HUDM6122 Homework_03

Chenguang Pan

2023-02-23

0.1 Github Address

All my up-to-date homework can be found on Github: https://github.com/cgpan/hudm6122_homeworks . Thanks for checking if interested.

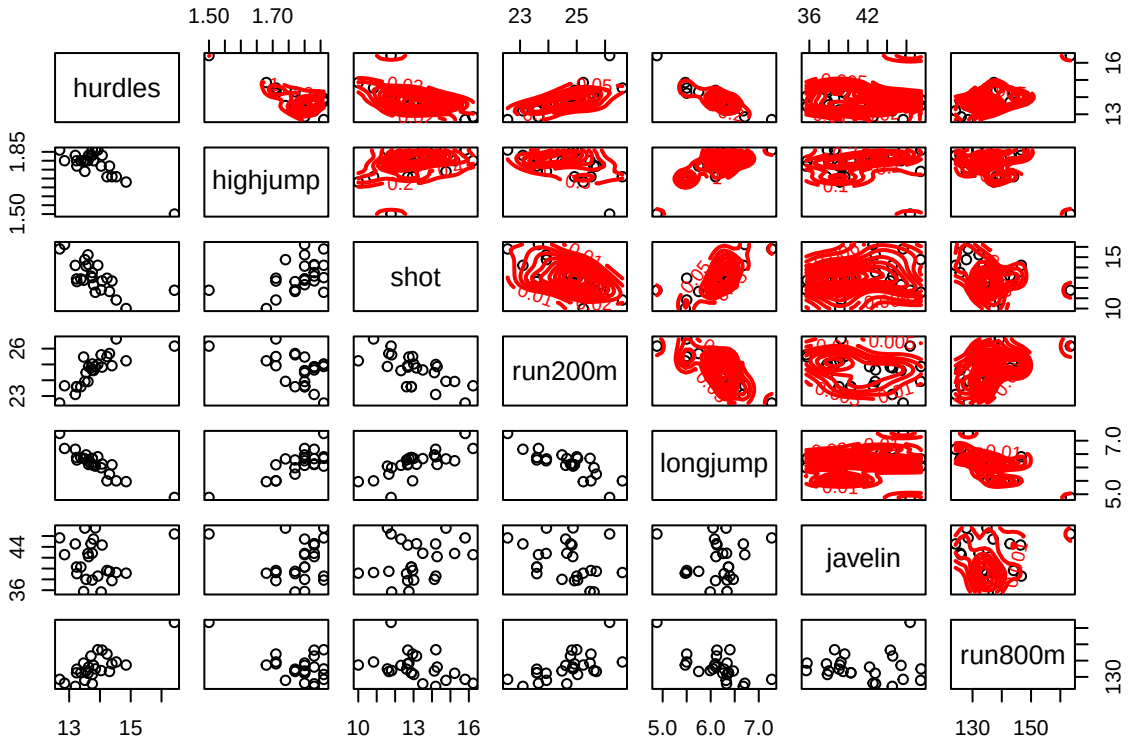
0.2 Ex 3.1

Construct the scatterplot of the heptathlon data showing the contours of the estimated bivariate density function on each panel. Is this graphic more useful than the unenhanced scatterplot matrix?

MY SOLUTION:

Here, I use the `MASS::kde2d()` function to estimate the bivariate density of the data, and plot the contours of the density using the `contour()` function.

```
> # import the package
> library(MVA)
> library(HSAUR2)
> data(heptathlon)
> # Create a scatterplot matrix with density contours
> pairs(heptathlon[, -ncol(heptathlon)], upper.panel = function(x, y) {
+   points(x, y)
+   den <- MASS::kde2d(x, y)
+   contour(den, add = TRUE, col = "red", lwd = 2)})
```



Comparing to the unenhanced scatter plot matrix, this mixed graph can help to easily find the specific characteristics of joint distribution of each pair, like the center of the distribution.

0.3 Ex 3.2

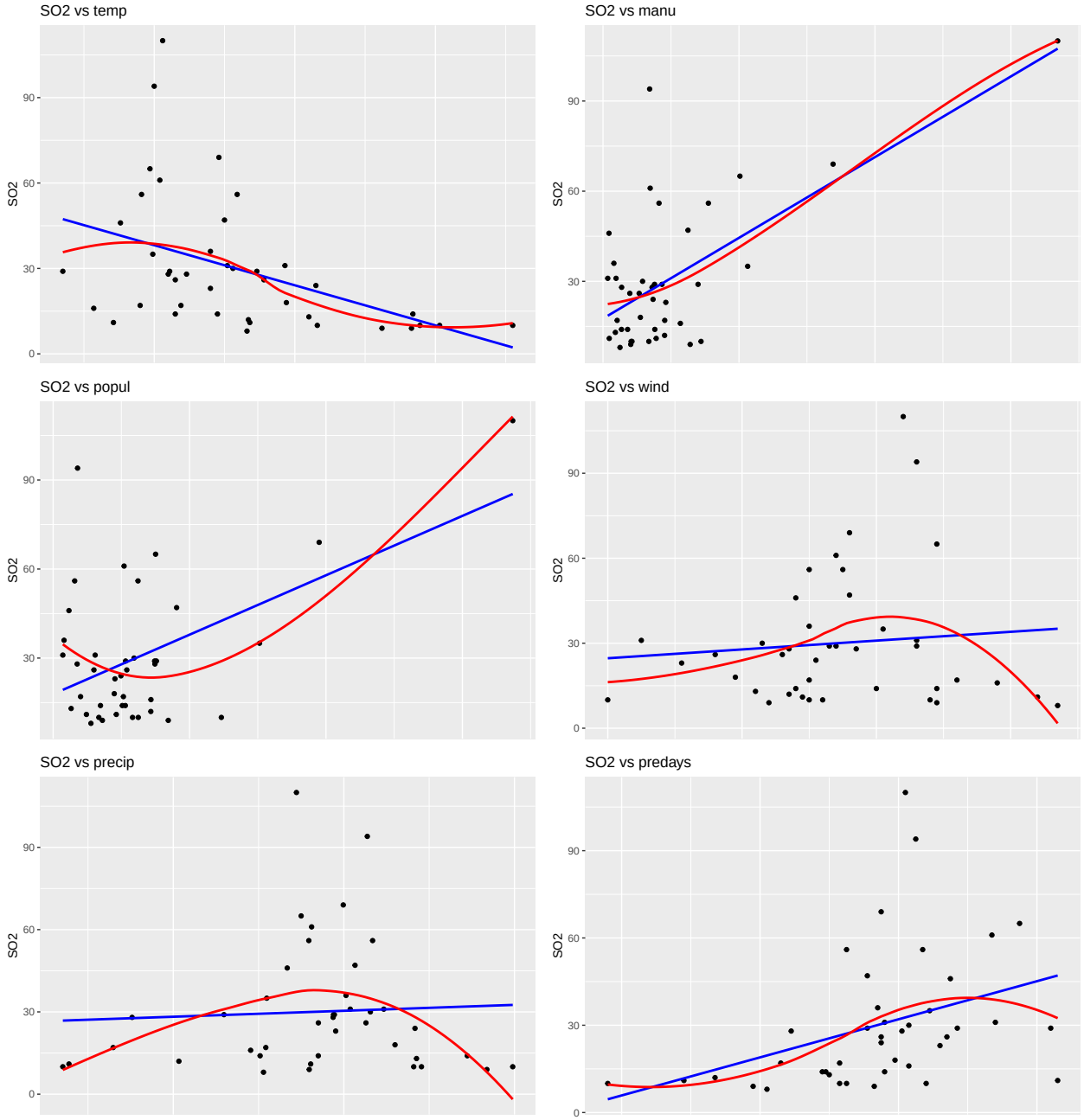
Construct a diagram that shows the SO2 variable in the air pollution data plotted against each of the six explanatory variables, and in each of the scatterplots show the fitted linear regression and a fitted locally weighted regression. Does this diagram help in deciding on the most appropriate model for determining the variables most predictive of sulphur dioxide levels?

MY SOLUTION:

To solve this questions, I used the `ggplot2` to draw each graph.

```
> # try to use a for-loop to get all the maps in fewer lines
> par(mfrow=c(2,3))
> for (i in c(2:7)) {
+   p <- ggplot(USairpollution, aes(x = USairpollution[,i], y = SO2)) +
+                 geom_point() +
+                 geom_smooth(method = "lm", se = FALSE, color = "blue") +
+                 geom_smooth(span = 1, se = FALSE, color = "red") +
+                 labs(title = paste0("SO2 vs ", colnames(USairpollution)[i]))+
+                 # to remove the un-elegant x-axis name
+                 theme(axis.title.x = element_blank(),
+                       axis.text.x = element_blank(),
+                       axis.ticks.x = element_blank())
```

```
+    # assign(var_name, p)
+    print(p)
+ }
```



From six graphs above, we can not easily tell what the strongest predictor is for predicting the SO_2 concentration, since there are some outliers with high leverage in each graph.

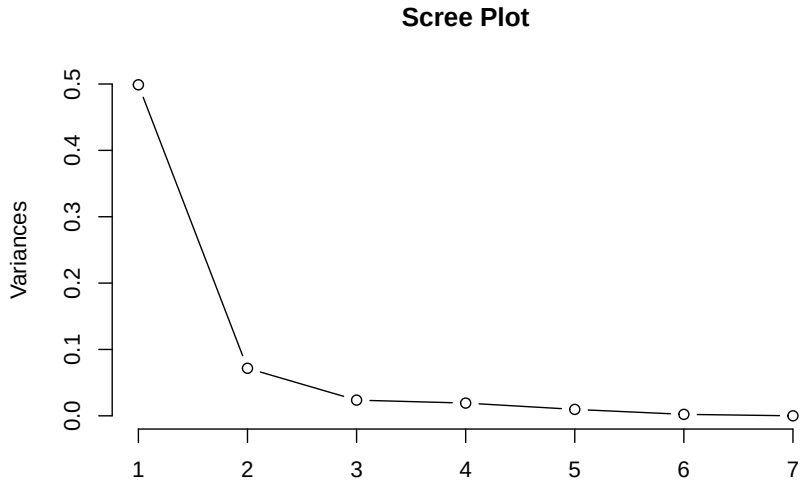
0.4 Ex 3.3

Find the principal components of the following correlation matrix given by MacDonnell (1902) from measurements of seven physical char- acteristics in each of 3000 convicted criminals: How would you interpret the derived components?

MY SOLUTION

First, by using `forceSymmetric()` function in the package `Matrix` to transform the low triangular correlation matrix into a complete correlation matrix. And then, I use `prcomp()` to get the components.

```
> library(Matrix)
> # import the correlation matrix
> corr_lower <- matrix(c(1, 0, 0, 0, 0, 0, 0,
+                        0.402, 1, 0,0,0,0,0,
+                        0.396, 0.618, 1,0,0,0,0,
+                        0.301, 0.150, 0.321, 1,0,0,0,
+                        0.305, 0.135, 0.289, 0.846, 1,0,0,
+                        0.339, 0.206, 0.363, 0.759, 0.797, 1,0,
+                        0.340, 0.183, 0.345, 0.661, 0.800, 0.736, 1),7,7, byrow = T)
> # generate a complete correlation matrix
> corr_symmetric <- forceSymmetric(corr_lower, uplo="L")
> # use prcomp to calculate the principal components
> pca <- prcomp(corr_symmetric, scale. = FALSE)
> # get the PCA results
> summary(pca)
Importance of components:
                          PC1    PC2     PC3     PC4     PC5     PC6       PC7
Standard deviation     0.7063 0.2677 0.15376 0.13851 0.09832 0.04686 3.458e-17
Proportion of Variance 0.7979 0.1146 0.03781 0.03069 0.01546 0.00351 0.000e+00
Cumulative Proportion  0.7979 0.9125 0.95034 0.98103 0.99649 1.00000 1.000e+00
> # draw the scree plot
> plot(pca, type = "l",
+      main = "Scree Plot")
```



The first two components can explain 91.25% variance of the total. Therefore, I will choose to use the first two components to represent this dataset.

0.5 Ex 3.4

Not all canonical correlations may be statistically significant. An approximate test proposed by Bartlett (1947) can be used to deter- mine how many significant relationships exist. The test statistic for testing that at least one canonical correlation is significant is

MY SOLUTION

This dataset is called `frets` included in the package `boot`. The `l1` and `l2` variables represent the length and the `b1` and `b2` are for the breadth. The index number `1` represents the first son and `2` for the second son in one family. Note, the code for `headsize.std` provided on *Page.97* in the textbook MVA is **wrong**! Each columns divided by the standard deviation cannot be called "standardized"! One should let each column subtract the mean first!!

First, I write a function to calculate the chi-square value. Although the book does not mention, one should notice that the `n`is the number of observations.

```
> cc_test <- function(eigenvalues, n, q1,q2){
+   # n is the number of observations
+   # write a for-loop to load the sum of log eigenvalues
+   sum_log_eigen <- 0
+   for (i in c(1:min(q1,q2))) {
+     sum_log_eigen <- sum_log_eigen + log(1-eigenvalues[i])
+   }
+
+   phi_2 <- -(n - 0.5*(q1+q2+1))*sum_log_eigen
+   p_value <- pchisq(q = phi_2, df = q1*q2, lower.tail = F)
+   return(p_value)
+ }
```

Then write a separate code chunk to calculate the eigenvalues of `headsize` dataset. Based on the dimension of dataset, we can easily find the `q1` $= 2$, `q2` $= 2$.

```
> # import the data
> data("frets", package = "boot")
> headsize <- frets
> # use scale to get the standardized headsize dataset
> headsize_std <- as.data.frame(scale(headsize))
> # get the correlation matrix
> R <- cor(headsize_std)
> # subset the correlation matrix to get the cor matrix for all first son
> r11 <- R[1:2, 1:2]
> r22 <- R[-(1:2), -(1:2)]
> r12 <- R[1:2, -(1:2)]
> r21 <- R[-(1:2), 1:2]
> E1 <- solve(r11)%*%r12%*%solve(r22)%*%r21
> E2 <- solve(r22)%*%r21%*%solve(r11)%*%r12
> # get the eigenvector of two dataset
> e1 <- eigen(E1)$values
> e2 <- eigen(E2)$values
```

Now, plug the values from the analysis above to the initial `cc_test` function.

```
> cc_test(e1, # the eigenvalues are quite identical, here I use e1
+         25, # number of observations
```

```
+         2, # number of q1
+         2) # number of q2
[1] 0.0002060779
```

The p value of a chi-square test at the degree of freedom 4 is far less than than the significant level .05. Therefore, we reject the null hypothesis. That is, at least one of the canonical correlation is significant.

Follow the same method, I first analyzed the dataset `LAdepr` to get the basic information and then plug them into the `cc_test`.

```
> # import the data
> depr <- c(
+            0.212,
+            0.124,  0.098,
+           -0.164,  0.308,  0.044,
+           -0.101, -0.207, -0.106, -0.208,
+           -0.158, -0.183, -0.180, -0.192, 0.492)
> LAdepr <- diag(6) / 2
> LAdepr[upper.tri(LAdepr)] <- depr
> LAdepr <- LAdepr + t(LAdepr)
> rownames(LAdepr) <- colnames(LAdepr) <- c("CESD", "Health", "Gender", "Age", "Edu", "Income")
> # LAdepr <- as.data.frame(LAdepr)
> r11 <- LAdepr[1:2, 1:2]
> r22 <- LAdepr[-(1:2), -(1:2)]
> r12 <- LAdepr[1:2, -(1:2)]
> r21 <- LAdepr[-(1:2), 1:2]
> E1 <- solve(r11)%*%r12%*%solve(r22)%*%r21
> E2 <- solve(r22)%*%r21%*%solve(r11)%*%r12
> e1 <- eigen(E1)$values
> e2 <- eigen(E2)$values
> cc_test(e1,294,2,4)
[1] 1.578454e-11
```

The p value of a chi-square test at the degree of freedom 8 is far less than than the significant level .05. Therefore, we reject the null hypothesis. That is, at least one of the canonical correlation is significant.

0.6 Ex 3.5

Repeat the regression analysis for the air pollution data described in the text after removing whatever cities you think should be regarded as outliers. For the results given in the text and the results from the outliers-removed data, produce scatterplots of sulphur dioxide concentration against each of the principal component scores. Interpret your results.

MY SOLUTION

```
> # import the data
> data("USairpollution")
> head(USairpollution)
           SO2 temp manu popul wind precip predays
Albany      46 47.6   44   116  8.8  33.36     135
Albuquerque 11 56.8   46   244  8.9   7.77      58
Atlanta     24 61.5  368   497  9.1  48.34     115
Baltimore   47 55.0  625   905  9.6  41.31     111
```

```
Buffalo      11 47.1  391   463 12.4  36.11     166
Charleston   31 55.2   35    71  6.5  40.75     148
> # drop the outliers
> out_cities <- c("Chicago", "Phoenix", "Philadelphia")
> df <- USairpollution[!(row.names(USairpollution) %in% out_cities),]
> dim(df)
[1] 38  7
> # to get the correlation matrix
> cor(df[,-1])
               temp        manu       popul        wind      precip     predays
temp     1.00000000 -0.17663356  0.05950795 -0.24910331  0.59678776 -0.33153165
manu    -0.17663356  1.00000000  0.83132399  0.31627175 -0.09682994  0.18024864
popul    0.05950795  0.83132399  1.00000000  0.27804217 -0.02047793  0.02622103
wind    -0.24910331  0.31627175  0.27804217  1.00000000 -0.19721151 -0.02599336
precip   0.59678776 -0.09682994 -0.02047793 -0.19721151  1.00000000  0.38212783
predays -0.33153165  0.18024864  0.02622103 -0.02599336  0.38212783  1.00000000
> # get the PCAs
> usair_pca <- princomp(df[,-1], cor = TRUE)
> # check the PCAs' details
> summary(usair_pca, loadings =T)
Importance of components:
                          Comp.1    Comp.2    Comp.3    Comp.4     Comp.5
Standard deviation     1.4600118 1.2635806 1.1299528 0.8675059 0.37362330
Proportion of Variance 0.3552724 0.2661060 0.2127989 0.1254278 0.02326573
Cumulative Proportion  0.3552724 0.6213784 0.8341773 0.9596050 0.98287073
                           Comp.6
Standard deviation     0.32058638
Proportion of Variance 0.01712927
Cumulative Proportion  1.00000000

Loadings:
        Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6
temp     0.330  0.524  0.467  0.122  0.101  0.611
manu    -0.596  0.286        -0.238 -0.680  0.209
popul   -0.529  0.385  0.232 -0.214  0.626 -0.282
wind    -0.414 -0.114         0.896
precip   0.272  0.653 -0.216  0.283 -0.246 -0.558
predays -0.102  0.235 -0.820         0.272  0.432
```

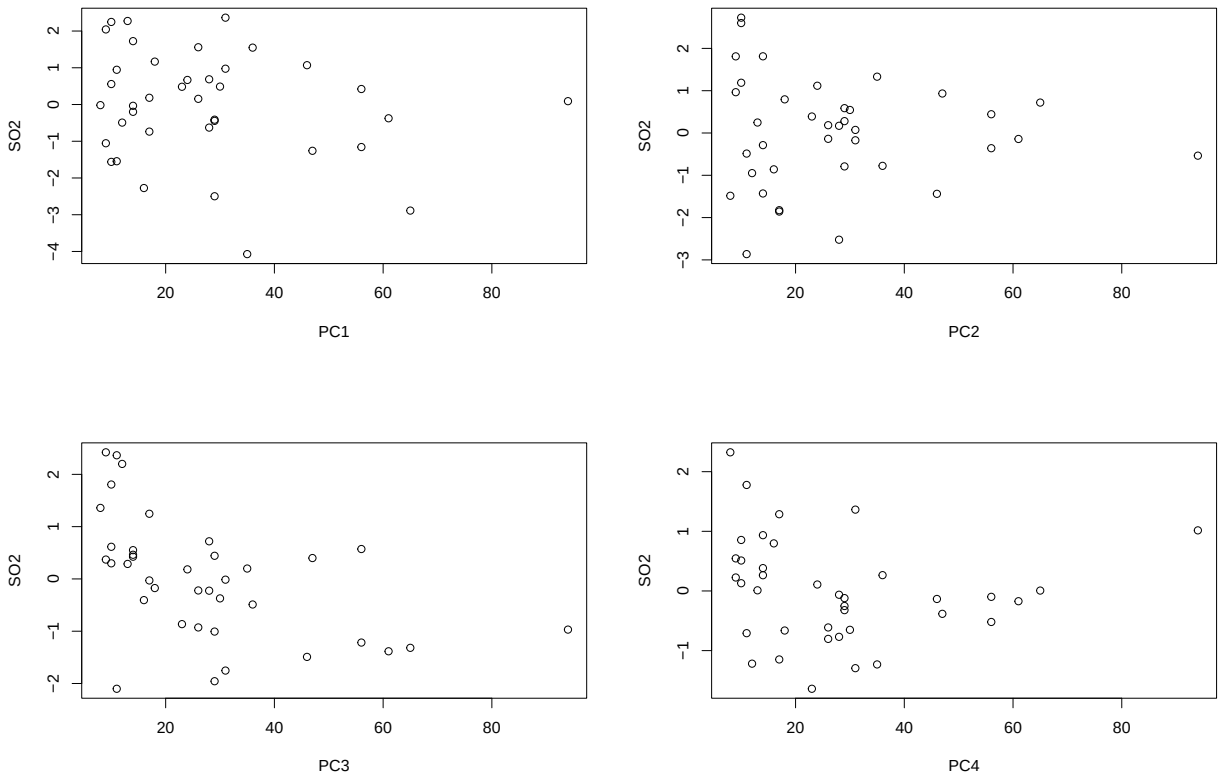
The PCA analysis looks good. Next, I run the regression analysis.

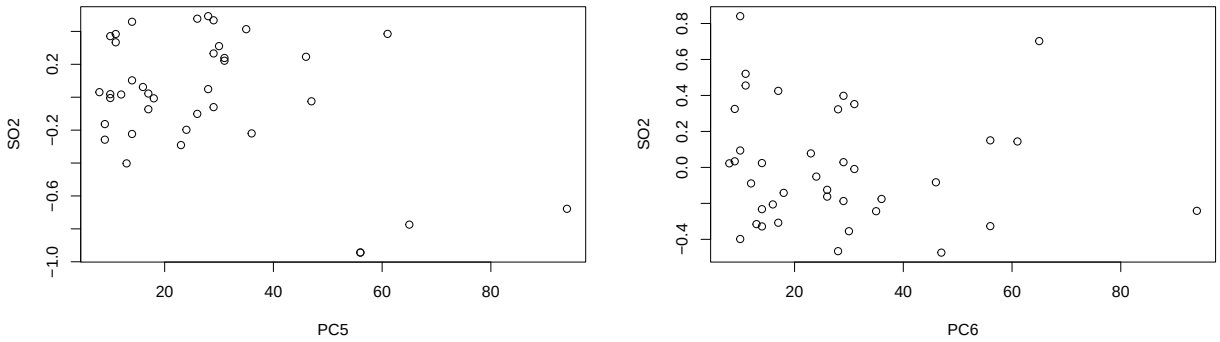
```
> # run the regression function
> usair_reg <- lm(SO2 ~ usair_pca$scores, data = df)
> summary(usair_reg)

Call:
lm(formula = SO2 ~ usair_pca$scores, data = df)

Residuals:
    Min      1Q  Median      3Q     Max
-22.828  -8.998  -2.146   6.729  45.087

Coefficients:
```

```
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)              27.4474     2.3942  11.464 1.11e-12 ***
usair_pca$scoresComp.1   -2.6115     1.6399  -1.592 0.121427
usair_pca$scoresComp.2   -0.8596     1.8948  -0.454 0.653234
usair_pca$scoresComp.3   -8.2642     2.1189  -3.900 0.000482 ***
usair_pca$scoresComp.4   -2.6787     2.7599  -0.971 0.339281
usair_pca$scoresComp.5  -21.6998     6.4082  -3.386 0.001941 **
usair_pca$scoresComp.6   -5.1290     7.4683  -0.687 0.497336
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.76 on 31 degrees of freedom
Multiple R-squared:  0.4987,    Adjusted R-squared:  0.4016
F-statistic: 5.139 on 6 and 31 DF,  p-value: 0.0009116
```

Next, I draw the scatter plots of SO2 against each principle components.

```
> #par(mfrow=c(3,2))
> out <- sapply(1:6, function(i) {
+   plot(df$SO2, usair_pca$scores[,i],
+        xlab = paste("PC",i,sep = ""),
+        ylab = "SO2")
+ })
```





After dropping the outliers, the results from the principle component regression show that the third and the fifth components are significantly associated with the SO2. In addition, the first principle component is no longer the most predictive of SO2. It also indicates that principle component with small variance may have large correlations with the outcome.