HW2 Solution

Rick Farouni

September 20, 2016

Question 1

Step 1: Load the data

```
Q1_df <- read.table("~/Documents/repos/PSYCH7821/assets/assignments/HW2/ex02_question1_data.txt",
                    header = TRUE)
```

Step 2: Inspect the data

```
head(Q1_df)
```

```
##   id sex sctyp civics motiv locus concept read write math science
## 1  1   2     1   40.6  0.67  0.29    0.88 33.6  43.7 40.2    39.0
## 2  2   1     1   45.6  0.33 -0.42    0.03 46.9  35.9 41.9    36.3
## 3  6   1     1   35.6  0.00  0.46    0.03 49.5  46.3 46.2    41.7
## 4  7   1     1   55.6  0.33  0.44   -0.47 62.7  64.5 48.0    63.4
## 5  8   2     1   55.6  1.00  0.68    0.25 44.2  51.5 36.9    49.8
## 6  9   1     1   55.6  0.33  0.06    0.56 46.9  41.1 45.3    47.1
```

We see that the variables *sex* and *sctyp* are categorical variables; *civics* and *motiv* need more inspection. Let's compute the contingency table of the counts:

```
table(Q1_df$motiv)
```

```
##
##    0 0.33 0.67    1
##   41   65   84  127
```

```
table(Q1_df$civics)
```

```
##
## 25.7 30.6 33.1 35.6 36.9 39.4 40.6 41.9 43.1 45.6 46.8 48.1 49.3 50.6 51.8
##    1    9    1   18    2    1   24    2    1   50    2    2    1   53    2
## 53.1 55.6 56.8 58.1 59.3 60.5 61.8 65.5 66.8 70.5
##    1   46    5    1    1   53    1   30    2    8
```

Now we can see that *motiv* is a an ordinal variable. More specifically, it is a polytomous variable with four levels. The variable *civics* on the other hand seems to have a mixture distribution. This leads me to think that the data has been previoulsy preprocessed in a way that might affect the analysis.

NOTE 1: The optimal coefficient β the regression fit gives us back depends on the distributions of the predictor variables!

Since we have standardized the data, let's compute the correlation matrix. Note that I am leaving out *civics* and *motiv* since correlation is a measure of linear assoication and these two variables are not really continuous.

```
cor(Q1_df[ , 6:11])
```

```
##             locus    concept      read      write      math   science
## locus   1.0000000 0.10943434 0.3752034 0.35061385 0.3618460 0.3365816
## concept 0.1094343 1.00000000 0.1112732 0.05590149 0.1338509 0.1061843
## read    0.3752034 0.11127317 1.0000000 0.63420287 0.6628906 0.6621116
## write   0.3506139 0.05590149 0.6342029 1.00000000 0.6391226 0.5480410
## math    0.3618460 0.13385089 0.6628906 0.63912259 1.0000000 0.6514850
## science 0.3365816 0.10618426 0.6621116 0.54804100 0.6514850 1.0000000
```

If we plan to fit a regression model, when need to worry about collinearity. The four test ability variables seem correlated among each other, so we need to deal with collinearity. One solution is to obtain a second dataset with uncorrelated predictor variables. A second approach is regularization and shrinkage methods, which are commonly used for high-dimensional datasets (n<<p). A third approach is to identify the correlated variables and just consider a single one to include in the model. Since this approach is not so straightforward, principal components analysis can be used instead to reduce the correlated variables into one or two independent variables.

Step 3: Plot the data

This the most important step in data analysis. If you do not plot your data first, you are just stumbling in the dark. The data we have is multidimensional, so a pairs plot is a good way to try to see any patterns in the data. You can use the *pairs* function to plot the data, but I am writing my own function so I can use web graphics instead.

```
library(rbokeh)
plotPairs <- function(df, colms, colr){

nms <- expand.grid(names(df)[colms],
                   rev(names(df)[colms]),
                   stringsAsFactors = FALSE)
splom_list <- vector("list", length(nms))
for (ii in seq_len(nrow(nms))) {
  splom_list[[ii]] <- figure(width = 110,
                             height = 110,
                             tools = c("pan", "box_zoom", "reset"),
                             xlab = nms$Var1[ii],
                             ylab = nms$Var2[ii]) %>%
                      ly_points(nms$Var1[ii],
                                nms$Var2[ii],
                                data = df,
                                color = as.factor(df[ , colr]),
                                size = 5,
                                legend = FALSE)
}
```

```
p <- grid_plot(splom_list,
          ncol = length(colms),
          same_axes = TRUE,
          link_data = TRUE)

return(p)
}
```

Let's plot columns 4 to 11 and color the observations by the sex (Variable 2)

```
plotPairs(Q1_df, 4:11, 2)
```

Not all the variables are on the same scale. That is problematic if we are running PCA. Note: *read*, *write*, *math*, and *science* seem to be measured on the same scale so there is no need to standardize the variable for PCA. That said, since we are analyzing the entire dataset which contain variables at differnt scales, it makes more sense to standardize the entire dataset (except of course the categorical variable).

Next we standardize and plot the data.

```
Q1_df[ , 4:11] <- scale(Q1_df[ , 4:11],
                               center = TRUE,
                               scale = TRUE)
plotPairs(Q1_df, 4:11, 2)
```

Step 4: Fit model

Dimensionality Reduction

Let's reduce the 4 dimensional data into fewer dimensions. You can fit the model with a single call to the function prcomp like this

```
pca_fit <- prcomp(~read + write + math + science,
                  data = Q1_df,
                  center = TRUE,
                  scale = FALSE)
```

Scale is FALSE since the data has been standardized already. The proportion of variance explained can be obtained using the summary function

```
summary(pca_fit)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.7030 0.6736 0.58044 0.55596
## Proportion of Variance 0.7251 0.1134 0.08423 0.07727
## Cumulative Proportion  0.7251 0.8385 0.92273 1.00000
```

Question 1A

Now pca_fit is a list with three objects (rotation, sdev, X). You can access the objects with the $ operator, so for example, Q is the loading matrix (eigenvectors)

```
Q <- pca_fit$rotation
Q
```

```
##               PC1          PC2         PC3        PC4
## read    0.5115308 -0.089313051  0.64949737 -0.5554391
## write   0.4845572  0.754212731  0.09266702  0.4333362
## math    0.5104171  0.001790006 -0.75452230 -0.4125134
## science 0.4929654 -0.650525273  0.01618916  0.5775291
```

Z is the matrix of sample PC scores. sigma is the vector of standard deviations

```
X <- scale(Q1_df[ , 8:11],
           center = TRUE,
           scale = FALSE)
Z <- X %*% Q
# or equivalently
Z <-  pca_fit$x
# standardized
sigma <- pca_fit$sdev
Zs <- Z %*% diag(1/sigma)
```

The first standardized first principle component can be obtained by subsetting

```
z1s <- Zs[ , 1]
```

Question 1B

Lets add PC1 to the dataset

```
Q1_df <- cbind(Q1_df,z1s)
```

plot the data

```
plotPairs(Q1_df, c(4:7,12), 2)
```

The predictor variables don't seem to be correlated.

Now we can fit the model

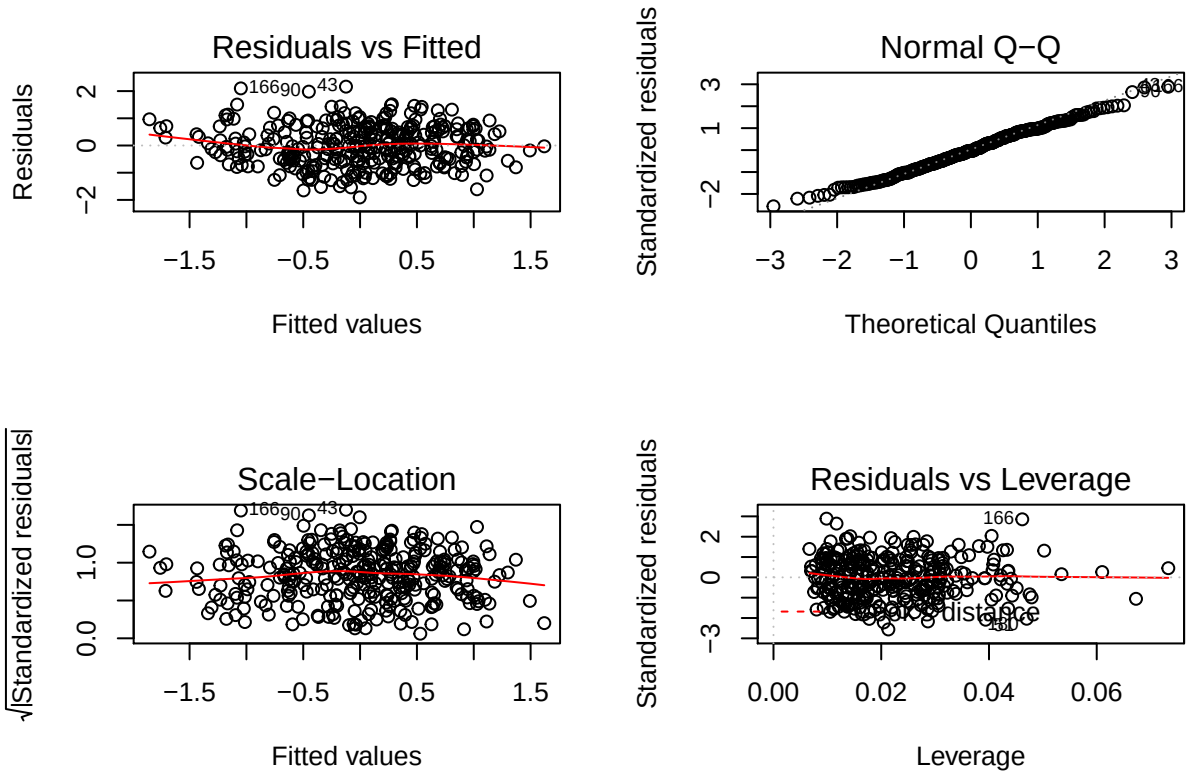
```
fit1 <- lm(z1s ~ sex + sctyp + civics + motiv + locus + concept,
           data = Q1_df)
summary(fit1)
```

```
##
## Call:
## lm(formula = z1s ~ sex + sctyp + civics + motiv + locus + concept,
```

```
##     data = Q1_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.90895 -0.54151 -0.02537  0.58217  2.16337
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.03870    0.19587  -0.198    0.843
## sex         -0.05733    0.08762  -0.654    0.513
## sctyp        0.10971    0.11377   0.964    0.336
## civics       0.52172    0.04422  11.798  < 2e-16 ***
## motiv        0.05322    0.04587   1.160    0.247
## locus        0.26479    0.04493   5.894 9.84e-09 ***
## concept      0.04411    0.04441   0.993    0.321
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.753 on 310 degrees of freedom
## Multiple R-squared:  0.4437, Adjusted R-squared:  0.433
## F-statistic: 41.21 on 6 and 310 DF,  p-value: < 2.2e-16
```

Step 5: Dignostics

```
# plot diagnostics
par(mfrow = c(2,2))
plot(fit1)
```



Models assumptions seem to hold for the most part. More about how to interpret these plots here.

Step 6: Model Selection

To find which of these 6 variables are the most important and statistically significant predictors, we can perform stepwise regression using the AIC.

> NOTE 2: The step function in R takes proper account of the number of parameters fit by dropping or adding variables (sometimes in a group) that minimizes the AIC score. If the package you are using does selection on F-statistics instead, by adding "significant" terms and dropping "non-significant" terms, then multiple comparison issues are not properly dealt with by your package (Friedman, J., Hastie, T., & Tibshirani, R., 2001) and I would recommend switching to another package. A better option would be to take a totally different approach and use ridge regression.

```
stepFit <- step(fit1)
```

```
## Start:  AIC=-172.93
## z1s ~ sex + sctyp + civics + motiv + locus + concept
##
##           Df Sum of Sq    RSS      AIC
## - sex      1     0.243 176.02 -174.488
## - sctyp    1     0.527 176.31 -173.976
## - concept  1     0.559 176.34 -173.918
## - motiv    1     0.763 176.54 -173.552
## <none>                 175.78 -172.925
## - locus    1    19.697 195.48 -141.257
## - civics   1    78.932 254.71  -57.349
##
## Step:  AIC=-174.49
## z1s ~ sctyp + civics + motiv + locus + concept
##
##           Df Sum of Sq    RSS      AIC
## - sctyp    1     0.533 176.56 -175.529
## - motiv    1     0.636 176.66 -175.345
## - concept  1     0.693 176.72 -175.243
## <none>                 176.02 -174.488
## - locus    1    19.547 195.57 -143.107
## - civics   1    78.693 254.72  -59.345
##
## Step:  AIC=-175.53
## z1s ~ civics + motiv + locus + concept
##
##           Df Sum of Sq    RSS      AIC
## - motiv    1     0.676 177.23 -176.317
## - concept  1     0.677 177.23 -176.316
## <none>                 176.56 -175.529
## - locus    1    19.871 196.43 -143.721
## - civics   1    79.382 255.94  -59.828
##
## Step:  AIC=-176.32
## z1s ~ civics + locus + concept
```

```
##
##           Df Sum of Sq    RSS      AIC
## - concept  1     1.098 178.33 -176.360
## <none>                 177.23 -176.317
## - locus    1    21.960 199.19 -141.289
## - civics   1    81.696 258.93  -58.146
##
## Step:  AIC=-176.36
## z1s ~ civics + locus
##
##          Df Sum of Sq    RSS      AIC
## <none>                178.33 -176.360
## - locus   1    23.151 201.48 -139.667
## - civics  1    82.369 260.70  -57.985
```

The final model is

```
summary(stepFit)
```

```
##
## Call:
## lm(formula = z1s ~ civics + locus, data = Q1_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0128 -0.5817 -0.0034  0.5573  2.1460
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.692e-17  4.233e-02   0.000        1
## civics       5.289e-01  4.391e-02  12.043  < 2e-16 ***
## locus        2.804e-01  4.391e-02   6.385 6.17e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7536 on 314 degrees of freedom
## Multiple R-squared:  0.4357, Adjusted R-squared:  0.4321
## F-statistic: 121.2 on 2 and 314 DF,  p-value: < 2.2e-16
```

Question 2

Step 1: Load the data

```
Q2_df <- read.table("~/Documents/repos/PSYCH7821/assets/assignments/HW2/ex02_question2_data.txt",
                    header = TRUE)
```

Step 2: Inspect the data

```
head(Q2_df)
```

```
##   id gpa spatial2d spatial3d mech_res arith computat
## 1  6  51        25        23       11    21       14
## 2  9  40         9         9       15     7        8
## 3 21  36         3         5        8     6       11
## 4 24  53        18        26        7    13       10
## 5 25  40        14        11        7    11       14
## 6 42  59        20         5       12    24       12
```

```
summary(Q2_df)
```

```
##        id              gpa          spatial2d       spatial3d
##  Min.   :  6.0   Min.   :21.00   Min.   : 3.00   Min.   : 2.00
##  1st Qu.:101.0   1st Qu.:36.00   1st Qu.:10.50   1st Qu.: 8.00
##  Median :240.0   Median :43.00   Median :16.00   Median :12.00
##  Mean   :221.4   Mean   :42.38   Mean   :16.53   Mean   :13.28
##  3rd Qu.:325.0   3rd Qu.:49.00   3rd Qu.:24.00   3rd Qu.:17.50
##  Max.   :425.0   Max.   :68.00   Max.   :30.00   Max.   :29.00
##     mech_res         arith          computat
##  Min.   : 2.00   Min.   : 6.00   Min.   : 5.00
##  1st Qu.: 8.00   1st Qu.:16.00   1st Qu.:11.00
##  Median :10.00   Median :21.00   Median :12.00
##  Mean   :10.03   Mean   :20.09   Mean   :12.91
##  3rd Qu.:12.00   3rd Qu.:24.00   3rd Qu.:14.00
##  Max.   :20.00   Max.   :35.00   Max.   :25.00
```

Except for *gpa*, the other variables seem to gave been measured on the same reference-normed scale. Accordingly, there doesn't seem to be a need for standardization. For PCA we also keep the optional scaling off (the default).

Step 3: Plot the data

```
plotPairs(Q2_df, 2:7, 1)
```

The 6 variables all seem to be highly correlated.

Step 4: fit the data

Question 2a

```
fit2 <- lm(gpa ~ spatial2d + spatial3d + mech_res + arith + computat,
           data = Q2_df)
summary(fit2)
```

```
##
## Call:
## lm(formula = gpa ~ spatial2d + spatial3d + mech_res + arith +
##     computat, data = Q2_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.4046  -5.3191   0.8052   4.2488  18.6125
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 21.37285    3.90386   5.475 5.89e-07 ***
## spatial2d    0.11915    0.12831   0.929  0.35612
## spatial3d    0.08029    0.16004   0.502  0.61740
## mech_res     0.06643    0.26552   0.250  0.80314
## arith        0.31333    0.18503   1.693  0.09464 .
## computat     0.85278    0.30327   2.812  0.00632 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.749 on 73 degrees of freedom
## Multiple R-squared:  0.346,  Adjusted R-squared:  0.3012
## F-statistic: 7.723 on 5 and 73 DF,  p-value: 7.116e-06
```

> NOTE 3: We see that only *computat* shows statistical significance, but that is expected given that the variables are correlated. Even if we have a situation where each of two highly correlated predictor variables perfectly predicts the reponse, you will see that only one of the two variables will have a highly significant p-value while the other won't.

We need to deal with multicollinearity first. PCA is an option.

Question 2b (Dimensionality Reduction)

First we run PCA on the 5 variables

```
pca_fit2 <- prcomp(~spatial2d + spatial3d + mech_res + arith + computat,
                   data = Q2_df,
                   center = TRUE,
                   scale = FALSE)
```

The proportion of variance explained can be obtained using the summary function

```
summary(pca_fit2)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5
## Standard deviation     10.1111 6.2720 4.9561 3.27497 2.66315
## Proportion of Variance  0.5558 0.2139 0.1335 0.05831 0.03856
## Cumulative Proportion   0.5558 0.7696 0.9031 0.96144 1.00000
```

Now pca_fit is a list with three objects (rotation, sdev, X). You can access the objects with the $ operator, so for example:

Q is the loading matrix (eigenvectors)

```
Q <- pca_fit2$rotation
Q
```

```
##                  PC1         PC2          PC3         PC4         PC5
## spatial2d -0.7086904  0.61291400  0.309177934 -0.12176929  0.10805352
## spatial3d -0.5329105 -0.19911301 -0.809081555 -0.07012241 -0.12973180
## mech_res  -0.1369442  0.03961033 -0.007586042  0.98965660  0.01412676
## arith     -0.4086301 -0.65739245  0.495785960 -0.02081940 -0.39321678
## computat  -0.1674277 -0.38853138  0.062727794 -0.02003615  0.90370078
```

Z is the matrix of sample PC scores.

```
Z <-  pca_fit2$x
```

Question 2c

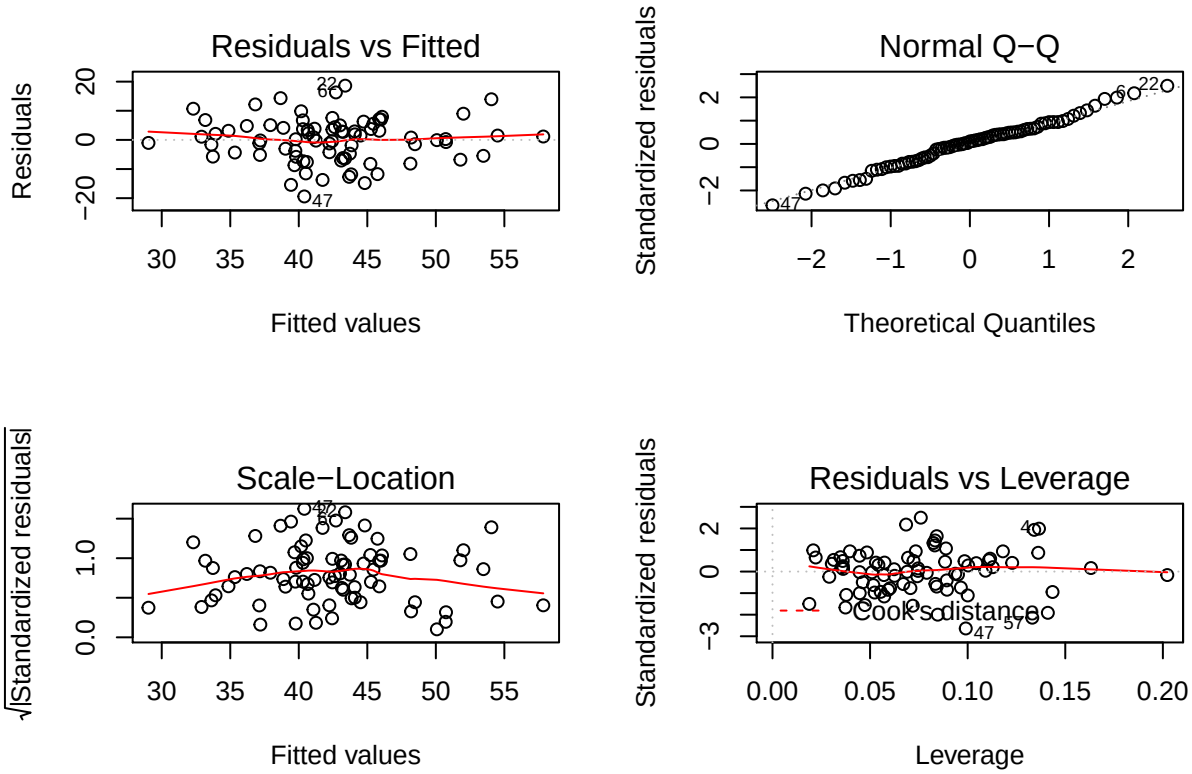
First we add the five principle components to the dataframe. Next, we fit the model with all five PCs

```
Q2_df <- cbind(Q2_df, Z)
fit3 <- lm(gpa ~ PC1 + PC2 + PC3 + PC4 + PC5,
           data = Q2_df)
summary(fit3)
```

```
##
## Call:
## lm(formula = gpa ~ PC1 + PC2 + PC3 + PC4 + PC5, data = Q2_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.4046  -5.3191   0.8052   4.2488  18.6125
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 42.37975    0.87187  48.608  < 2e-16 ***
## PC1         -0.40714    0.08678  -4.692 1.23e-05 ***
## PC2         -0.47764    0.13990  -3.414  0.00105 **
## PC3          0.18021    0.17704   1.018  0.31208
## PC4          0.02200    0.26792   0.082  0.93480
## PC5          0.65085    0.32947   1.975  0.05200 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.749 on 73 degrees of freedom
## Multiple R-squared:  0.346,  Adjusted R-squared:  0.3012
## F-statistic: 7.723 on 5 and 73 DF,  p-value: 7.116e-06
```

Dignostics

```
# plot diagnostics
par(mfrow = c(2, 2))
plot(fit3)
```



Nothing alarming! We move on to model selection

```
stepFit3 <- step(fit3)
```

```
## Start:  AIC=329.28
## gpa ~ PC1 + PC2 + PC3 + PC4 + PC5
##
##        Df Sum of Sq    RSS    AIC
## - PC4   1      0.40 4384.2 327.29
## - PC3   1     62.22 4446.0 328.39
## <none>              4383.8 329.28
## - PC5   1    234.34 4618.1 331.40
## - PC2   1    700.00 5083.8 338.98
## - PC1   1   1321.86 5705.6 348.10
##
## Step:  AIC=327.29
## gpa ~ PC1 + PC2 + PC3 + PC5
##
##        Df Sum of Sq    RSS    AIC
## - PC3   1     62.22 4446.4 326.40
## <none>              4384.2 327.29
## - PC5   1    234.34 4618.5 329.40
## - PC2   1    700.00 5084.2 336.99
## - PC1   1   1321.86 5706.1 346.11
##
## Step:  AIC=326.4
## gpa ~ PC1 + PC2 + PC5
##
##        Df Sum of Sq    RSS    AIC
```

```
## <none>              4446.4 326.40
## - PC5   1    234.34 4680.7 328.46
## - PC2   1    700.00 5146.4 335.95
## - PC1   1   1321.86 5768.3 344.96
```

The final model is

```
summary(stepFit3)
```

```
##
## Call:
## lm(formula = gpa ~ PC1 + PC2 + PC5, data = Q2_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.5761  -4.8354   0.6633   4.3220  18.0740
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 42.37975    0.86628  48.921  < 2e-16 ***
## PC1         -0.40714    0.08622  -4.722 1.06e-05 ***
## PC2         -0.47764    0.13900  -3.436 0.000965 ***
## PC5          0.65085    0.32736   1.988 0.050446 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.7 on 75 degrees of freedom
## Multiple R-squared:  0.3366, Adjusted R-squared:  0.3101
## F-statistic: 12.69 on 3 and 75 DF,  p-value: 8.596e-07
```

We see that the fifth principle component was included in the model. What to do? For a more interpertable model, just keep the first two.

> NOTE 4: Note that I haven't mentioned R-squared here at all. The reason is people have begining to realize that R-squared is a useless criterion for model fit.

The reason for that are as follows:

1. R-squared does not measure goodness of fit.
2. R-squared does not measure predictive error.
3. R-squared does not allow you to compare models using transformed responses.
4. R-squared does not measure how one variable explains another.

These reasons are explained really well here.

==========