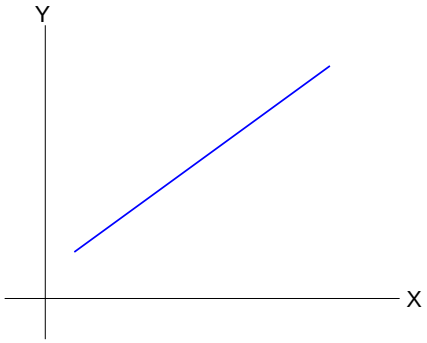
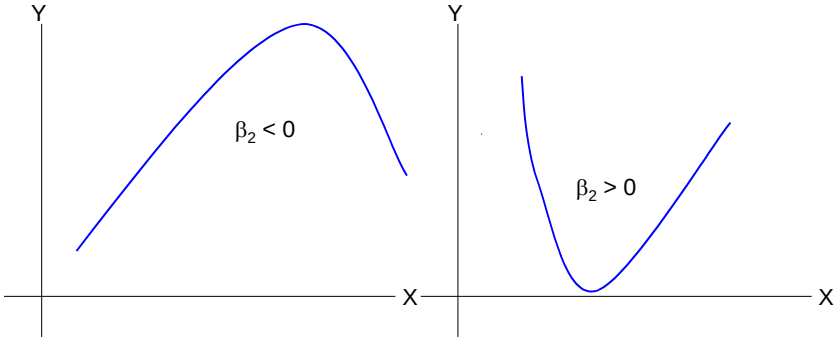
Regression models for quantitative and qualitative predictors

8.1 Polynomial regression models

First-order model: $E(Y_i) = \beta_0 + \beta_1 X_{i1} = \beta_0 + \beta_1 X_{i1}^1$



Second-order model: $E(Y_i) = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i1}^2$



X_{i1}^2 is called the second-order or "quadratic" term of the model. It allows for curvature in the relationship between X and Y.

The sign of β_2 determines if the curve opens upwards or downwards.

Since X_{i1}^2 is a transformation of X_{i1}, these two model terms can be highly correlated leading to multicollinearity and problems with inverting $\mathbf{X}'\mathbf{X}$. To partially avoid this, the predictor variable can be transformed to be deviations from its mean, $Z_{i1}=X_{i1}-\bar{X}_1$. The second order model becomes, $E(Y_i) = \beta_0 + \beta_1 Z_{i1} + \beta_2 Z_{i1}^2$.

Notes:

- KNN use a lowercase "script X" for Z here.
- Usually, I will only use this transformation when I see signs of multicollinearity and its effects of inferences. For example, if I see a VERY large standard error for a model coefficient when the transformation is not made, I will examine if this still occurs when the transformation is made.

Example: NBA guard data (nba_ch8.R)

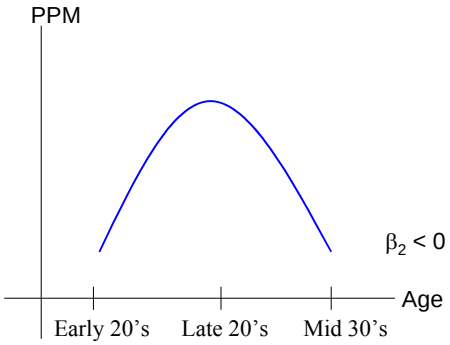
Examine the age variable.

Some basketball fans may believe:

1) Younger guards are learning the game and do not perform as well as more experienced guards
2) Older guards performance decreases past a certain age

3) Guards reach a peak performance level in their late 20's.

In the above statements were true, one might expect to see a plot something like:



Consider the model $E(Y_i) = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i1}^2$ where Y_i=PPM and X_{i1}=Age. Also, consider the model $E(Y_i) = \beta_0 + \beta_1 Z_{i1} + \beta_2 Z_{i1}^2$ where Y_i=PPM and $Z_{i1}=X_{i1}-\bar{X}_1$.

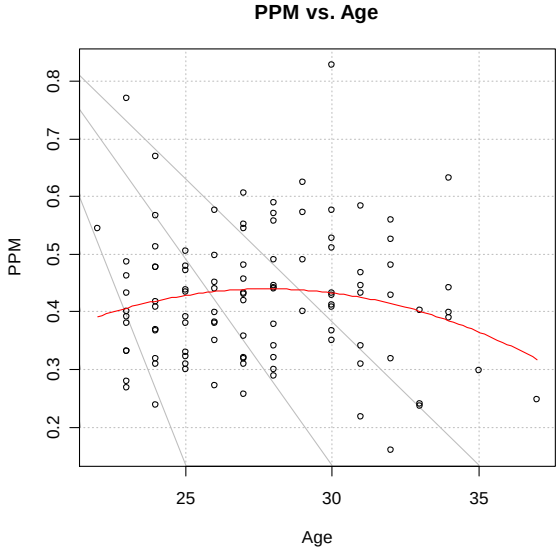
> NOTE: The β's are NOT the same for the two different models! I used the "β" notation to be consistent with the notation used before. If you are uncomfortable with this notation, β' could be substituted for β in the second model.

Fit the models and determine if Age^2 should be in the model.

R code and output:

```
> nba<-read.table(file =
         "C:\\chris\\UNL\\STAT870\\Chapter6\\nba_data.txt",
          header=TRUE, sep = "")

> mod.fit1<-lm(formula = PPM ~ age + I(age^2), data = nba)
> summary(mod.fit1)

Call:
lm(formula = PPM ~ age + I(age^2), data = nba)

Residuals:
      Min        1Q    Median        3Q       Max
-0.255059 -0.083069 -0.001772  0.058228  0.396231

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.6760076  0.7220455  -0.936    0.351
age          0.0802913  0.0514092   1.562    0.121
I(age^2)    -0.0014443  0.0009053  -1.595    0.114

Residual standard error: 0.1155 on 102 degrees of freedom
Multiple R-Squared: 0.02634,  Adjusted R-squared: 0.007247
F-statistic:  1.38 on 2 and 102 DF,  p-value: 0.2563


> mod.fit2<-lm(formula = PPM ~ I(age-mean(age)) + I((age-
                mean(age))^2), data = nba)
> summary(mod.fit2)

Call:
lm(formula = PPM ~ I(age - mean(age)) + I((age -
mean(age))^2),
    data = nba)

Residuals:
      Min        1Q    Median        3Q       Max
-0.255059 -0.083069 -0.001772  0.058228  0.396231

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)           0.4397844  0.0151822  28.967   <2e-16 ***
I(age - mean(age))    0.0007589  0.0036605   0.207    0.836
I((age - mean(age))^2) -0.0014443  0.0009053  -1.595    0.114
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1155 on 102 degrees of freedom
Multiple R-Squared: 0.02634,  Adjusted R-squared: 0.007247
F-statistic:  1.38 on 2 and 102 DF,  p-value: 0.2563

> mean(nba$age)
[1] 27.53333
> mod.fit2$coefficients
          (Intercept)     I(age - mean(age)) I((age - mean(age))^2)
         0.4397843689          0.0007589219         -0.0014442934

> cor(x = nba$age, y = nba$age^2)
[1] 0.9978604
> cor(x = nba$age - mean(nba$age),
      y = (nba$age- mean(nba$age))^2)
[1] 0.3960341

> plot(x = nba$age, y = nba$PPM, xlab = "Age", ylab =
       "PPM", main = "PPM vs. Age", panel.first = grid(col
       = "gray", lty = "dotted"))
> curve(expr = predict(object = mod.fit1, newdata =
        data.frame(age = x)), col = "red", lty = "solid",
        lwd = 1, add = TRUE, from = min(nba$age), to =
        max(nba$age))
```



Notes:

1) Notice how the *I()* (identity) function is used in the formula statements of *lm()*. The *I()* function helps to protect the meaning of what is inside of it. Note that just saying age^2 without the function will not work properly. We will see later on that syntax like (var1 + var2)^2 means to include var1, var2, and var1∗var2 in the model (all "main effects" and "interactions"). Thus, age^2 just means age to R because there are no other terms with it.
2) The *scale()* function can also be used to find the mean-adjusted values. See example in program.
3) There is strong positive correlation between age and age2. There is not as strong of correlation between the mean adjusted age terms.
4) The sample regression model using the mean adjusted age variable is:

$$\hat{Y} = 0.4397843689 + 0.0007589219Z - 0.0014442934Z^2$$
$$= 0.4397843689 + 0.0007589219(X - 27.5\bar{3}) - 0.0014442934(X - 27.5\bar{3})^2$$
$$= -0.6760076 + 0.08029134X - 0.0014442934X^2$$

Therefore, the model using the transformed X (Z) is the same as just using X.

5) The overall F test has a p-value of 0.2563 for both models indicating there is not a significant relationship between PPM and age, age^2. If the overall F test rejected H_0, then a test for age^2 would be appropriate to determine if there is a quadratic relationship between age and PPM.

6) For illustrative purposes, the p-value for testing $H_0:\beta_2 = 0$ vs. $H_a:\beta_2 \neq 0$ is 0.114 for both models. This would indicate there is marginal evidence that age^2 is needed.
7) Given we received the same p-values for the overall F-test (shown in 5)) and both models are the same (shown in 4)), one does not need to worry about potential problems with using non-transformed predictor variables.
8) The scatter plot with the sample regression model does not show a strong relationship between PPM and age.
9) The interpretation of the b_j values can not be done the same way as before (i.e., for every one unit increase in X_j, $\hat{Y}$ increases by b_j). The age term itself is not interpreted directly due to the quadratic age term being present. Typically, one will just look at the sign on the squared term to determine if the sample model is concave up or down.
10) More complicated models could be considered, like $E(PPM) = \beta_o + \beta_1 MPG + \beta_2 Height + \beta_3 FTP + \beta_4 AGE + \beta_5 AGE^2$. To determine if there is a relationship between Age and PPM, partial F tests can be used to test: $H_0:\beta_4=\beta_5=0$ vs. H_a: At least one $\beta \neq 0$.

Estimation of E(Y) and prediction of Y with these types of models can be done in a similar manner as in the past. However, you need to be careful about how the *predict()* function is used. For example,

```
>   #Predict at age = 20 with mod.fit1
>   predict(object = mod.fit1, newdata = data.frame(age =
     20))
[1] 0.3521019
>   c(1,20,20^2)%*%mod.fit1$coefficients
          [,1]
[1,] 0.3521019
```

Notice that age^2 did not need to be entered into the *data.frame()*. Next is an example of working with the transformation where R does not do the calculations as we would like.

```
>   predict(object = mod.fit2, newdata = data.frame(age =
      20))  #Incorrect answer
[1] 0.4397844

>   c(1,20-mean(nba$age),(20-mean(nba$age))^2) %*%
      mod.fit2$coefficients  #Correct
          [,1]
[1,] 0.3521019

>   c(1,20,20^2)%*%mod.fit2$coefficients  #Incorrect answer
           [,1]
[1,] -0.1227545

>   c(1,0,0)%*%mod.fit2$coefficients  #Incorrect answer -
                          this is what predict()does above
          [,1]
[1,] 0.4397844
```

The *predict()* finds `I(age-mean(age))` due to the formula given in the *lm()* function and substitutes age = 20 everywhere age is given . This leads to values of 20 – mean(20) = 0.

To correct this problem, one needs to specify the model a little different in the *lm()* function:

```
>   #Mean adjusting age - lm() part works AND predict()
      part does as well
>   mod.fit3<-lm(formula = PPM ~ I(age-mean(nba$age)) +
      I((age-mean(nba$age))^2), data = nba)
>   summary(mod.fit3)

Call:
lm(formula = PPM ~ I(age - mean(nba$age)) + I((age -
mean(nba$age))^2), data = nba)

Residuals:
      Min        1Q    Median        3Q       Max
-0.255059 -0.083069 -0.001772  0.058228  0.396231

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                0.4397844  0.0151822  28.967   <2e-16 ***
I(age - mean(nba$age))     0.0007589  0.0036605   0.207    0.836
I((age - mean(nba$age))^2) -0.0014443  0.0009053  -1.595    0.114
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1155 on 102 degrees of freedom
Multiple R-Squared: 0.02634,   Adjusted R-squared: 0.007247
F-statistic:  1.38 on 2 and 102 DF,  p-value: 0.2563

>   predict(object = mod.fit3, newdata = data.frame(age =
      20))
[1] 0.3521019
>   c(1,20-mean(nba$age),(20-mean(nba$age))^2) %*%
      mod.fit3$coefficients  #Correct
         [,1]
[1,] 0.3521019
```

Now, *nba$age* in the formula option tells R to always use the nba *data.frame* and pull out age. Alternatively, one

could just find the mean of the ages before the *lm()* function and implement the following code:

```
mean.age<-mean(nba$age)
mod.fit4<-lm(formula = PPM ~ I(age-mean.age) +
   I((age-mean.age)^2), data = nba)
```

What about the confidence interval for E(Y)? As you would expect, these are the same for the first (*mod.fit1*) and third (*mod.fit3*) ways to fit the model.

```
>   #Compare C.I.s - notice they are the same
>   predict(object = mod.fit1, newdata = data.frame(age =
      20), se.fit = TRUE, interval = "confidence")
$fit
           fit       lwr       upr
[1,] 0.3521019 0.2350136 0.4691902

$se.fit
[1] 0.0590313

$df
[1] 102

$residual.scale
[1] 0.1154669

>   predict(object = mod.fit3, newdata = data.frame(age =
      20), se.fit = TRUE, interval = "confidence")
$fit
           fit       lwr       upr
[1,] 0.3521019 0.2350136 0.4691902

$se.fit
[1] 0.0590313

$df
[1] 102

$residual.scale
```

```
[1] 0.1154669
```

You may think that the first way would have a larger variance because the variability of the b_j's is larger. However, since the covariances need to be incorporated, the same $\text{Var}(\hat{Y}_h)$ is found. For the first model,

$$\begin{aligned}\text{Var}(\hat{Y}_h) &= \text{Var}(b_0 + b_1X + b_2X^2) \\ &= \text{Var}(b_0) + X^2\text{Var}(b_1) + X^4\text{Var}(b_2) + 2X\text{Cov}(b_0,b_1) \\ &\quad +2X^2\text{Cov}(b_0,b_2) + 2X^3\text{Cov}(b_1,b_2)\end{aligned}$$

and for the third model,

$$\begin{aligned}\text{Var}(\hat{Y}_h) &= \text{Var}(b_0 + b_1(X-\bar{X}) + b_2(X-\bar{X})^2) \\ &= \text{Var}(b_0) + (X-\bar{X})^2\text{Var}(b_1) \\ &\quad +(X-\bar{X})^4\text{Var}(b_2) + 2(X-\bar{X})\text{Cov}(b_0,b_1) \\ &\quad +2(X-\bar{X})^2\text{Cov}(b_0,b_2) + 2(X-\bar{X})^3\text{Cov}(b_1,b_2)\end{aligned}$$

where b_0, b_1, and b_2 are defined for the particular model (thus, they are NOT the same). In matrix notation, we found the variance to be $\text{Var}(\hat{Y}_h) = \sigma^2 * (\mathbf{X}_h'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}_h)$ in general for $\mathbf{X}_h = (1, X, X^2)$ or $\mathbf{X}_h = (1,(X-\bar{X}),(X-\bar{X})^2)'$. Using R to find the estimated quantities produces,

```
>   #Examine estimated covariance matrices for b
>   sum.fit1<-summary(mod.fit1)
>   cov.b.fit1<-sum.fit1$sigma^2 * sum.fit1$cov.unscaled

>   sum.fit3<-summary(mod.fit3)
>   cov.b.fit3<-sum.fit3$sigma^2 * sum.fit3$cov.unscaled
>   x<-20
>   x.bar<-mean(nba$age)
>   var.Y.hat.fit1<-cov.b.fit1[1,1] + x^2*cov.b.fit1[2,2] +
       x^4*cov.b.fit1[3,3] + 2*x*cov.b.fit1[1,2] +
       2*x^2*cov.b.fit1[1,3] + 2*x^3*cov.b.fit1[2,3]
>   var.Y.hat.fit3<-cov.b.fit3[1,1] +
       (x-x.bar)^2*cov.b.fit3[2,2] +
       (x-x.bar)^4*cov.b.fit3[3,3] +
       2*(x-x.bar)*cov.b.fit3[1,2] +
       2*(x-x.bar)^2*cov.b.fit3[1,3] +
       2*(x-x.bar)^3*cov.b.fit3[2,3]
>   sqrt(var.Y.hat.fit1)
[1] 0.0590313
>   sqrt(var.Y.hat.fit3)
[1] 0.0590313

>   #Using matrices
>   X.h<-c(1, 20, 20^2)
>   X<-cbind(1, nba$age, nba$age^2)
>   sqrt(as.numeric(sum.fit1$sigma^2)*X.h%*%solve(t(X)%*%X)
         %*%X.h)
          [,1]
[1,] 0.0590313

>   X.adj.h<-c(1, 20-x.bar, (20-x.bar)^2)
>   X.adj<-cbind(1, nba$age-x.bar, (nba$age-x.bar)^2)
>   sqrt(as.numeric(sum.fit3$sigma^2)*X.adj.h
      %*%solve(t(X.adj)%*%X.adj)%*%X.adj.h)
          [,1]
[1,] 0.0590313
```

Second order models can become more complicated with additional predictor variables.

1) Third order model with 1 predictor variable:

$$E(Y_i) = \beta_0 + \beta_1 Z_{i1} + \beta_2 Z_{i1}^2 + \beta_3 Z_{i1}^3$$

2) Second order model with 2 predictor variables:

$$E(Y_i) = \beta_0 + \beta_1 Z_{i1} + \beta_2 Z_{i2} + \beta_3 Z_{i1} Z_{i2} + \beta_4 Z_{i1}^2 + \beta_5 Z_{i2}^2$$

3) Second order model with 3 predictor variables:

$$E(Y_i) = \beta_0 + \beta_1 Z_{i1} + \beta_2 Z_{i2} + \beta_3 Z_{i3} + \beta_4 Z_{i1} Z_{i2} + \beta_5 Z_{i1} Z_{i3} + \beta_6 Z_{i2} Z_{i3} + \beta_7 Z_{i1}^2 + \beta_8 Z_{i2}^2 + \beta_9 Z_{i3}^2$$

Notice the last two models above contain "interaction" terms. Along with the "squared" terms, these are considered to be second order model terms as well.

There is a hierarchical approach to fitting the regression model. For example, if $\beta_3 \neq 0$ in 1), then Z_{i1}^2 and Z_{i1} are kept in the mode in addition to Z_{i1}^3. See p. 299 of KNN for a discussion.

Residual plots should always be examined to evaluate the assumptions of the model. When a squared or higher order term is in the model corresponding to Z_1, only the plot of e_i vs. Z_{i1} needs to be examined for the "linearity" assumption. This is because e_i vs. Z_{i2}^2, e_i vs. Z_{i1}^3, … give the same information as e_i vs. Z_1

See p.300 of KNN for an example of a second order model with two predictor variables.

Example: NBA data (nba_ch8.R)

Consider the following model: $E(PPM) = \beta_0 + \beta_1 MPG + \beta_2 Age + \beta_3 Age*MPG + \beta_4 MPG^2 + \beta_5 Age^2$. Find the corresponding sample regression model and perform a partial F test to determine if a second order model should be used instead of a first order model. Use $\alpha=0.05$.

Below is the R code and output. Note how I got the interaction term in the model.

```
>   mod.fit.comp<-lm(formula = PPM ~ age + MPG + age:MPG +
                          I(age^2) + I(MPG^2), data = nba)
>   sum.fit.comp<-summary(mod.fit.comp)
>   sum.fit.comp

Call:
lm(formula = PPM ~ age + MPG + age:MPG + I(age^2) +
I(MPG^2), data = nba)

Residuals:
     Min       1Q   Median       3Q      Max
-0.18857 -0.07244 -0.01223  0.05517  0.30057

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.1943663  0.7416479   1.610 0.110490
age         -0.0332123  0.0512598  -0.648 0.518535
MPG         -0.0335019  0.0090391  -3.706 0.000347 ***
I(age^2)     0.0002166  0.0008646   0.251 0.802680
I(MPG^2)     0.0003221  0.0001150   2.800 0.006142 **
age:MPG      0.0008442  0.0003401   2.482 0.014746 *
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1

Residual standard error: 0.09997 on 99 degrees of freedom
Multiple R-Squared: 0.2916,     Adjusted R-squared: 0.2558
F-statistic: 8.149 on 5 and 99 DF,  p-value: 1.787e-06

>   sum.fit.comp$sigma^2
[1] 0.009994667

>   mod.fit.red<-lm(formula = PPM ~ age + MPG, data = nba)
>   summary(mod.fit.red)

Call:
lm(formula = PPM ~ age + MPG, data = nba)

Residuals:
      Min        1Q    Median        3Q       Max
-0.199997 -0.072681 -0.004975  0.051162  0.409765

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.422674   0.088204   4.792 5.63e-06 ***
age         -0.003906   0.003209  -1.217    0.226
MPG          0.004461   0.001097   4.068 9.35e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1

Residual standard error: 0.1084 on 102 degrees of freedom
Multiple R-Squared: 0.1414,     Adjusted R-squared: 0.1245
F-statistic: 8.396 on 2 and 102 DF,  p-value: 0.0004211

>   anova(mod.fit.red, mod.fit.comp)
Analysis of Variance Table

Model 1: PPM ~ age + MPG
Model 2: PPM ~ age + MPG + age * MPG + I(age^2) + I(MPG^2)
  Res.Df     RSS  Df Sum of Sq      F    Pr(>F)
1    102 1.19927
2     99 0.98947   3   0.20980 6.9971 0.0002568 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1

>   p<-length(mod.fit.comp$coefficients)  #Number of betas
                                           in model is p
>   g<-length(mod.fit.red$coefficients)-1 #Number of
                                variables remaining in the
                                reduced model is g
>   qf(p = 0.95, df1=p-1-g, df2=mod.fit.comp$df.residual)
[1] 2.696469
```

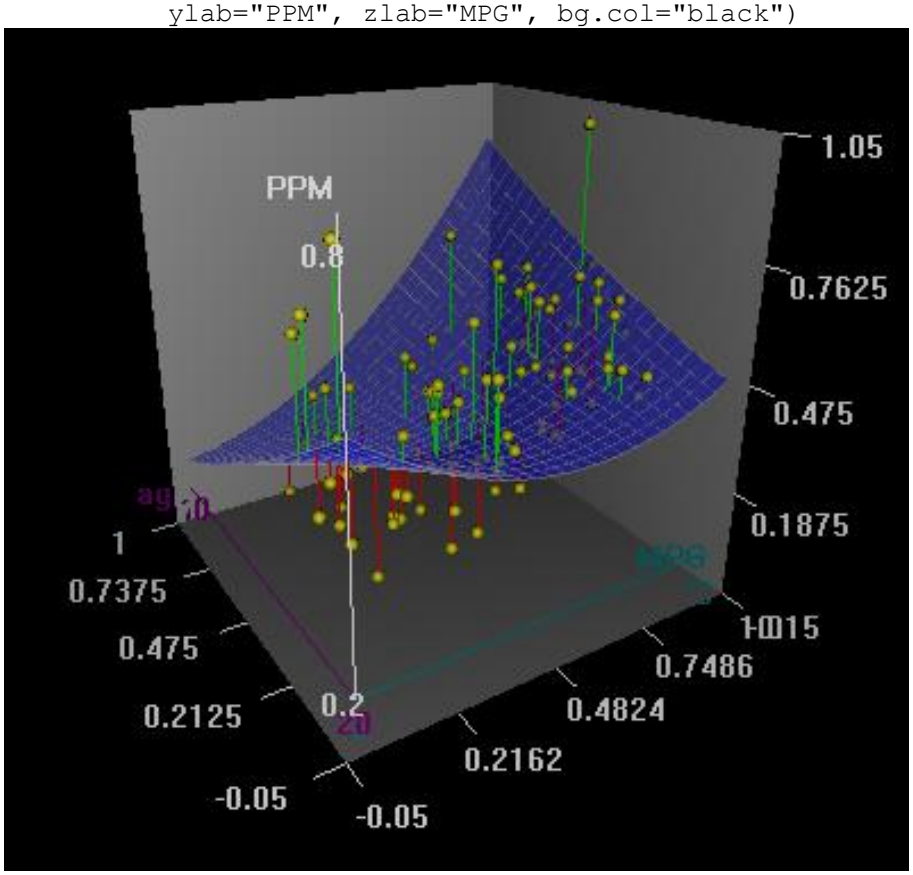
1) H_0: $E(PPM) = \beta_0 + \beta_1 MPG + \beta_2 Age$
   H_a: $E(PPM) = \beta_0 + \beta_1 MPG + \beta_2 Age + \beta_3 Age*MPG + \beta_4 MPG^2 + \beta_5 Age^2$
2) $$F^* = \frac{\left(SSE(MPG, Age) - SSE(MPG, Age, MPG*Age, MPG^2, Age^2)\right)/3}{MSE(MPG, Age, MPG*Age, MPG^2, Age^2)}$$
   $$= \frac{(1.19927 - 0.98947)/3}{0.00999} = 6.9971$$
3) $F(0.95, 3, 99)=2.69647$
4) Since $6.9971>2.7$, reject H_0
5) At least one of MPG^2, Age^2, and $MPG*Age$ are important to the model.

The p-value of 0.0002568 could be used more simply as well.

A 3D scatter plot of the data with the regression plane is shown below.

```
library(Rcmdr)
library(rgl)
```

```
rgl.clear("all") #Clears plot window
rgl.light() #Gray background
rgl.bbox()  #Puts numbers on plot and box around it
scatter3d(formula = PPM ~ age + MPG, data = nba,
        fit="quadratic", grid=TRUE, xlab="age",
        ylab="PPM", zlab="MPG", bg.col="black")
```



One can also put both the first and second-order models on the plot by using *c("linear", "quadratic")* for the *fit* option.

8.2 Interaction regression models

The effect of one predictor variable on the response variable depends on another predictor variable.

Example: Suppose there are two predictor variables

Consider the first order model: $E(Y_i)=\beta_0+\beta_1X_{i1}+\beta_2X_{i2}$. The effect of X_1 on E(Y) is measured by β_1.

Consider the model $E(Y_i)=\beta_0+\beta_1X_{i1}+\beta_2X_{i2}+\beta_3X_{i1}X_{i2}$. The effect of X_1 on E(Y) is measured only by β_1 and β_3X_{i2}. Since X_2 is a predictor variable, the effect of X_1 on E(Y) is dependent on X_2. Similarly, the effect of X_2 on E(Y) is dependent on X_1. Thus, we say there is an "interaction" between X_1 and X_2.

For a model containing an interaction term, the regression function is no longer a flat plane.

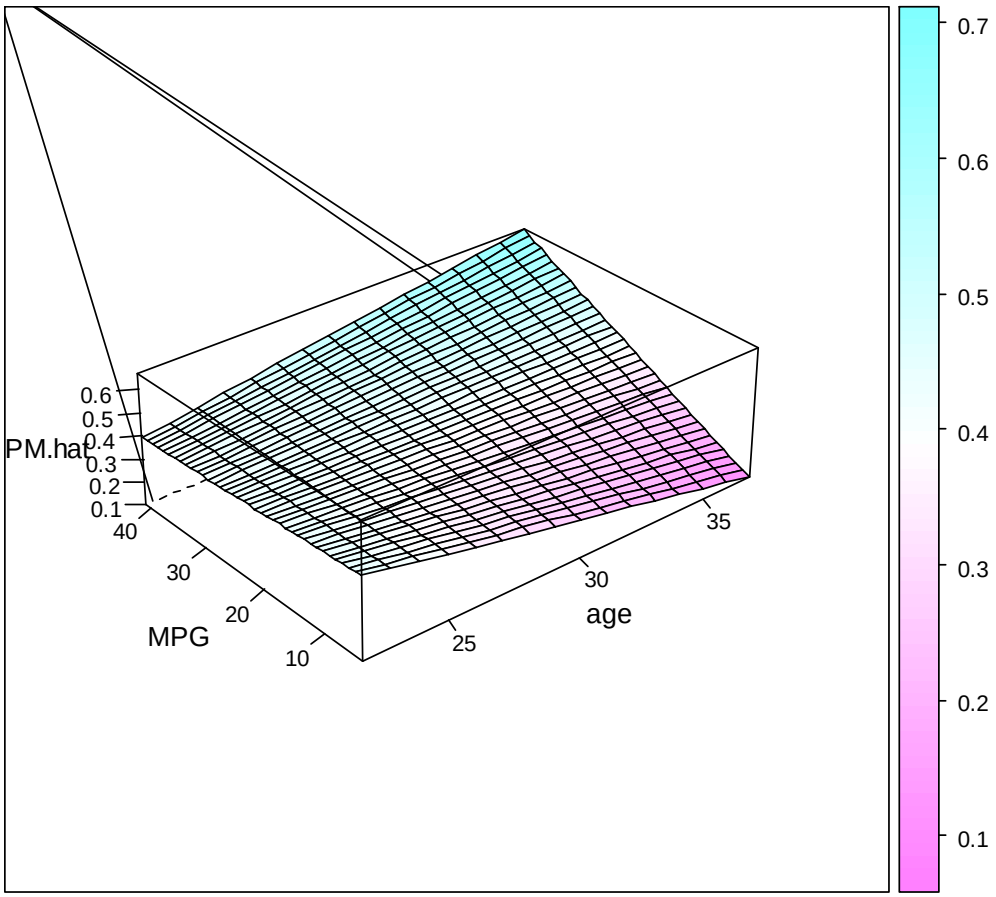
Example: NBA data (nba_ch7.R)

Find the estimated regression model for $E(PPM) = \beta_0 + \beta_1MPG + \beta_2Age + \beta_3MPG*Age$.

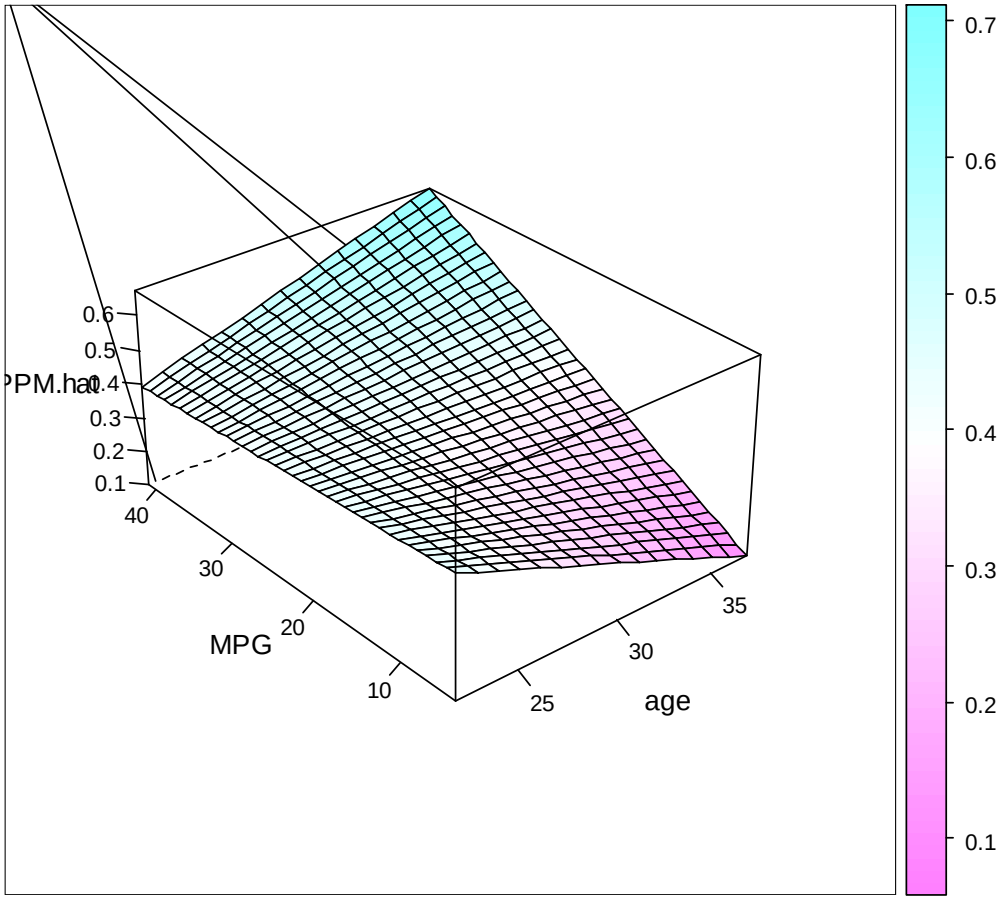
R code and output:

```
>   mod.fit.inter<-lm(formula = PPM ~ age + MPG + age:MPG,
                      data = nba)
>   summary(mod.fit.inter)

Call:
lm(formula = PPM ~ age + MPG + age:MPG, data = nba)

Residuals:
      Min        1Q    Median        3Q       Max
-0.181469 -0.086163 -0.004787  0.052240  0.344910

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.0710195  0.2037211   5.257 8.19e-07 ***
age         -0.0279646  0.0075348  -3.711 0.000338 ***
MPG         -0.0264535  0.0089165  -2.967 0.003756 **
age:MPG      0.0011338  0.0003248   3.491 0.000715 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1

Residual standard error: 0.1029 on 101 degrees of freedom
Multiple R-Squared: 0.2338,     Adjusted R-squared: 0.2111
F-statistic: 10.27 on 3 and 101 DF,  p-value: 5.806e-06

>   library(lattice)
>   save.xyz<-expand.grid(age = min(nba$age):max(nba$age),
      MPG = floor(min(nba$MPG)):ceiling(max(nba$MPG)))
>   save.xyz$PPM.hat<-predict(object = mod.fit.inter,
      newdata = save.xyz)

>   win.graph(width = 8, height = 6, pointsize = 10)
>   wireframe(PPM.hat ~ age + MPG, data = save.xyz, scales
            = list(arrows = FALSE), drape = TRUE,
            colorkey = TRUE, aspect = c(0.7, 0.3))
```

8.20




```
> wireframe(PPM.hat ~ age + MPG, data = save.xyz, scales =
            list(arrows = FALSE), drape = TRUE,
            colorkey = TRUE, aspect = c(1.3, 0.6))
```



Notes:

1) The interaction term is represented as age*MPG in the formula statement of *lm()*. There are other ways to represent this as well.

```
>   mod.fit.inter1<-lm(formula = PPM ~ age + MPG + age:MPG,
                       data = nba)
>   mod.fit.inter1$coefficients
 (Intercept)          age          MPG      age:MPG
 1.071019547 -0.027964607 -0.026453471  0.001133794
```

```
>   mod.fit.inter2<-lm(formula = PPM ~ age*MPG, data = nba)
>   mod.fit.inter2$coefficients
 (Intercept)          age          MPG      age:MPG
 1.071019547 -0.027964607 -0.026453471  0.001133794

>   mod.fit.inter3<-lm(formula = PPM ~ (age+MPG)^2, data =
                       nba)
>   mod.fit.inter3$coefficients
 (Intercept)          age          MPG      age:MPG
 1.071019547 -0.027964607 -0.026453471  0.001133794
```

age*MPG – this puts both the interaction AND the "main-effects" (first-order terms) into the model. Thus, our original use in the formula statement of

```
formula = PPM ~ age + MPG + age * MPG
```

was actually redundant. R recognizes this and adjusts it accordingly.

age:MPG – this puts ONLY the interaction term into the model. Notice in the output how R states the interaction as age:MPG.

(age+MPG)^2 – this also puts the interaction AND the main-effects into the model. Note that (age+MPG+var3)^3 would put in all main-effects, all two-way interactions, and the three-way interaction into the model.

2) The sample model is

$$\hat{PPM} = 1.0710 - 0.02796age - 0.02645MPG + 0.001134age * MPG$$

3) The interaction term is significant.
4) The estimated regression model is not a flat plane. Some people call it a “twisted” plane. I was unable to adjust the *scatterplot3d()* or the *scatter3d()* functions to construct the plot.

Sometimes problems with multicollinearity can occur when interaction terms are in the model. Similar to polynomial models, a transformation of $Z_{ij}=X_{ij}-\bar{X}_j$ can be done to partially remedy the problem.

8.3 Qualitative predictors

Quantitative variables – Variables measured on a numerical scale.

Example: Height measured in inches

Qualitative variables – Variables that can not be measured on a numerical scale (i.e. measured on a categorical scale).

Example: Gender – Male or female

Code the data values for qualitative variables with indicator variables (0 or 1)

Example: Gender

X = 1 if female (F)
0 if male (M)

Notice there is only one indicator variable for 2 levels (male or female) of the qualitative variable; however, each level was a unique coding.

Suppose the model is $E(Y) = \beta_0 + \beta_1 X$.

If X = 0 (Male), then $E(Y) = \mu_M = \beta_0$.
If X = 1 (Female), then $E(Y) = \mu_F = \beta_0 + \beta_1$.

where μ_M and μ_F are the mean of the response variable for male and female, respectively.

Then, $\mu_F - \mu_M = \beta_0 + \beta_1 - \beta_0 = \beta_1$

This means that the t-test for a hypothesis test of $H_0:\beta_1=0$ is equivalent to a hypothesis test for $H_0:\mu_F - \mu_M=0$ (hypothesis test for differences between means of independent samples).

The model is simply a one-factor (or one-way) analysis of variance model. You may be more used to seeing the model in the form of

$$E(Y) = \mu + \alpha_i \text{ for } i = 1, 2$$

where μ is the grand mean and α_i represents the "effect" of the i^{th} treatment level. In this setting, one often will let $\alpha_1 = 0$ for identifiability purposes. Then $E(Y) = \mu$ for $i = 1$ and $E(Y) = \mu + \alpha_2$ for $i = 2$. Notice how this matches up with $E(Y) = \beta_0$ and $E(Y) = \beta_0 + \beta_1$ shown previously. Thus, a regression model with one qualitative variable is equivalent to a one-factor ANOVA model!

Example: Political party affiliation (Republican, Democrat, and Independent)

X_1 =1 if Republican, 0 otherwise
X_2 =1 if Democrat, 0 otherwise

Notice there are only two indicator variable for 3 levels (Republican, Democrat, and Independent) of the qualitative variable; however, each level was a unique coding.

Party	X_1	X_2
Republican	1	0
Democrat	0	1
Independent	0	0

Suppose the model is $E(Y) = \beta_0 + \beta_1 X_1 + \beta_2 X_2$. Note that this is a special case of a multiple linear regression model. The first subscript on X denotes the observation number, and the second subscript denotes the variable number. The hypothesis test for $H_0: \beta_1 = \beta_2 = 0$ uses a F-test which is equivalent to the F-test for $H_0: \mu_R = \mu_D = \mu_I$ in a one-factor, completely randomized design, ANOVA model. Since political party affiliation is ONE variable, a F-test must be used to determine if it is an important variable. T-tests only test one indicator variable at a time. **DO NOT PERFORM INDIVIDUAL T-TESTS FOR INDICATOR VARIBALES!**

Note:

Republican: $E(Y) = \beta_0 + \beta_1$

Democrat: $E(Y) = \beta_0 + \beta_2$

Independent: $E(Y) = \beta_0$

In general, if there are c different levels for a qualitative variable, then c-1 indicator variables are needed. No matter what you choose as the coding for the indicator variables (Republican, Democrat, or Independent is the "all 0 level"), you will get the same $\hat{Y}$ and hypothesis tests involving the indicator variables will be the same.

Example: Car Highway MPG (car_mpg.R)

Estimate the highway MPG for cars based on their class.

Below is a partial listing of a data set that contains MPG and other information about 1998 cars.

Commented [b1]: http://www.eren.doe.gov

Obs.	Make	Model	MPG	Class	Trans.	Engine	Cylinders	Speed
1	Acura	2.5TL/3.2TL	25	Compact	L	2.5	5	4
2	Acura	2.5TL/3.2TL	24	Compact	L	3.2	6	4
3	Acura	3.5RL	25	Mid-Size	L	3.5	6	4
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
376	Volvo	V70	20	Station Wagon	L	2.4	5	4

Response Variable	MPG: Highway miles per gallon
Predictor Variables	Trans.: transmission type with M=Manual, A=Automatic, and L=Automatic Lockup
	Speed: number of gears
	Engine: engine size (liters)
	Cylinder: number of cylinders for the engine
	Class: the class of car with levels of two-seater, minicompact, subcompact, compact, mid-size, large, and station wagon.

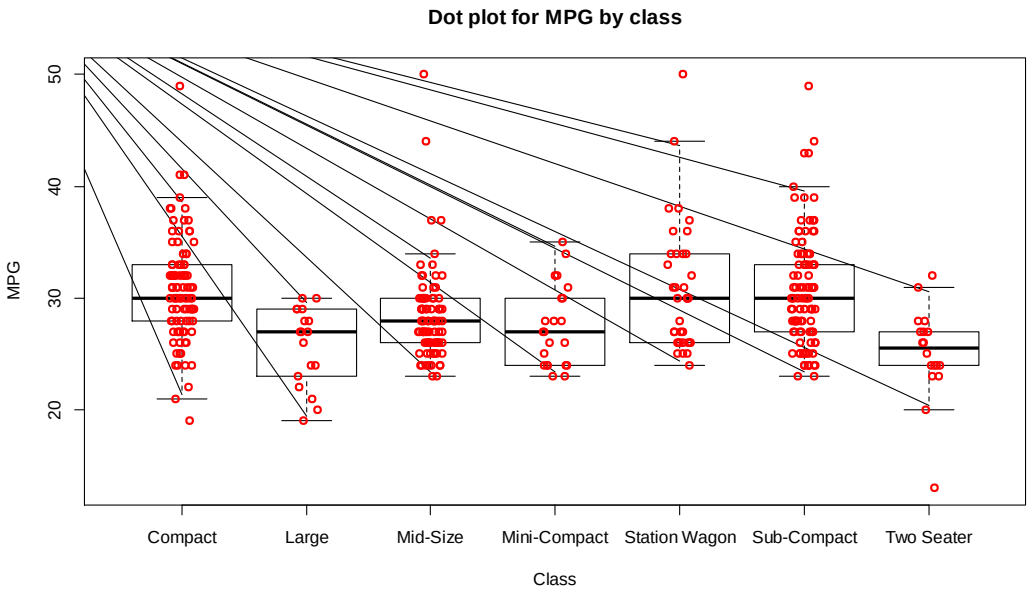
The data is stored in an Excel file called car_data98.xls. The file can be read into R using the following code:

```
> library(RODBC)
> z<-odbcConnectExcel("C:\\chris\\UNL\\STAT870\\
    Chapter8\\car_data98.xls")
> car.data<-sqlFetch(z, "Sheet1")
> close(z)
> head(car.data)
  Obs       Model  Make MPG       Class Transmission Engine Cylinders Speed Abbr
1   1 2.5TL/3.2TL Acura  25     Compact            L    2.5         5     4    P
2   2 2.5TL/3.2TL Acura  24     Compact            L    3.2         6     4    P
3   3       3.5RL Acura  25    Mid-Size            L    3.5         6     4    P
4   4 2.3CL/3.0CL Acura  28 Sub-Compact            L    3.0         6     4
5   5     Integra Acura  31 Sub-Compact            L    1.8         4     4
6   6     Integra Acura  31 Sub-Compact            M    1.8         4     5
```

Suppose we are interested in the MPG by class. Below is a box plot with a dot plot overlay:

```
> boxplot(formula = MPG ~ Class, data = car.data, main =
```

```
      "Dot plot for MPG by class", ylab = "MPG", xlab =
      "Class", pars = list(outpch=NA))
> stripchart(x = car.data$MPG ~ car.data$Class, lwd = 2,
    col = "red", method = "jitter", vertical = TRUE, pch =
    1, xlab = "Class", ylab = "MPG", main = "Dot plot for
    MPG by class", add = TRUE)
```



Are there differences in the mean MPG among car classes?

Some summary statistics:

```
> aggregate(formula = MPG ~ Class, data = car.data, FUN =
    mean)
          Class      MPG
1       Compact 30.73118
2         Large 25.72222
3      Mid-Size 28.35632
4  Mini-Compact 27.50000
5 Station Wagon 30.81081
```

```
6   Sub-Compact 30.67327
7    Two Seater 25.11111

> aggregate(formula = MPG ~ Class, data = car.data, FUN =
    sd)
          Class      MPG
1       Compact 4.658185
2         Large 3.528132
3      Mid-Size 4.043303
4  Mini-Compact 3.635146
5 Station Wagon 5.685448
6   Sub-Compact 5.034102
7    Two Seater 4.185253
```

Examine what happens when we use the Class variable to predict MPG with the *lm()* function.

```
> mod.fit<-lm(formula = MPG ~ Class, data = car.data)
> summary(mod.fit)

Call:
lm(formula = MPG ~ Class, data = car.data)

Residuals:
    Min      1Q  Median      3Q     Max
-12.111  -2.947  -0.500   2.290  21.644

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        30.73118    0.47940  64.103  < 2e-16 ***
ClassLarge         -5.00896    1.19049  -4.207 3.25e-05 ***
ClassMid-Size      -2.37486    0.68957  -3.444 0.000639 ***
ClassMini-Compact  -3.23118    1.09607  -2.948 0.003402 **
ClassStation Wagon  0.07963    0.89861   0.089 0.929438
ClassSub-Compact   -0.05792    0.66442  -0.087 0.930586
ClassTwo Seater    -5.62007    1.19049  -4.721 3.34e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1

Residual standard error: 4.623 on 369 degrees of freedom
```

```
Multiple R-Squared: 0.1271,     Adjusted R-squared: 0.1129
F-statistic: 8.958 on 6 and 369 DF,  p-value: 3.831e-09
```

R will automatically create 6 indicator variables to represent the 7-level Class predictor variable. For example "ClassLarge" above is an indicator variable of the form

$$X_1 = \begin{cases} 1 \text{ if class = large} \\ 0 \quad \text{otherwise} \end{cases}$$

To see exactly how R codes all of the predictor variables, use the contrasts function:

```
> contrasts(car.data$Class)
             Large Mid-Size Mini-Compact Station Wagon Sub-Compact Two Seater
Compact          0        0            0             0           0          0
Large            1        0            0             0           0          0
Mid-Size         0        1            0             0           0          0
Mini-Compact     0        0            1             0           0          0
Station Wagon    0        0            0             1           0          0
Sub-Compact      0        0            0             0           1          0
Two Seater       0        0            0             0           0          1
```

Here's a table showing the different indicator variables.

	X_1	X_2	X_3	X_4	X_5	X_6
Compact	0	0	0	0	0	0
Large	1	0	0	0	0	0
Mid-Size	0	1	0	0	0	0
Minicompact	0	0	1	0	0	0
Station Wagons	0	0	0	1	0	0
Subcompact	0	0	0	0	1	0

Two-seater	0	0	0	0	0	1

By default, R will put the levels of a qualitative variable in alphabetical order and set the first level to 0 for all indicator variables. Thus, “Compact” is the base level from which comparisons are made to.

To see the ordering among the levels of class, the *levels()* function can be used:

```
> levels(car.data$Class)
[1] "Compact"       "Large"         "Mid-Size"      "Mini-
     Compact"  "Station Wagon" "Sub-Compact"
[7] "Two Seater"
```

The sample regression model is:

$$\hat{Y} = 30.7312 - 5.0090X_1 - 2.3749X_2 - 3.2312X_3 + 0.07963X_4 - 0.05792X_5 - 5.6201X_6.$$

A more descriptive way of writing this model is:

$$\hat{MPG} = 30.7312 - 5.0090\text{Large} - 2.3749\text{Mid-size} - 3.2312\text{Mini-compact} + 0.07963\text{StationWagon} - 0.05792\text{Sub-compact} - 5.6201\text{Two-seater}.$$

Questions:

1. Is class linearly related to MPG?
2. What is the estimated MPG for compact cars?
3. What is the estimated MPG for two-seaters?

Commented [CRB2]: Yes, F-test p-value is 3.8*10^(-9)

Commented [CRB3]: 30.73 = b_0

Commented [CRB5]: 30.7312 - 5.6201 (notice it is just a deviation from compact cars)

4. Which car gets the worst gas mileage?

Commented [CRB4]: 25.11 = two-seater

To estimate the mean response for a class, we can use the *predict()* function again:

```
> predict(object = mod.fit, newdata = data.frame(Class =
    "Compact"), se.fit = TRUE, interval = "confidence",
    level = 0.95)
$fit
       fit      lwr      upr
1 30.73118 29.78848 31.67389

$se.fit
[1] 0.4794041

$df
[1] 369

$residual.scale
[1] 4.623205
```

Notice that $\hat{Y}$ = 30.73, which is exactly the same as the sample mean for compact cars!

Additional output:

```
> mpg.compact<-car.data$MPG[car.data$Class == "Compact"]
> t.test(x = mpg.compact, conf.level = 0.95)

        One Sample t-test

data:  mpg.compact
t = 63.6215, df = 92, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 29.77184 31.69052
sample estimates:
mean of x
```

```
 30.73118

> sd(mpg.compact)/sqrt(length(mpg.compact))
[1] 0.4830313
```

Notes:

- The confidence interval using the sample regression model is a little smaller in width than the "usual" t-based interval that you learned about in STAT 801. What are the differences between the formulas?
- $\sqrt{MSE}$ = 0.4794 which is a little less than $s/\sqrt{n}$ = 0.4830
- An important advantage to using the sample regression model rather than analyzing MPG one class at a time is that you obtain one overall measure of the importance of class.
- Another important advantage is that one can easily incorporate other important predictor variables into the analysis with a sample regression model.

Commented [cb26]: t dist value and regression model calculations variance differently (uses information across all classes - assumes equal variances)

Suppose you wanted to compare the mean of compact to the mean of large.

$\mu_{compact}$: $E(Y) = \beta_0$
μ_{Large}: $E(Y) = \beta_0 + \beta_1$

Thus, $\mu_{Large} - \mu_{compact} = \beta_1$. A test of H_0: $\beta_1 = 0$ vs. H_a: $\beta_1 \neq 0$ tests for equality of means. From the output, we see that the p-value was $3.25 * 10^{-5}$. There is strong evidence

of a difference in mean MPG for large and compact cars. Compare this result back to the box and dot plot.

Are there mean MPG differences between compact and the other car classes?

To compare classes other than compact, we can use the following methods:
1)Use the *relevel()* function to change the base level:

To compare to the Large class:

```
> car.data$Class<-relevel(x = car.data$Class, ref =
    "Large")
> levels(car.data$Class)
[1] "Large"         "Compact"       "Mid-Size"      "Mini-
    Compact"  "Station Wagon" "Sub-Compact"
[7] "Two Seater"
> mod.fit.relevel<-lm(formula = MPG ~ Class, data =
    car.data)
> summary(mod.fit.relevel)

Call:
lm(formula = MPG ~ Class, data = car.data)

Residuals:
    Min      1Q  Median      3Q     Max
-12.111  -2.947  -0.500   2.290  21.644

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)         25.7222     1.0897  23.605  < 2e-16 ***
ClassCompact         5.0090     1.1905   4.207 3.25e-05 ***
ClassMid-Size        2.6341     1.1971   2.200  0.02840 *
ClassMini-Compact    1.7778     1.4694   1.210  0.22709
ClassStation Wagon   5.0886     1.3286   3.830  0.00015 ***
ClassSub-Compact     4.9510     1.1828   4.186 3.56e-05 ***
```

```
ClassTwo Seater     -0.6111     1.5411  -0.397  0.69193
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
’ 1

Residual standard error: 4.623 on 369 degrees of freedom
Multiple R-squared: 0.1271,     Adjusted R-squared: 0.1129
F-statistic: 8.958 on 6 and 369 DF,  p-value: 3.831e-09
```

Notice the p-value for Compact is exactly the same as it was before. Also, the b_1 estimate is 5.009 for Compact, where the estimate for Large was -5.009 with the previous model.

All confidence intervals for E(Y), prediction intervals Y, and values of $\hat{Y}$ remain the same as before.

2)Use the multcomp package

```
> library(package = multcomp)
> #Large vs. Mid-size cars - Contrast between E(Y) = beta0
    + beta1 and E(Y) = beta0 + beta2
> K<-matrix(data = c(0,1,-1,0,0,0,0), nrow = 1, ncol = 7,
    byrow = TRUE)
> compare.means<-glht(model = mod.fit, linfct = K)
> summary(compare.means, test = adjusted("none"))

         Simultaneous Tests for General Linear Hypotheses

Fit: lm(formula = MPG ~ Class, data = car.data)

Linear Hypotheses:
       Estimate Std. Error t value Pr(>|t|)
1 == 0   -2.634      1.197    -2.2   0.0284 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
’ 1
(Adjusted p values reported -- none method)
```

```
> #Large vs. Mid-size and Large vs. Compact
> K<-matrix(data = c(0,1,-1,0,0,0,0,
                     0,1, 0,0,0,0,0), nrow = 2, ncol = 7,
    byrow = TRUE)
> compare.means<-glht(model = mod.fit, linfct = K)
> summary(compare.means, test = adjusted("none"))

         Simultaneous Tests for General Linear Hypotheses

Fit: lm(formula = MPG ~ Class, data = car.data)

Linear Hypotheses:
       Estimate Std. Error t value Pr(>|t|)
1 == 0   -2.634      1.197  -2.200   0.0284 *
2 == 0   -5.009      1.190  -4.207 3.25e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
’ 1
(Adjusted p values reported -- none method)

>  K<-matrix(data = c(0,1,-1,0,0,0,0,
                      0,1, 0,0,0,0,0), nrow = 2, ncol = 7,
     byrow = TRUE)
> compare.means<-glht(model = mod.fit, linfct = K)
> summary(compare.means, test = adjusted("bonferroni"))

         Simultaneous Tests for General Linear Hypotheses

Fit: lm(formula = MPG ~ Class, data = car.data)

Linear Hypotheses:
       Estimate Std. Error t value Pr(>|t|)
1 == 0   -2.634      1.197  -2.200   0.0568 .
2 == 0   -5.009      1.190  -4.207 6.49e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
’ 1
(Adjusted p values reported -- bonferroni method)
```

The sample regression model here is equivalent to what is often referred to as a single-factor ANOVA model. The ANOVA model is sometimes written as

$$E(y) = \mu + \alpha_i \text{ for } i = 1, \ldots, 7$$

where μ is the "grand mean" and α_i is a parameter representing the deviation from the grand mean for each car class. Similar to what we did for the regression model, the α_1 is set to 0 for compact cars so that we have

	Model
Compact	$E(Y) = \mu$
Large	$E(Y) = \mu + \alpha_2$
Mid-Size	$E(Y) = \mu + \alpha_3$
Minicompact	$E(Y) = \mu + \alpha_4$
Station Wagons	$E(Y) = \mu + \alpha_5$
Subcompact	$E(Y) = \mu + \alpha_6$
Two-seater	$E(Y) = \mu + \alpha_7$

Thus, μ is equivalent to our β_0, α_2 is equivalent to our β_1, …, and α_7 is equivalent to our β_6.

STAT 802 discusses ANOVA models in more detail.

Suppose there is a qualitative variable of interest that has numbers for its values. While there is no variables like this here, suppose we treat Cylinders as it if were. Examine the two sets of output below.

Commented [b7]: For example, Likert scale measured variable may be one place where this occurs; Also, multiple choice question where the answers are coded as 1, 2, 3, 4, 5 while answer 5 is not 4 units greater than answer 1.

```
>   mod.fit1<-lm(formula = MPG ~ Cylinders, data =
                 car.data)
>   summary(mod.fit1)

Call:
lm(formula = MPG ~ Cylinders, data = car.data)

Residuals:
     Min       1Q   Median       3Q      Max
-7.95345 -1.95345  0.04655  1.36348 18.04655

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  40.2704     0.6733   59.81   <2e-16 ***
Cylinders    -2.0792     0.1243  -16.72   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1

Residual standard error: 3.718 on 374 degrees of freedom
Multiple R-Squared: 0.4278,     Adjusted R-squared: 0.4263
F-statistic: 279.7 on 1 and 374 DF,  p-value: < 2.2e-16

>   mod.fit2<-lm(formula = MPG ~ factor(Cylinders), data =
                 car.data)
>   summary(mod.fit2)

Call:
lm(formula = MPG ~ factor(Cylinders), data = car.data)

Residuals:
    Min      1Q  Median      3Q     Max
-8.2613 -2.2613 -0.2613  1.7387 17.7387

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)           49.000      3.570  13.724  < 2e-16 ***
factor(Cylinders)4   -16.739      3.579  -4.676 4.10e-06 ***
factor(Cylinders)5   -22.286      3.817  -5.839 1.15e-08 ***
factor(Cylinders)6   -21.976      3.584  -6.131 2.24e-09 ***
factor(Cylinders)8   -24.568      3.618  -6.790 4.49e-11 ***
factor(Cylinders)12  -30.800      3.911  -7.875 3.82e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.57 on 370 degrees of freedom
Multiple R-Squared: 0.478,      Adjusted R-squared: 0.4709
F-statistic: 67.76 on 5 and 370 DF,  p-value: < 2.2e-16

>   factor(car.data$Cylinders)
  [1] 5  6  6  6  4  4  4  6  6  4  4  6  6  4  4  6  6  6
6  8  8  6  6  6  6  4  4  6  6  8  8  8
 [33] 12 8  6  6  6  6  4  4  6  6  6  6  6  6  4  4  6  6
4  6  6  6  6  6  6  6  6  8  6  8  8  4
```

Edited

```
 [353] 6  6  4  4  4  4  6  6  6  4  4  4  4  6  4  4  6  5
5  5  5  5  6  5
Levels: 3 4 5 6 8 12

>   contrasts(factor(car.data$Cylinders))
   4 5 6 8 12
3  0 0 0 0  0
4  1 0 0 0  0
5  0 1 0 0  0
6  0 0 1 0  0
8  0 0 0 1  0
12 0 0 0 0  1
```

The first model is $\widehat{MPG} = 40.2704 - 2.0792\text{Cylinders}$ where Cylinders is treated correctly as a quantitative variable. For illustrative purposes only, one can have Cylinders treated as a qualitative variable by using the *factor()* function. The second model does and creates 5

indicator variables to represent the 6 different cylinder values in the data set.

Question: Why is treating Cylinders as a quantitative variable better?

8.4 Some considerations in using indicator variables

Please read on your own.

8.5 Modeling interactions between quantitative and qualitative variables and 8.6 More complex models

The interaction terms between a quantitative and qualitative variable involves all indicator variables multiplied by the quantitative variable.

The interaction terms between two qualitative variables involves all of their corresponding indicator variables multiplied by each other. Thus, if the first variable has c levels and the second variable has d levels, there are (c-1)∗(d-1) terms needed to represent their interaction.

Example: Car Highway MPG (car_mpg.R)

Suppose engine size and class are used to predict MPG.

```
>   mod.fit3<-lm(formula = MPG ~ Engine + Class + Engine:Class, data
                 = car.data)
>   summary(mod.fit3)

Call:
lm(formula = MPG ~ Engine + Class + Engine:Class, data = car.data)

Residuals:
    Min      1Q  Median      3Q     Max
-8.0764 -2.0802 -0.3274  1.5063 19.0958

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                41.1196     1.2338  33.327  < 2e-16 ***
Engine                     -4.3835     0.4969  -8.822  < 2e-16 ***
ClassLarge                 -0.8976     4.6215  -0.194 0.846113
ClassMid-Size              -5.3575     1.9689  -2.721 0.006821 **
ClassMini-Compact          -2.6585     3.0719  -0.865 0.387385
ClassStation Wagon          8.7483     3.1419   2.784 0.005644 **
```

```
ClassSub-Compact           -2.9217     1.5431  -1.893 0.059096 .
ClassTwo Seater           -10.4879     2.2823  -4.595 5.97e-06 ***
Engine:ClassLarge           0.8277     1.1822   0.700 0.484299
Engine:ClassMid-Size        1.8267     0.7143   2.558 0.010949 *
Engine:ClassMini-Compact    0.1230     1.1644   0.106 0.915943
Engine:ClassStation Wagon  -3.8828     1.3242  -2.932 0.003581 **
Engine:ClassSub-Compact     1.3227     0.6069   2.180 0.029929 *
Engine:ClassTwo Seater      2.6877     0.7270   3.697 0.000252 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.553 on 362 degrees of freedom
Multiple R-Squared: 0.4942,     Adjusted R-squared: 0.4761
F-statistic: 27.21 on 13 and 362 DF,  p-value: < 2.2e-16
```

The sample model is

$$\widehat{MPG} = 41.1196 - 4.3835Engine - 0.8976Large - 5.3575Mid\text{-}size - 2.6585Mini\text{-}compact + 8.7483StationWagon - 2.9217Sub\text{-}compact - 10.4879Two\text{-}seater + 0.8277Engine*Large + 1.8267Engine*Mid\text{-}size + 0.1230Engine*Mini\text{-}compact - 3.8828Engine*StationWagon + 1.3227Engine*Sub\text{-}compact + 2.6877Engine*Two\text{-}seater$$

A partial F test can be performed to determine if the interaction term is needed in the model. Let $\alpha = 0.05$.

```
>   mod.fit.red<-lm(formula = MPG ~ Engine + Class, data =
                    car.data)

>   anova(mod.fit.red, mod.fit3)
Analysis of Variance Table
```

```
Model 1: MPG ~ Engine + Class
Model 2: MPG ~ Engine + Class + Engine:Class
  Res.Df    RSS  Df Sum of Sq      F    Pr(>F)
1    368 4994.1
2    362 4570.1   6     424.0 5.5979 1.432e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1
```

1) H_0: $\beta_8=\beta_9=\beta_{10}=\beta_{11}=\beta_{12}=\beta_{13}=0$
   H_a: At least one of the β's does not equal to 0
2) $F^* = 5.5979$ and p-value = $1.4*10^{-5}$
3) $\alpha = 0.05$
4) Since $1.4*10^{-5} < 0.05$, reject H_0
5) There is sufficient evidence to indicate an interaction between engine and class.

Since there is only one quantitative variable, we can create a 2D plot of the model. There will be a line on the plot for each class.

Class	**Model**
Compact	$\hat{MPG} = 41.1196 - 4.3835Engine$
Large	$\hat{MPG} = 41.1196 - 4.3835Engine$ $- 0.8976*1 + 0.8277Engine*1$ $= 40.2220 - 3.5558Engine$
⋮	
Two-seater	$\hat{MPG} = 41.1196 - 4.3835Engine$ $- 10.4879Two\text{-}seater + 2.6877Engine*1$

	= 30.6317 – 1.6958Engine

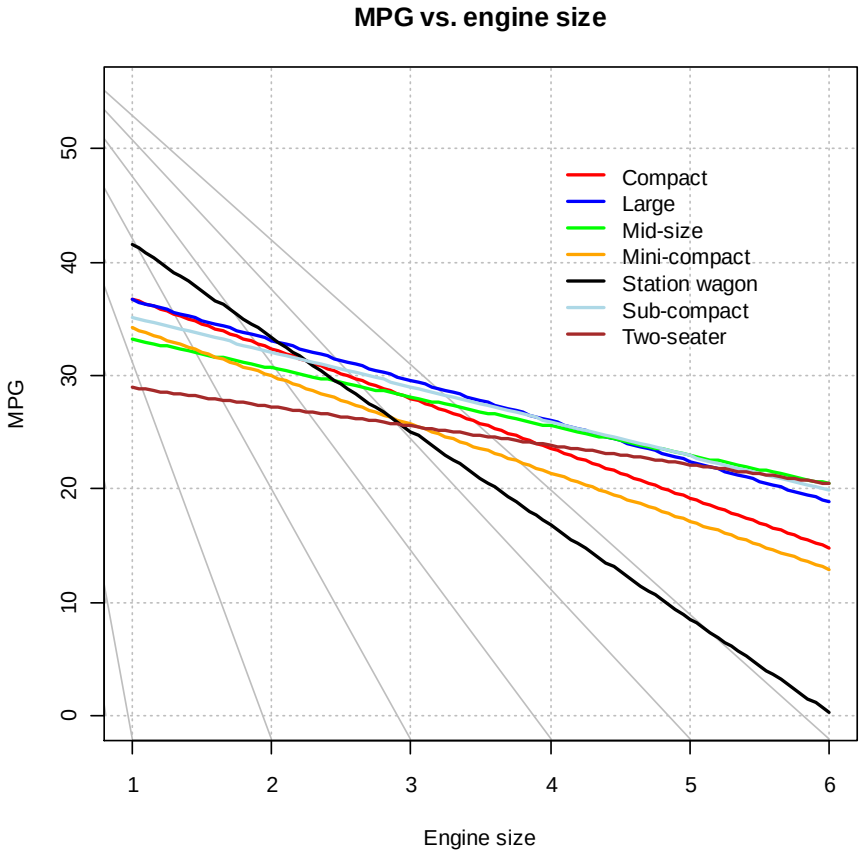
R code:

```
#Sample model plot
win.graph(width = 6, height = 6, pointsize = 10)
#Compact
curve(expr = mod.fit3$coefficients[1] +
            mod.fit3$coefficients[2]*x, col = "red", lty =
      "solid", lwd = 2, xlim = c(1,6), ylim = c(0,55), xlab =
      "Engine size", ylab = "MPG", main = "MPG vs. engine
      size", panel.first = grid(col = "gray", lty = "dotted"))
#Large
curve(expr = mod.fit3$coefficients[1] +
            mod.fit3$coefficients[2]*x +
            mod.fit3$coefficients[3] +
            mod.fit3$coefficients[9]*x, col = "blue", lty =
      "solid", lwd = 2, add = TRUE, from = 1, to = 6)
```

Code excluded

```
#Two-seater
curve(expr = mod.fit3$coefficients[1] +
           mod.fit3$coefficients[2]*x +
           mod.fit3$coefficients[8] +
           mod.fit3$coefficients[14]*x,
    col = "brown", lty = "solid", lwd = 2, add = TRUE, from =
    1, to = 6)
legend(locator(1), legend = c("Compact", "Large", "Mid-size",
     "Mini-compact", "Station wagon", "Sub-compact", "Two-
     seater"), col = c("red", "blue", "green", "orange",
     "black", "lightblue", "brown"), lty = rep(x = "solid",
     times = 7), bty = "n", cex = 1, lwd = 2)
```

EASIER WAY: Use *predict()* with *curve()*. How can the two be used together?



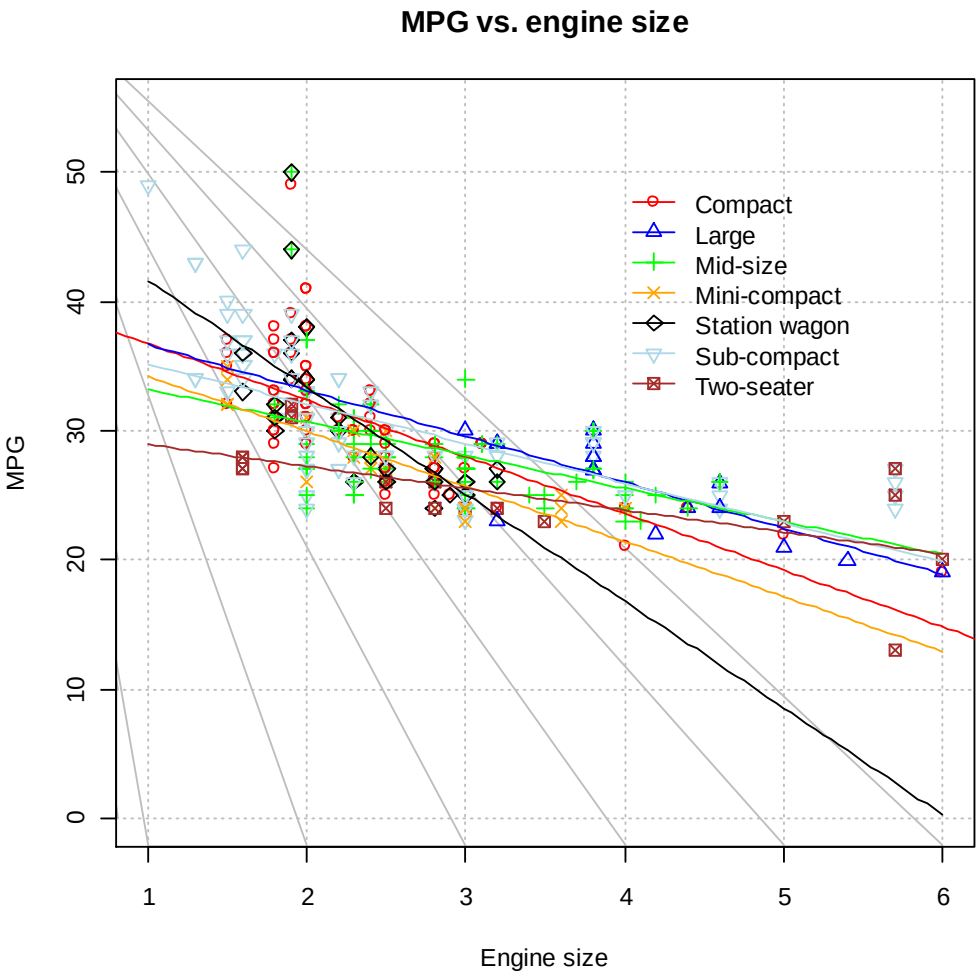
Questions:

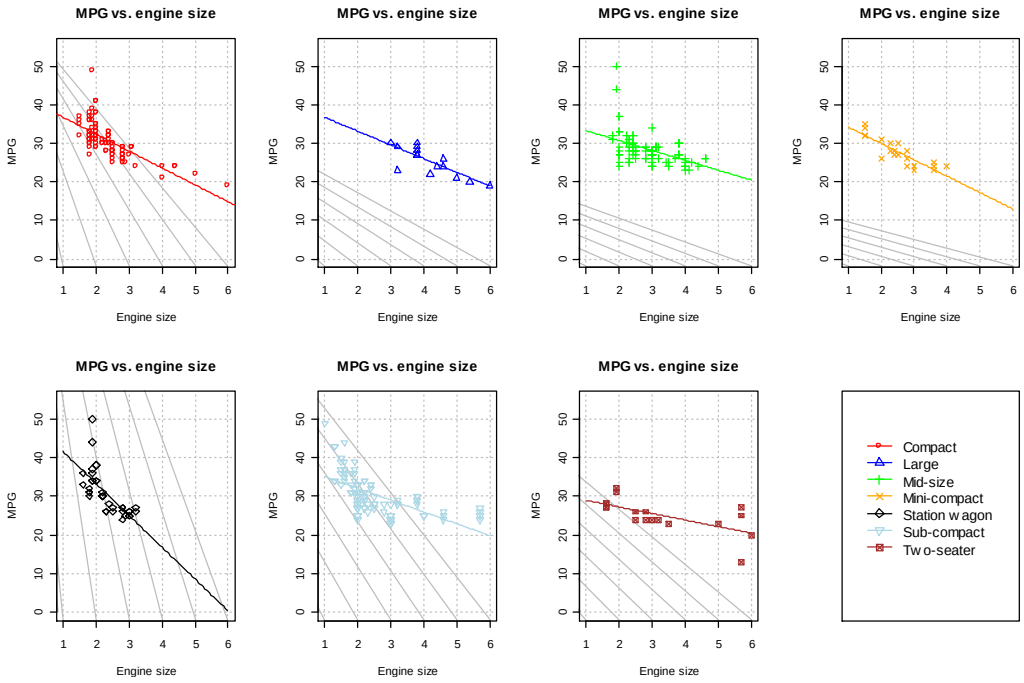
1) Why are the lines not parallel?
2) Is there extrapolation beyond the range of the engine sizes?
3) What would you expect the plot to look like if there was not a significant interaction and class effect?

Commented [b8]: Shows example of multiple reg. problem of extrapolating beyond range of data - not all classes have engines of size 1 to 6 - see observed data plot

Next are two more plots to help you understand this model and the observed data. For the code, please see

the program (make sure to see the use of the *recode()* function).





Other notes about qualitative variables:

1) Residual vs. indicator variable plots are not necessary since the indicator variable has only 2 levels.
2) When both qualitative and quantitative predictor variables are present, the regression models are the same as analysis of covariance (ANCOVA) models.

8.7 Comparison of two or more regression functions

Please read on your own.