,  p-value: 4.198e-16
```

Residual and normality analysis

Residual Analysis:

- Residuals vs Fitted Values (Predicted)
- Normal Q-Q plot: Theoretical Quantile vs Standardized Residuals
- Scale Location: Fitted values vs Square Roots of the Standardized residuals
- Residuals vs Leverage: Leverage vs Standardized Residuals

```
par(mfrow=c(2, 2)) # Returns a window with four graphs side by side
plot(regressor)
```



Note: The lm() model returns a list containing a lot of useful information.

You can access them with the regressor object you have created, followed by the $ sign and the information you want to extract.

- coefficients: `regressor$coefficients`
- residuals: `regressor$residuals`
- fitted value: `regressor$fitted.values`

```
regressor$coefficients
```

```
##  (Intercept)          VMT 
## 3.896792e+09 2.304147e-02
```

```
regressor$residuals
```

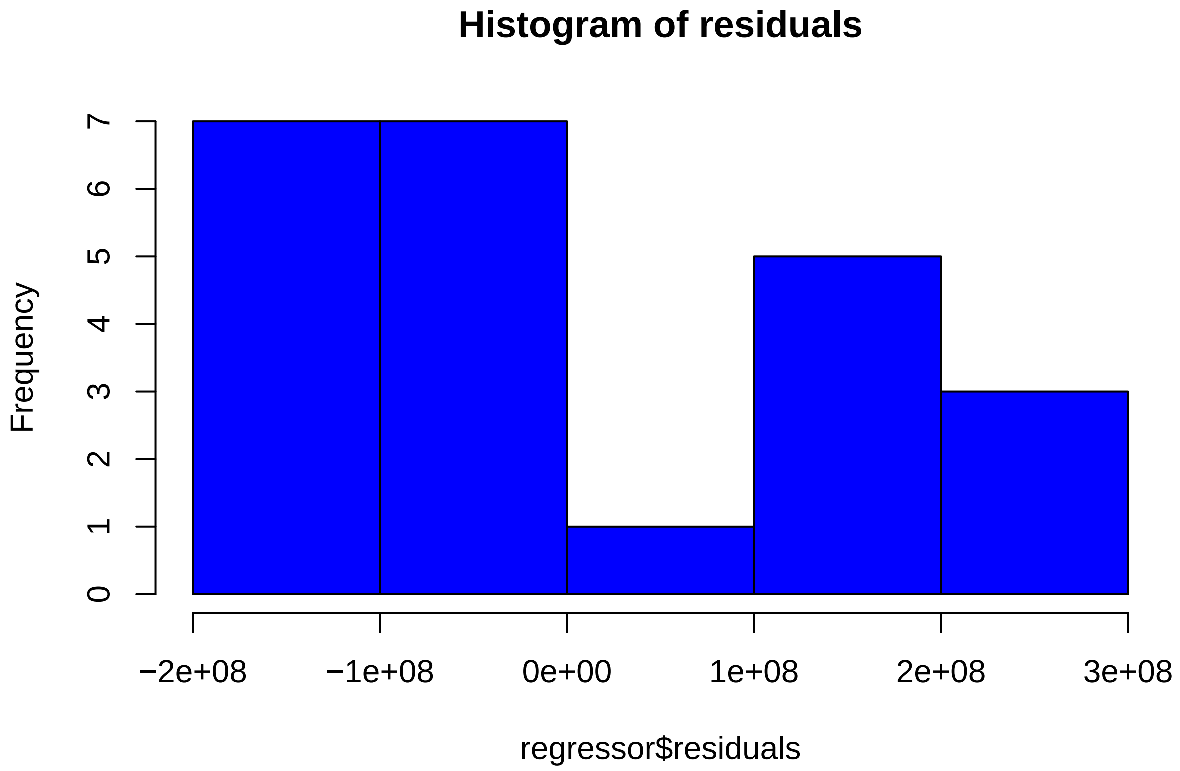
```
##          1          2          3          4          5          6          7 
## -147046058  -97623911  -64387271  -14316203  141484335  263447125  243894428 
##          8          9         10         11         12         13         14 
## -123588713  -13261915    6342688  132233525  164530664  105741368  -13778174 
##         15         16         17         18         19         20         21 
## -194551021 -100511192 -161081024 -164711383 -142618938  -89612951  -94193197
##         22         23
##  141646212  221961602
```

```
regressor$fitted.values
```

```
##          1          2          3          4          5          6          7
## 6682542058 6841685911 6883824271 6967478203 7046118665 7142603875 7349725572
##          8          9         10         11         12         13         14
## 7837560713 8013858915 8174254312 8417193475 8527536336 8592242632 8630001174
##         15         16         17         18         19         20         21
## 8457899014 8422194702 8407279267 8317413028 8294277938 8336925951 8529059319
##         22         23
## 8659959286 8842481398
```

Histogram of residuals

```
par(mfrow=c(1, 1))
hist(regressor$residuals, col="blue", main="Histogram of residuals")
```


Histogram of residuals
Frequency
−2e+08
−1e+08
0e+00
1e+08
2e+08
3e+08
regressor$residuals