Final Project

Zhaoqi Li

2022/12/20

```
library(openintro)
```

```
## Loading required package: airports
```

```
## Loading required package: cherryblossom
```

```
## Loading required package: usdata
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
```

```
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(car)
```

```
## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
##
## The following object is masked from 'package:openintro':
##
##     densityPlot
```

Clean data and remove missing value

```
# Data clean
my_data <- read_csv("MY2022 Fuel Consumption Ratings.csv")
```

```
## Rows: 946 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): Make, Model, Vehicle Class, Transmission, Fuel Type
## dbl (10): Model Year, Engine Size(L), Cylinders, Fuel Consumption (City (L/1...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
fuel_cons <-rename(my_data, engine_size = 'Engine Size(L)'
                   , fuel_type = 'Fuel Type'
                   , fuel_consumption_comb = 'Fuel Consumption(Comb (L/100 km))'
                   , co2_emissions_gkm = 'CO2 Emissions(g/km)'
                   , fuel_consumption_hwy = 'Fuel Consumption(Hwy (L/100 km))'
                   , fuel_consumption_city = 'Fuel Consumption (City (L/100 km)'
                   , co2_rating = 'CO2 Rating'
                   , fuel_consumption_mpg = 'Fuel Consumption(Comb (mpg))'
                   , model_year = 'Model Year'
                   , smog_rating = 'Smog Rating') %>%
  mutate(co2_emissions = co2_emissions_gkm*100,
         transmission = case_when(Transmission == "A10" ~ "A",
                                  Transmission == "A9" ~"A",
                                  Transmission == "A7" ~"A",
                                  Transmission == "A6" ~"A",
                                  Transmission == "AM6" ~"AM",
                                  Transmission == "AM7" ~"AM",
                                  Transmission == "AM8" ~"AM",
                                  Transmission == "AS10" ~"AS",
                                  Transmission == "AS9" ~"AS",
                                  Transmission == "AS8" ~"AS",
                                  Transmission == "AS7" ~"AS",
                                  Transmission == "AS6" ~"AS",
                                  Transmission == "AS5" ~"AS",
                                  Transmission == "AV1" ~"AV",
                                  Transmission == "AV10" ~"AV",
                                  Transmission == "AV8" ~"AV",
                                  Transmission == "AV7" ~"AV",
                                  Transmission == "AV6" ~"AV",
                                  Transmission == "M5" ~"M",
                                  Transmission == "M6" ~"M",
                                  Transmission == "M7" ~"M")) %>%
  drop_na()
```

Step 0: Divide data into training and testing

```
# Create training and test set
set.seed(147)

# Count the number of observations in the data
```

```r
n <- nrow(fuel_cons)

# Randomly choose 80% as training and round number
training <- sample(1:n, size = round(0.8*n))

# Create a training set
train <- fuel_cons[training,]

# Create a testing set
test <- fuel_cons[-training,]
```

Description of Important Variables

```r
# Table of the important variables

Variable <-(c("Fuel Consumption","Smog Rating","Transmission", "Cylinders","CO2 Rating","Fuel Type "))
Type <-(c("Numerical Variable","Numerical Variable","Categorical Variable","Numerical Variable","Numeri
Description <-(c("The combine fuel consumption (city 55%, hwy 45%) in liters per 100 kilometers(L/100km
knitr::kable(tibble(Variable,Type, Description),caption="The Description of Important Variables")
```

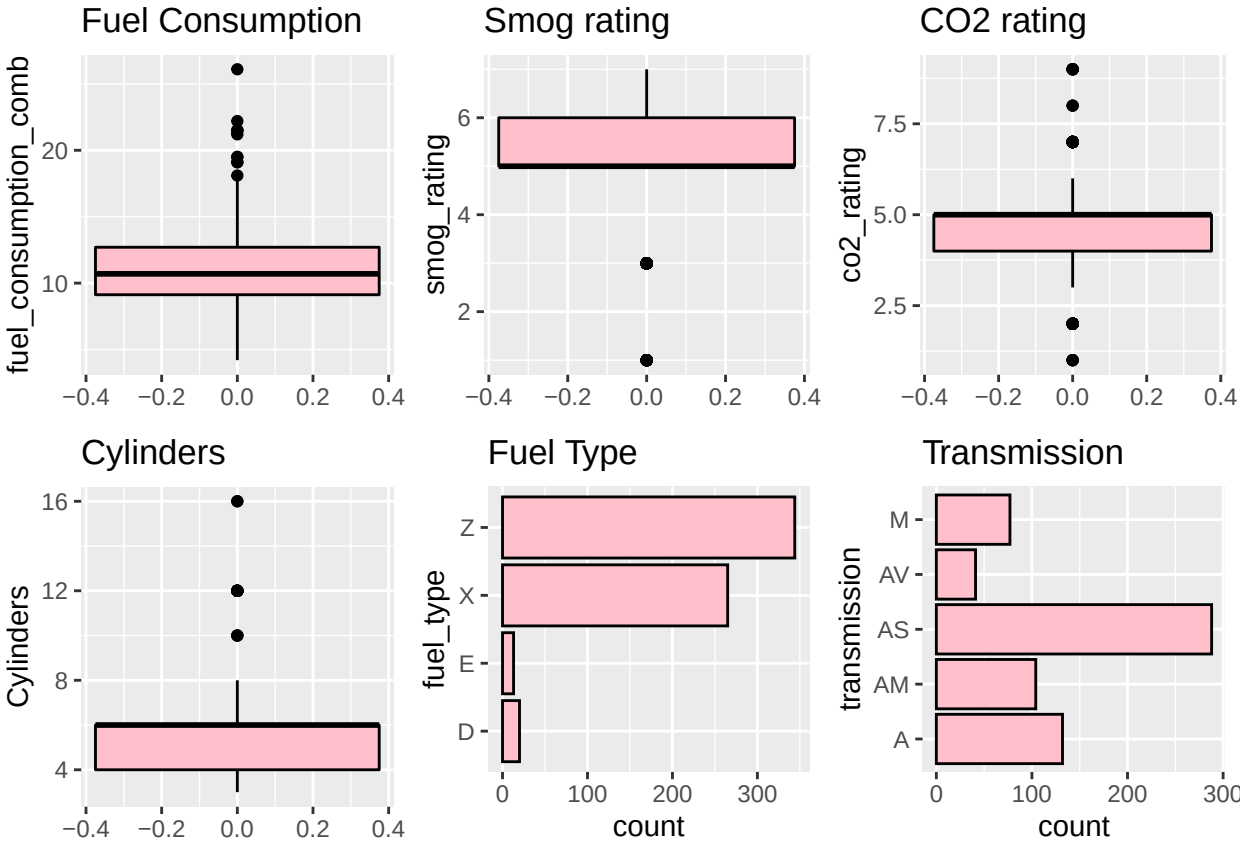
Table 1: The Description of Important Variables

Variable	Type	Description
Fuel Consumption	Numerical Variable	The combine fuel consumption (city 55%, hwy 45%) in liters per 100 kilometers(L/100km)
Smog Rating	Numerical Variable	The emissions of smog-forming pollutants rated on a scale from 1 (worst) to 10 (best)
Transmission	Categorical Variable	The type of transmissions.A = automatic; M = manual; AM = automated manual; AS = automatic with select shift; AV = continuously
Cylinders	Numerical Variable	The size of engine (L)
CO2 Rating	Numerical Variable	The emissions of CO2 pollutants rated on a scale from 1 (worst) to 10 (best)
Fuel Type	Categorical Variable	The fuel type of vehicle. X = regular gasoline; Z = premium gasoline; D = diesel ; E = ethanol (E85);N = natural gas

Exploratory Data Analysis of training set

```r
# Box plot and bar plot of predictors in training dataset
bar_transmission <- train %>%
  ggplot(aes(x=transmission)) +
  geom_bar(color='black', fill='pink') +
  labs(title="Transmission") +
  coord_flip()

bar_fuel_type <- train %>%
  ggplot(aes(x=fuel_type)) +
  geom_bar(color='black', fill='pink') +
  labs(title="Fuel Type") +
  coord_flip()
```

```
box_fuel_cons <- train %>%
  ggplot(aes(y=fuel_consumption_comb)) +
  geom_boxplot(color= "black", fill="pink") +
  labs(title="Fuel Consumption")
box_smog_rating <- train %>%
  ggplot(aes(y=smog_rating)) +
  geom_boxplot(color="black", fill="pink") +
  labs(title="Smog rating")

box_co2_rating <- train %>%
  ggplot(aes(y=co2_rating)) +
  geom_boxplot(color="black", fill="pink") +
  labs(title="CO2 rating")

box_cylinders <- train %>%
  ggplot(aes(y=Cylinders)) +
  geom_boxplot(color="black", fill="pink") +
  labs(title="Cylinders")

grid.arrange(box_fuel_cons,box_smog_rating,box_co2_rating,box_cylinders,bar_fuel_type,bar_transmission,
```

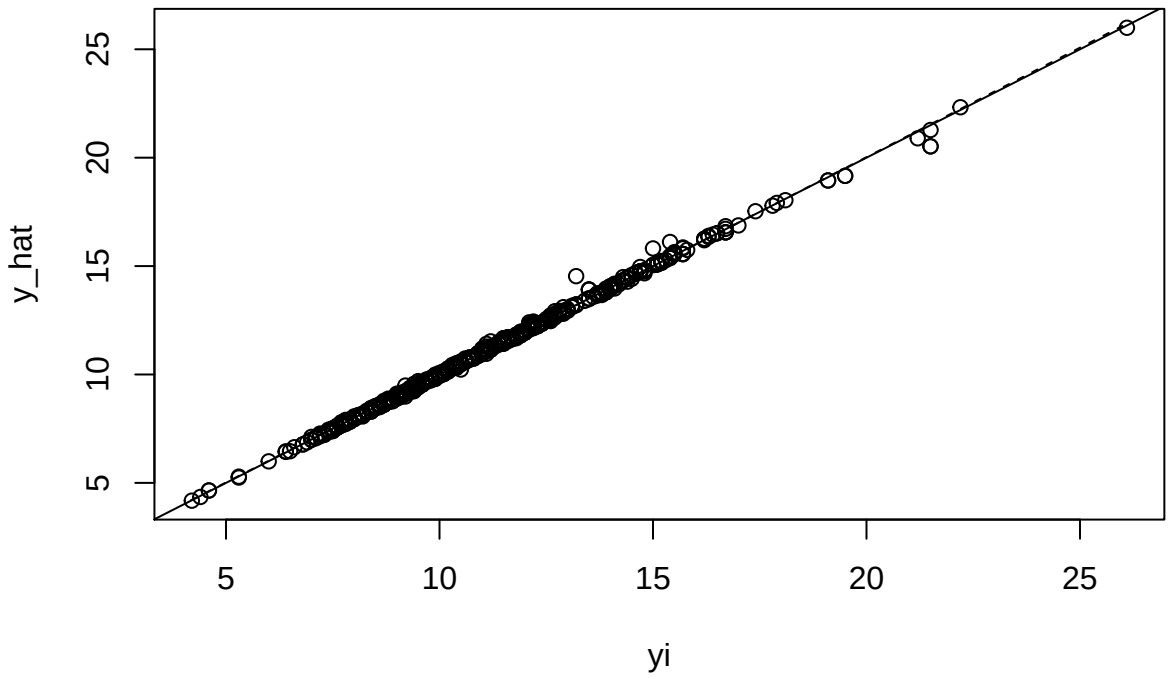


Step 1: Choose a starting model

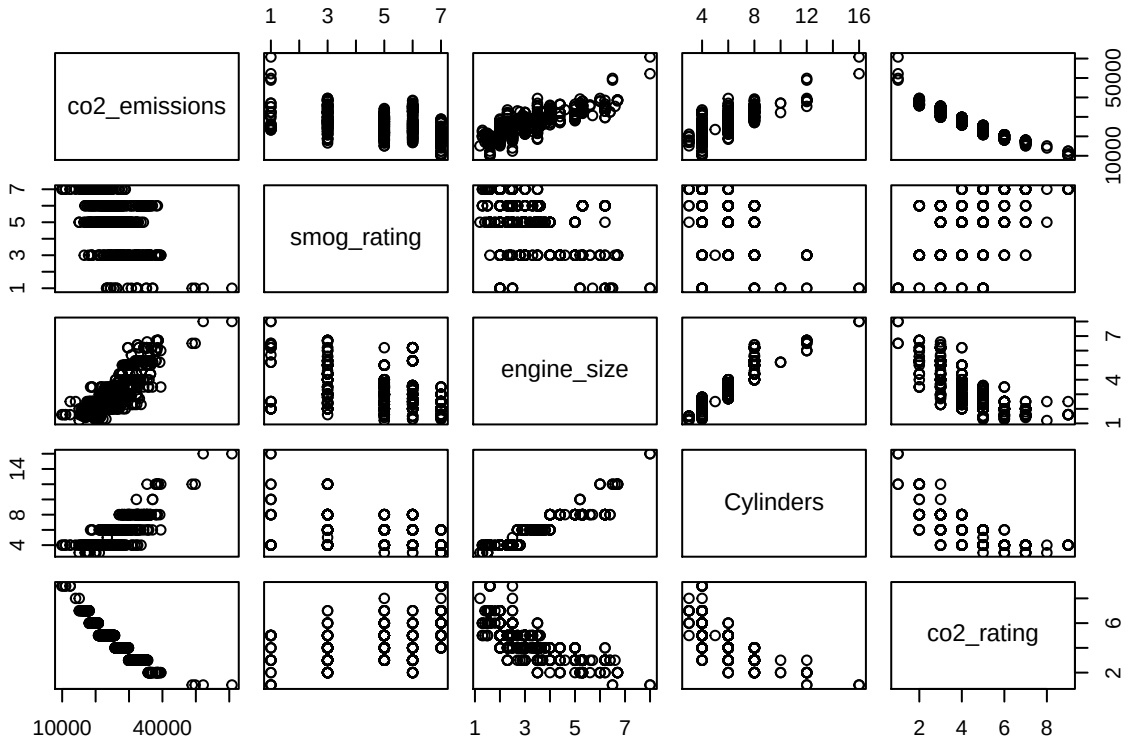
```
# Full model with all possible predictors
model_full <- lm(fuel_consumption_comb ~ co2_emissions + smog_rating + transmission + engine_size + Cyli
```

Step 2: Check condition 1&2 and assumptions

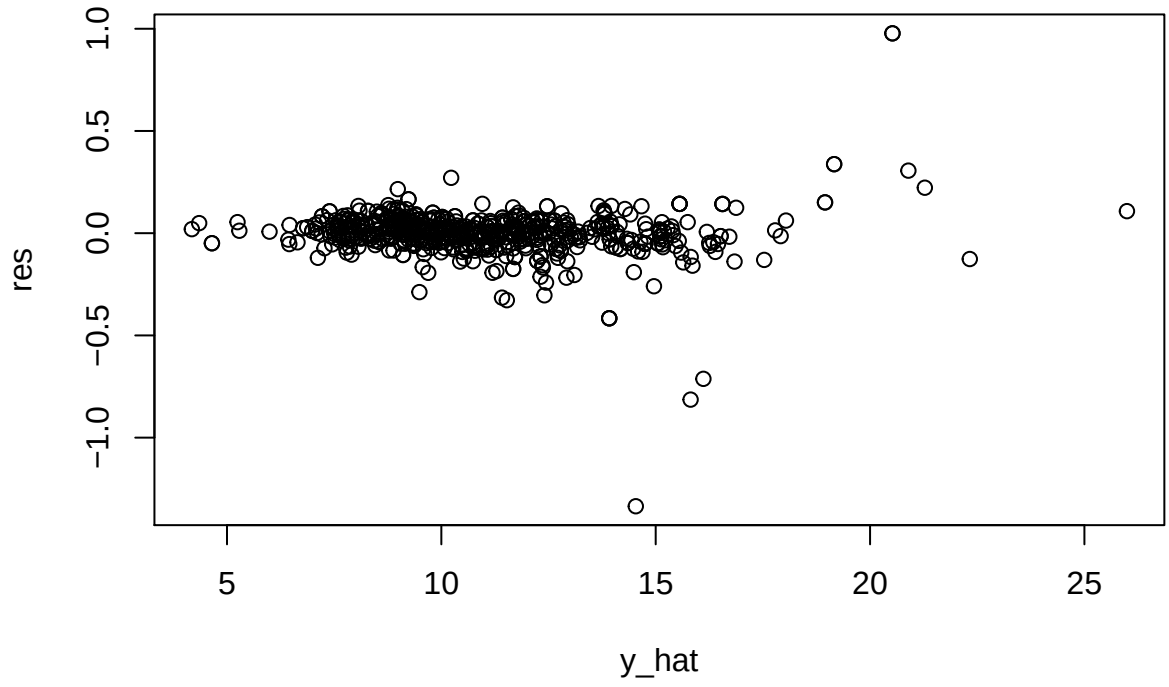
```
## Condition 1: draw a scatter plot between yi and y_hat
y_hat <- fitted(model_full)
yi <- train$fuel_consumption_comb
plot(yi,y_hat)
abline(a = 0, b = 1)
lines(lowess(yi ~ y_hat), lty=2)
```



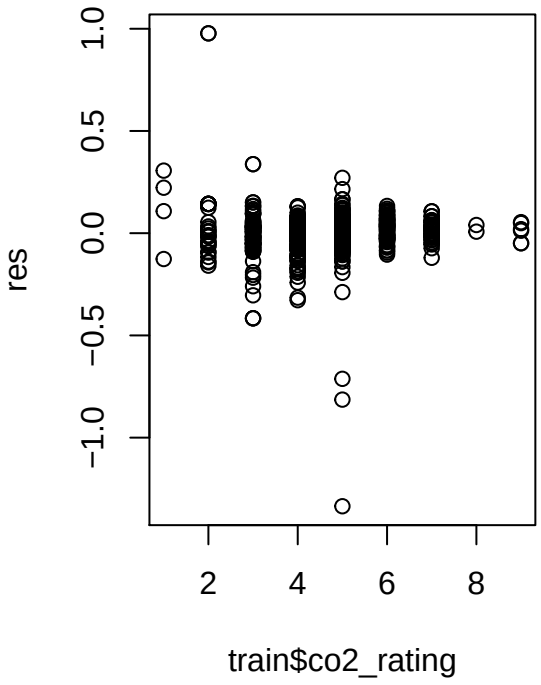

```
# Condition 2: draw scatter plots between predictors (numerical predictors)
pairs(~co2_emissions + smog_rating + engine_size + Cylinders + co2_rating, data=train)
```

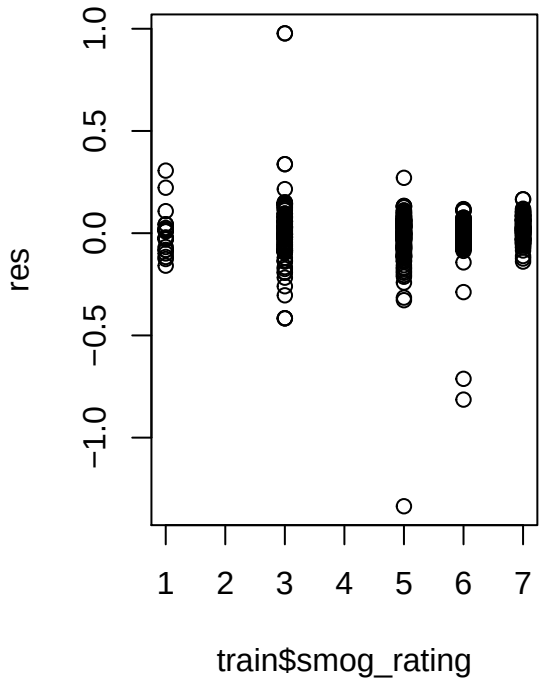



```
## Residual vs. Fitted model
res <- model_full$residuals
y_hat <- fitted(model_full)
plot(y_hat, res)
```

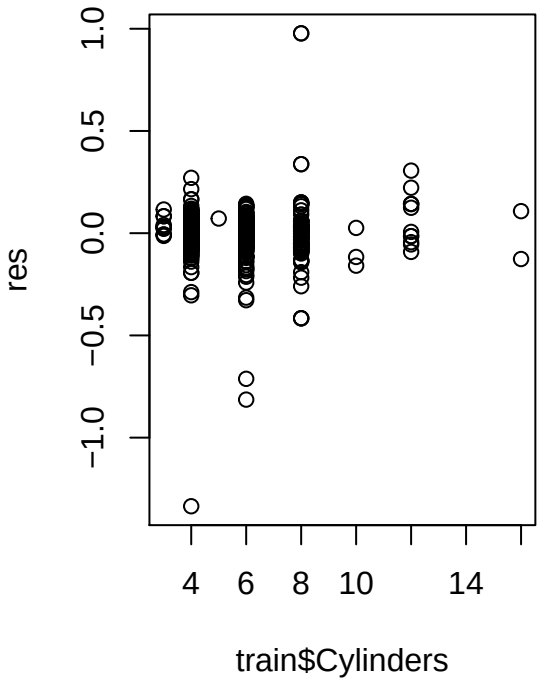


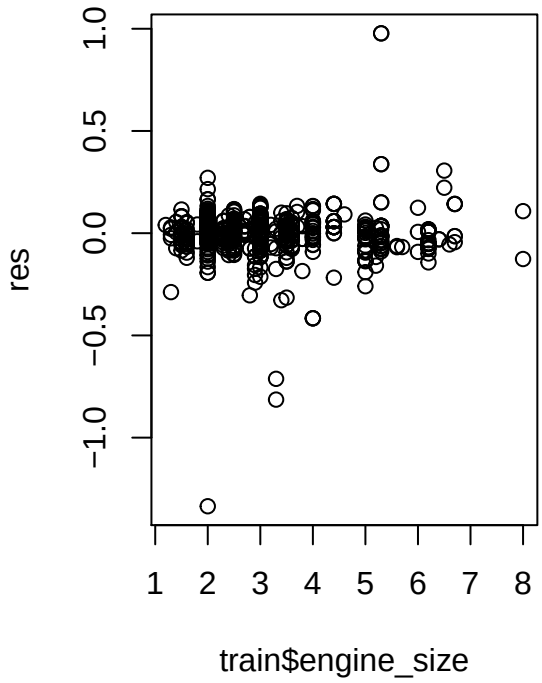

```
## Residual vs. Predictors
par(mfrow = c(1, 2))
plot(train$co2_rating, res)
plot(train$smog_rating,res)
```



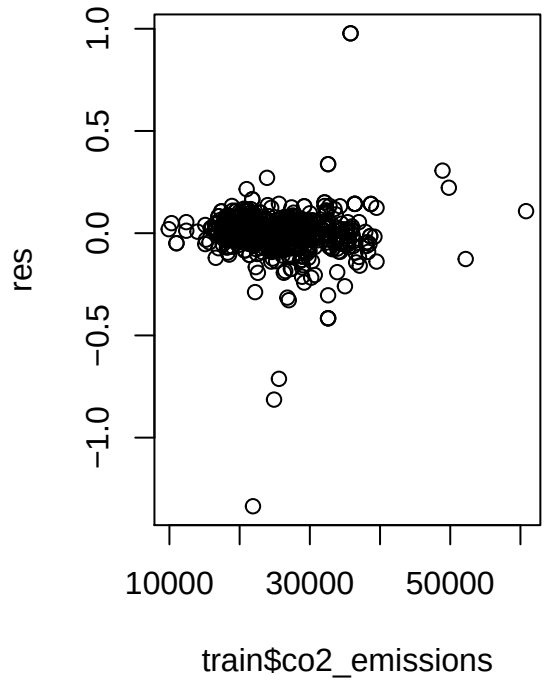



```
plot(train$Cylinders,res)
plot(train$engine_size,res)
```

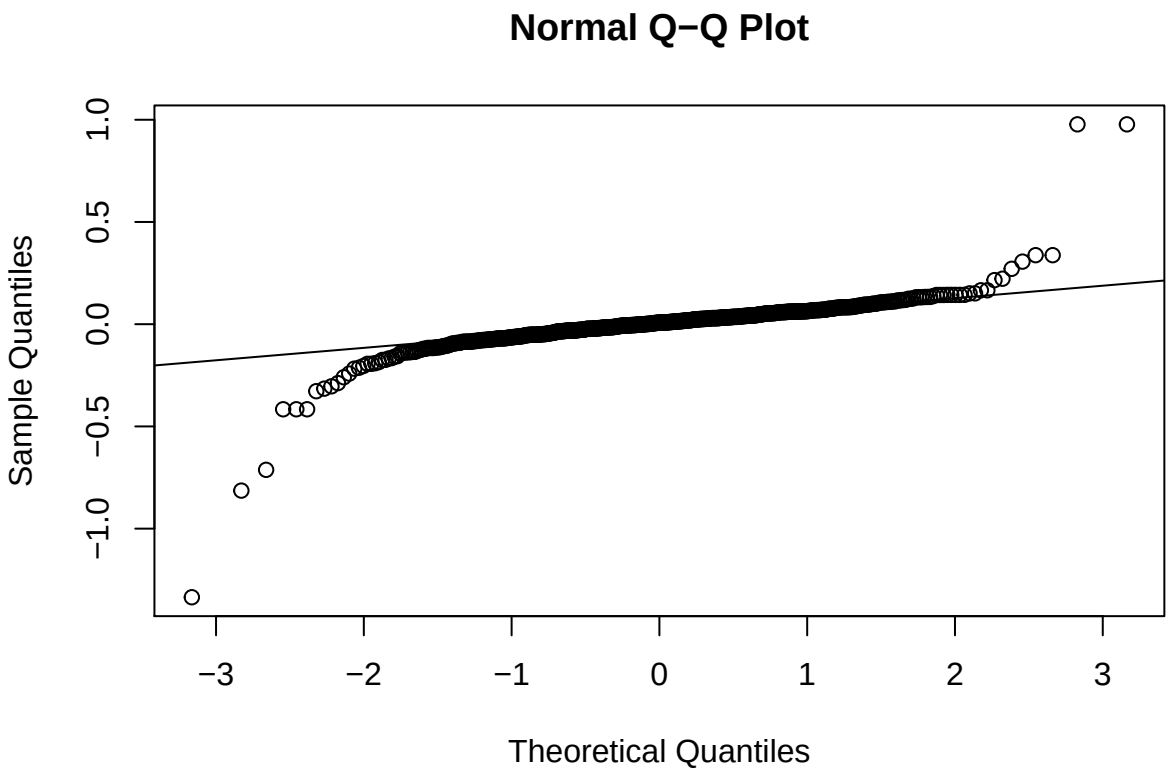





```
plot(train$co2_emissions,res)
```




```
qqnorm(res)
qqline(res)
```



Step 3 : Model transformations to correct assumption violations

```
# Summary box-cox transformation for numerical variables
summary(powerTransform(cbind(train$fuel_consumption_comb,
                             train$engine_size,
                             train$Cylinders,
                             train$co2_emissions,
                             train$co2_rating,
                             train$smog_rating)))
```

```
## bcPower Transformations to Multinormality
##    Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## Y1   -0.4579       -0.50      -0.5793      -0.3366
## Y2   -0.0871        0.00      -0.1997       0.0255
## Y3   -0.3608       -0.50      -0.5219      -0.1996
## Y4   -0.2196       -0.33      -0.3427      -0.0966
## Y5    1.0480        1.00       0.9335       1.1624
## Y6    1.7042        1.70       1.4975       1.9108
##
## Likelihood ratio test that transformation parameters are equal to 0
##  (all log transformations)
##                                      LRT df       pval
## LR test, lambda = (0 0 0 0 0 0) 922.3271  6 < 2.22e-16
##
## Likelihood ratio test that no transformations are needed
##                                      LRT df       pval
## LR test, lambda = (1 1 1 1 1 1) 1205.101  6 < 2.22e-16
```

```r
# Create transformed variables
train_trans <- train %>%
  mutate(trans_fuel_consumption_comb = fuel_consumption_comb^(-0.5),
         trans_engine_size =log(engine_size),
         trans_Cylinders = Cylinders^(-0.5),
         trans_co2_emissions = co2_emissions^(-0.33),
         trans_smog_rating = smog_rating^(1.7))
```

```r
# Fit a new model with the transformed variables
model_full_trans <- lm(trans_fuel_consumption_comb ~trans_co2_emissions + trans_smog_rating + transmiss
```

Step 4: Ensure no multicollinearity is present in the model

```r
# Check the VIF of the model and remove the predictors with high VIF
vif(model_full_trans)
```

```
##                           GVIF Df GVIF^(1/(2*Df))
## trans_co2_emissions 19.995553  1        4.471639
## trans_smog_rating    1.776572  1        1.332881
## transmission         1.879148  4        1.082045
## trans_engine_size    9.217127  1        3.035972
## trans_Cylinders      8.435009  1        2.904309
## co2_rating          18.686629  1        4.322803
## fuel_type            1.761690  3        1.098976
```

```r
# Remove transformed CO2 emissions because it has a high VIF
model_full_trans1 <- lm(trans_fuel_consumption_comb ~ trans_smog_rating + transmission + trans_engine_s

vif(model_full_trans1)
```

```
##                       GVIF Df GVIF^(1/(2*Df))
## trans_smog_rating 1.758585  1        1.326116
## transmission      1.816370  4        1.077458
## trans_engine_size 9.000382  1        3.000064
## trans_Cylinders   8.416469  1        2.901115
## co2_rating        3.377959  1        1.837922
## fuel_type         1.733179  3        1.095992
```

```r
# Remove transformed engine size  because it has a high VIF
model_full_trans2 <- lm(trans_fuel_consumption_comb ~ trans_smog_rating + transmission + trans_Cylinder

vif(model_full_trans2)
```

```
##                       GVIF Df GVIF^(1/(2*Df))
## trans_smog_rating 1.758526  1        1.326094
## transmission      1.754288  4        1.072785
## trans_Cylinders   2.796586  1        1.672299
## co2_rating        3.049340  1        1.746236
## fuel_type         1.699488  3        1.092412
```

Step 5: Model selection

```r
# Manually select the predictors with low P-value as reduced model
summary(model_full_trans2)
```

```
##
## Call:
## lm(formula = trans_fuel_consumption_comb ~ trans_smog_rating +
##     transmission + trans_Cylinders + co2_rating + fuel_type,
##     data = train_trans)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.021023 -0.006644 -0.000865  0.006011  0.063686
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        2.049e-01  3.535e-03  57.966  < 2e-16 ***
## trans_smog_rating  1.533e-04  6.157e-05   2.489 0.013058 *
## transmissionAM     5.526e-03  1.387e-03   3.984 7.56e-05 ***
## transmissionAS     2.666e-03  1.065e-03   2.504 0.012524 *
## transmissionAV     6.973e-03  1.783e-03   3.911 0.000102 ***
## transmissionM      1.748e-03  1.465e-03   1.193 0.233237
## trans_Cylinders    1.616e-02  9.038e-03   1.788 0.074245 .
## co2_rating         2.456e-02  4.542e-04  54.078  < 2e-16 ***
## fuel_typeE        -6.434e-02  3.344e-03 -19.241  < 2e-16 ***
## fuel_typeX        -1.947e-02  2.404e-03  -8.099 2.86e-15 ***
## fuel_typeZ        -2.264e-02  2.386e-03  -9.486  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.008989 on 631 degrees of freedom
## Multiple R-squared:  0.9455, Adjusted R-squared:  0.9446
## F-statistic:  1094 on 10 and 631 DF,  p-value: < 2.2e-16
```

```
# Remove the predictors transmission and cylinders becuase they are not significant
model_reduced <- lm(trans_fuel_consumption_comb ~ trans_smog_rating + co2_rating + fuel_type, data = tra
```

```
# Compare the F test of full model and reduced model
anova(model_reduced, model_full_trans2)
```

```
## Analysis of Variance Table
##
## Model 1: trans_fuel_consumption_comb ~ trans_smog_rating + co2_rating +
##     fuel_type
## Model 2: trans_fuel_consumption_comb ~ trans_smog_rating + transmission +
##     trans_Cylinders + co2_rating + fuel_type
##   Res.Df      RSS Df Sum of Sq      F    Pr(>F)
## 1    636 0.053306
## 2    631 0.050984  5 0.0023223 5.7483 3.368e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

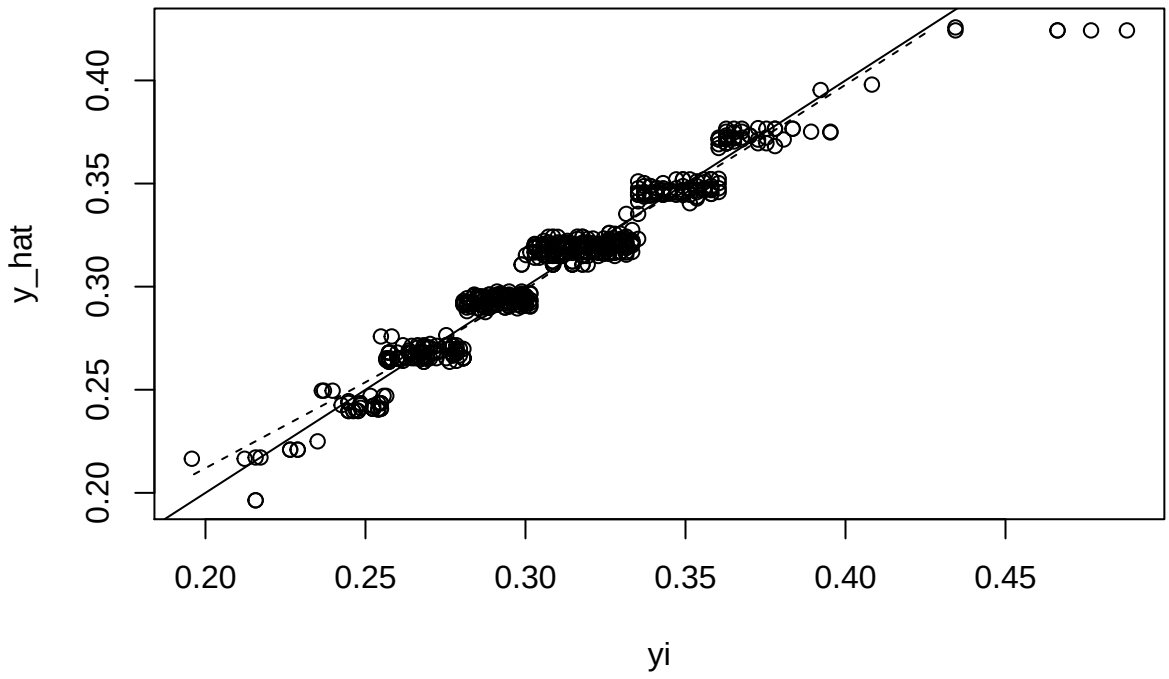
```
# Add back a removed predictor to ensure F test is large than 0.05
model_reduced2 <- lm(trans_fuel_consumption_comb ~ trans_smog_rating + co2_rating + fuel_type + transmi
anova(model_reduced2, model_full_trans2)
```

```
## Analysis of Variance Table
##
## Model 1: trans_fuel_consumption_comb ~ trans_smog_rating + co2_rating +
##     fuel_type + transmission
```

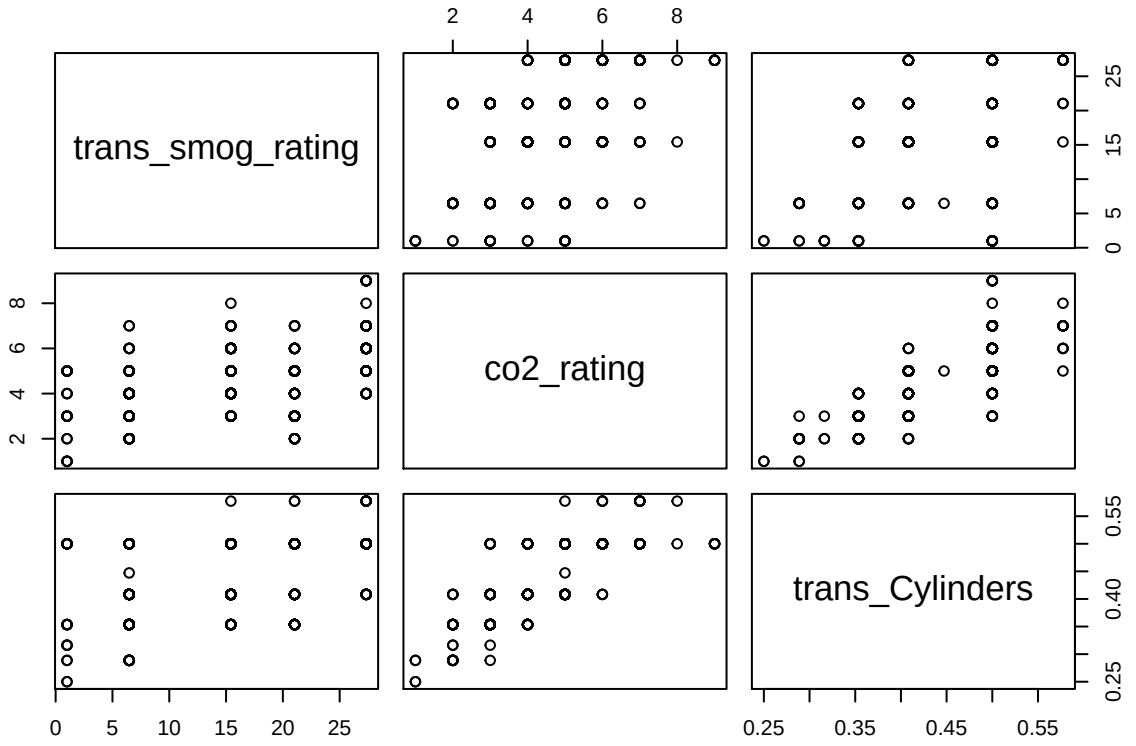
```
## Model 2: trans_fuel_consumption_comb ~ trans_smog_rating + transmission +
##     trans_Cylinders + co2_rating + fuel_type
##   Res.Df      RSS Df  Sum of Sq      F  Pr(>F)
## 1    632 0.051242
## 2    631 0.050984  1 0.00025833 3.1972 0.07425 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Check the Condition 1&2 and assumptions of full model

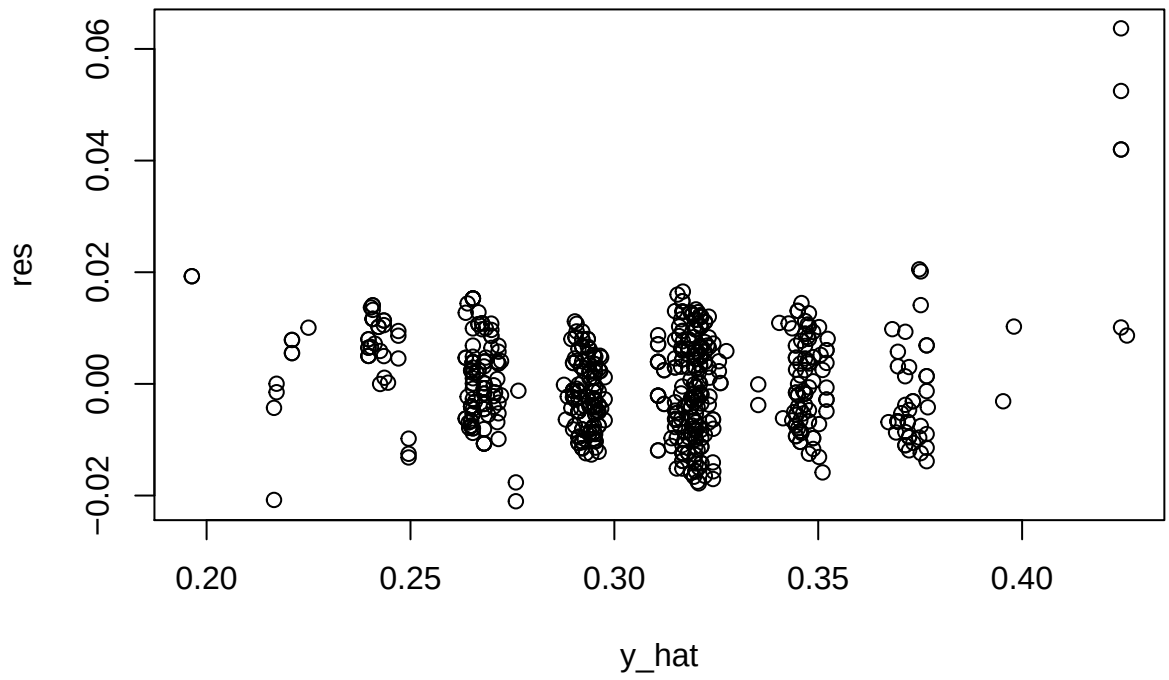
```
## Condition 1: draw a scatter plot between yi and y_hat
y_hat <- fitted(model_full_trans2)
yi <- train_trans$trans_fuel_consumption_comb
plot(yi,y_hat)
abline(a = 0, b = 1)
lines(lowess(yi ~ y_hat), lty=2)
```



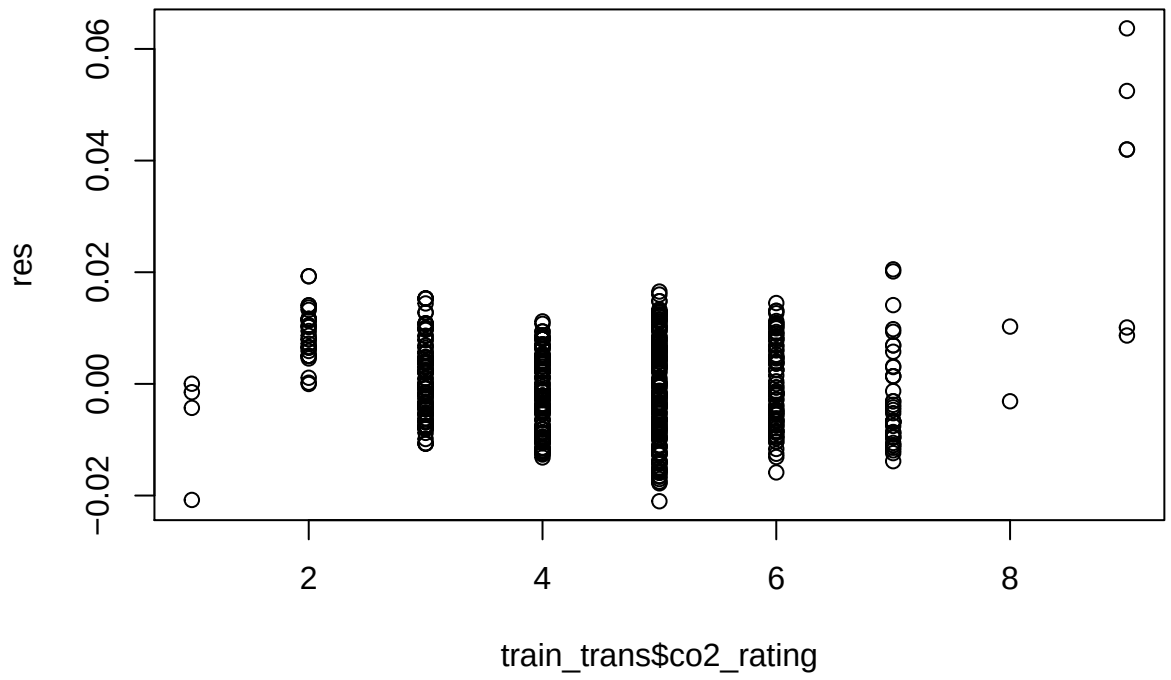

```
# Condition 2: draw scatter plots between predictors (numerical predictors)
pairs(~trans_smog_rating+co2_rating+trans_Cylinders, data=train_trans)
```



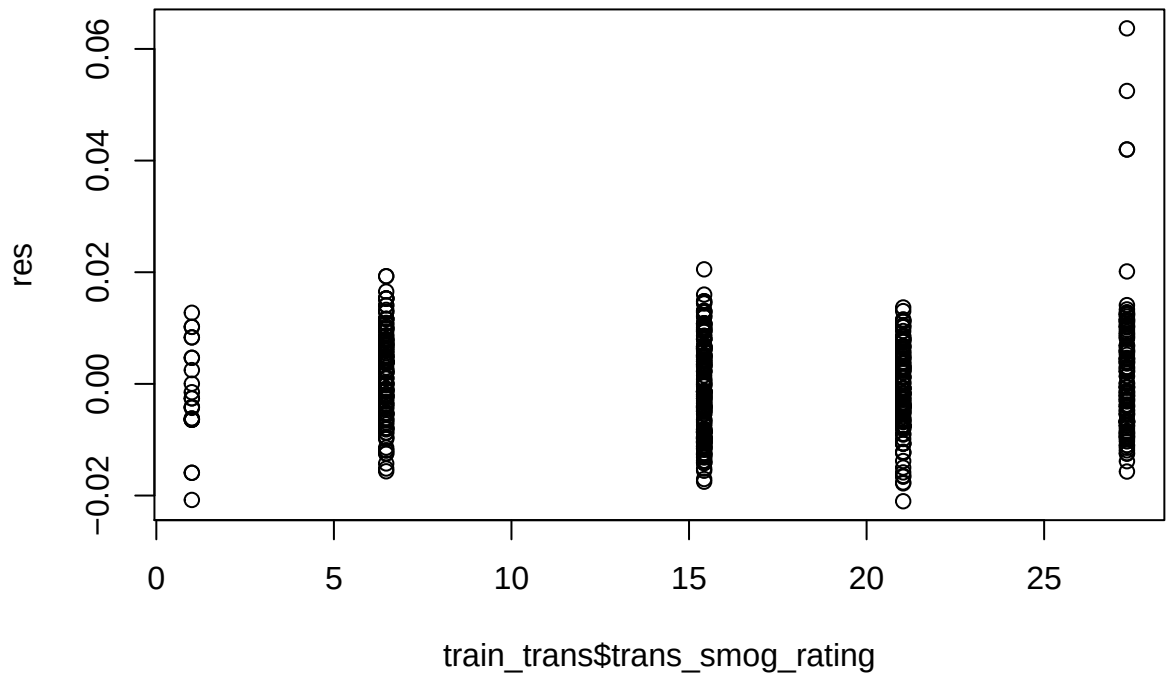

```
## Residual vs. Fitted model
res <- model_full_trans2$residuals
y_hat <- fitted(model_full_trans2)
plot(y_hat, res)
```



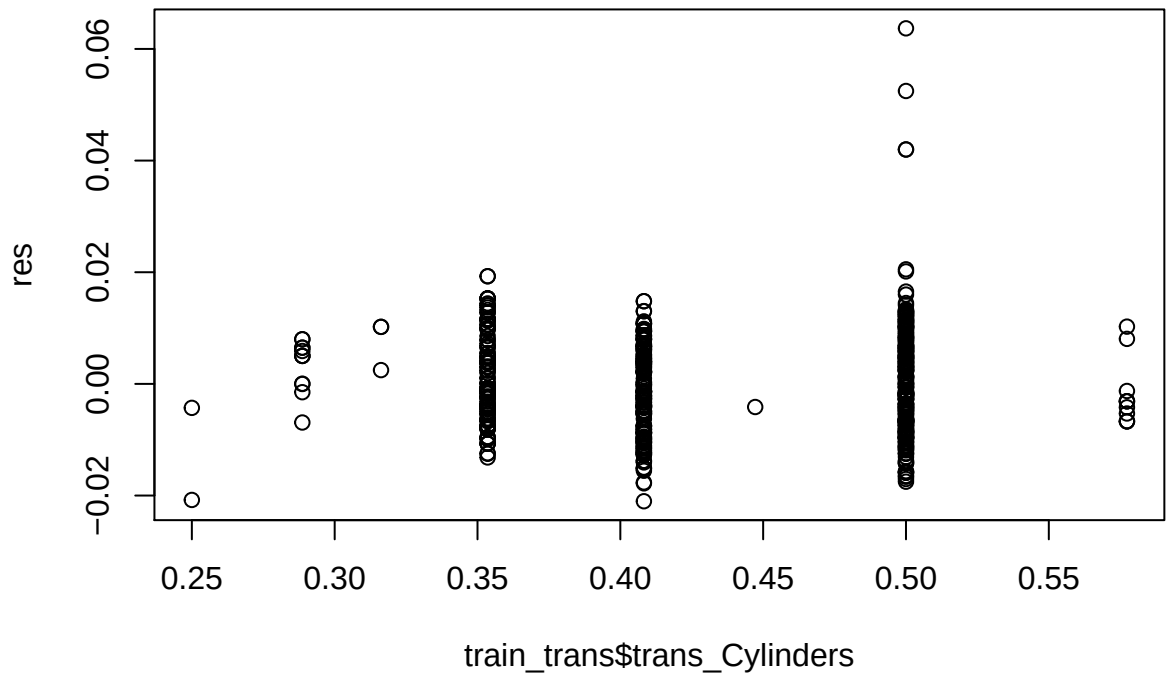

```
## Residual vs. Predictors
plot(train_trans$co2_rating, res)
```



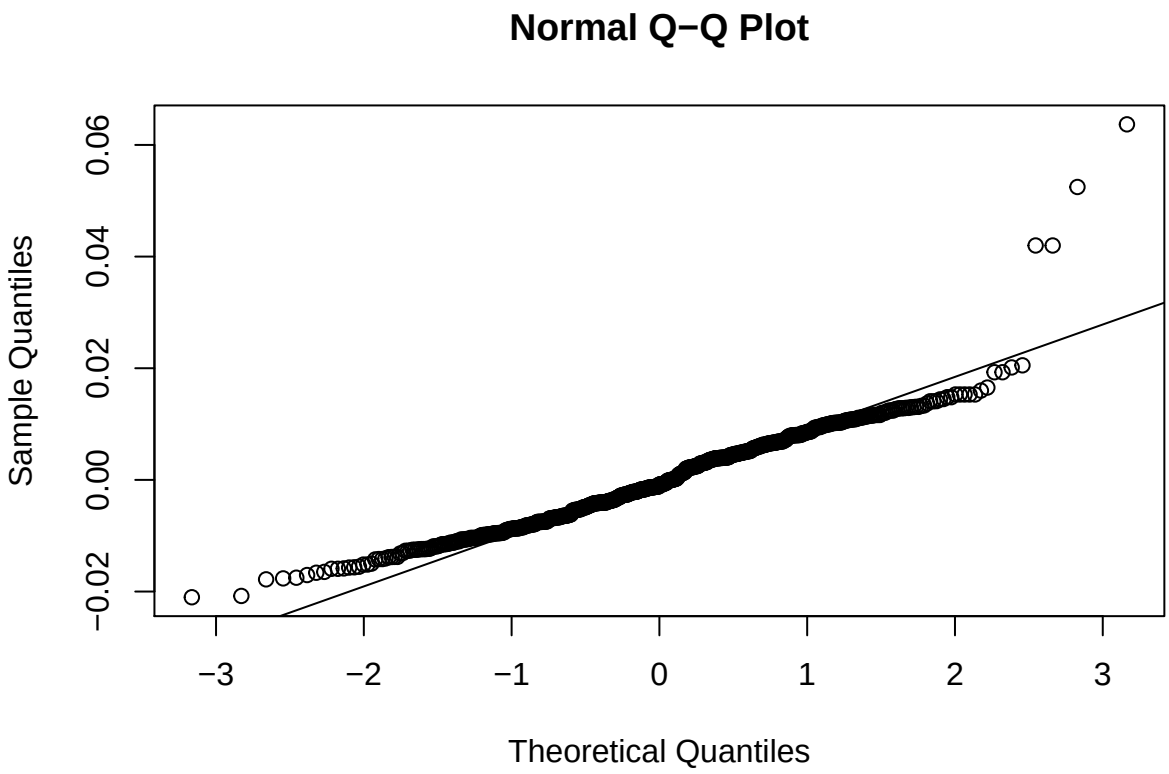

```
plot(train_trans$trans_smog_rating,res)
```




```
plot(train_trans$trans_Cylinders,res)
```

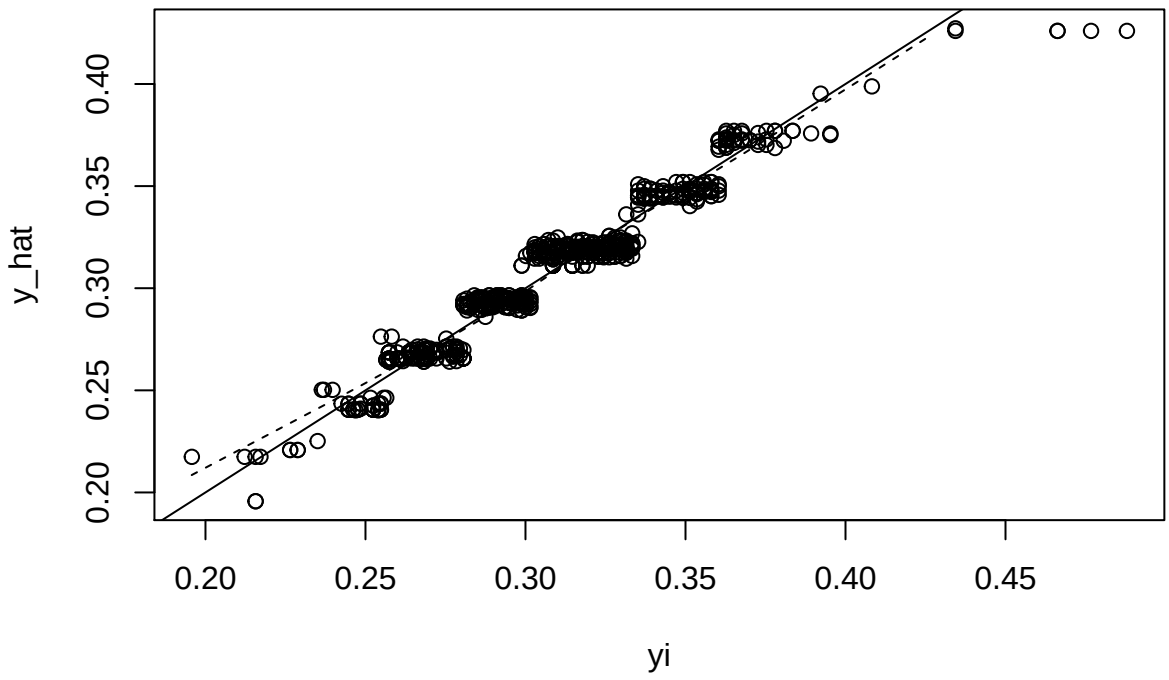



```
# Use normal QQ plot check normality
qqnorm(res)
qqline(res)
```

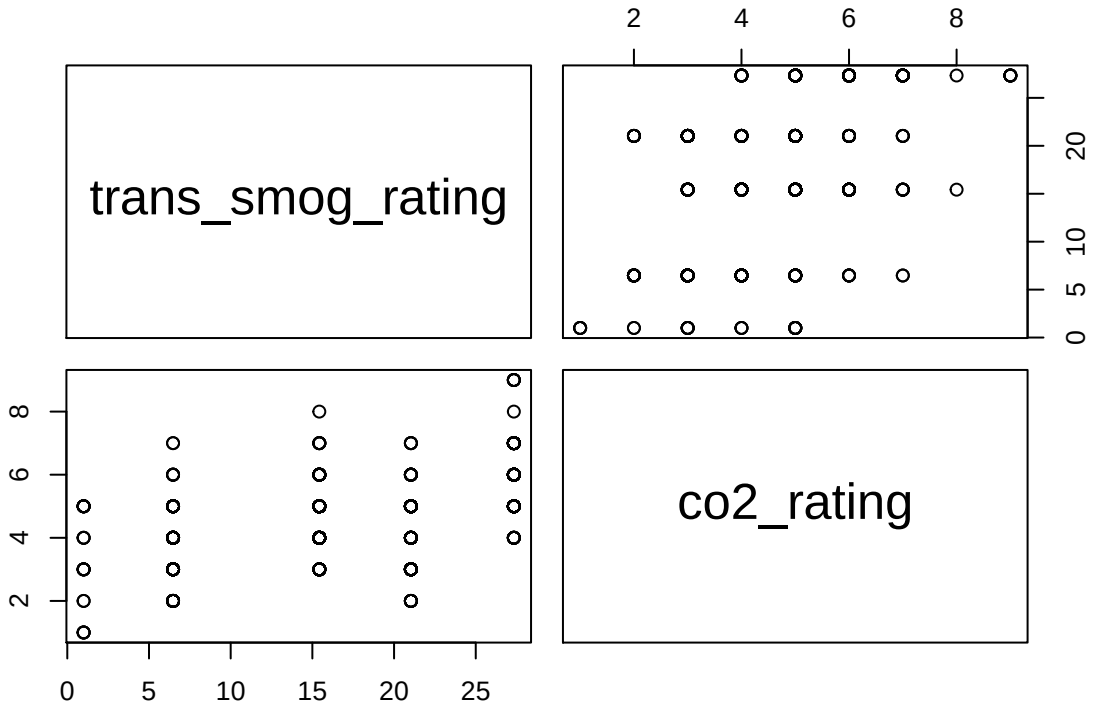


Check the Condition 1&2 and assumptions of reduced model

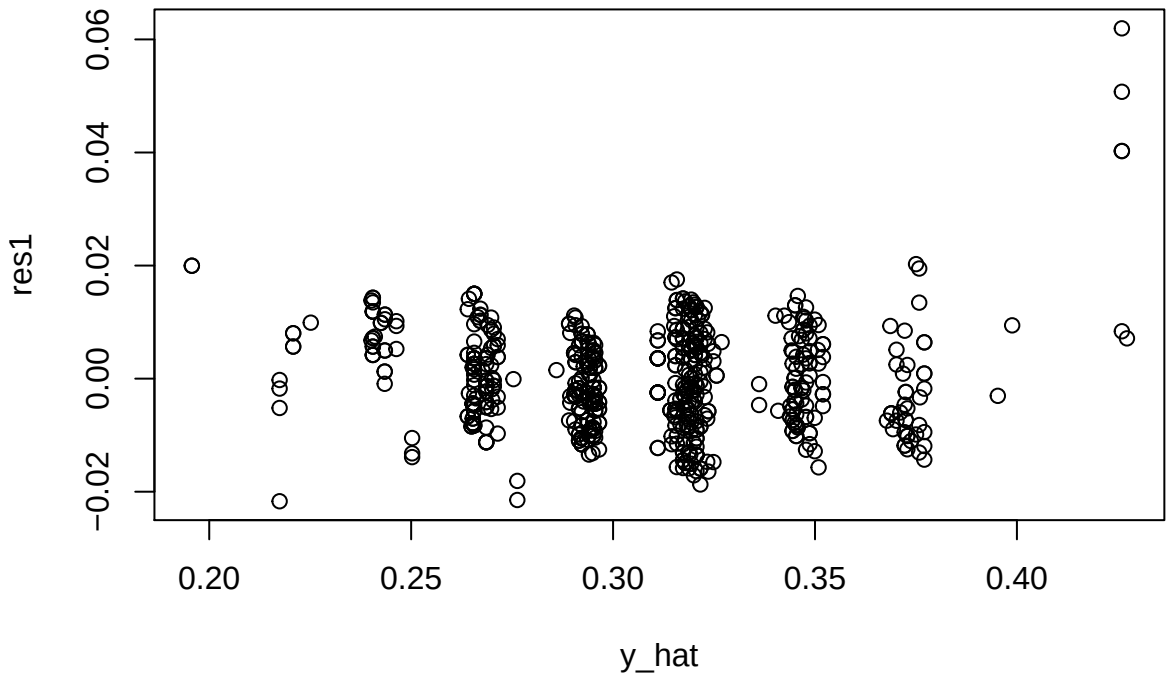
```
## Condition 1: draw a scatter plot between yi and y_hat
y_hat <- fitted(model_reduced2)
yi <- train_trans$trans_fuel_consumption_comb
plot(yi,y_hat)
abline(a = 0, b = 1)
lines(lowess(yi ~ y_hat), lty=2)
```



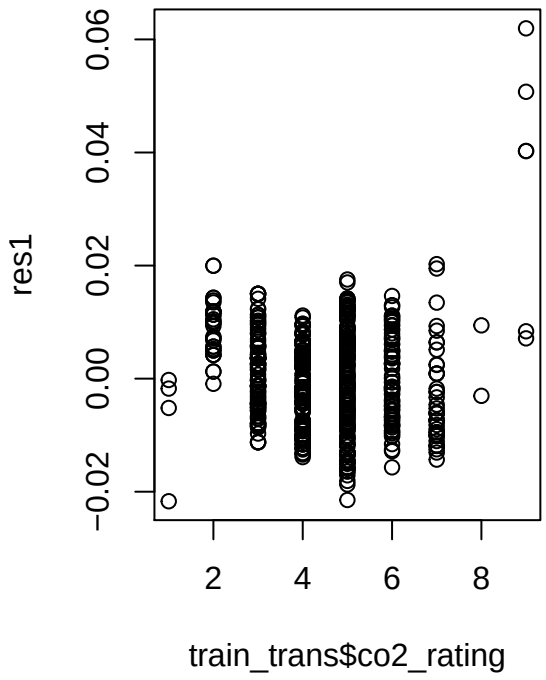

```
# Condition 2: draw scatterplots between predictors (can only be done for numerical predictors)
pairs(~trans_smog_rating+co2_rating, data=train_trans)
```

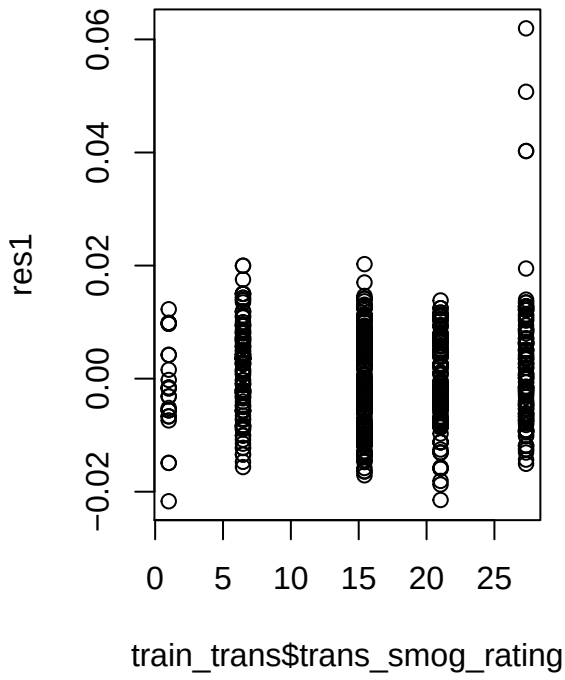



```
## Residual vs. Fitted
res1 <- model_reduced2$residuals
y_hat <- fitted(model_reduced2)
plot(y_hat, res1)
```

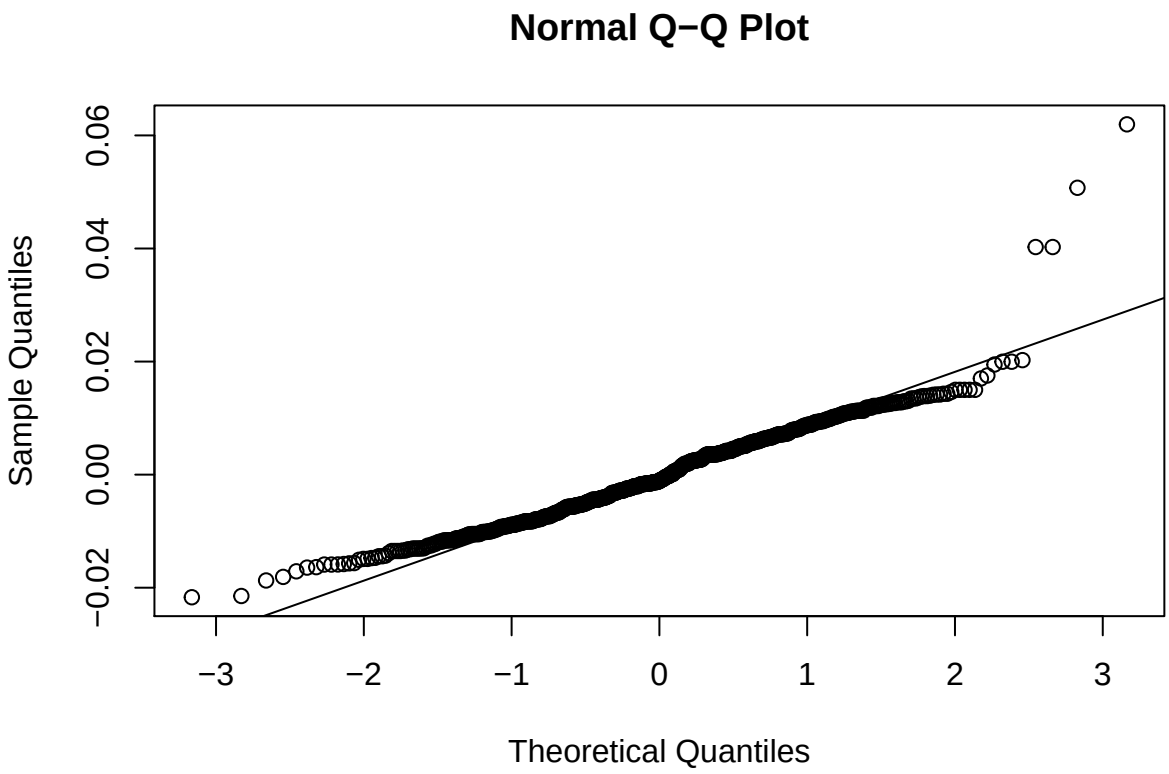



```
## Residual vs. Predictors
par(mfrow = c(1, 2))
plot(train_trans$co2_rating, res1)
plot(train_trans$trans_smog_rating,res1)
```






```r
# Use normal QQ plot check normality
qqnorm(res1)
qqline(res1)
```



Compare the R^2, AIC, and BIC

```
summary(model_full_trans2)$adj.r.squared
```

```
## [1] 0.9446026
```

```
summary(model_reduced2)$adj.r.squared
```

```
## [1] 0.94441
```

```
AIC(model_full_trans2)
```

```
## [1] -4215.099
```

```
AIC(model_reduced2)
```

```
## [1] -4213.854
```

```
BIC(model_full_trans2)
```

```
## [1] -4161.524
```

```
BIC(model_reduced2)
```

```
## [1] -4164.743
```

Step 6:Leverage Point, Outlier, and Influential Point

```
# Leverage Point
h <- hatvalues(model_full_trans2)
```

```r
Hcut <- 2*(length(model_full_trans2$coefficients)/nrow(train_trans))
which(h > Hcut)
```

```
##  14  18  27  34  48  78  80  87  97 113 137 172 184 213 216 251 252 270 274 281
##  14  18  27  34  48  78  80  87  97 113 137 172 184 213 216 251 252 270 274 281
## 285 289 295 297 300 306 338 356 371 374 377 428 447 466 479 494 502 512 543 547
## 285 289 295 297 300 306 338 356 371 374 377 428 447 466 479 494 502 512 543 547
## 573 578 589 592 596 610 611 613 635
## 573 578 589 592 596 610 611 613 635
```

```r
 # Outlier
r <- rstandard(model_full_trans2)
which(r < -4 | r > 4)
```

```
##  78 113 589 596
##  78 113 589 596
```

```r
# Cooks's Distance
D <- cooks.distance(model_full_trans2)
Dcut <- qf(0.5,
              length(model_full_trans2$coefficients),
           nrow(train_trans)-length(model_full_trans2$coefficients))
which(D > Dcut)
```

```
## named integer(0)
```

```r
# Leverage Point
h <- hatvalues(model_reduced2)
Hcut <- 2*(length(model_reduced$coefficients)/nrow(train_trans))
which(h > Hcut)
```

```
##   5   6   7  12  14  16  18  23  25  27  30  34  42  47  48  49  51  53  58  63
##   5   6   7  12  14  16  18  23  25  27  30  34  42  47  48  49  51  53  58  63
##  78  80  85  87  91  95  97 113 126 128 130 133 137 157 160 170 172 184 190 197
##  78  80  85  87  91  95  97 113 126 128 130 133 137 157 160 170 172 184 190 197
## 204 213 216 217 229 246 247 251 252 261 262 270 272 274 281 285 288 289 295 297
## 204 213 216 217 229 246 247 251 252 261 262 270 272 274 281 285 288 289 295 297
## 300 306 307 320 322 332 336 338 340 341 344 355 356 357 358 371 374 377 391 398
## 300 306 307 320 322 332 336 338 340 341 344 355 356 357 358 371 374 377 391 398
## 399 401 409 415 422 426 428 430 447 459 463 465 466 479 487 489 491 494 497 499
## 399 401 409 415 422 426 428 430 447 459 463 465 466 479 487 489 491 494 497 499
## 502 507 512 519 523 543 547 551 554 558 560 571 573 575 578 583 587 588 589 592
## 502 507 512 519 523 543 547 551 554 558 560 571 573 575 578 583 587 588 589 592
## 596 605 607 609 610 611 613 621 624 630 631 635 637 641
## 596 605 607 609 610 611 613 621 624 630 631 635 637 641
```

```r
 # Outlier
r <- rstandard(model_reduced2)
which(r < -4 | r > 4)
```

```
##  78 113 589 596
##  78 113 589 596
```

```r
# Cooks's Distance
D <- cooks.distance(model_reduced2)
Dcut <- qf(0.5,
              length(model_reduced2$coefficients),
           nrow(train_trans)-length(model_reduced2$coefficients))
```

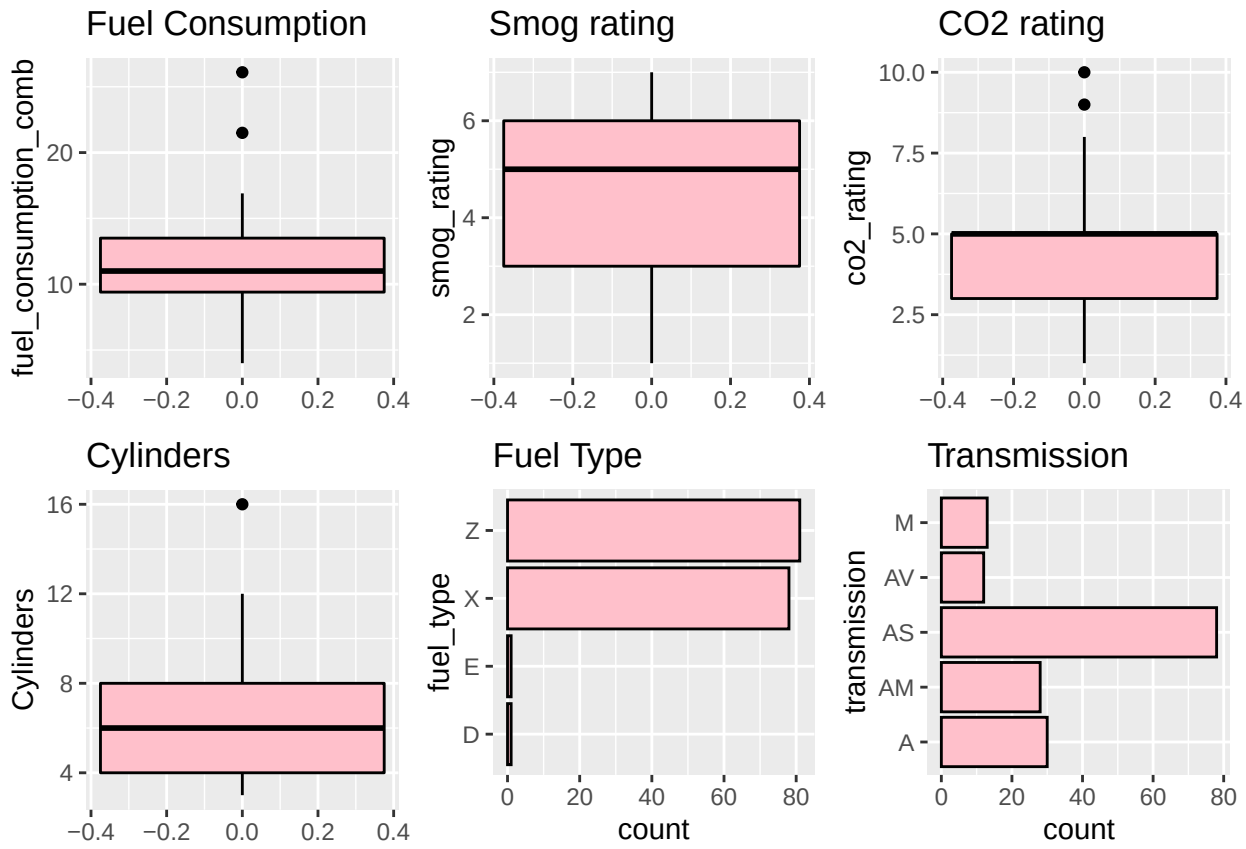
```
which(D > Dcut)
```

```
## named integer(0)
```

Step 7: Model Validation

Compare EDA

```
bar_transmission <- test %>%
  ggplot(aes(x=transmission)) +
  geom_bar(color='black', fill='pink') +
  labs(title="Transmission") +
  coord_flip()

bar_fuel_type <- test %>%
  ggplot(aes(x=fuel_type)) +
  geom_bar(color='black', fill='pink') +
  labs(title="Fuel Type") +
  coord_flip()
box_fuel_cons <- test %>%
  ggplot(aes(y=fuel_consumption_comb)) +
  geom_boxplot(color= "black", fill="pink") +
  labs(title="Fuel Consumption")
box_smog_rating <- test %>%
  ggplot(aes(y=smog_rating)) +
  geom_boxplot(color="black", fill="pink") +
  labs(title="Smog rating")

box_co2_rating <- test %>%
  ggplot(aes(y=co2_rating)) +
  geom_boxplot(color="black", fill="pink") +
  labs(title="CO2 rating")

box_cylinders <- test %>%
  ggplot(aes(y=Cylinders)) +
  geom_boxplot(color="black", fill="pink") +
  labs(title="Cylinders")

grid.arrange(box_fuel_cons,box_smog_rating,box_co2_rating,box_cylinders,bar_fuel_type,bar_transmission,
```



###Apply the same transformation on testing data

```
test_trans <- test %>%
  mutate(trans_fuel_consumption_comb = fuel_consumption_comb^(-0.5),
         trans_engine_size =log(engine_size),
         trans_Cylinders = Cylinders^(-0.5),
         trans_co2_emissions = co2_emissions^(-0.33),
         trans_smog_rating = smog_rating^(1.7))
```

###Refit the model on testing dat

```
model_full_test <- lm(trans_fuel_consumption_comb ~ trans_smog_rating + transmission + trans_Cylinders+
```

Compare the coefficients of training model and testing model

```
summary(model_full_trans2)
```

```
##
## Call:
## lm(formula = trans_fuel_consumption_comb ~ trans_smog_rating +
##     transmission + trans_Cylinders + co2_rating + fuel_type,
##     data = train_trans)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.021023 -0.006644 -0.000865  0.006011  0.063686
##
```

```
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        2.049e-01  3.535e-03  57.966  < 2e-16 ***
## trans_smog_rating  1.533e-04  6.157e-05   2.489 0.013058 *
## transmissionAM     5.526e-03  1.387e-03   3.984 7.56e-05 ***
## transmissionAS     2.666e-03  1.065e-03   2.504 0.012524 *
## transmissionAV     6.973e-03  1.783e-03   3.911 0.000102 ***
## transmissionM      1.748e-03  1.465e-03   1.193 0.233237
## trans_Cylinders    1.616e-02  9.038e-03   1.788 0.074245 .
## co2_rating         2.456e-02  4.542e-04  54.078  < 2e-16 ***
## fuel_typeE        -6.434e-02  3.344e-03 -19.241  < 2e-16 ***
## fuel_typeX        -1.947e-02  2.404e-03  -8.099 2.86e-15 ***
## fuel_typeZ        -2.264e-02  2.386e-03  -9.486  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.008989 on 631 degrees of freedom
## Multiple R-squared:  0.9455, Adjusted R-squared:  0.9446
## F-statistic:  1094 on 10 and 631 DF,  p-value: < 2.2e-16
```

```
summary(model_full_test)
```

```
##
## Call:
## lm(formula = trans_fuel_consumption_comb ~ trans_smog_rating +
##     transmission + trans_Cylinders + co2_rating + fuel_type,
##     data = test_trans)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.021010 -0.004554  0.000000  0.004859  0.039015
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.2096897  0.0102065  20.545  < 2e-16 ***
## trans_smog_rating  0.0002793  0.0001206   2.316  0.02189 *
## transmissionAM     0.0084263  0.0026961   3.125  0.00213 **
## transmissionAS     0.0037115  0.0020550   1.806  0.07291 .
## transmissionAV     0.0057024  0.0033917   1.681  0.09479 .
## transmissionM     -0.0013993  0.0031359  -0.446  0.65609
## trans_Cylinders   -0.0220062  0.0182026  -1.209  0.22858
## co2_rating         0.0265233  0.0008696  30.501  < 2e-16 ***
## fuel_typeE        -0.0587159  0.0125415  -4.682 6.31e-06 ***
## fuel_typeX        -0.0189943  0.0089985  -2.111  0.03645 *
## fuel_typeZ        -0.0226666  0.0090380  -2.508  0.01321 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.008717 on 150 degrees of freedom
## Multiple R-squared:  0.9573, Adjusted R-squared:  0.9545
## F-statistic: 336.6 on 10 and 150 DF,  p-value: < 2.2e-16
```

```
Variable <-(c("Intercept","Smog Rating","TransmissionAM","TransmissionAS","TransmissionAV","Transmissio
Coefficient_Estimate_Train <-(c("0.2049","0.0001533","0.005526","0.002666","0.006973","0.001748","0.016
Coefficient_Estimate_Test <-(c("0.2096897","0.0002793","0.0084263","0.0037115","0.0057024","-0.0013993"
```

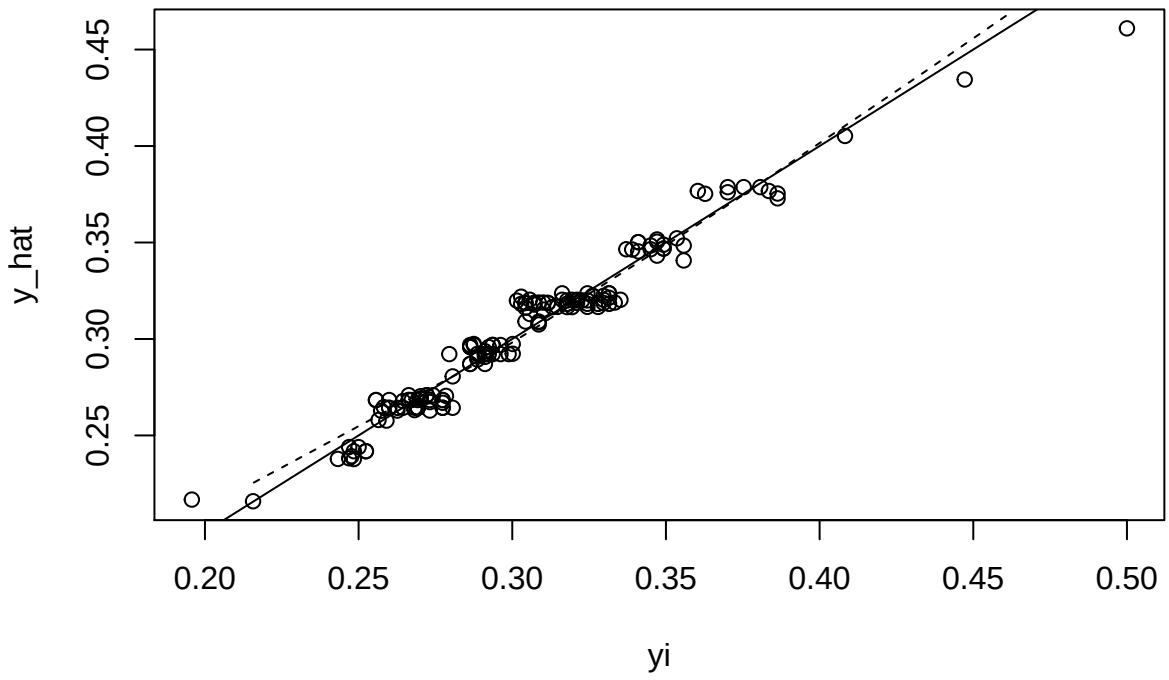
```
knitr::kable(tibble(Variable,Coefficient_Estimate_Train,Coefficient_Estimate_Test),caption="The Summary
```

Table 2: The Summary of Coefficient Estimate for Train and Test

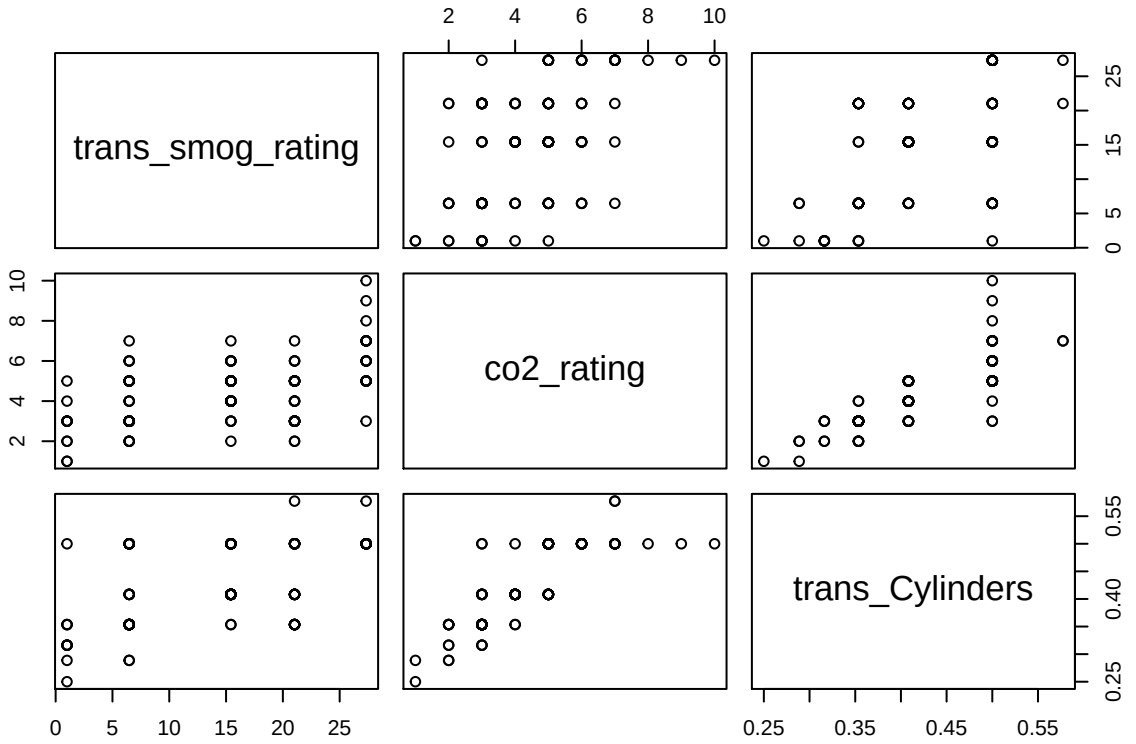
Variable	Coefficient_Estimate_Train	Coefficient_Estimate_Test
Intercept	0.2049	0.2096897
Smog Rating	0.0001533	0.0002793
TransmissionAM	0.005526	0.0084263
TransmissionAS	0.002666	0.0037115
TransmissionAV	0.006973	0.0057024
TransmissionM	0.001748	-0.0013993
Cylinders	0.01616	-0.0220062
CO2 Rating	0.02456	0.0265233
Fuel Type E	-0.06434	-0.0587159
Fuel Type X	-0.01947	-0.0189943
Fuel Type Z	-0.02264	-0.0226666

Check condition 1&2 and assumptions of test model

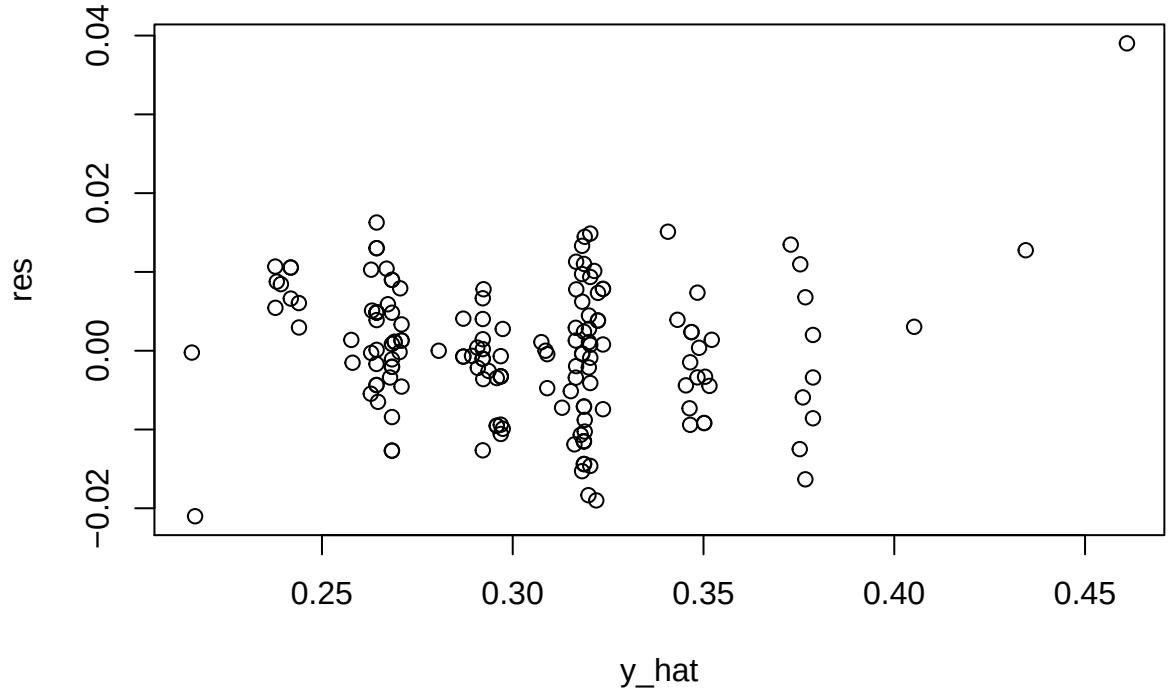
```
# Condition 1
y_hat <- fitted(model_full_test)
yi <- test_trans$trans_fuel_consumption_comb
plot(yi,y_hat)
abline(a = 0, b = 1)
lines(lowess(yi ~ y_hat), lty=2)
```



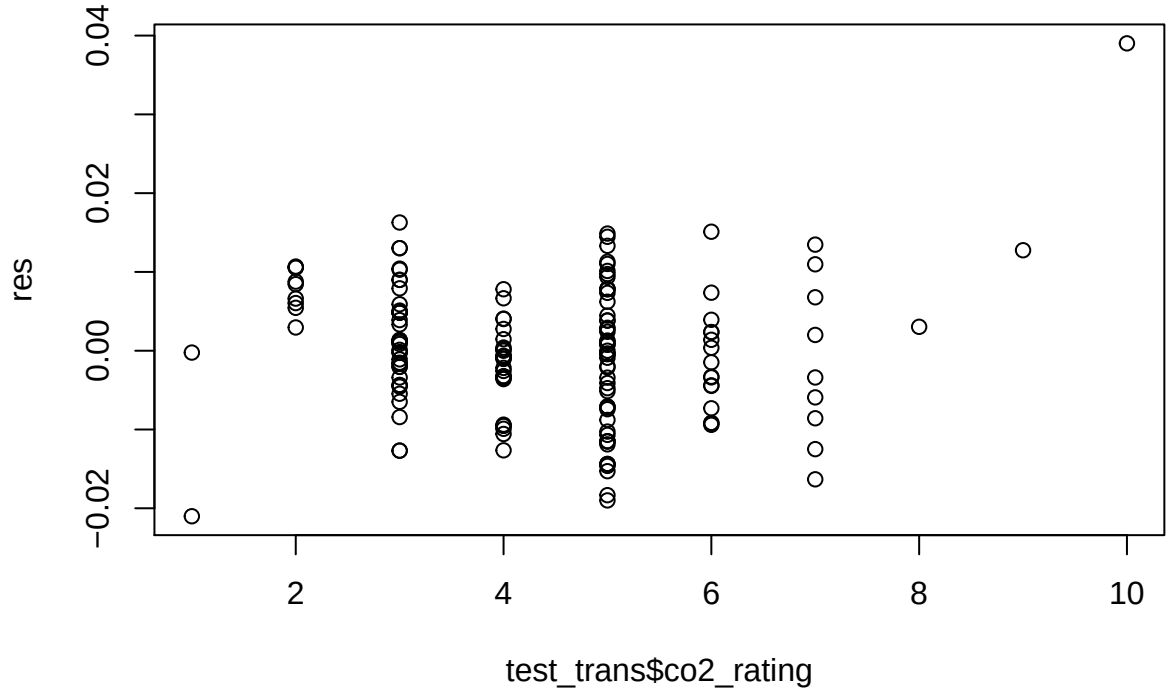

```
# Condition 2: draw scatt erplots between predictors (numerical predictors)
pairs(~trans_smog_rating+co2_rating+trans_Cylinders, data=test_trans)
```



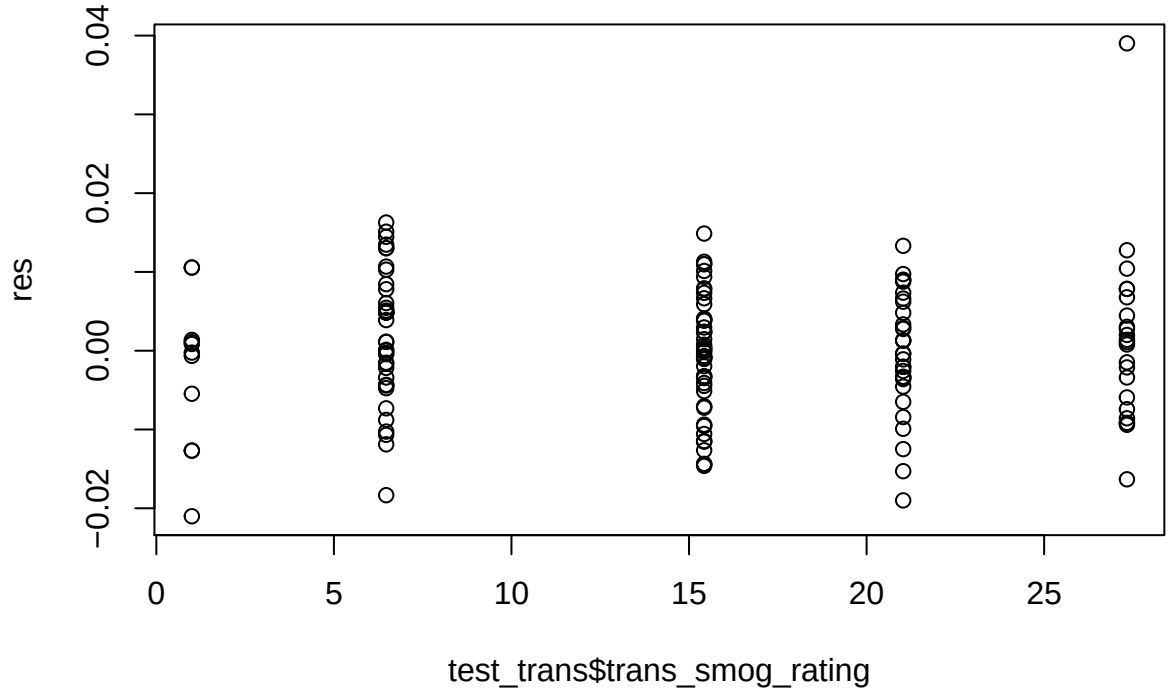

```
## Residual vs. Fitted
res <- model_full_test$residuals
y_hat <- fitted(model_full_test)
plot(y_hat, res)
```



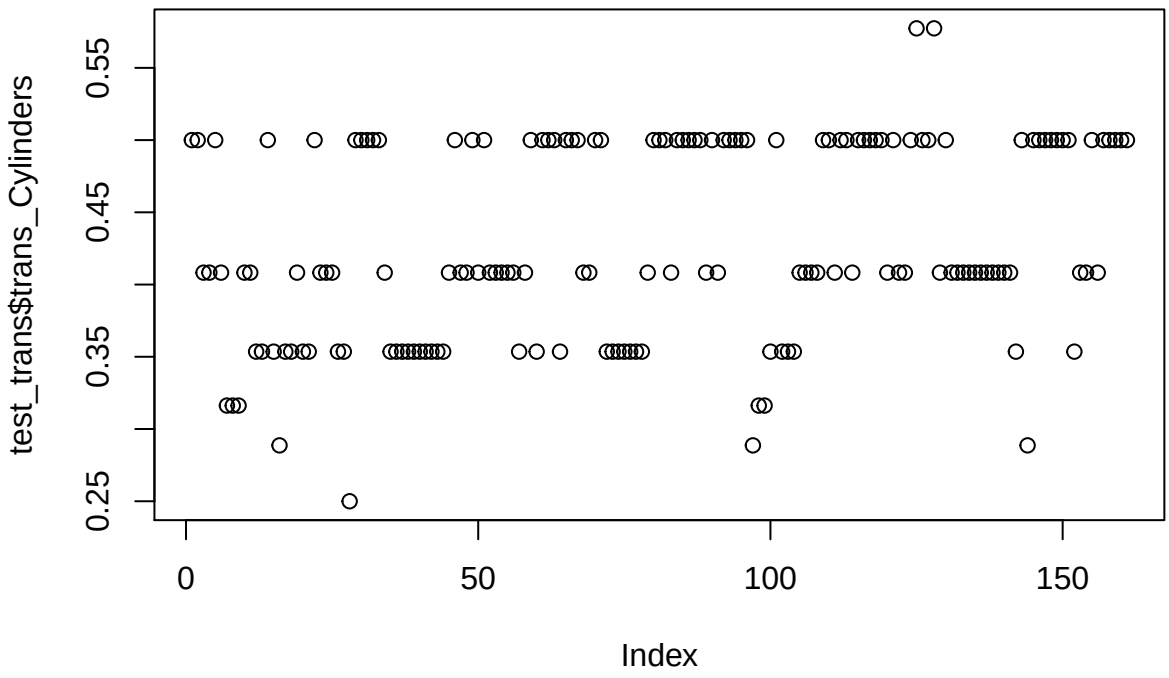

```
## Residual vs. Predictors
plot(test_trans$co2_rating, res)
```



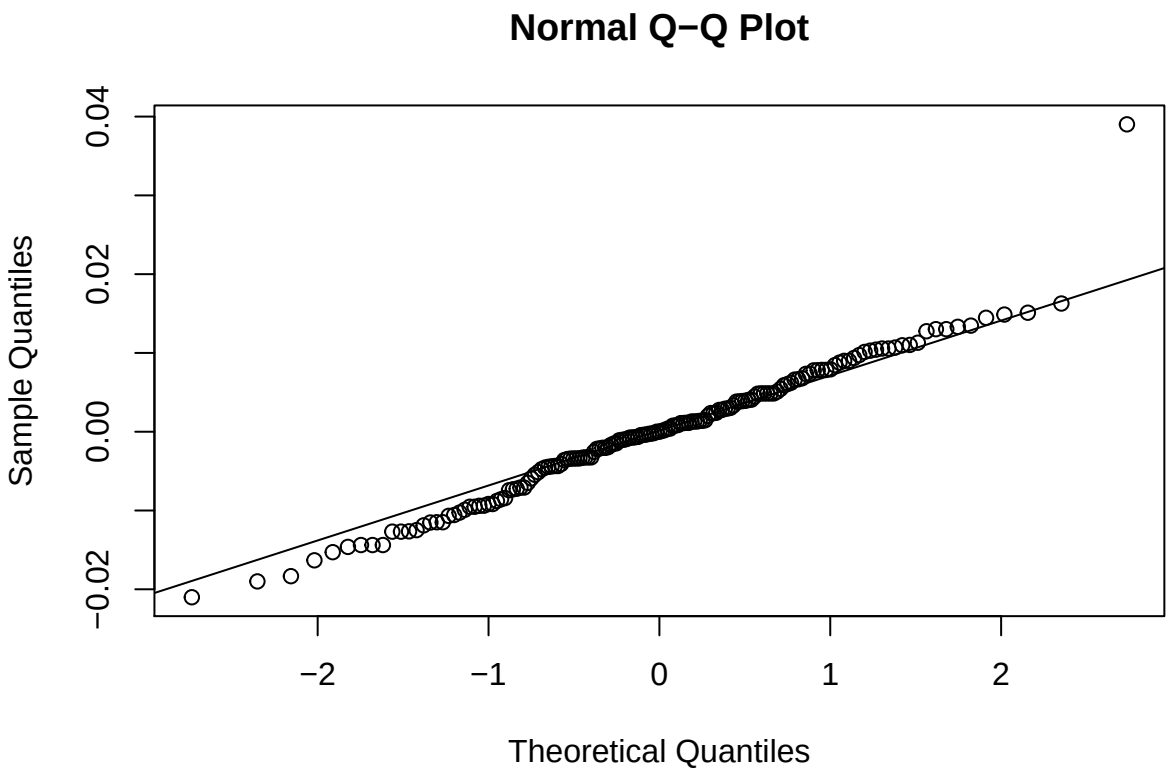

```
plot(test_trans$trans_smog_rating,res)
```




```
plot(test_trans$trans_Cylinders)
```




```
# Use normal QQ plot check normality
qqnorm(res)
qqline(res)
```


Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles
0.04
0.02
0.00
−0.02
−2
−1
0
1
2