

# HUDM6122 Homework\_03

Chenguang Pan

2023-02-19

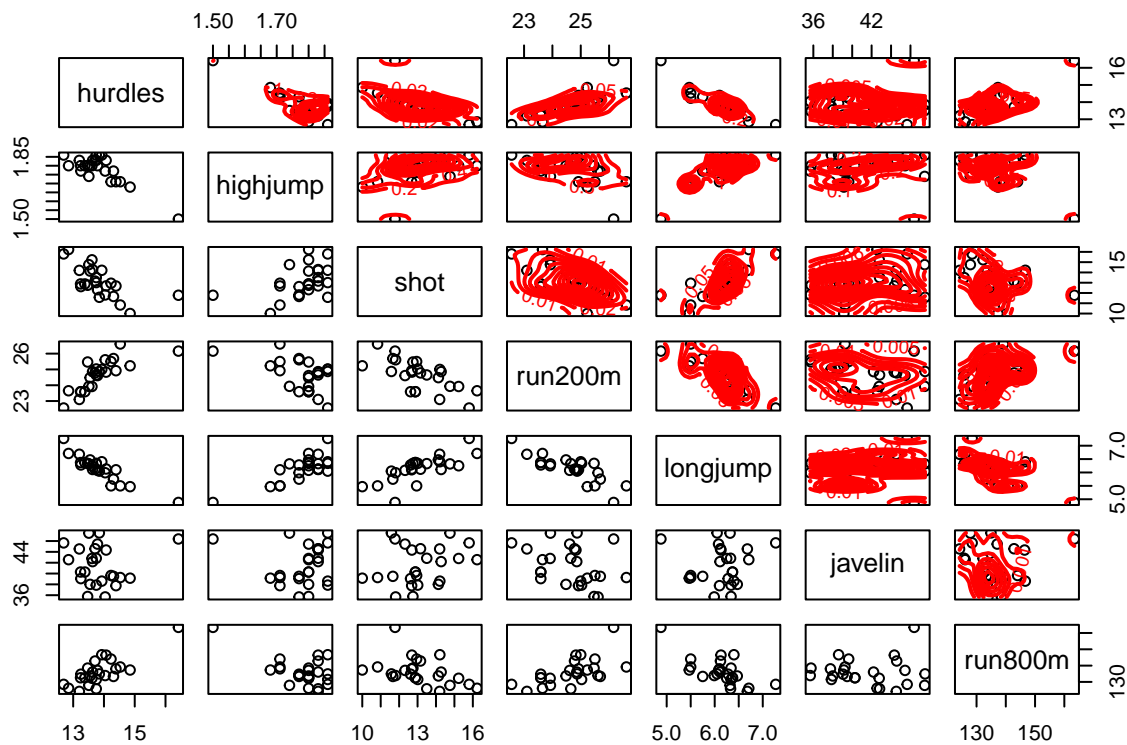
## 0.1 Ex 3.1

*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 where is the center of the distribution.

## 0.2 Ex 3.2

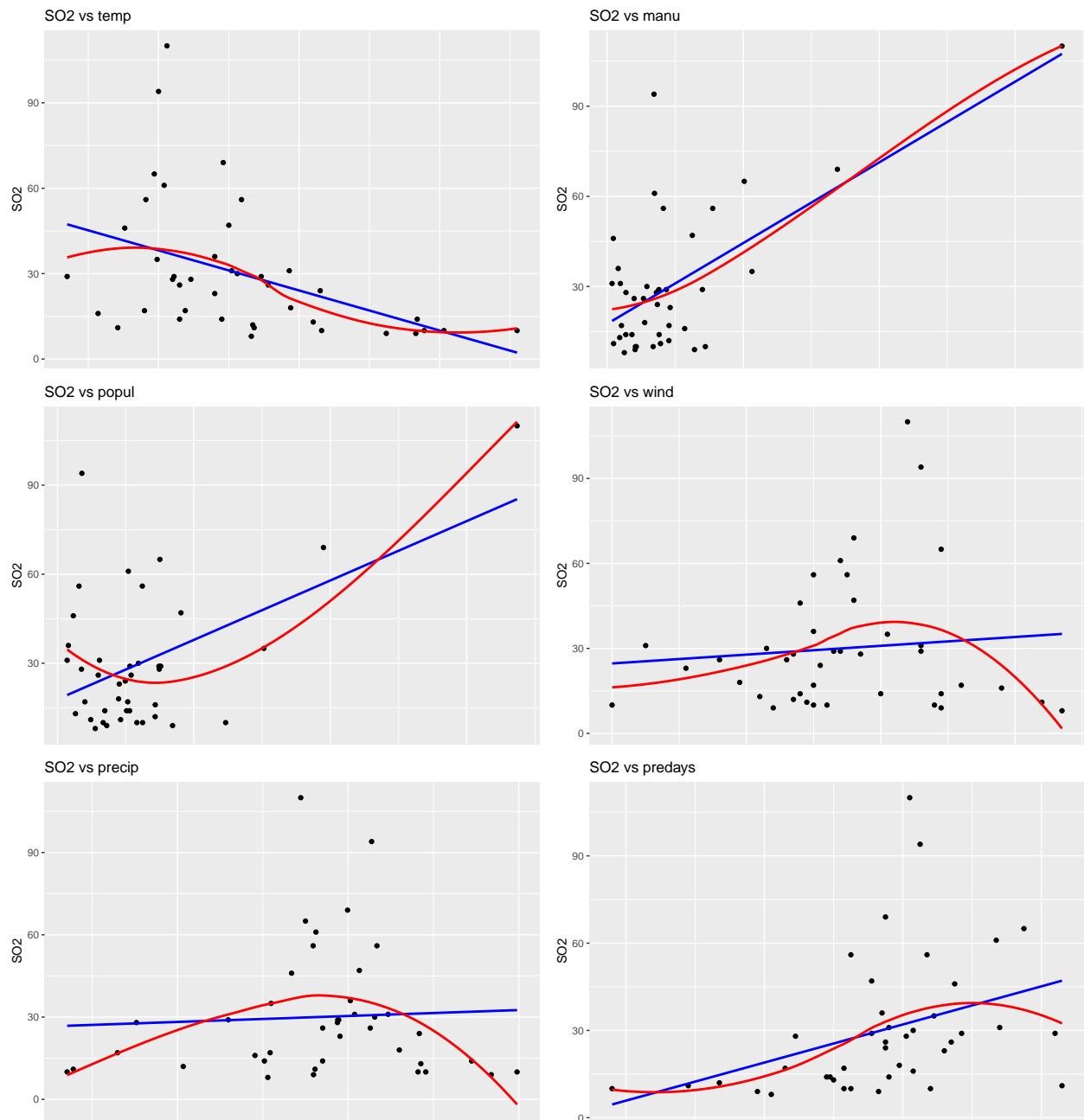
Construct a diagram that shows the  $SO_2$  variable in the air pollution data plotted against each of the six explanatory variables, and in each of the scatterplots show the fitted linear regression and a fitted locally weighted regression. Does this diagram help in deciding on the most appropriate 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.

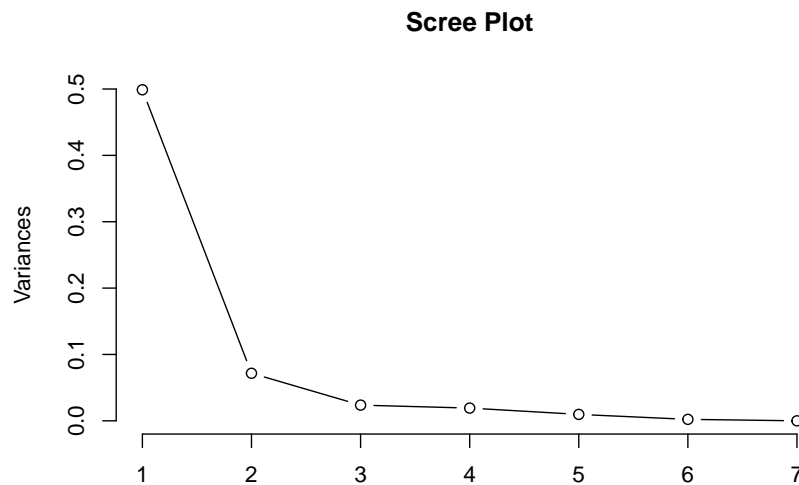
### 0.3 Ex 3.3

Find the principal components of the following correlation matrix given by MacDonnell (1902) from measurements of seven physical characteristics 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.

## 0.4 Ex 3.4

*Not all canonical correlations may be statistically significant. An approximate test proposed by Bartlett (1947) can be used to determine 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 `nis` is the number of observations.

```
> cc_test <- function(eigenvalues, n, q1,q2){
+   # eigenvalues is a list of a eigenvalues computed by hand.
+   # s_df is the df of the s matrix, ie., the n in the equation
+   # p_q is
+   # 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)
+ }
```

Write separate code chunk to calculate the eigenvalues of `headsize` dataset. Based on the dimension of dataset, we can easily find the  $q1 = 2$ ,  $q2 = 2$ , and the `df` of `s` matrix is

```
> # 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]
> # since the eigenvalues are the same for E1 and E2, here I only compute E1
> 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.