Complete Example: Cars2010

Class of 20213

EDA

- Explore variables univariately
- Anything that looks unusual?
- Missing values?
- Data types

EDA

```
> summary(cars2010)
    EngDispl         NumCyl        Transmission   FE                AirAspirationMethod
 Min.   :1.000   Min.   : 2.000   S6     :213     Min.   :17.50   NaturallyAspirated:921
 1st Qu.:2.400   1st Qu.: 4.000   M6     :167     1st Qu.:29.09   Supercharged      : 18
 Median :3.500   Median : 6.000   A4     :143     Median :34.51   Turbocharged      :168
 Mean   :3.507   Mean   : 5.971   A6     :126     Mean   :34.71
 3rd Qu.:4.300   3rd Qu.: 8.000   A5     :114     3rd Qu.:39.20
 Max.   :8.400   Max.   :16.000   M5     :101     Max.   :69.64
                                   (Other):243

 NumGears      TransLockup        TransCreeperGear                   DriveDesc
 Min.   :1.000   Min.   :0.0000    Min.   :0.00000    AllWheelDrive         :205
 1st Qu.:5.000   1st Qu.:0.0000    1st Qu.:0.00000    FourWheelDrive        :159
 Median :6.000   Median :1.0000    Median :0.00000    ParttimeFourWheelDrive: 11
 Mean   :5.268   Mean   :0.6802    Mean   :0.04878    TwoWheelDriveFront    :382
 3rd Qu.:6.000   3rd Qu.:1.0000    3rd Qu.:0.00000    TwoWheelDriveRear     :350
 Max.   :8.000   Max.   :1.0000    Max.   :1.00000
```

Multicollinearity

- See if multicollinearity is an issue
- If so, how do you want to deal with it?

```
> cor(cars2010[,c(1,2,4,6,7,8,10,11,13,14)])
                       EngDispl      NumCyl  FE            NumGears
EngDispl             1.00000000  0.906260027 -0.78739383  0.211730489
NumCyl               0.90626003  1.000000000 -0.74021798  0.288711440
FE                  -0.78739383 -0.740217981  1.00000000 -0.211284876
NumGears             0.21173049  0.288711440 -0.21128488  1.000000000
TransLockup          0.22839513  0.208771908 -0.27193887  0.001353611
TransCreeperGear     0.02666562  0.025520828 -0.06962168  0.043595219
IntakeValvePerCyl   -0.42235745 -0.248509452  0.28034403  0.177960634
ExhaustValvesPerCyl -0.47843804 -0.339851831  0.33565285  0.152819250
VarValveTiming      -0.06825603  0.005399291  0.12495278  0.090839722
VarValveLift        -0.08657142 -0.059461008  0.09621127  0.130719422

                     TransLockup IntakeValvePerCyl
EngDispl             0.228395128 -0.42235745
NumCyl               0.208771908 -0.24850945
FE                  -0.271938867  0.28034403
NumGears             0.001353611  0.17796063
TransLockup          1.000000000 -0.13132599
TransCreeperGear     0.092328478 -0.07767916
IntakeValvePerCyl   -0.131325993  1.00000000
ExhaustValvesPerCyl -0.158326003  0.91148782
VarValveTiming      -0.094772029  0.24082398
VarValveLift        -0.097809395  0.15485588
```

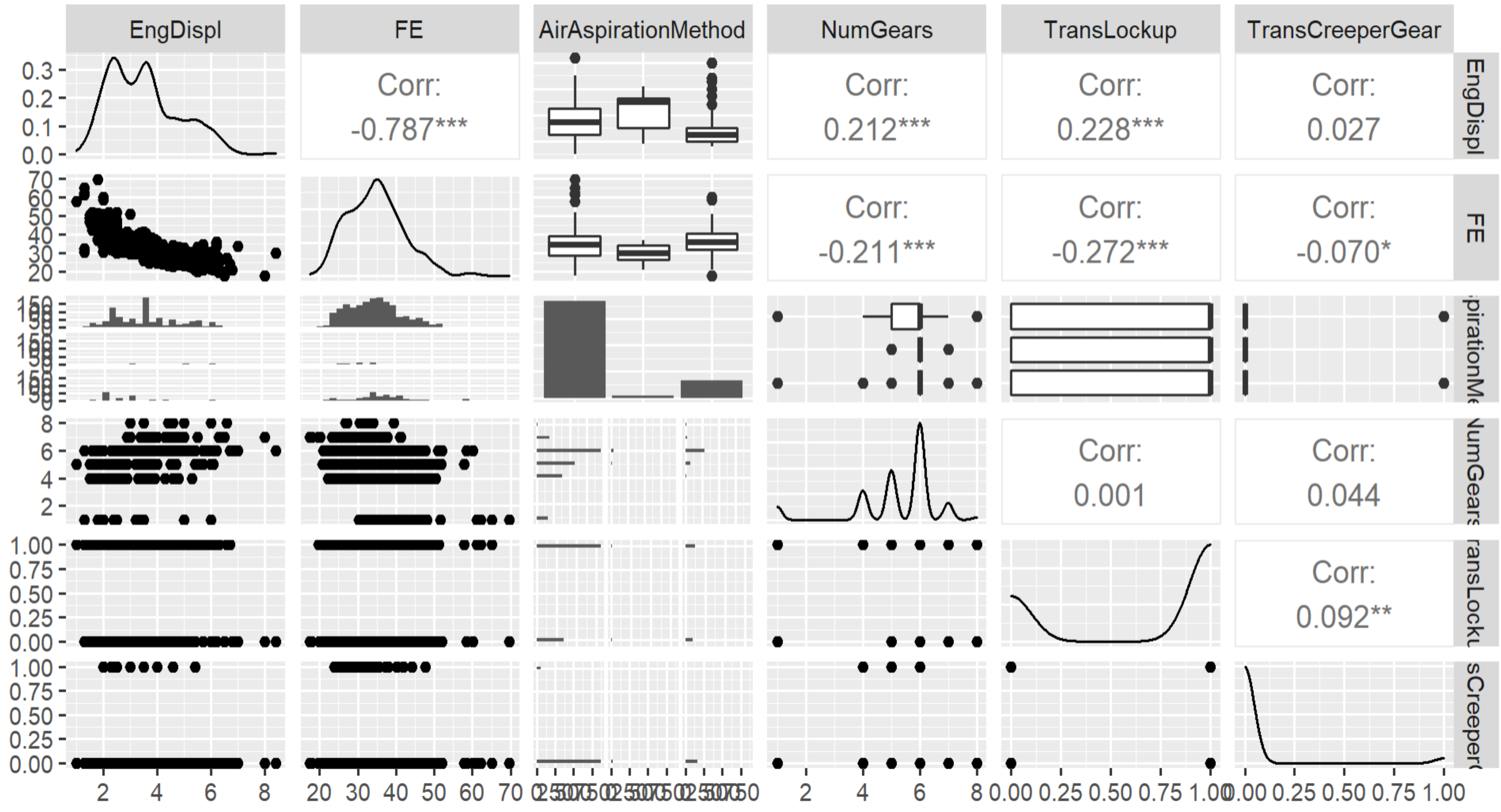
Going to remove NumCyl and IntakeValvePerCyl

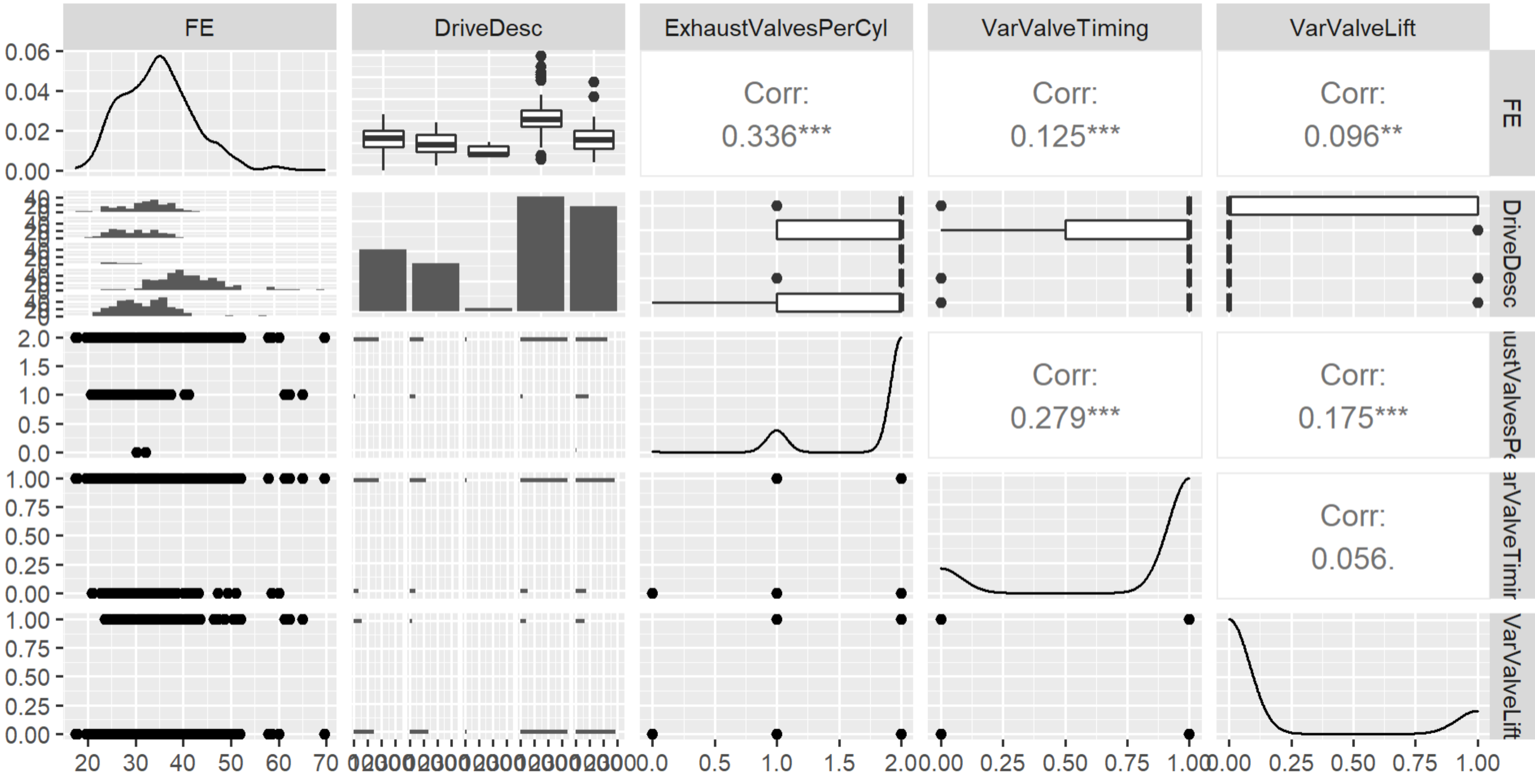
```
> cars2010.1=cars2010[,-c(2,10)]
> collin.test=lm(FE~.,data=cars2010.1)
> vif(collin.test)
                          GVIF  Df  GVIF^(1/(2*Df))
EngDispl              2.492719   1  1.578835
Transmission        327.630688  15  1.212959
AirAspirationMethod   1.442853   2  1.095987
NumGears             26.097874   1  5.108608
TransLockup           3.015590   1  1.736545
TransCreeperGear      1.210922   1  1.100419
DriveDesc             9.381876   4  1.322928
ExhaustValvesPerCyl   2.085180   1  1.444015
CarlineClassDesc     22.735388  16  1.102547
VarValveTiming        1.339877   1  1.157531
VarValveLift          1.416643   1  1.190228
```

```
> table(cars2010$Transmission,cars2010$NumGears)

         1   4   5   6   7   8
  Other   2   0   2   0   2   0
  A4      0 143   0   0   0   0
  A5      0   0 114   0   0   0
  A6      0   0   0 126   0   0
  A7      0   0   0   0  59   0
  AM6     0   0   0  11   0   0
  AM7     0   0   0   0   5   0
  AV     54   0   0   0   0   1
  AVS6   13   0   0   0   0   0
  M5      0   0 101   0   0   0
  M6      0   0   0 167   0   0
  S4      0  13   0   0   0   0
  S5      0   0  48   0   0   0
  S6      0   0   0 213   0   0
  S7      0   0   0   0  22   0
  S8      0   0   0   0   0  11
```

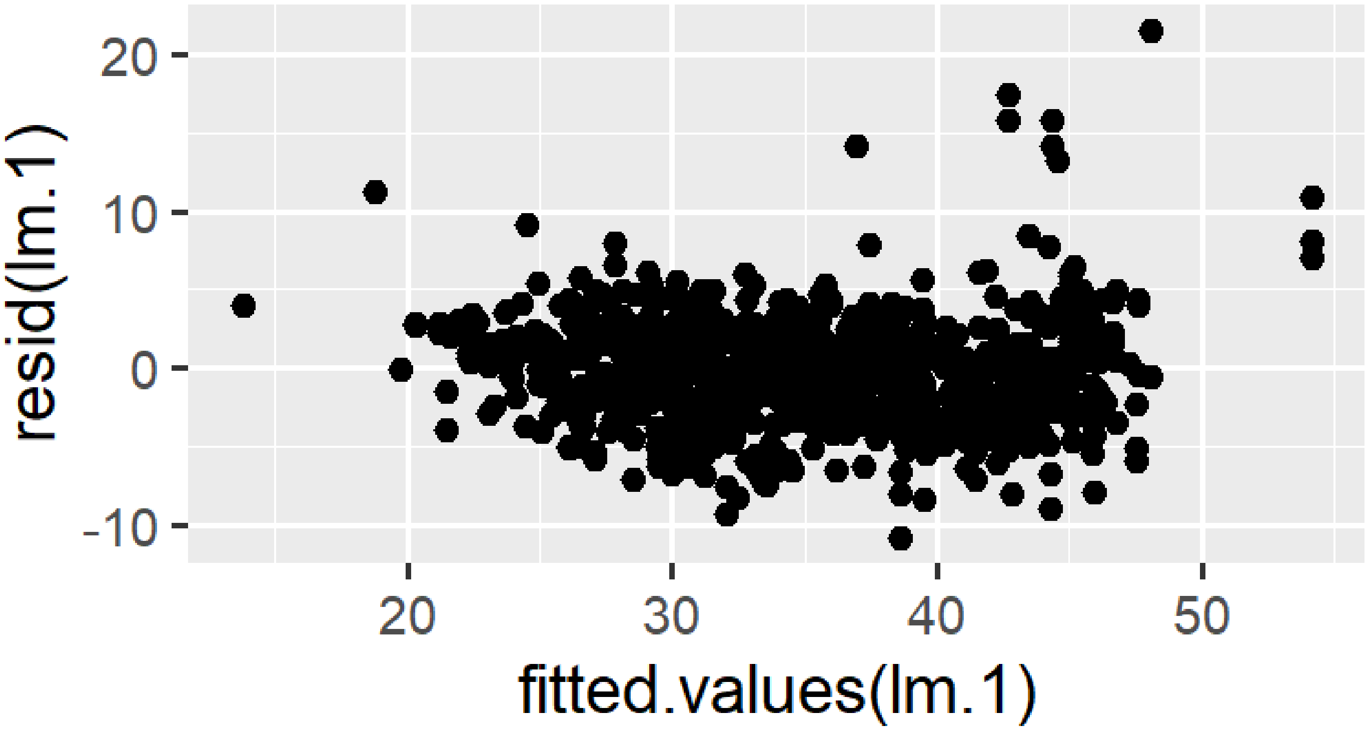
Going to also get rid of Transmission!!


EngDispl
FE
AirAspirationMethod
NumGears
TransLockup
TransCreeperGear
Corr:
-0.787***
Corr:
0.212***
Corr:
0.228***
Corr:
0.027
Corr:
-0.211***
Corr:
-0.272***
Corr:
-0.070*
Corr:
0.001
Corr:
0.044
Corr:
0.092**
EngDispl
FE
pirationMe
NumGears
ransLocku
sCreeperG
0.3
0.2
0.1
0.0
70
60
50
40
30
20
8
6
4
2
1.00
0.75
0.50
0.25
0.00
2
4
6
8
20
30
40
50
60
70
2
4
6
8
0.00
0.25
0.50
0.75
1.00

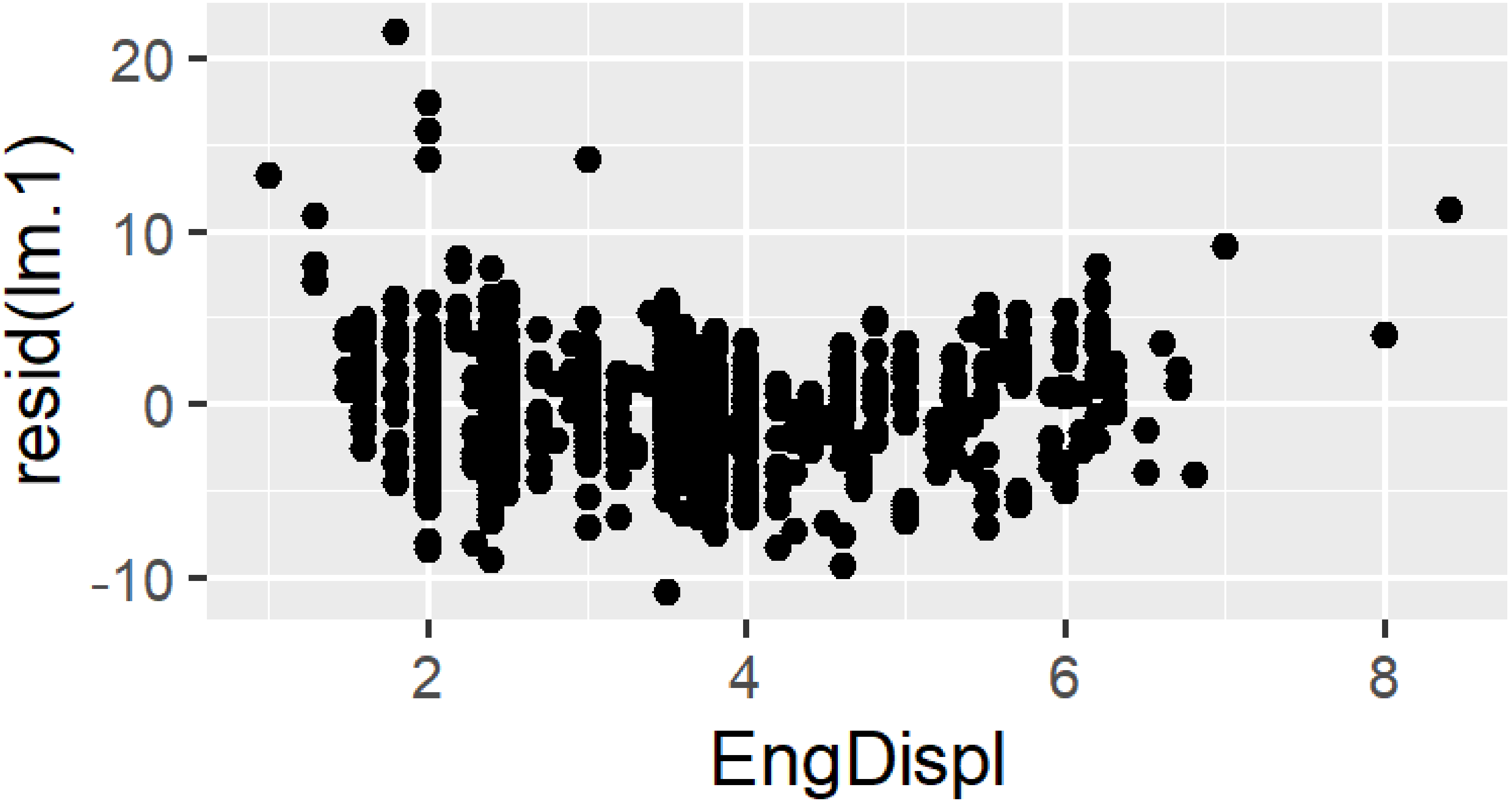

FE
DriveDesc
ExhaustValvesPerCyl
VarValveTiming
VarValveLift
FE
DriveDesc
ustValvesPe
arValveTimir
VarValveLift
Corr:
0.336***
Corr:
0.125***
Corr:
0.096**
Corr:
0.279***
Corr:
0.175***
Corr:
0.056.
0.06
0.04
0.02
0.00
40
20
0
2.0
1.5
1.0
0.5
0.0
1.00
0.75
0.50
0.25
0.00
1.00
0.75
0.50
0.25
0.00
20
30
40
50
60
70
0.0
0.5
1.0
1.5
2.0
0.00
0.25
0.50
0.75
1.00
0.00
0.25
0.50
0.75
1.00

Also, observations 1279 and 1280 have 0 intake AND 0 exhaust valves per cylinder (recording error).
Going to remove these two observations for the analysis. Also going to make NumGears a factor too.
Let’s try some automated search algorithms….

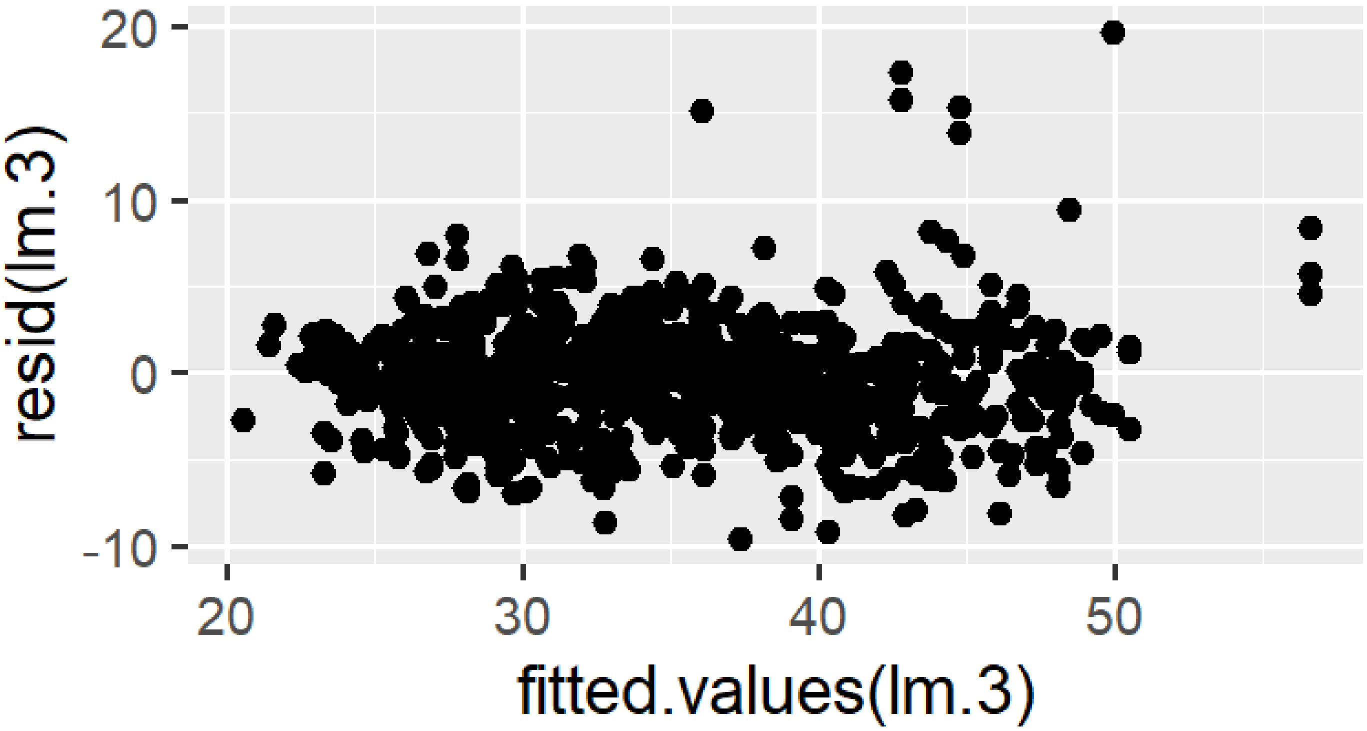
```
>
lm.1=lm(FE~EngDispl+CarlineClassDesc+DriveDesc+ExhaustValvesPerCyl+NumGears+TransCreeperGear+AirAspirati
onMethod,data=cars2010.3)
> ggplot(lm.1,aes(x=fitted.values(lm.1),y=resid(lm.1)))+geom_point()
```

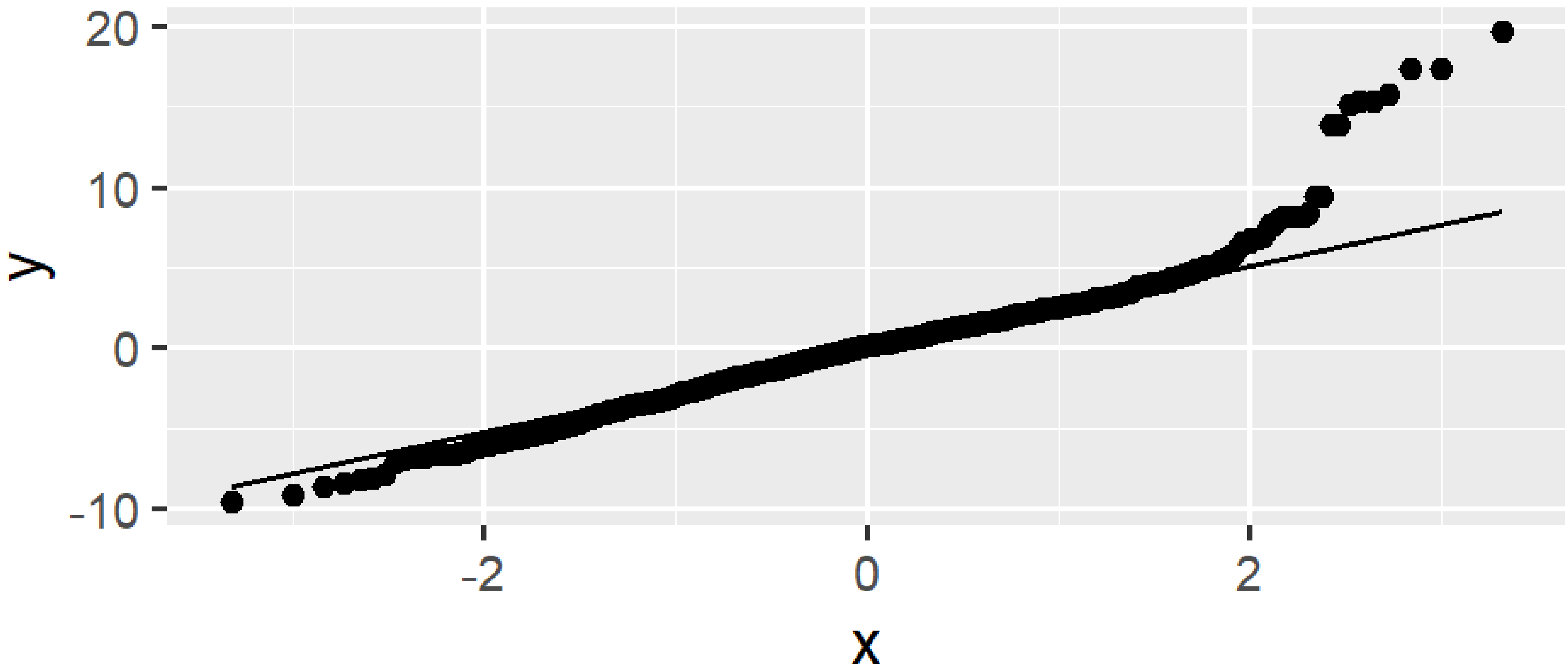


```
ggplot(lm.1,aes(x=EngDispl,y=resid(lm.1)))+geom_point()
```

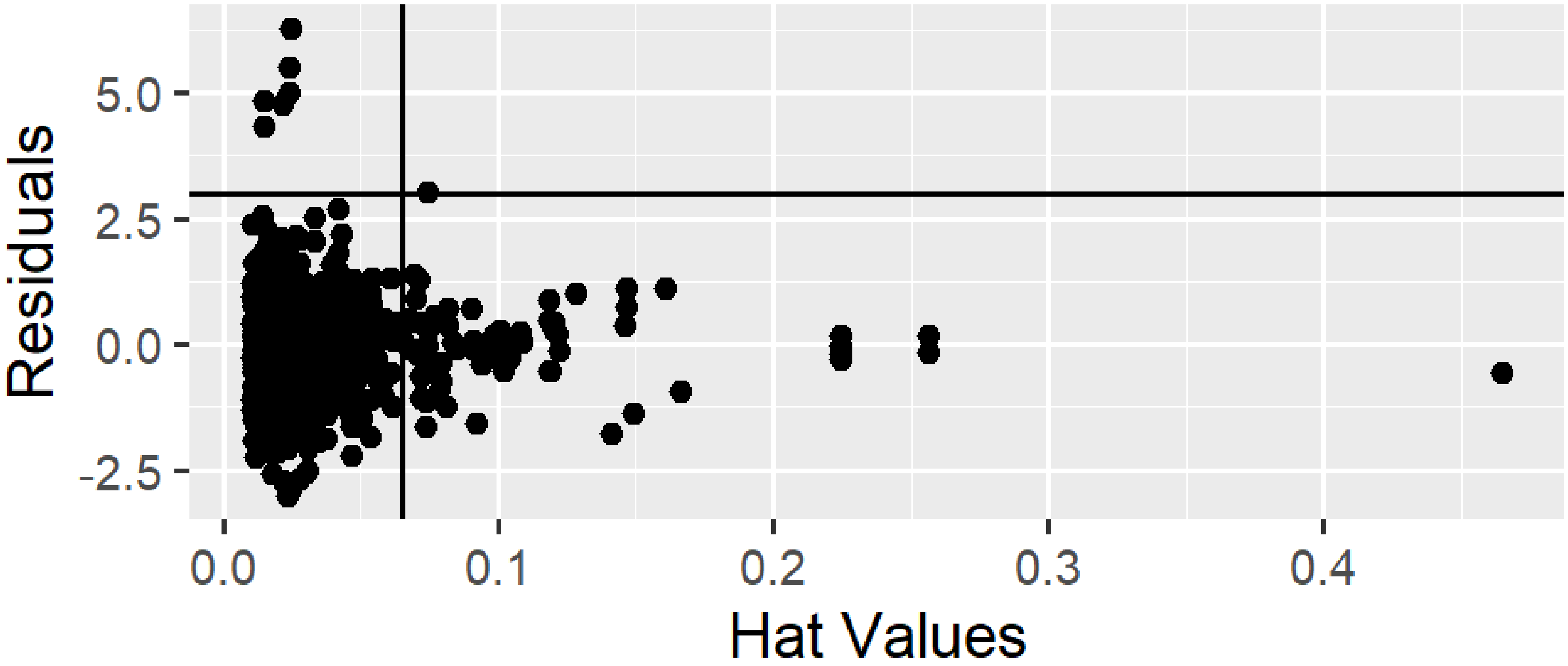


```
> m.engdispl=mean(cars2010.3$EngDispl)
> cars2010.3$c.EngDispl=cars2010.3$EngDispl-m.engdispl
>
lm.4=lm(FE~c.EngDispl+I(c.EngDispl^2)+CarlineClassDesc+DriveDesc+ExhaustValvesPerC
yl+NumGears+TransCreeperGear+AirAspirationMethod+DriveDesc:c.EngDispl +
c.EngDispl:NumGears,data=cars2010.3)
```




20
10
0
-10
y
-2
0
2
x

```
ggplot(lm.4,aes(x=hatvalues(lm.4),y=rstudent(lm.4)))+geom_po
int()+geom_hline(yintercept=3)+geom_vline(xintercept=0.065)
+labs(x="Hat Values",y="Residuals")
```



```
> cars2011.1=cars2011[,-c(2,3,10)]
> cars2011.1$c.EngDispl=cars2011.1$EngDispl-m.engdispl
> cars2011.1$n.index=seq(1,nrow(cars2011.1))
> cars2011.1$NumGears=as.factor(cars2011.1$NumGears)
> valid.fit=predict(lm.4,newdata = cars2011.1)
```

```
> MAE=mean(abs(cars2011.1$FE-valid.fit))
> MAE
[1] 2.678234
```