**Fuel Economy Data**

**Metadata Updated:** [June 23, 2017](https://catalog.data.gov/dataset/fuel-economy-data#sec-dates) 

Fuel economy data are the result of vehicle testing done at the Environmental Protection Agency's National Vehicle and Fuel Emissions Laboratory in Ann Arbor, Michigan, and by vehicle manufacturers with oversight by EPA. The Find a Car vehicle table contains fuel economy information for 1984-current model year vehicles. The data are available for download in CSV and XML formats:

<https://catalog.data.gov/dataset/fuel-economy-data>

<https://fueleconomy.gov/feg/ws/>

**Variables description:**

Var 1: barrels08, Type: numeric, Low/High: (0.0600, 47.0871)

Var 2: barrelsA08, Type: numeric, Low/High: (0.0000, 18.3117)

Var 3: charge120, Type: numeric, Low/High: (0.0000, 0.0000)

Var 4: charge240, Type: numeric, Low/High: (0.0000, 13.0000)

Var 5: city08, Type: integer, Low/High: (6, 150)

Var 6: city08U, Type: numeric, Low/High: (0.0000, 150.0000)

Var 7: cityA08, Type: integer, Low/High: (0, 145)

Var 8: cityA08U, Type: numeric, Low/High: (0.0000, 145.0835)

Var 9: cityCD, Type: numeric, Low/High: (0.0000, 5.3500)

Var 10: cityE, Type: numeric, Low/High: (0.0000, 122.0000)

Var 11: cityUF, Type: numeric, Low/High: (0.0000, 0.9270)

Var 12: co2, Type: integer, Low/High: (-1, 847)

Var 13: co2A, Type: integer, Low/High: (-1, 713)

Var 14: co2TailpipeAGpm, Type: numeric, Low/High: (0.0000, 713.0000)

Var 15: co2TailpipeGpm, Type: numeric, Low/High: (0.0000, 1269.5714)

Var 16: comb08, Type: integer, Low/High: (7, 136)

Var 17: comb08U, Type: numeric, Low/High: (0.0000, 136.0000)

Var 18: combA08, Type: integer, Low/High: (0, 133)

Var 19: combA08U, Type: numeric, Low/High: (0.0000, 133.2662)

Var 20: combE, Type: numeric, Low/High: (0.0000, 121.0000)

Var 21: combinedCD, Type: numeric, Low/High: (0.0000, 4.8000)

Var 22: combinedUF, Type: numeric, Low/High: (0.0000, 0.9200)

Var 23: cylinders, Type: integer, Low/High: (2, 16)

Var 24: displ, Type: numeric, Low/High: (0.0000, 8.4000)

Var 25: drive

8 factor levels: 2-Wheel Drive 4-Wheel Drive 4-Wheel or All-Wheel Drive All-Wheel Drive Front-Wheel Drive Part-time 4-Wheel Drive Rear-Wheel Drive

Var 26: engId, Type: integer, Low/High: (0, 69102)

Var 27: eng\_dscr

558 factor levels: (121) (FFS) (122) (FFS) (16-VALVE) (FFS) (16-VALVE) (FFS) (MPFI) ... VTEC VTEC-E VTEC-E (FFS) VTEC (FFS) Z/28

Var 28: feScore, Type: integer, Low/High: (-1, 10)

Var 29: fuelCost08, Type: integer, Low/High: (500, 7050)

Var 30: fuelCostA08, Type: integer, Low/High: (0, 3900)

Var 31: fuelType

14 factor levels: CNG Diesel Electricity Gasoline or E85 Gasoline or natural gas ... Premium Gas or Electricity Premium or E85 Regular Regular Gas and Electricity Regular Gas or Electricity

Var 32: fuelType1

6 factor levels: Diesel Electricity Midgrade Gasoline Natural Gas Premium Gasoline Regular Gasoline

Var 33: ghgScore, Type: integer, Low/High: (-1, 10)

Var 34: ghgScoreA, Type: integer, Low/High: (-1, 8)

Var 35: highway08, Type: integer, Low/High: (9, 123)

Var 36: highway08U, Type: numeric, Low/High: (0.0000, 123.3400)

Var 37: highwayA08, Type: integer, Low/High: (0, 121)

Var 38: highwayA08U, Type: numeric, Low/High: (0.0000, 121.2005)

Var 39: highwayCD, Type: numeric, Low/High: (0.0000, 4.0600)

Var 40: highwayE, Type: numeric, Low/High: (0.0000, 120.0000)

Var 41: highwayUF, Type: numeric, Low/High: (0.0000, 0.9100)

Var 42: hlv, Type: integer, Low/High: (0, 49)

Var 43: hpv, Type: integer, Low/High: (0, 195)

Var 44: id, Type: integer, Low/High: (1, 41372)

Var 45: lv2, Type: integer, Low/High: (0, 41)

Var 46: lv4, Type: integer, Low/High: (0, 55)

Var 47: make

136 factor levels: Acura Alfa Romeo AM General American Motors Corporation ASC Incorporated ... Volkswagen Volvo VPG Wallace Environmental Yugo

Var 48: model

4040 factor levels: 1-Ton Truck 2WD 100 100 quattro 100 quattro Wagon 100 Wagon ... Z4 sDrive35i Z4 sDrive35is Z8 ZDX 4WD Zephyr

Var 49: mpgData

2 factor levels: N Y

Var 50: phevBlended

2 factor levels: false true

Var 51: pv2, Type: integer, Low/High: (0, 194)

Var 52: pv4, Type: integer, Low/High: (0, 192)

Var 53: range, Type: integer, Low/High: (0, 335)

Var 54: rangeCity, Type: numeric, Low/High: (0.0000, 333.1115)

Var 55: rangeCityA, Type: numeric, Low/High: (0.0000, 135.2800)

Var 56: rangeHwy, Type: numeric, Low/High: (0.0000, 346.9000)

Var 57: rangeHwyA, Type: numeric, Low/High: (0.0000, 114.7600)

Var 58: trany

38 factor levels: Automatic (A1) Automatic (AM-S6) Automatic (AM-S7) Automatic (AM-S8) ... Manual 4-spd Manual 4-spd Doubled Manual 5-spd Manual 6-spd Manual 7-spd

Var 59: UCity, Type: numeric, Low/High: (0.0000, 224.8000)

Var 60: UCityA, Type: numeric, Low/High: (0.0000, 207.2622)

Var 61: UHighway, Type: numeric, Low/High: (0.0000, 182.7000)

Var 62: UHighwayA, Type: numeric, Low/High: (0.0000, 173.1436)

Var 63: VClass

34 factor levels: Compact Cars Large Cars Midsize-Large Station Wagons Midsize Cars Midsize Station Wagons ... Two Seaters Vans Vans Passenger Vans, Cargo Type Vans, Passenger Type

Var 64: year, Type: integer, Low/High: (1984, 2020)

Var 65: youSaveSpend, Type: integer, Low/High: (-27750, 5000)

Var 66: guzzler

4 factor levels: G S T

Var 67: trans\_dscr

53 factor levels: 2LKUP 2MODE 2MODE 2LKUP 2MODE 3LKUP ... SIL EMS VLKUP VMODE VMODE CLKUP VMODE VLKUP

Var 68: tCharger, Type: logical, Low/High: (1, 1)

Var 69: sCharger

2 factor levels: S

Var 70: atvType

9 factor levels: Bifuel (CNG) Bifuel (LPG) CNG Diesel EV FFV Hybrid Plug-in Hybrid

Var 71: fuelType2

5 factor levels: E85 Electricity Natural Gas Propane

Var 72: rangeA

229 factor levels: 10 11 110 119 ... 53 69 72 9 97

Var 73: evMotor

149 factor levels: 100 kW AC PMSM 100 kW DCPM 101V Ni-MH 102kW AC Induction ... 95 kW and 116 kW DC Brushless 96kW 96kW AC Induction Two 444V Li-Ion Two 480V Li-Ion

Var 74: mfrCode

49 factor levels: ADX ASX AZD BEX ... TVP TYX VGA VVX VWX

Var 75: c240Dscr

6 factor levels: 3.6 kW charger 6.6 kW charger 7.2 kW charger single charger standard charger

Var 76: charge240b, Type: numeric, Low/High: (0.0000, 8.5000)

Var 77: c240bDscr

5 factor levels: 3.6 kW charger 6.6 kW charger 80 amp dual charger dual charger

Var 78: createdOn

262 factor levels: Fri Apr 11 00:00:00 EDT 2014 Fri Apr 25 00:00:00 EDT 2014 Fri Aug 03 00:00:00 EDT 2018 Fri Aug 10 00:00:00 EDT 2018 Fri Aug 17 00:00:00 EDT 2018 ... Wed Oct 28 00:00:00 EDT 2015 Wed Sep 09 00:00:00 EDT 2015 Wed Sep 10 00:00:00 EDT 2014 Wed Sep 16 00:00:00 EDT 2015 Wed Sep 27 00:00:00 EDT 2017

Var 79: modifiedOn

131 factor levels: Fri Apr 01 00:00:00 EDT 2016 Fri Apr 11 00:00:00 EDT 2014 Fri Aug 03 00:00:00 EDT 2018 Fri Aug 10 00:00:00 EDT 2018 Fri Aug 17 00:00:00 EDT 2018 ... Wed Oct 12 00:00:00 EDT 2016 Wed Oct 17 00:00:00 EDT 2018 Wed Oct 19 00:00:00 EDT 2016 Wed Oct 26 00:00:00 EDT 2016 Wed Sep 27 00:00:00 EDT 2017

Var 80: startStop

3 factor levels: N Y

Var 81: phevCity, Type: integer, Low/High: (0, 97)

Var 82: phevHwy, Type: integer, Low/High: (0, 81)

Var 83: phevComb, Type: integer, Low/High: (0, 88)

## Data Description

1. [atvtype](https://fueleconomy.gov/feg/ws/#atvtype)- type of alternative fuel or advanced technology vehicle
2. barrels08 - annual petroleum consumption in barrels for [fuelType1](https://fueleconomy.gov/feg/ws/#fuelType1) [(1)](https://fueleconomy.gov/feg/ws/#ft1)
3. barrelsA08 - annual petroleum consumption in barrels for [fuelType2](https://fueleconomy.gov/feg/ws/#fuelType2) [(1)](https://fueleconomy.gov/feg/ws/#ft1)
4. charge120 - time to charge an electric vehicle in hours at 120 V
5. charge240 - time to charge an electric vehicle in hours at 240 V
6. city08 - city MPG for [fuelType1](https://fueleconomy.gov/feg/ws/#fuelType1) [(2)](https://fueleconomy.gov/feg/ws/#ft2), [(11)](https://fueleconomy.gov/feg/ws/#ft11)
7. city08U - unrounded city MPG for [fuelType1](https://fueleconomy.gov/feg/ws/#fuelType1) [(2)](https://fueleconomy.gov/feg/ws/#ft2), [(3)](https://fueleconomy.gov/feg/ws/#ft3)
8. cityA08 - city MPG for [fuelType2](https://fueleconomy.gov/feg/ws/#fuelType2) [(2)](https://fueleconomy.gov/feg/ws/#ft2)
9. cityA08U - unrounded city MPG for [fuelType2](https://fueleconomy.gov/feg/ws/#fuelType2) [(2)](https://fueleconomy.gov/feg/ws/#ft2), [(3)](https://fueleconomy.gov/feg/ws/#ft3)
10. cityCD - city gasoline consumption (gallons/100 miles) in charge depleting mode [(4)](https://fueleconomy.gov/feg/ws/#ft4)
11. cityE - city electricity consumption in kw-hrs/100 miles
12. cityUF - EPA city utility factor (share of electricity) for PHEV
13. co2 - tailpipe CO2 in grams/mile for [fuelType1](https://fueleconomy.gov/feg/ws/#fuelType1) [(5)](https://fueleconomy.gov/feg/ws/#ft5)
14. co2A - tailpipe CO2 in grams/mile for [fuelType2](https://fueleconomy.gov/feg/ws/#fuelType2)[(5)](https://fueleconomy.gov/feg/ws/#ft5)
15. co2TailpipeAGpm - tailpipe CO2 in grams/mile for [fuelType2](https://fueleconomy.gov/feg/ws/#fuelType2) [(5)](https://fueleconomy.gov/feg/ws/#ft5)
16. co2TailpipeGpm- tailpipe CO2 in grams/mile for [fuelType1](https://fueleconomy.gov/feg/ws/#fuelType1) [(5)](https://fueleconomy.gov/feg/ws/#ft5)
17. comb08 - combined MPG for [fuelType1](https://fueleconomy.gov/feg/ws/#fuelType1) [(2),](https://fueleconomy.gov/feg/ws/#ft2)[(11)](https://fueleconomy.gov/feg/ws/#ft11)
18. comb08U - unrounded combined MPG for [fuelType1](https://fueleconomy.gov/feg/ws/#fuelType1) [(2)](https://fueleconomy.gov/feg/ws/#ft2), [(3)](https://fueleconomy.gov/feg/ws/#ft3)
19. combA08 - combined MPG for [fuelType2](https://fueleconomy.gov/feg/ws/#fuelType2) [(2)](https://fueleconomy.gov/feg/ws/#ft2)
20. combA08U - unrounded combined MPG for [fuelType2](https://fueleconomy.gov/feg/ws/#fuelType2) [(2)](https://fueleconomy.gov/feg/ws/#ft2), [(3)](https://fueleconomy.gov/feg/ws/#ft3)
21. combE - combined electricity consumption in kw-hrs/100 miles
22. combinedCD - combined gasoline consumption (gallons/100 miles) in charge depleting mode [(4)](https://fueleconomy.gov/feg/ws/#ft4)
23. combinedUF - EPA combined utility factor (share of electricity) for PHEV
24. cylinders - engine cylinders
25. displ - engine displacement in liters
26. [drive](https://fueleconomy.gov/feg/ws/#drive)- drive axle type
27. [emissionsList](https://fueleconomy.gov/feg/ws/#emissions)
28. engId - EPA model type index
29. eng\_dscr - engine descriptor; see http://www.fueleconomy.gov/feg/findacarhelp.shtml#engine
30. evMotor - electric motor (kw-hrs)
31. feScore - EPA Fuel Economy Score (-1 = Not available)
32. fuelCost08 - annual fuel cost for [fuelType1](https://fueleconomy.gov/feg/ws/#fuelType1) ($) [(7)](https://fueleconomy.gov/feg/ws/#ft7)
33. fuelCostA08 - annual fuel cost for [fuelType2](https://fueleconomy.gov/feg/ws/#fuelType2) ($) [(7)](https://fueleconomy.gov/feg/ws/#ft7)
34. fuelType - fuel type with [fuelType1](https://fueleconomy.gov/feg/ws/#fuelType1) and [fuelType2](https://fueleconomy.gov/feg/ws/#fuelType2) (if applicable)
35. fuelType1 - fuel type 1. For single fuel vehicles, this will be the only fuel. For dual fuel vehicles, this will be the conventional fuel.
36. fuelType2 - fuel type 2. For dual fuel vehicles, this will be the alternative fuel (e.g. E85, Electricity, CNG, LPG). For single fuel vehicles, this field is not used
37. ghgScore - EPA GHG score (-1 = Not available)
38. ghgScoreA - EPA GHG score for dual fuel vehicle running on the alternative fuel (-1 = Not available)
39. guzzler- if G or T, this vehicle is subject to the gas guzzler tax
40. highway08 - highway MPG for [fuelType1](https://fueleconomy.gov/feg/ws/#fuelType1) [(2)](https://fueleconomy.gov/feg/ws/#ft2), [(11)](https://fueleconomy.gov/feg/ws/#ft11)
41. highway08U - unrounded highway MPG for [fuelType1](https://fueleconomy.gov/feg/ws/#fuelType1) [(2)](https://fueleconomy.gov/feg/ws/#ft2), [(3)](https://fueleconomy.gov/feg/ws/#ft3)
42. highwayA08 - highway MPG for [fuelType2](https://fueleconomy.gov/feg/ws/#fuelType2) [(2)](https://fueleconomy.gov/feg/ws/#ft2)
43. highwayA08U - unrounded highway MPG for [fuelType2](https://fueleconomy.gov/feg/ws/#fuelType2) [(2)](https://fueleconomy.gov/feg/ws/#ft2),[(3)](https://fueleconomy.gov/feg/ws/#ft3)
44. highwayCD - highway gasoline consumption (gallons/100miles) in charge depleting mode [(4)](https://fueleconomy.gov/feg/ws/#ft4)
45. highwayE - highway electricity consumption in kw-hrs/100 miles
46. highwayUF - EPA highway utility factor (share of electricity) for PHEV
47. hlv - hatchback luggage volume (cubic feet) [(8)](https://fueleconomy.gov/feg/ws/#ft8)
48. hpv - hatchback passenger volume (cubic feet) [(8)](https://fueleconomy.gov/feg/ws/#ft8)
49. id - vehicle record id
50. lv2 - 2 door luggage volume (cubic feet) [(8)](https://fueleconomy.gov/feg/ws/#ft8)
51. lv4 - 4 door luggage volume (cubic feet) [(8)](https://fueleconomy.gov/feg/ws/#ft8)
52. make - manufacturer (division)
53. mfrCode - 3-character manufacturer code
54. model - model name (carline)
55. mpgData - has My MPG data; see [yourMpgVehicle](https://fueleconomy.gov/feg/ws/" \l "ympgVehicle) and [yourMpgDriverVehicle](https://fueleconomy.gov/feg/ws/" \l "ympgDriverVehicle)
56. phevBlended - if true, this vehicle operates on a blend of gasoline and electricity in charge depleting mode
57. pv2 - 2-door passenger volume (cubic feet) [(8)](https://fueleconomy.gov/feg/ws/#ft8)
58. pv4 - 4-door passenger volume (cubic feet) [(8)](https://fueleconomy.gov/feg/ws/#ft8)
59. rangeA - EPA range for [fuelType2](https://fueleconomy.gov/feg/ws/#fuelType2)
60. rangeCityA - EPA city range for [fuelType2](https://fueleconomy.gov/feg/ws/#fuelType2)
61. rangeHwyA - EPA highway range for [fuelType2](https://fueleconomy.gov/feg/ws/#fuelType2)
62. trans\_dscr - transmission descriptor; see http://www.fueleconomy.gov/feg/findacarhelp.shtml#trany
63. trany - transmission
64. UCity - unadjusted city MPG for [fuelType1](https://fueleconomy.gov/feg/ws/#fuelType1); see the description of the [EPA test procedures](https://fueleconomy.gov/feg/fe_test_schedules.shtml)
65. UCityA - unadjusted city MPG for [fuelType2](https://fueleconomy.gov/feg/ws/#fuelType2); see the description of the [EPA test procedures](https://fueleconomy.gov/feg/fe_test_schedules.shtml)
66. UHighway - unadjusted highway MPG for [fuelType1](https://fueleconomy.gov/feg/ws/#fuelType1); see the description of the [EPA test procedures](https://fueleconomy.gov/feg/fe_test_schedules.shtml)
67. UHighwayA - unadjusted highway MPG for [fuelType2](https://fueleconomy.gov/feg/ws/#fuelType2); see the description of the [EPA test procedures](https://fueleconomy.gov/feg/fe_test_schedules.shtml)
68. [VClass](https://fueleconomy.gov/feg/ws/#VClass)- EPA vehicle size class
69. year - model year
70. youSaveSpend - you save/spend over 5 years compared to an average car ($). Savings are positive; a greater amount spent yields a negative number. For dual fuel vehicles, this is the cost savings for gasoline
71. sCharger - if S, this vehicle is supercharged
72. tCharger - if T, this vehicle is turbocharged
73. c240Dscr - electric vehicle charger description
74. charge240b - time to charge an electric vehicle in hours at 240 V using the alternate charger
75. c240bDscr - electric vehicle alternate charger description
76. createdOn - date the vehicle record was created (ISO 8601 format)
77. modifiedOn - date the vehicle record was last modified (ISO 8601 format)
78. startStop - vehicle has start-stop technology (Y, N, or blank for older vehicles)
79. phevCity - EPA composite gasoline-electricity city MPGe for plug-in hybrid vehicles
80. phevHwy - EPA composite gasoline-electricity highway MPGe for plug-in hybrid vehicles
81. phevComb - EPA composite gasoline-electricity combined city-highway MPGe for plug-in hybrid vehicles

**R-code for Data Analysis**

#Setting working directory

setwd("C:\\Dati\\VBA for AI")

#Creating a .XDF file from the original CSV using rxImport() functionto import only the interesting variables

fuelEconomy <- "Fuel Economy EPA vehicles.csv"

fuelEconomyCSV <- read.csv(fuelEconomy)

fuelEconomyXDF <- "FuelEconomy.xdf"

xdfSource <- RxXdfData(fuelEconomyXDF)

fuelEconomyXDF <- rxImport(fuelEconomyCSV, outFile = fuelEconomyXDF, varsToDrop = c("barrels08", "barrelsA08", "charge120", "charge240", "city08", "city08u", "cityA08", "cityA08U", "cityCD", "cityE", "cityUF", "engId", "eng\_dscr", "feScore", "fuelCost08", "fuelCostA08", "fuelType", "hlv", "hpv", "id", "lv2", "lv4", "youSaveSpend", "guzzler", "trans\_dscr", "mfrCode", "charge240b", "modifiedOn"))

rxGetVarInfo(fuelEconomyXDF)

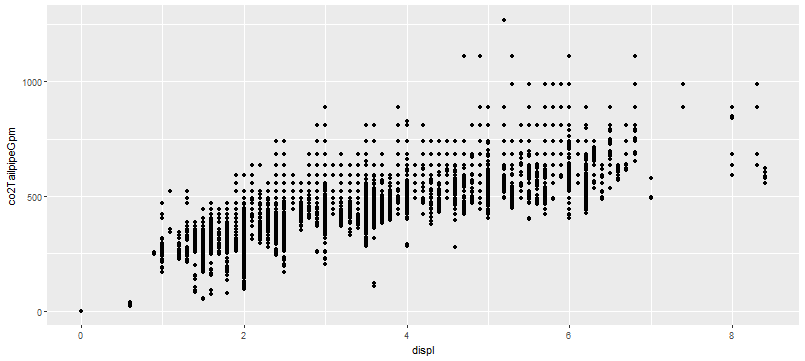
#first activity on CSV file with ggplot2 library

#Importing ggplot2 library

library(ggplot2)

#Creates a scatter plot with CO2 tail pipe for fuel Type2 on y-axis and engine displacement on x-axis

ggplot(fuelEconomyCSV) + geom\_point(mapping = aes(x = displ, y = co2TailpipeGpm))

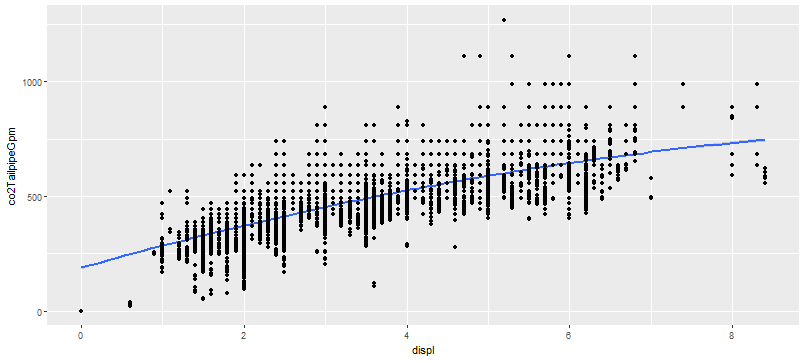


#Creates the same scatter plot with a regression 2nd grade polynomial function

ggplot(fuelEconomyCSV, mapping = aes(x = displ, y = co2TailpipeGpm)) +

geom\_smooth(method = "lm", formula = y ~ poly(x, 2)) +

geom\_point()

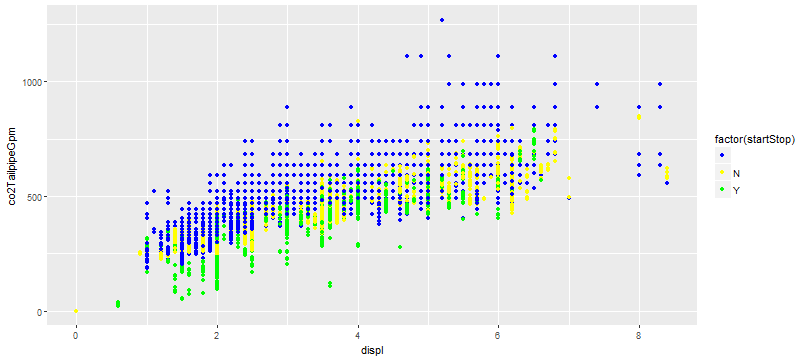


#Creates the same scatter plot with different colors for Start&Stop (Y, N, NA). NA is assumed to be "no S&S", will be changed with rxImport

ggplot(fuelEconomyCSV, mapping = aes(x = displ, y = co2TailpipeGpm, color = factor(startStop))) +

scale\_color\_manual(values = c("blue", "yellow", "green")) +

geom\_point()



#Creates the same scatter plot with facets for Star&Stop (Y, N, NA). NA is "empty" so the code is not working

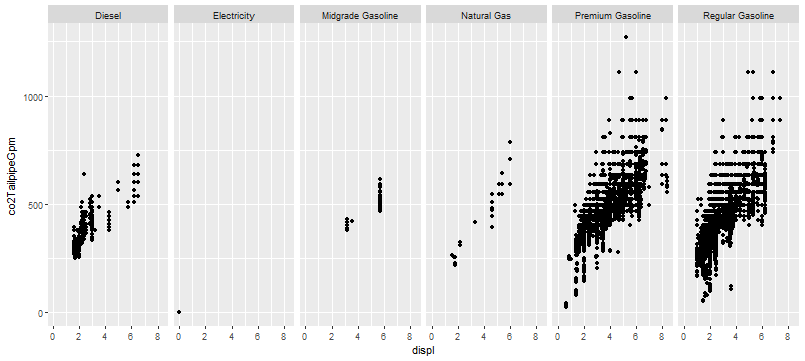
ggplot(fuelEconomyCSV) +geom\_point(mapping = aes(x = displ, y = co2TailpipeGpm)) +

facet\_grid(.~startStop)

#Creates faceted scatter plot for different fuel type 1.

ggplot(fuelEconomyCSV) + geom\_point(mapping = aes(x = displ, y = co2TailpipeGpm)) +

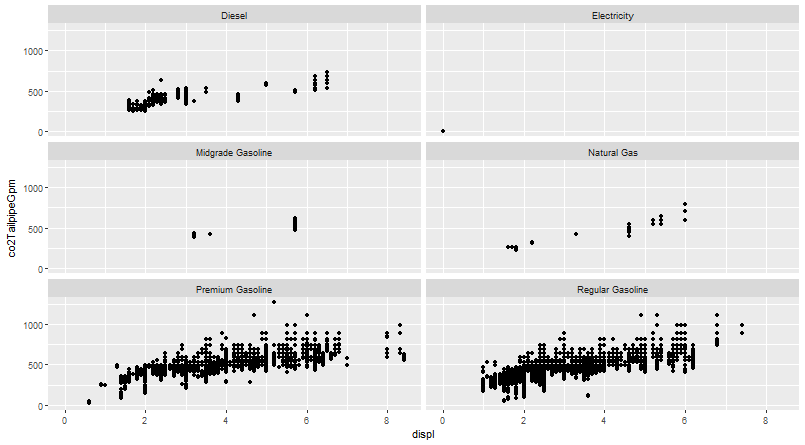
facet\_grid(. ~ fuelType1)



#Creates the same scatter plot for different fuel type 1 but in 2 columns with facet\_wrap().

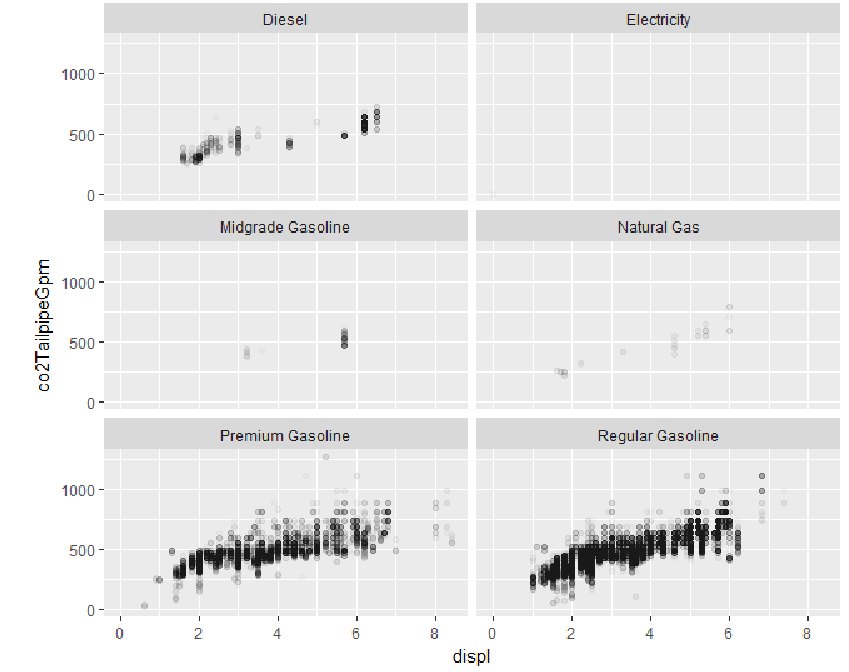
ggplot(fuelEconomyCSV) + geom\_point(mapping = aes(x = displ, y = co2TailpipeGpm)) +

facet\_wrap(~ fuelType1, ncol = 2)



#Creates the same scatter plot for different fuel type 1 but in 2 columns with facet\_wrap() and with transparency linked to number of data in the sinlge points.

ggplot(fuelEconomyCSV) + geom\_point(mapping = aes(x = displ, y = co2TailpipeGpm), alpha=1/50) + facet\_wrap(~ fuelType1, ncol = 2)

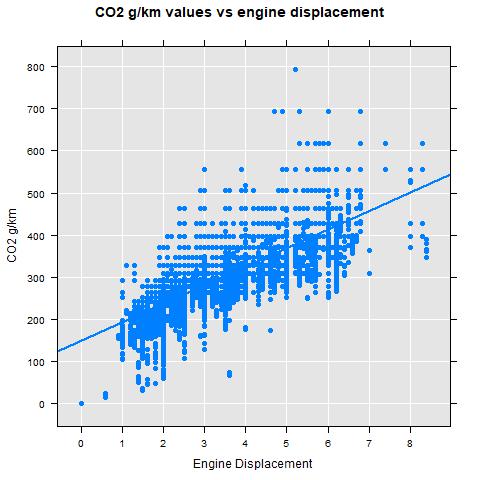


#Let's now use rx functions for the xdf analysis, saving the plots in jpeg files

jpeg("testplot.jpg")

rxLinePlot(CO2grams\_km ~ displ, fuelEconomyXDF, title = "CO2 g/km values vs engine displacement", xTitle = "Engine Displacement", yTitle = "CO2 g/km", type = c("p", "r"))

dev.off()



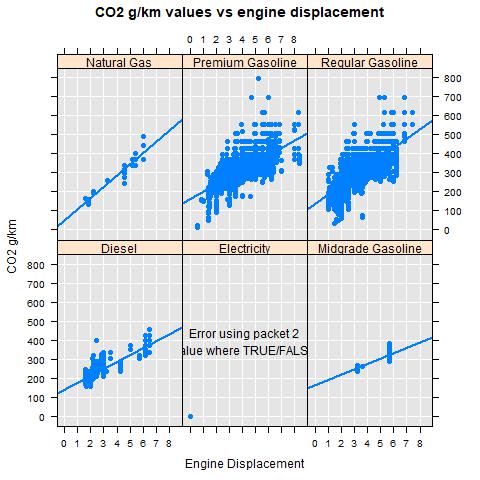
#Creates trellis plots by fuel type:

jpeg("testplot.jpg")

rxLinePlot(CO2grams\_km ~ displ | fuelType1, fuelEconomyXDF, title = "CO2 g/km values vs engine displacement",

xTitle = "Engine Displacement", yTitle = "CO2 g/km", type = c("p", "r"))

dev.off()



#the variable “trany” contains too many variants of trasnmission. Let’s re-import the data creating a variable “TransType” which contains the first 3 characters of the variable trany:

fuelEconomyXDF <- rxDataStep(fuelEconomyCSV, outFile = fuelEconomyXDF, overwrite = TRUE, varsToDrop = c("barrels08", "barrelsA08", "charge120", "charge240", "city08", "city08u", "cityA08", "cityA08U", "cityCD", "cityE", "cityUF", "engId", "eng\_dscr", "feScore", "fuelCost08", "fuelCostA08", "fuelType", "hlv", "hpv", "id", "lv2", "lv4", "youSaveSpend", "guzzler", "trans\_dscr", "mfrCode", "charge240b", "modifiedOn"), transforms = list(CO2grams\_km = co2TailpipeGpm / 1.6, TransType = substr(trany, 1,3)))

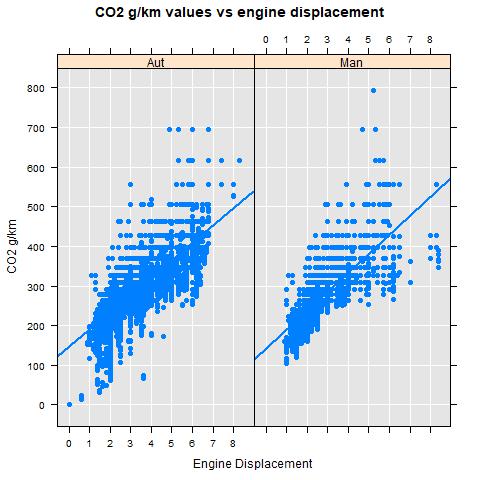
#Creates trellis plots by Transmission (Auto or Manual):

jpeg("testplot.jpg")

rxLinePlot(CO2grams\_km ~ displ | TransType, fuelEconomyXDF, title = "CO2 g/km values vs engine displacement",

xTitle = "Engine Displacement", yTitle = "CO2 g/km", type = c("p", "r"))

dev.off()



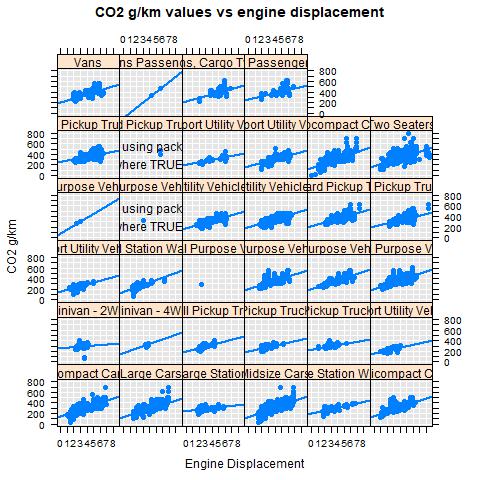
#Creates trellis plots by Vehicle class, this variable has 34 factor levels:

jpeg("testplot.jpg")

rxLinePlot(CO2grams\_km ~ displ | VClass, fuelEconomyXDF, title = "CO2 g/km values vs engine displacement",

xTitle = "Engine Displacement", yTitle = "CO2 g/km", type = c("p", "r"))

dev.off()



#Selects only "electricity" as factor of the variable fuelType1

fuelEconomyXDF2 <- "FuelEconomy2.xdf"

xdfSource2 <- RxXdfData(fuelEconomyXDF2)

fuelEconomyXDF2 <- rxDataStep(fuelEconomyCSV, outFile = fuelEconomyXDF, overwrite = TRUE,

varsToDrop = c("barrels08", "barrelsA08", "charge120", "charge240", "city08", "city08u", "cityA08", "cityA08U", "cityCD", "cityE", "cityUF", "engId", "eng\_dscr", "feScore", "fuelCost08", "fuelCostA08", "fuelType", "hlv", "hpv", "id", "lv2", "lv4", "youSaveSpend", "guzzler", "trans\_dscr", "mfrCode", "charge240b", "modifiedOn"),

transforms = list(CO2grams\_km = co2TailpipeGpm / 1.6, TransType = substr(trany, 1, 3)), rowSelection = (fuelType1 == "Electricity"))

rxGetInfo(fuelEconomyXDF2)

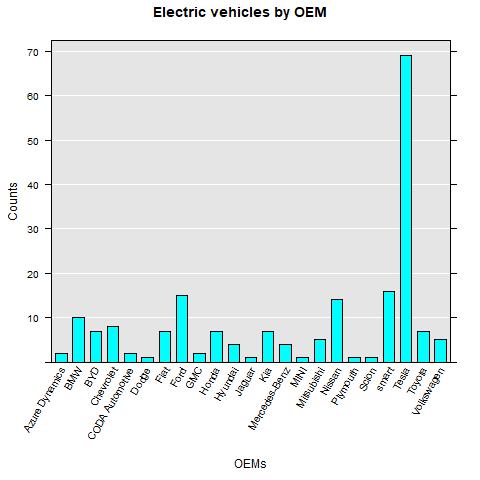
#Creates histogram with count of all tests done on electric vehicles by OEM

jpeg("testplot.jpg")

rxHistogram(~ make, fuelEconomyXDF2, title = "Electric vehicles by OEM",

xTitle = "OEMs", yTitle = "Counts")

dev.off()



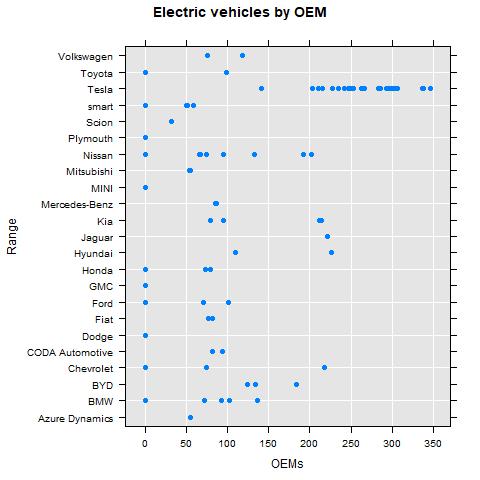
#Creates histogram with range of all tests done on electric vehicles by OEM

jpeg("testplot.jpg")

rxLinePlot(make ~ rangeHwy, fuelEconomyXDF2, title = "Electric vehicles by OEM",

xTitle = "OEMs", yTitle = "Range", type = "p", legend = TRUE )

dev.off()



#Creates scatter plot of CO2 vs number of cylinders with "smooth" regression line

jpeg("testplot.jpg")

rxLinePlot(CO2grams\_km ~ cylinders, fuelEconomyXDF, title = "CO2 g/km splitted by number of cylinders",

xTitle = "nr. cylinders", yTitle = "CO2 g/km", type = c("p","smooth"), legend = TRUE, lineColor = "red")

dev.off()

