0908766HW3.R

Owner

Fri Nov 11 22:48:54 2016

```
library(leaps)
library(e1071)
library(broom)
library(ggplot2)
library(car)

corolla = read.csv(file = "Data/ToyotaCorolla.csv")

# Become familiar with the data

head(corolla)
```

```
##   Price Age    KM FuelType HP MetColor Automatic   CC Doors Weight
## 1 13500  23 46986   Diesel 90        1         0 2000     3   1165
## 2 13750  23 72937   Diesel 90        1         0 2000     3   1165
## 3 13950  24 41711   Diesel 90        1         0 2000     3   1165
## 4 14950  26 48000   Diesel 90        0         0 2000     3   1165
## 5 13750  30 38500   Diesel 90        0         0 2000     3   1170
## 6 12950  32 61000   Diesel 90        0         0 2000     3   1170
```

```
# Recode categorical variable into numeric so I can run calculations
corolla$FuelCode[corolla$FuelType=="CNG"] <- 1
corolla$FuelCode[corolla$FuelType=="Diesel"] <- 2
corolla$FuelCode[corolla$FuelType=="Petrol"] <- 3
corolla$FuelCode[corolla$FuelType=="NA"] <- 0
corolla <- corolla[-4]
head(corolla)
```

```
##   Price Age    KM HP MetColor Automatic   CC Doors Weight FuelCode
## 1 13500  23 46986 90        1         0 2000     3   1165        2
## 2 13750  23 72937 90        1         0 2000     3   1165        2
## 3 13950  24 41711 90        1         0 2000     3   1165        2
## 4 14950  26 48000 90        0         0 2000     3   1165        2
## 5 13750  30 38500 90        0         0 2000     3   1170        2
## 6 12950  32 61000 90        0         0 2000     3   1170        2
```

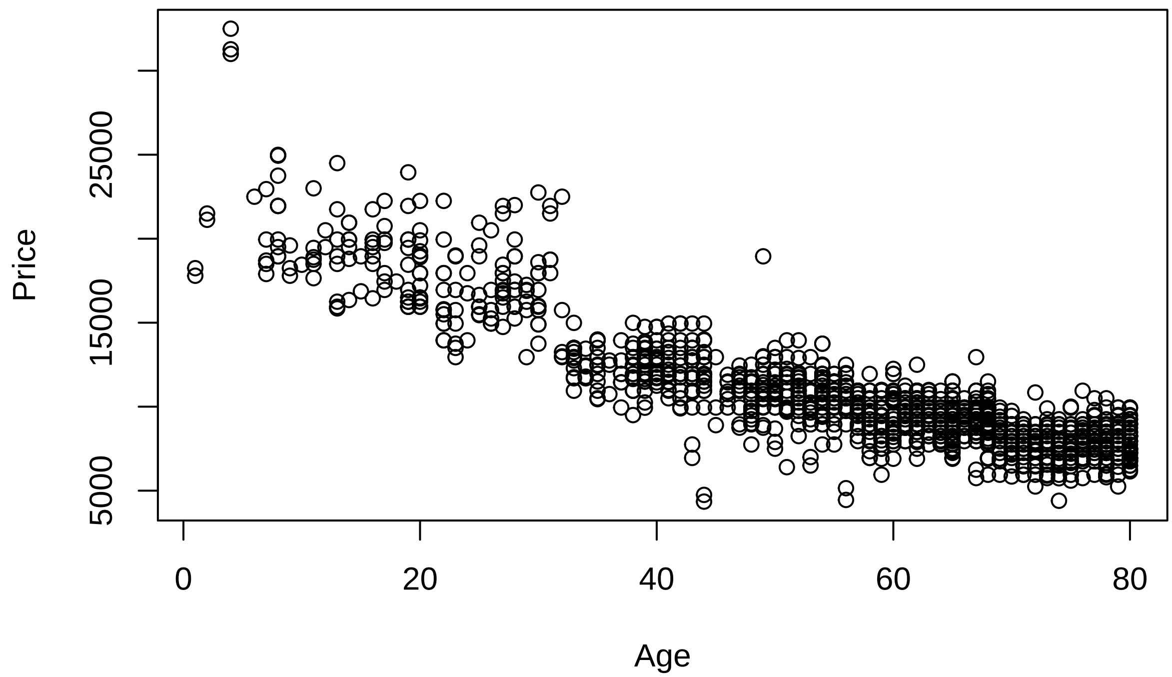
```
# Drop the textcolumn


# Verify it worked
head(corolla)
```

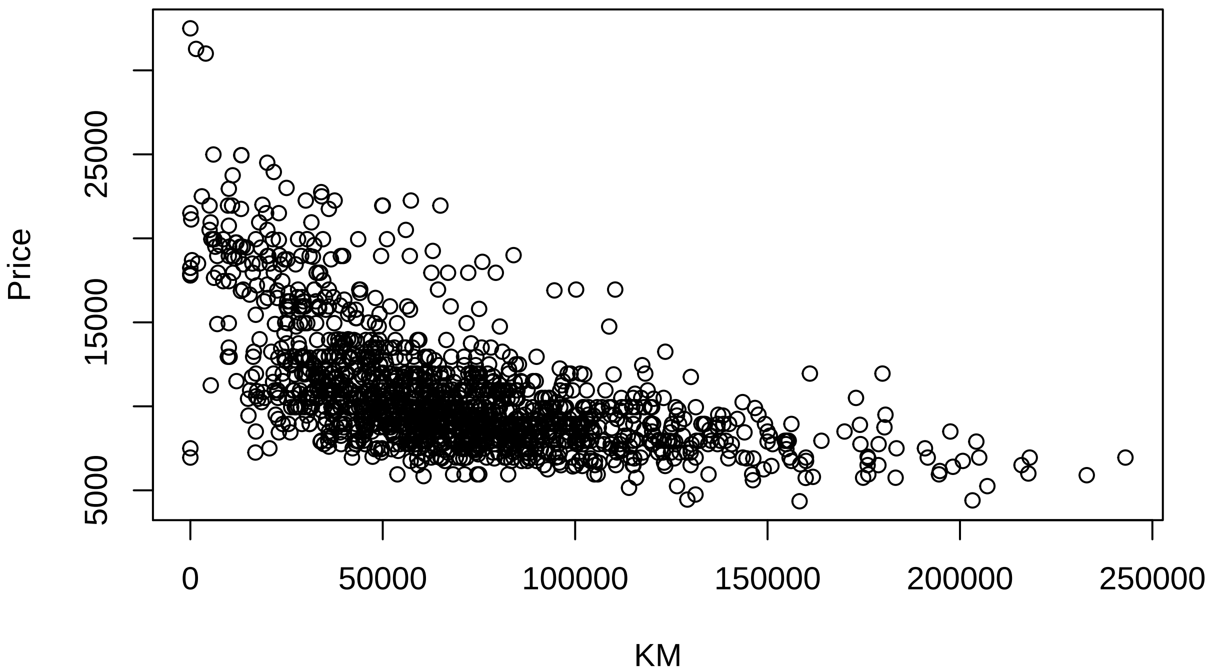
```
##   Price Age    KM HP MetColor Automatic   CC Doors Weight FuelCode
## 1 13500  23 46986 90        1         0 2000     3   1165        2
## 2 13750  23 72937 90        1         0 2000     3   1165        2
## 3 13950  24 41711 90        1         0 2000     3   1165        2
```

```
## 4 14950   26 48000 90         0           0 2000      3    1165          2
## 5 13750   30 38500 90         0           0 2000      3    1170          2
## 6 12950   32 61000 90         0           0 2000      3    1170          2
```

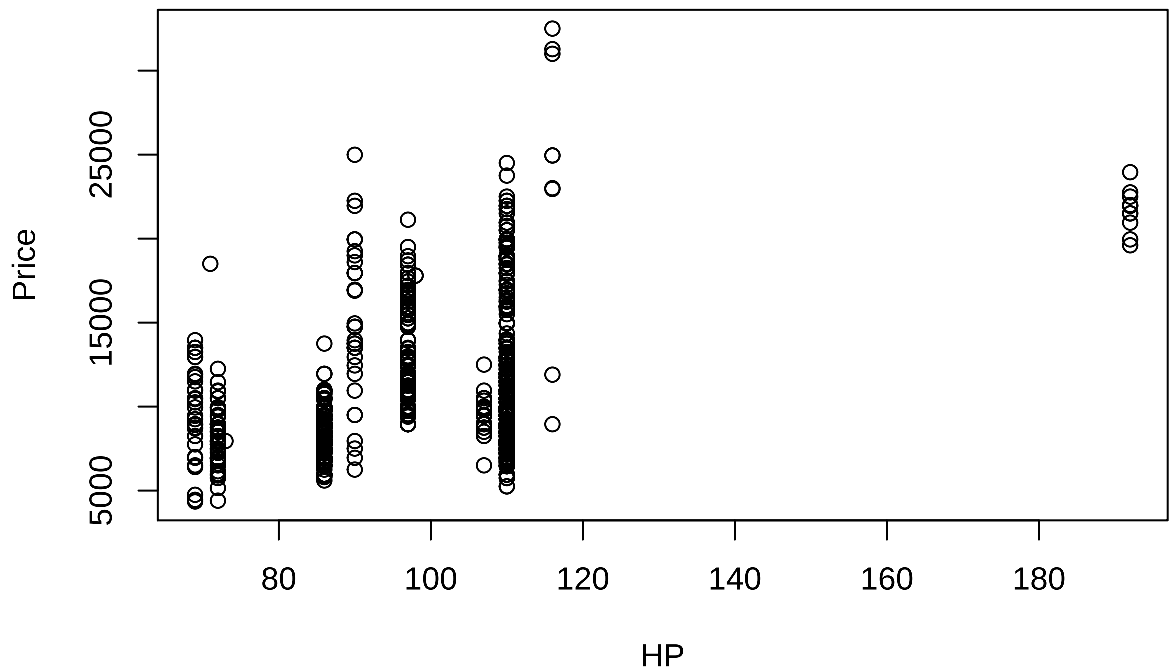
```r
# Descriptive stats (in lieu of 9 separate descriptive stats runs, simply graph them)
plot(Price ~ Age, data = corolla)
```



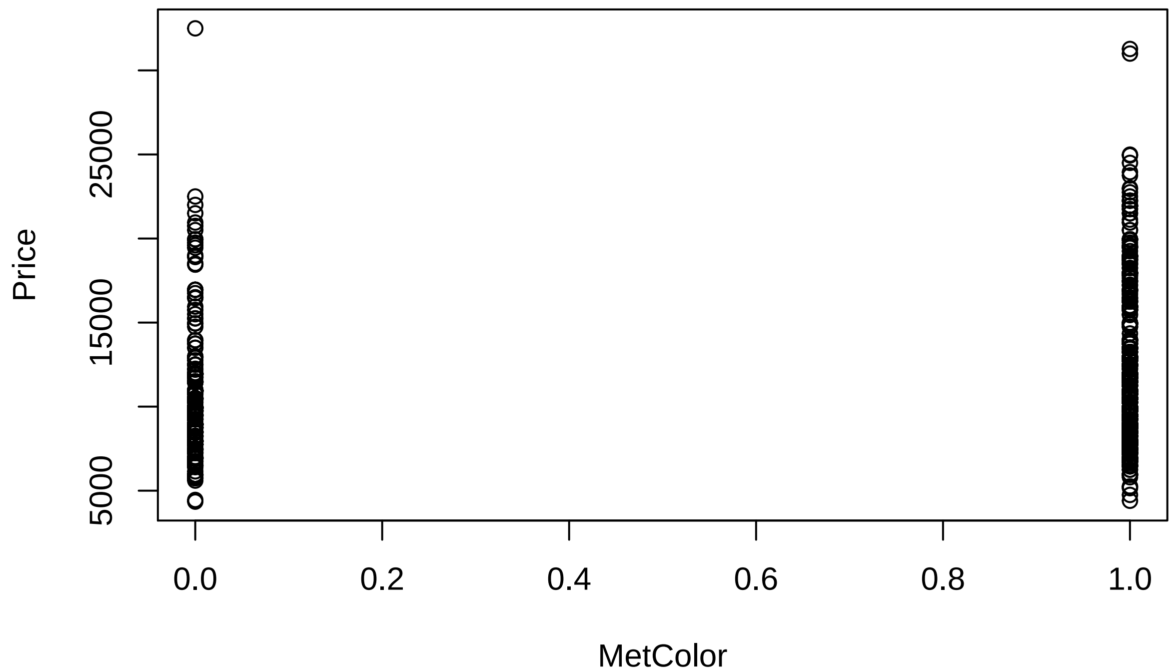

```r
plot(Price ~ KM, data = corolla)
```



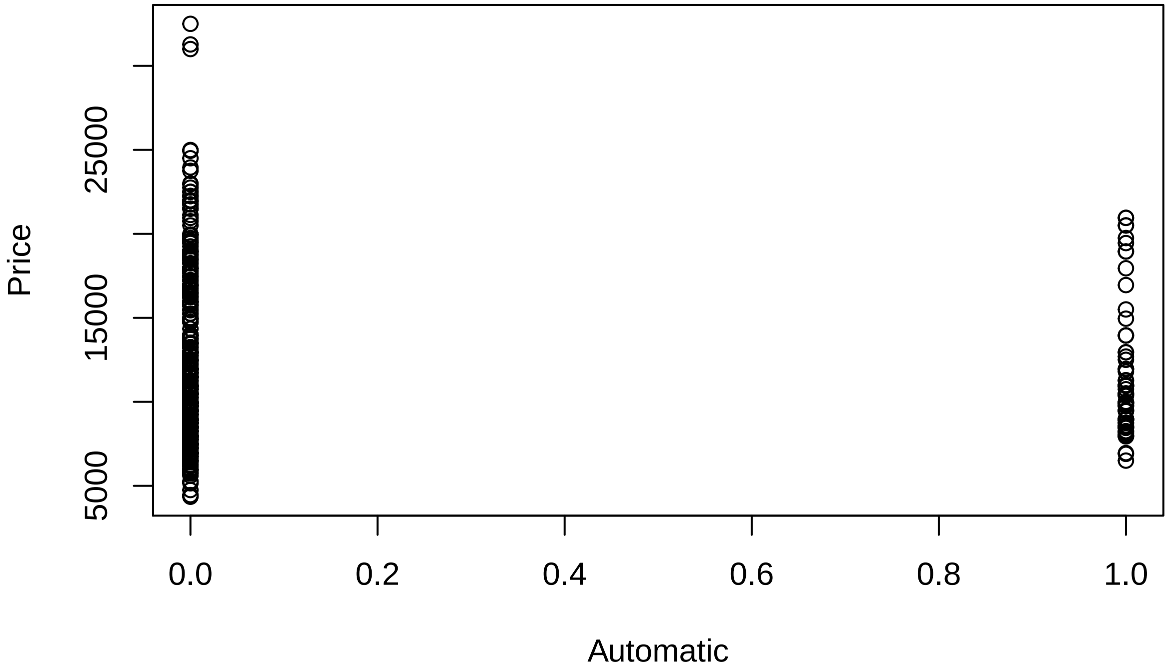

```
plot(Price ~ HP, data = corolla)
```



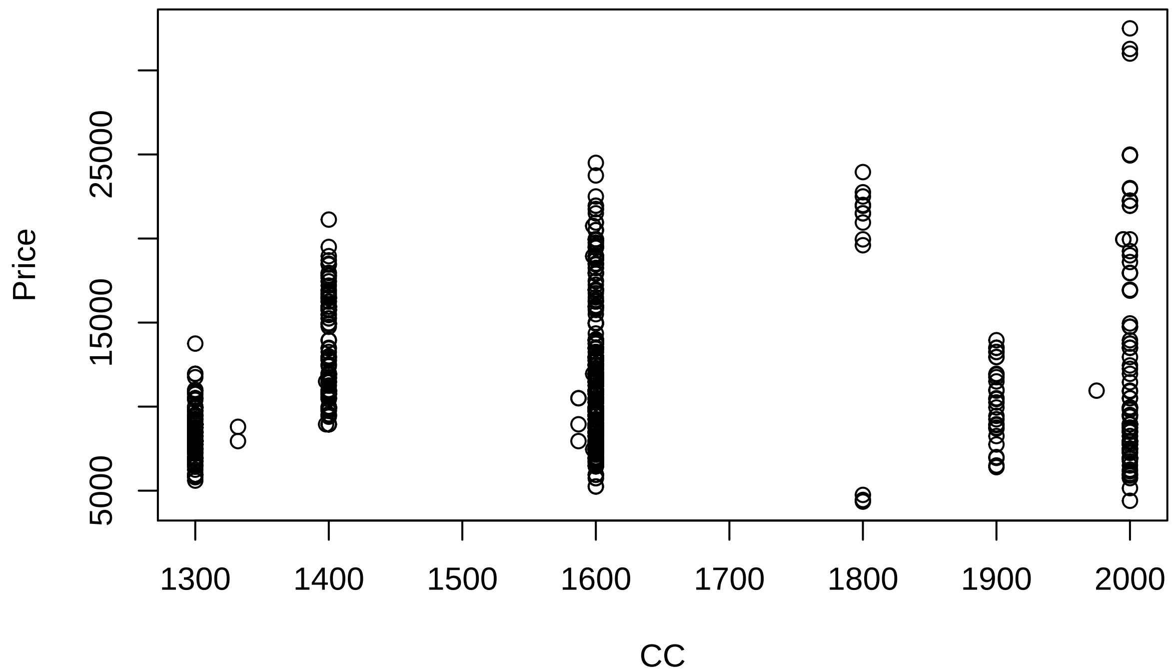

```
plot(Price ~ MetColor, data = corolla)
```



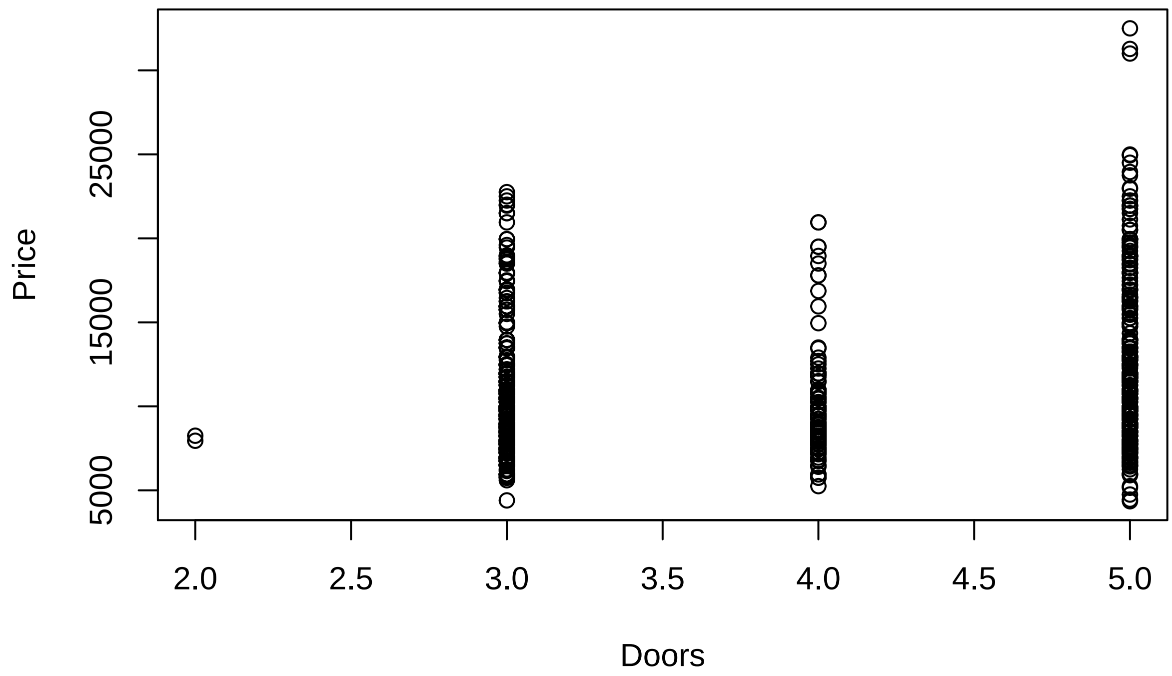

```r
plot(Price ~ Automatic, data = corolla)
```



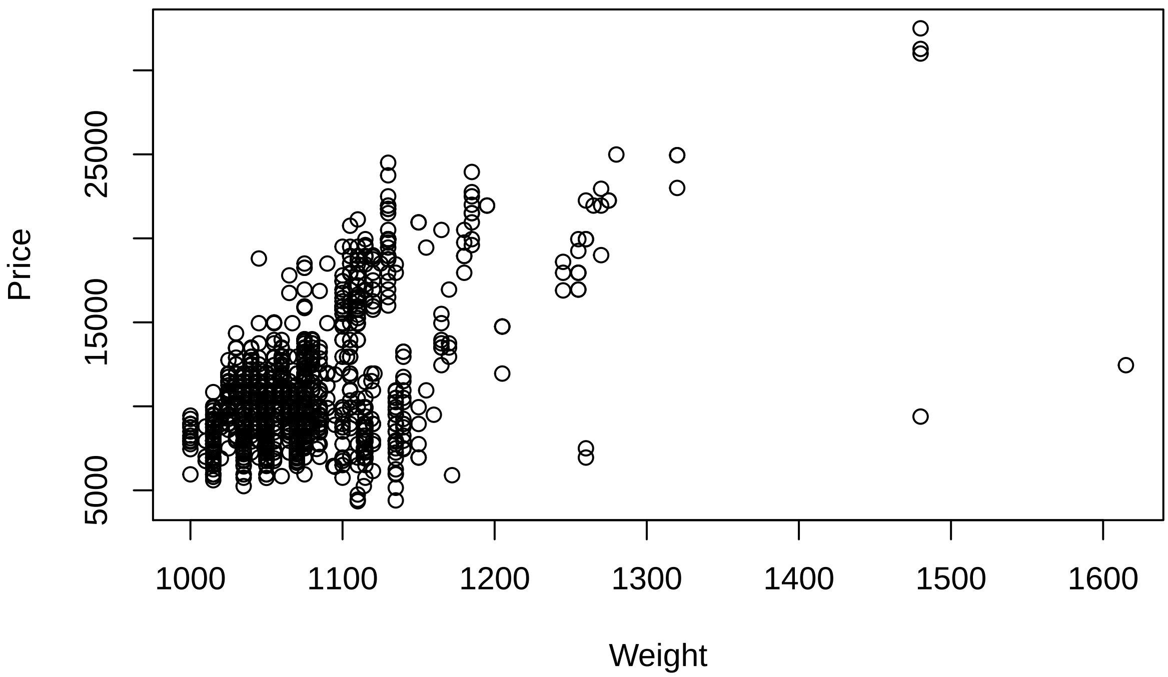

```
plot(Price ~ CC, data = corolla)
```



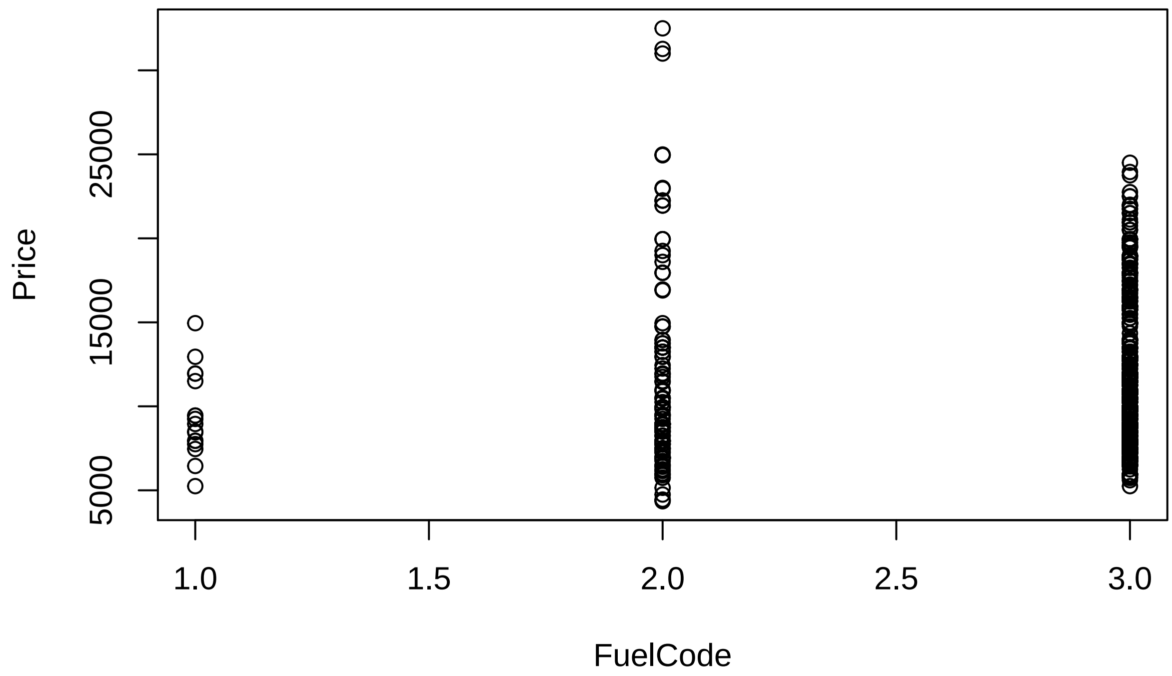

```r
plot(Price ~ Doors, data = corolla)
```




```r
plot(Price ~ Weight, data = corolla)
```




```
plot(Price ~ FuelCode, data = corolla)
```




```
# Build a linear model using all variables
corolla.m1 <- lm(Price ~ ., data = corolla)
corolla.m1.summary <- summary(corolla.m1)

# Check confidence interval
corolla.m1.confint <- confint(corolla.m1)

# Get it ready to plot
x <- corolla[,2:10] # Independent variables
y <- corolla[,1] # Dependent variables

# Check correlation

corolla.cor <- cor(corolla) #Age accounts for >87% of variation in price, so we will build a separate l

# Build a regtab with the model to check which variables have a significant influence
corolla.out <- summary(regsubsets(x, y, nbest = 1, nvmax = ncol(x),force.in = NULL, force.out = NULL, me
corolla.regtab <- cbind(corolla.out$which, corolla.out$rsq, corolla.out$adjr2, corolla.out$cp)
colnames(corolla.regtab) <- c("(Intercept)","Age","KM","HP", "MetColor", "Automatic", "CC", "Doors", "We
                              "R-Sq", "R-Sq (adj)", "Cp")

# Create a second model with just age because of the high correlation
corolla.m2 <- lm(Price ~ Age, data = corolla)
corolla.m2.summary <- summary(corolla.m2)

# Show results
```

```r
print(corolla.m2.summary)
```

```
##
## Call:
## lm(formula = Price ~ Age, data = corolla)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8423.0  -997.4   -24.6   878.5 12889.7
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 20294.059    146.097  138.91   <2e-16 ***
## Age          -170.934      2.478  -68.98   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1746 on 1434 degrees of freedom
## Multiple R-squared:  0.7684, Adjusted R-squared:  0.7682
## F-statistic:  4758 on 1 and 1434 DF,  p-value: < 2.2e-16
```

```r
# Check second model's confidence interval
corolla.m2.confint <- confint(corolla.m2)
print(corolla.m2.confint)
```

```
##                  2.5 %     97.5 %
## (Intercept) 20007.4714 20580.6459
## Age          -175.7946  -166.0725
```

```r
## Build a third model dropping all the variables that did not have high enough P-values in model 1 to v
# Restructure data to drop insignificant variables
corollaM3 <-corolla[-5]
corollaM3 <-corollaM3[-7]
corollaM3 <-corollaM3[-5]
corollaM3 <-corollaM3[-7]

# Third model with restructured data
corolla.m3 <- lm(Price ~ ., data = corollaM3)
corolla.m3.summary <- summary(corolla.m3) # Show results
print(corolla.m3.summary)
```

```
##
## Call:
## lm(formula = Price ~ ., data = corollaM3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11992.2   -767.1    -16.8    769.2   6199.9
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.000e+03  9.852e+02  -6.090 1.45e-09 ***
```

```
## Age         -1.221e+02  2.594e+00 -47.086  < 2e-16 ***
## KM          -1.682e-02  1.287e-03 -13.061  < 2e-16 ***
## HP           3.247e+01  2.540e+00  12.784  < 2e-16 ***
## CC          -1.626e+00  2.771e-01  -5.869 5.46e-09 ***
## Weight       2.235e+01  1.026e+00  21.781  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1335 on 1430 degrees of freedom
## Multiple R-squared:  0.865,  Adjusted R-squared:  0.8646
## F-statistic:  1833 on 5 and 1430 DF,  p-value: < 2.2e-16
```

```
# Check third model's confidence interval
corolla.m3.confint <- confint(corolla.m3)
print(corolla.m3.confint)
```

```
##                     2.5 %        97.5 %
## (Intercept) -7.932079e+03 -4.066992e+03
## Age         -1.272147e+02 -1.170390e+02
## KM          -1.934143e-02 -1.429045e-02
## HP           2.748461e+01  3.744775e+01
## CC          -2.170030e+00 -1.082747e+00
## Weight       2.033614e+01  2.436175e+01
```

```
n <- length(corolla$Price) # Get the number of elements
diff <- dim(n) # Set the dimension of the container object
percdiff <- dim(n) # Set the dimension of the container object

for (k in 1:n) {
  train1 <- c(1:n)

  # the R expression "train1[train1 != k]" picks from train1 those
  # elements that are different from k and stores those elements in the
  # object train.
  # For k = 1, train consists of elements that are different from 1; that
  # is 2, 3, ..., n.
  train <- train1[train1 != k]

  # Create the linar model for the all but one element
  m1 <- lm(Price ~ ., data = corolla[train,])

  # Predict the missing value based on the model
  pred <- predict(m1, newdat = corolla[-train,])

  # What is the real value
  obs <- corolla$Price[-train]

  # Calculate the delta between observed and predicted
  diff[k] <- obs - pred

  # Calculate the relative difference between observed and predicted
  percdiff[k] <- abs(diff[k]) / obs
}
```

```r
corolla.m1.me <- mean(diff) # mean error
corolla.m1.rmse <- sqrt(mean(diff**2)) # root mean square error
corolla.m1.mape <- 100*(mean(percdiff)) # mean absolute percent error

# Repeat process, but for second model which only needs to check age
n <- length(corolla$Price)
diff <- dim(n)
percdiff <- dim(n)
for (k in 1:n) {
  train1 <- c(1:n)
  train <- train1[train1 !=k ]
  m2 <- lm(Price ~ Age, data = corolla[train,])
  pred <- predict(m2, newdat = corolla[-train,])
  obs <- corolla$Price[-train]
  diff[k] <- obs - pred
  percdiff[k] <- abs(diff[k]) / obs
}
corolla.m2.me <- mean(diff)
corolla.m2.rmse <- sqrt(mean(diff**2))
corolla.m2.mape <- 100*(mean(percdiff))

# Third repetition for third model
for (k in 1:n) {
  train1 <- c(1:n)
  train <- train1[train1 != k]
  m3 <- lm(Price ~ ., data = corollaM3[train,])
  pred <- predict(m3, newdat = corollaM3[-train,])
  obs <- corollaM3$Price[-train]
  diff[k] <- obs - pred
  percdiff[k] <- abs(diff[k]) / obs
}
corolla.m3.me <- mean(diff)
corolla.m3.rmse <- sqrt(mean(diff**2))
corolla.m3.mape <- 100*(mean(percdiff))

corolla.m1.me   # mean error
```

```
## [1] -2.494298
```

```r
corolla.m1.rmse # root mean square error
```

```
## [1] 1372.747
```

```r
corolla.m1.mape # mean absolute percent error
```

```
## [1] 9.662033
```

```r
corolla.m2.me   # mean error
```

```
## [1] 0.6085014
```

```r
corolla.m2.rmse # root mean square error
```

```
## [1] 1748.76
```

```r
corolla.m2.mape # mean absolute percent error
```

```
## [1] 12.13156
```

```r
corolla.m3.me   # mean error
```

```
## [1] -1.427223
```

```r
corolla.m3.rmse # root mean square error
```

```
## [1] 1364.132
```

```r
corolla.m3.mape # mean absolute percent error
```

```
## [1] 9.700578
```

```r
# Model 3 has the lowest root mean square error, therefore the best model

## To use model 3 to predict the car with Dr. Kalisch's specifications, we will need to fill in the blan
#
Median.Mileage <- median(corolla[["KM"]])
cc.vs.weight <- corollaM3[-1]
cc.vs.weight <- cc.vs.weight[-1]
cc.vs.weight <- cc.vs.weight[-1]
cc.vs.weight <- cc.vs.weight[-1]
# We found out earlier weight was moderately correlated with displacement, so we will find a rough esti
cc.weight.cor <- cor(cc.vs.weight)
predicted.weight <- cc.weight.cor [2,1] * 2000
Age <- 12
KM <- Median.Mileage
HP <- 185
CC <- 2000
Weight <- predicted.weight
kalischcar <- data.frame(Age, KM, HP, CC, Weight)
kalischcar.prediction <-predict(m3, kalischcar)

# Check if the assumptions are met...
## Create data frame with residuals
corolla.f <- fortify(corolla.m3)

## Linearity
### Residual vs Fitted Plot
p1 <- ggplot(corolla.f, aes(x = .fitted, y = .resid)) +
  geom_point() +
  stat_smooth(method = "loess") +
  geom_hline(yintercept = 0, col = "red", linetype = "dashed") +
```

```
  xlab("Fitted values") +
  ylab("Residuals") +
  ggtitle("Residual vs Fitted Plot")

## Normality
### Normal Q-Q Plot
p2 <- ggplot(corolla.f, aes(x = qqnorm(.stdresid)[[1]], y = .stdresid)) +
  geom_point(na.rm = TRUE) +
  geom_abline() +
  xlab("Theoretical Quantiles") +
  ylab("Standardized Residuals") +
  ggtitle("Normal Q-Q")

corolla.skew <- skewness(corolla.f$.resid)
corolla.kurt <- kurtosis(corolla.f$.resid)

## Equal variance
### Scale-Location Plot
p3 <- ggplot(corolla.f, aes(x = .fitted, y = sqrt(abs(.stdresid)))) +
  geom_point(na.rm=TRUE) +
  stat_smooth(method = "loess", na.rm = TRUE) +
  xlab("Fitted Value") +
  ylab(expression(sqrt("|Standardized residuals|"))) +
  ggtitle("Scale-Location")

## Independence
# Perform a Durbin-Watson F-test for autocorrelation
corolla.dw <- durbinWatsonTest(m1)

## Outlier influance
### Cook's Distance Histogram
p4 <- ggplot(corolla.f, aes(x = seq_along(.cooksd), y = .cooksd)) +
  geom_bar(stat="identity", position="identity") +
  xlab("Obs. Number") +
  ylab("Cook's distance") +
  ggtitle("Cook's distance")

p5 <- ggplot(corolla.f, aes(x =.hat, y = .stdresid)) +
  geom_point(aes(size=.cooksd), na.rm=TRUE) +
  stat_smooth(method="loess", na.rm=TRUE) +
  xlab("Leverage") +
  ylab("Standardized Residuals") +
  ggtitle("Residual vs Leverage Plot") +
  scale_size_continuous("Cook's Distance", range = c(1,5)) +
  theme(legend.position="bottom")

ggsave("graphs/linearityAssumption.pdf", p1)
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave("graphs/normalityAssumption.pdf", p2)
```

```
## Saving 6.5 x 4.5 in image
```

```r
ggsave("graphs/equalVarianceAssumptions.pdf", p3)
```

```
## Saving 6.5 x 4.5 in image
```

```r
ggsave("graphs/outlierInfluance1Assumptions.pdf", p4)
```

```
## Saving 6.5 x 4.5 in image
```

```r
ggsave("graphs/outlierInfluance2Assumptions.pdf", p4)
```

```
## Saving 6.5 x 4.5 in image
```