Solo Project

Abraham Hussein

October 14, 2019

Data Wrangling

This is Abraham's Solo project. My question is can I model the combined metric (city and highway) fuel economy of a subset of vehicles from 2000 - 2013, based on factors such as transmission, manufacturer, etc.? Our data is a sample which does not reflect electric vehicles (EVs), as some of the emissions being monitored (CO_2 and CO) are not emitted from EVs.

```
fueldata<-read.csv("C:/Users//Abraham Hussein//Documents//STAT 429//Solo Dolo Project//vroom.csv", sep =
sum(is.na(fueldata$engine_capacity))
```

```
## [1] 7
```

```
fueldata[1:3,]
```

```
##                     file year manufacturer     model        description
## 1 DatapartC_july2000.csv 2000   Alfa Romeo 145 Range 1.6 Twin Spark 16v
## 2 DatapartC_july2000.csv 2000   Alfa Romeo 145 Range 1.8 Twin Spark 16v
## 3 DatapartC_july2000.csv 2000   Alfa Romeo 145 Range         Cloverleaf
##   euro_standard tax_band transmission transmission_type engine_capacity
## 1             2                   M5                 M            1598
## 2             2                   M5                 M            1747
## 3             2                   M5                 M            1970
##   fuel_type urban_metric extra_urban_metric combined_metric urban_imperial
## 1    Petrol         11.1                6.5             8.2           25.4
## 2    Petrol         11.5                6.6             8.4           24.6
## 3    Petrol         12.4                6.6             8.8           22.8
##   extra_urban_imperial combined_imperial noise_level co2 thc_emissions
## 1                 43.5              34.4          74 195            NA
## 2                 42.8              33.6          74 200            NA
## 3                 42.8              32.1          74 210            NA
##   co_emissions nox_emissions thc_nox_emissions particulates_emissions
## 1          980            NA                NA
## 2         1105            NA                NA
## 3         1103            NA                NA
##   fuel_cost_12000_miles fuel_cost_6000_miles standard_12_months
## 1                    NA                  618                 NA
## 2                    NA                  633                 NA
## 3                    NA                  663                 NA
##   standard_6_months first_year_12_months first_year_6_months
## 1                NA                   NA                  NA
## 2                NA                   NA                  NA
## 3                NA                   NA                  NA
##   date_of_change
## 1
## 2
## 3
```

```r
#Establishing Brands to inspect (Cat variable)
levels(fueldata$manufacturer)
```

```
##  [1] "Abarth"                "Alfa Romeo"            "Aston Martin Lagonda"
##  [4] "Audi"                  "Bentley Motors"        "BMW"
##  [7] "Cadillac"              "Chevrolet"             "Chrysler Jeep"
## [10] "Citroen"               "Corvette"              "Dacia"
## [13] "Daewoo Cars"           "Daihatsu"              "Dodge"
## [16] "Ferrari"               "Fiat"                  "Ford"
## [19] "Honda"                 "Hummer"                "Hyundai"
## [22] "Infiniti"              "Isuzu"                 "Jaguar Cars"
## [25] "Kia"                   "Lamborghini"           "Land Rover"
## [28] "Lexus"                 "Lotus"                 "LTI"
## [31] "Maserati"              "Mazda"                 "McLaren"
## [34] "Mercedes-Benz"         "Metrocab"              "MG Motors Uk"
## [37] "MG Motors UK"          "MG Rover Group"        "Micro Compact Car"
## [40] "Mini"                  "Mitsubishi"            "Morgan Motor Company"
## [43] "Nissan"                "Perodua"               "Peugeot"
## [46] "Porsche"               "Proton"                "Renault"
## [49] "Rolls-Royce"           "Rover Group Limited"   "Saab"
## [52] "Seat"                  "Skoda"                 "Smart"
## [55] "SsangYong"             "Subaru"                "Suzuki"
## [58] "Tata"                  "Toyota"                "Vauxhall"
## [61] "Volkswagen"            "Volvo"
```

```r
#Choosing 10 random 'Makes':
set.seed(429) #For Consistency
sample(levels(fueldata$manufacturer),10,replace = F)
```

```
##  [1] "Nissan"    "Audi"      "Metrocab"  "Tata"      "BMW"
##  [6] "Toyota"    "SsangYong" "Mini"      "Cadillac"  "Fiat"
```

```r
#Creating an indicator if the car is of the sampled make.
indicat <- ifelse(
  fueldata$manufacturer=="Nissan"|fueldata$manufacturer=="Audi"|
  fueldata$manufacturer=="Metrocab"|fueldata$manufacturer=="Tata"|
  fueldata$manufacturer=="BMW"|fueldata$manufacturer=="Toyota"|
  fueldata$manufacturer=="SsangYong"|fueldata$manufacturer=="Mini"|
  fueldata$manufacturer=="Cadillac"|fueldata$manufacturer=="Fiat",
  1,0)
sum(indicat) #total cars being evaluated
```

```
## [1] 8792
```

```r
shortlist <- fueldata[indicat==1,]
shortlist <- shortlist[-c(8792),] #Removing 'Nissan Leaf'; too many NA obs.

#Alot of Data is incomplete; creating a dataframe for objects of interest.
attach(shortlist)
```

```
## The following object is masked from package:datasets:
##
##     co2
```

```r
workingdata<- data.frame(
combined_metric, manufacturer, engine_capacity, co2,
noise_level, co_emissions, transmission_type,
fuel_type,year)
detach(shortlist)
#Removing final missing values:
workingdata<- na.omit(workingdata)
names(workingdata)
```

```
## [1] "combined_metric"   "manufacturer"      "engine_capacity"
## [4] "co2"               "noise_level"       "co_emissions"
## [7] "transmission_type" "fuel_type"         "year"
```

```r
attach(workingdata)
```

```
## The following object is masked from package:datasets:
##
##     co2
```

Basic summaries, i.e. mean, range, and fivenum.

We have the following for each predictor

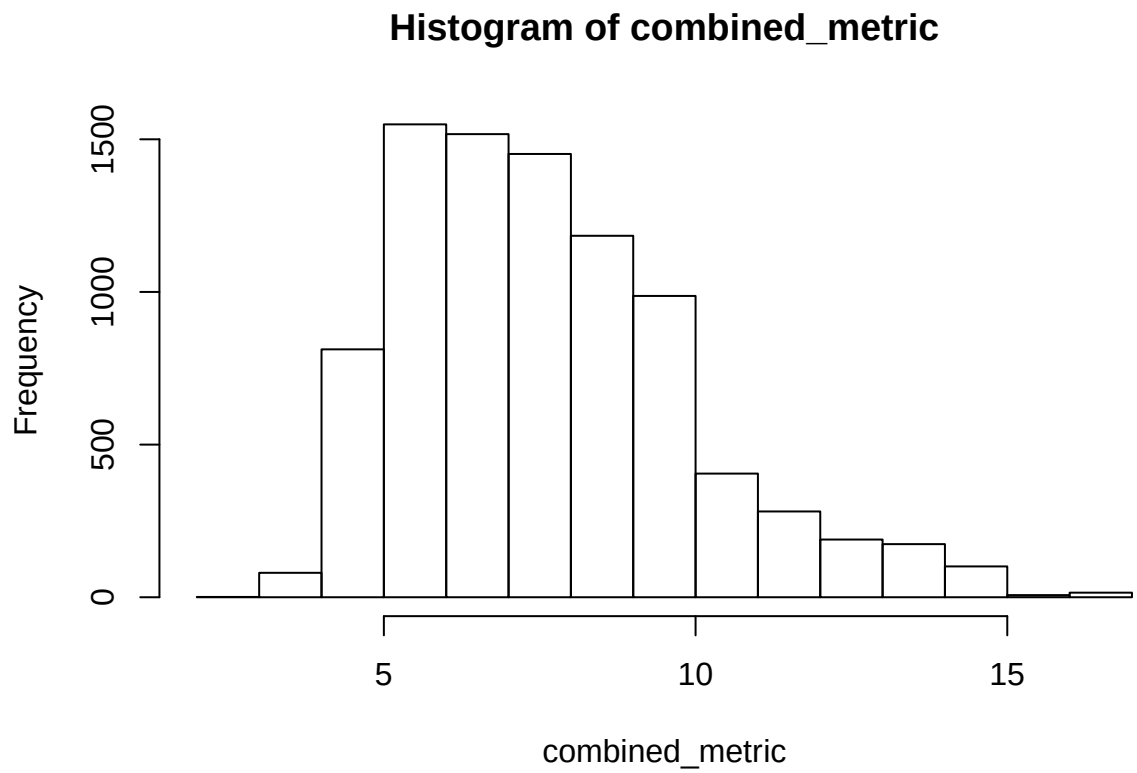
```r
#Response Variable:

range(combined_metric)
```

```
## [1]  2.1 16.7
```

```r
summary(combined_metric)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.10    5.90    7.30    7.66    9.00   16.70
```

```r
hist(combined_metric)
```




```r
#Variables:

range(year)
```

```
## [1] 2000 2013
```

```r
#Change year to 'age' or how old the car is (0 being made in 2013).
age<-year-2000
range(age)
```

```
## [1]  0 13
```

```r
#Manufacturers being evaluated and counted.
manufacturer<-as.factor(manufacturer)
manufacturer<-droplevels.factor(manufacturer)
table(manufacturer)
```

```
## manufacturer
##      Audi       BMW  Cadillac      Fiat  Metrocab      Mini    Nissan
##      2842      2941       172       702        20       371       665
## SsangYong      Tata    Toyota
##       168         7       866
```

```r
#Engine capacity by CC.
range(engine_capacity)
```

```
## [1]  875 6299
```

```r
#Carbon Dioxide emissions in grams per kilometer (g/km).
range(co2)
```

```
## [1]  49 405
```

```r
#Decibel noise level on the A scale of a noise meter.
range(noise_level)
```

```
## [1]  0.2 78.0
```

```r
#Carbon monoxide emissions in milligrams per kilometer (mg/km).
range(co_emissions)
```

```
## [1]    0.2 2100.0
```

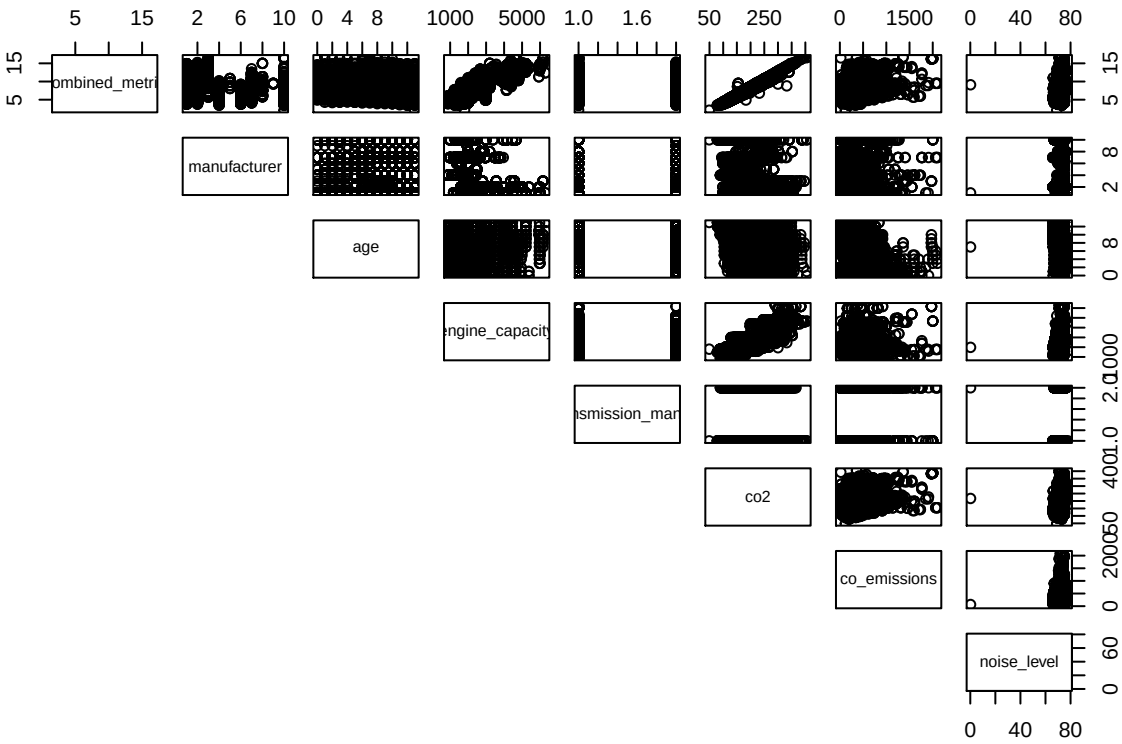
```r
#Dummy variable; car's transmission (auto or manual).
transmission_manual<-ifelse(transmission_type=="M",1,0)
transmission_manual<-as.factor(transmission_manual)
table(transmission_manual)
```

```
## transmission_manual
##    0    1
## 4127 4627
```

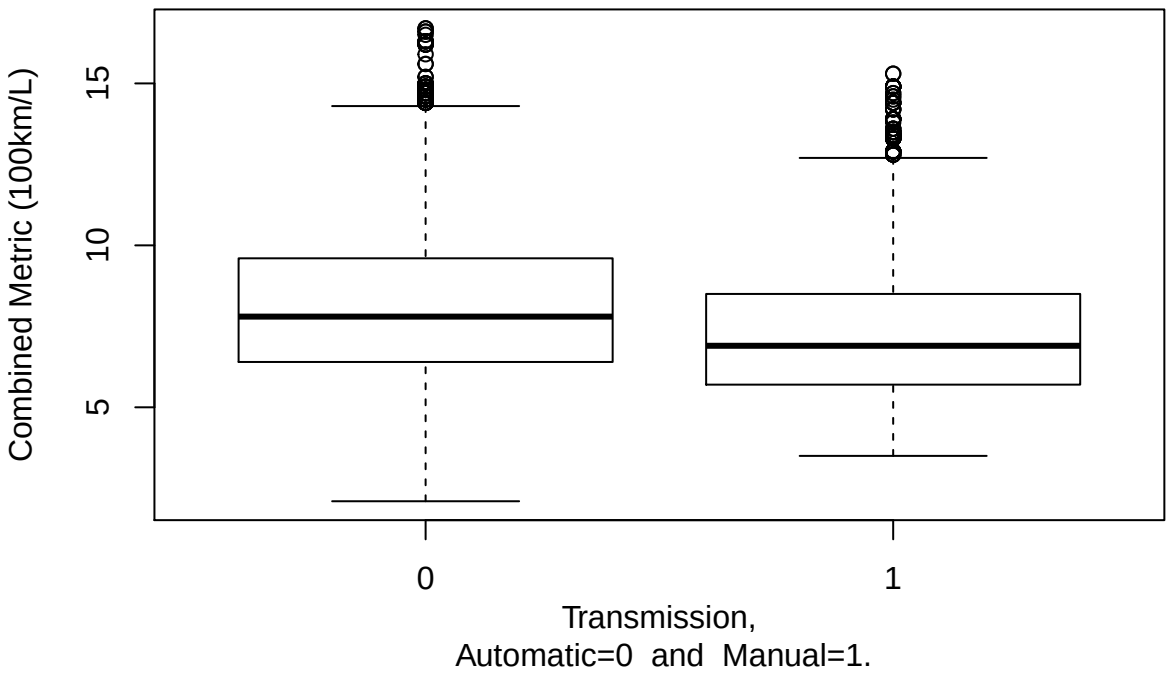
Looking for Pairs

We'll look at pairs to see trends:

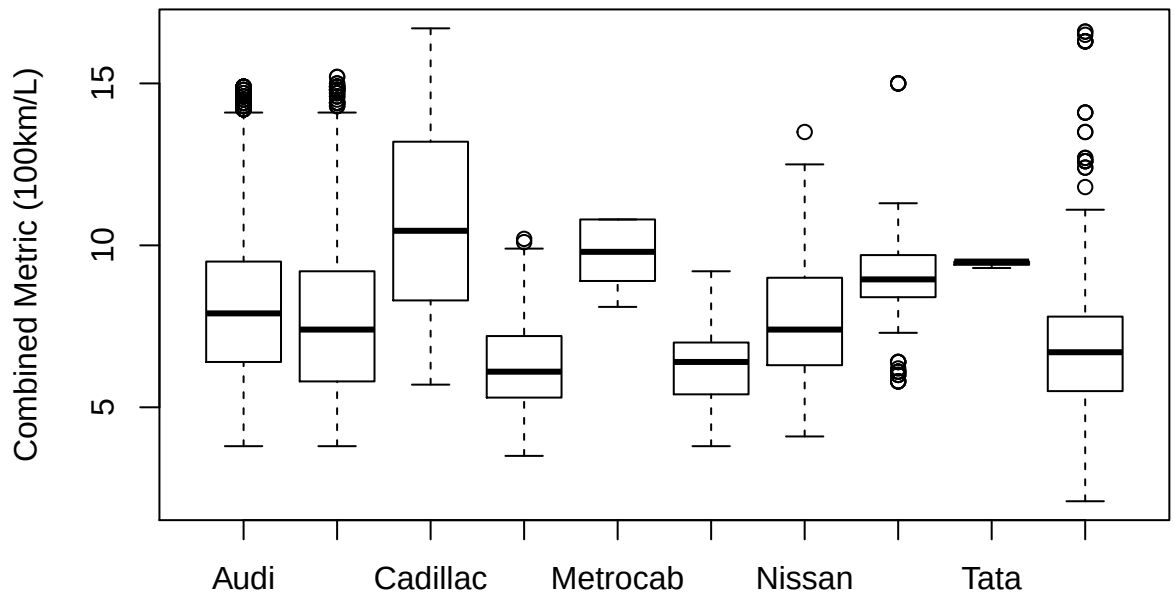
```r
pairs(combined_metric~manufacturer+age+engine_capacity+transmission_manual
      +co2+co_emissions+noise_level,lower.panel = NULL)
```




```
boxplot(combined_metric~transmission_manual,
        ylab="Combined Metric (100km/L)",xlab="Transmission,\n Automatic=0  and  Manual=1.")
```




```r
boxplot(combined_metric~manufacturer,ylab="Combined Metric (100km/L)")
```



Model Building/Selection

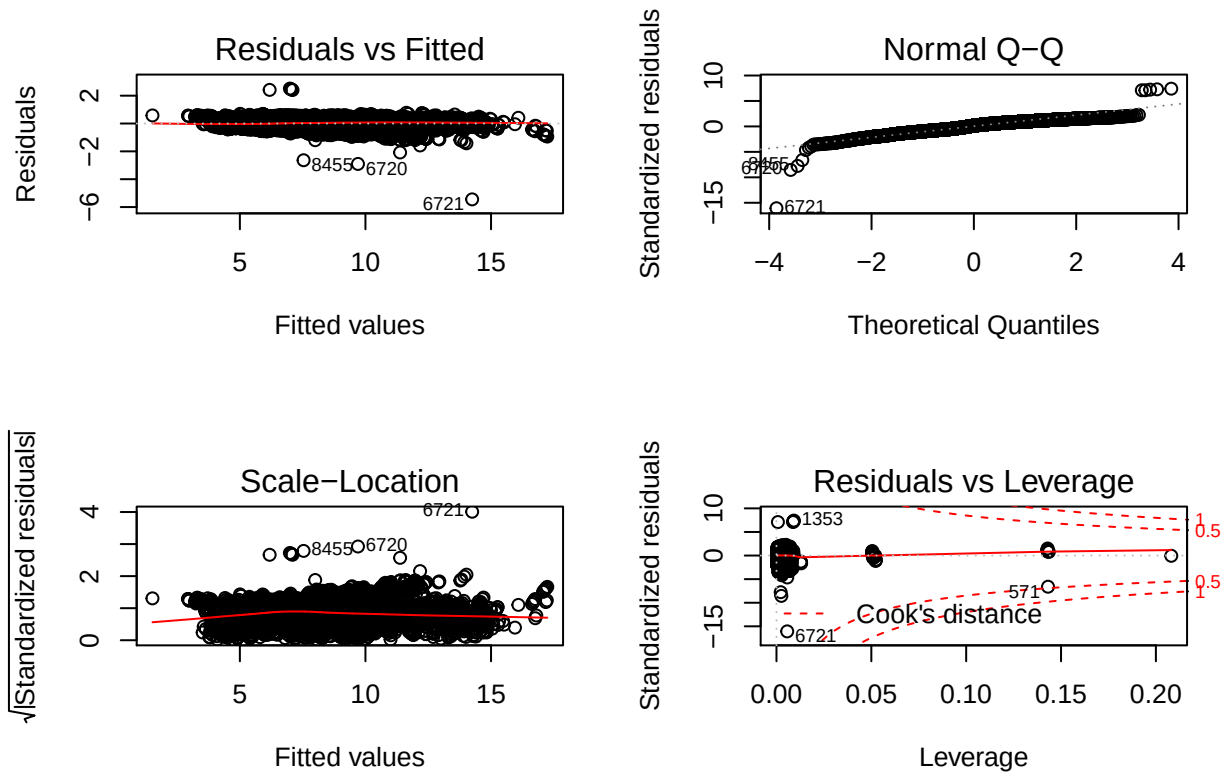
Let's construct a basic model with all possible parameters:

```
m1<-lm(combined_metric~manufacturer+age+engine_capacity+transmission_manual
       +co2+co_emissions+noise_level)
summary(m1)
```

```
##
## Call:
## lm(formula = combined_metric ~ manufacturer + age + engine_capacity +
##     transmission_manual + co2 + co_emissions + noise_level)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.448 -0.238  0.053  0.262  2.508
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -6.93e-01   1.50e-01   -4.63  3.8e-06 ***
## manufacturerBMW        7.98e-02   9.39e-03    8.50  < 2e-16 ***
## manufacturerCadillac  -5.84e-02   2.71e-02   -2.15   0.0316 *
## manufacturerFiat      -8.42e-03   1.52e-02   -0.55   0.5800
## manufacturerMetrocab  -7.56e-01   7.67e-02   -9.86  < 2e-16 ***
```

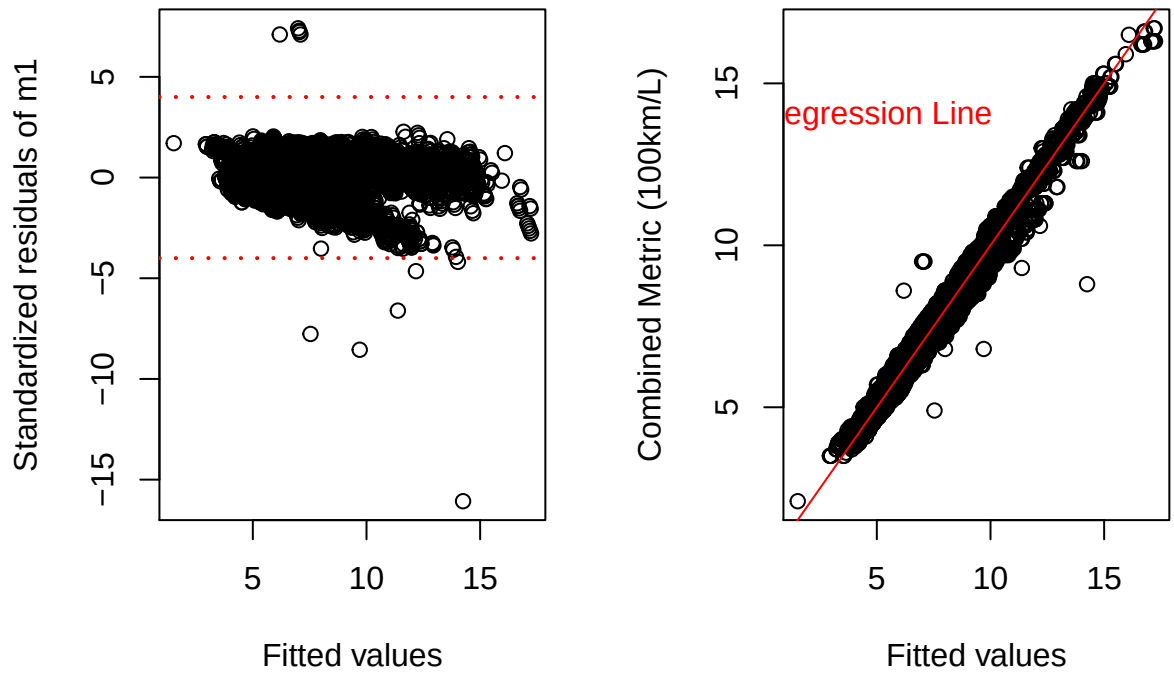
```
## manufacturerMini       6.09e-02   1.94e-02    3.13   0.0017 **
## manufacturerNissan    -1.06e-01   1.52e-02   -6.95  3.9e-12 ***
## manufacturerSsangYong -6.21e-01   2.74e-02  -22.63  < 2e-16 ***
## manufacturerTata      -1.25e+00   1.29e-01   -9.71  < 2e-16 ***
## manufacturerToyota     3.43e-02   1.37e-02    2.51   0.0122 *
## age                    4.21e-02   1.27e-03   33.09  < 2e-16 ***
## engine_capacity       -2.02e-04   8.72e-06  -23.17  < 2e-16 ***
## transmission_manual1  -1.23e-02   8.22e-03   -1.49   0.1361
## co2                    4.56e-02   1.52e-04  299.61  < 2e-16 ***
## co_emissions           6.16e-04   1.67e-05   36.86  < 2e-16 ***
## noise_level           -4.49e-03   2.12e-03   -2.11   0.0345 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.34 on 8738 degrees of freedom
## Multiple R-squared:  0.978,  Adjusted R-squared:  0.978
## F-statistic: 2.64e+04 on 15 and 8738 DF,  p-value: <2e-16
```

```r
par(mfrow=c(2,2))
plot(m1)
```




```r
par(mfrow=c(1,2))
plot(m1$fitted.values,rstandard(m1),xlab="Fitted values",ylab="Standardized residuals of m1")
abline(h=-4,lty=3,col="red",lwd=2)
abline(h=4,lty=3,col="red",lwd=2)
```

```r
plot(m1$fitted.values,combined_metric,xlab="Fitted values",ylab="Combined Metric (100km/L)")
abline(lsfit(m1$fitted.values,combined_metric),col=2)
text(5,14,"Regression Line",col=2)
```




```r
#Testing for normality; Shapiro cannot be used as n>5000.
ks.test(x=m1$residuals,y='pnorm',alternative='two.sided')
```

```
## Warning in ks.test(x = m1$residuals, y = "pnorm", alternative =
## "two.sided"): ties should not be present for the Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  m1$residuals
## D = 0.28, p-value <2e-16
## alternative hypothesis: two-sided
```

```r
vif(m1)
```

```
##                      GVIF Df GVIF^(1/(2*Df))
## manufacturer        1.811  9           1.034
## age                 1.745  1           1.321
## engine_capacity     4.069  1           2.017
```

```
## transmission_manual 1.276  1           1.130
## co2                 5.017  1           2.240
## co_emissions        1.307  1           1.143
## noise_level         1.255  1           1.120
```

There does not appear to collinearity between the predictors; however, from the Kolmogorov-Smirnovtest, our assumption of normality does not hold.

Variable Selection

Lets use a selection process to select significicant predictors.

```r
X<-cbind(as.factor(manufacturer),age,engine_capacity,transmission_manual,
         co2,co_emissions,noise_level)
mselect<-regsubsets(X,combined_metric)
summary(mselect)
```

```
## Subset selection object
## 7 Variables  (and intercept)
##                     Forced in Forced out
##                         FALSE      FALSE
## age                     FALSE      FALSE
## engine_capacity         FALSE      FALSE
## transmission_manual     FALSE      FALSE
## co2                     FALSE      FALSE
## co_emissions            FALSE      FALSE
## noise_level             FALSE      FALSE
## 1 subsets of each size up to 7
## Selection Algorithm: exhaustive
##              age engine_capacity transmission_manual co2 co_emissions
## 1  ( 1 ) " " " " " "             " "                 "*" " "
## 2  ( 1 ) " " " " " "             " "                 "*" "*"
## 3  ( 1 ) " " "*" " "             " "                 "*" "*"
## 4  ( 1 ) " " "*" "*"             " "                 "*" "*"
## 5  ( 1 ) "*" "*" "*"             " "                 "*" "*"
## 6  ( 1 ) "*" "*" "*"             "*"                 "*" "*"
## 7  ( 1 ) "*" "*" "*"             "*"                 "*" "*"
##          noise_level
## 1  ( 1 ) " "
## 2  ( 1 ) " "
## 3  ( 1 ) " "
## 4  ( 1 ) " "
## 5  ( 1 ) " "
## 6  ( 1 ) " "
## 7  ( 1 ) "*"
```

```r
#summary(powerTransform(cbind(combined_metric,age)~1))
```