```
#
# Make sure your working directory was set as "Source file location".
# Make sure your data files are in the same folder as this R script.
#
install.packages(c("dplyr", "ggplot2", "GGally"))
library(dplyr)
library(ggplot2)
library(GGally)
install.packages("car")
library(car) # for VIF

# Load data:
elantra <- read.csv("Elantra142-Fall2018.csv")

str(elantra)
head(elantra)

# Plot scatter matrix
ggscatmat(elantra, columns = 0:8, alpha = 0.8)


# split into training and test sets

elantra.train <- filter(elantra, Year <= 2015)
head(elantra.train)
tail(elantra.train)
elantra.test <- filter(elantra, Year <= 2018 & Year >= 2016)

# train the model
#lm(y~x1+x2+...,data)
help(lm)
mod1 <- lm(ElantraSales ~ ElantraQueries + CPI.All, data =
elantra.train)
summary(mod1)

# A better model...
mod2 <- lm(ElantraSales ~ MonthFactor + ElantraQueries + CPI.All +
CPI.Energy + Unemployment, data = elantra.train)
summary(mod2)
vif(mod2)

mod3 <- lm(ElantraSales ~ MonthFactor + ElantraQueries + CPI.All, data =
elantra.train)
summary(mod3)
vif(mod3)

# compute OSR^2

# this builds a vector of predicted values on the test set
SSE = sum((elantra.test$ElantraSales - elantraPredictions)^2)
```

```
SST = sum((elantra.test$ElantraSales -
mean(elantra.train$ElantraSales))^2)
OSR2 = 1 - SSE/SST

# Load NEW HONDA data:
honda <- read.csv("HondaOdysseySales.csv")

str(honda)
head(honda)

# Plot scatter matrix
ggscatmat(honda, columns = 0:10, alpha = 0.8)


# split into training and test sets

honda.train <- filter(honda, Year <= 2015)
head(honda.train)
tail(honda.train)
honda.test <- filter(honda, Year <= 2018 & Year >= 2016)

# train the model
mod4 <- lm(HondaSales ~ MonthFactor + HondaQueries + CPI.All +
Unemployment, data = honda.train)
summary(mod4)

# compute OSR^2

# this builds a vector of predicted values on the test set
SSE = sum((honda.test$HondaSales - hondaPredictions)^2)
SST = sum((honda.test$HondaSales - mean(honda.train$HondaSales))^2)
OSR2 = 1 - SSE/SST
OSR2

AugustEst <- read.csv("AugustEst.csv")

AugustEst.train <- filter(AugustEst, Year <= 2015)
head(AugustEst.train)
tail(AugustEst.train)
AugustEst.test <- filter(AugustEst, Year <= 2018 & Year >= 2016)

# train the model
elantramod <- lm(ElantraSales ~ MonthFactor + ElantraQueries + CPI.All,
data = AugustEst.train)
summary(elantramod)
hondamod <- lm(HondaSales ~ MonthFactor + HondaQueries + CPI.All +
Unemployment, data = AugustEst.train)
summary(hondamod)


elantraNewPredictions <- predict(elantramod, newdata=AugustEst.test)
```

```
hondaNewPredictions <- predict(hondamod, newdata=AugustEst.test)
elantraNewPredictions
hondaNewPredictions
```