MVA_HW1

陈子睿 15220212202842

March 2024

1 The alternative expression of the covariance matrix S

Note that $\mathbf{Y} = (y_1, y_2, \ldots, y_n)'$ and $\bar{\mathbf{y}} = \frac{1}{n}\mathbf{Y}'\mathbf{j}$.
Then we have

$$\begin{aligned}
S &= \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})(y_i - \bar{y})' \\
&= \frac{1}{n-1}(\sum_{i=1}^{n} y_i y_i' - n\bar{y}\bar{y}') \\
&= \frac{1}{n-1}(\mathbf{Y}'\mathbf{Y} - \frac{1}{n}\mathbf{Y}'\mathbf{j}\mathbf{j}'\mathbf{Y}) \\
&= \frac{1}{n-1}(\mathbf{Y}'\mathbf{Y} - \frac{1}{n}\mathbf{Y}'\mathbf{J}\mathbf{Y}) \\
&= \frac{1}{n-1}\mathbf{Y}'(\mathbf{I} - \frac{1}{n}\mathbf{J})\mathbf{Y}
\end{aligned}$$

Since $\bar{\mathbf{Y}} = \bar{\mathbf{y}}\mathbf{j}' = \frac{1}{n}\mathbf{Y}'\mathbf{j}\mathbf{j}'$, the second equation is trivial.

2 Some matrix algebra

(a) Given that $A = \begin{bmatrix} 4 & 8 & 8 \\ 3 & 6 & -9 \end{bmatrix}$, we have that $AA' = \begin{bmatrix} 144 & -12 \\ -12 & 126 \end{bmatrix}$. It's easy to find out that the eigenvalues of AA' are 150 and 120. The corresponding eigenvectors are $[\frac{2}{\sqrt{5}}, -\frac{1}{\sqrt{5}}]'$ and $[\frac{1}{\sqrt{5}}, \frac{2}{\sqrt{5}}]'$

(b) Similarly, for $A'A$, it's eigenvalues are 150, 120 and 0, the corresponding eigenvectors are $[1/\sqrt{30}, 2/\sqrt{30}, 5/\sqrt{30}]'$, $[1/\sqrt{6}, 2/\sqrt{6}, -1\sqrt{6}]'$ and $[-2/\sqrt{5}, 1/\sqrt{5}, 0]'$.

(c) Since we've obtained the eigenvalues and the corresponding eigenvectors of AA', we have that:

$$AA' = \begin{bmatrix} \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \\ -\frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \end{bmatrix} \begin{bmatrix} 150 & 0 \\ 0 & 120 \end{bmatrix} \begin{bmatrix} \frac{2}{\sqrt{5}} & -\frac{1}{\sqrt{5}} \\ \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \end{bmatrix}$$

(d) The singular value decomposition of A is $A = U\Sigma V'$, where U consists of the eigenvectors of AA', V consists of the eigenvectors of $A'A$. While Σ consists of the singular value of A. Therefore, we have

$$A = \begin{bmatrix} \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \\ -\frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \end{bmatrix} \begin{bmatrix} 5\sqrt{6} & 0 & 0 \\ 0 & 2\sqrt{30} & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{30}} & \frac{2}{\sqrt{30}} & \frac{5}{\sqrt{30}} \\ \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & -\frac{1}{\sqrt{6}} \\ -\frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} & 0 \end{bmatrix}$$

3 Linear combinations of variables

We have $Y = (Y_1, Y_2, Y_3, Y_4)'$, the population mean vector $\mu = (4, 3, 2, 1)'$ and the population covariance matrix

$$\Sigma = \begin{bmatrix} 3 & 0 & 2 & 2 \\ 0 & 1 & 1 & 0 \\ 2 & 1 & 9 & -2 \\ 2 & 0 & -2 & 4 \end{bmatrix}$$

By some calculation, we have:

(a) $E(y^{(1)}) = (4, 3)'$.

(b) $E(Ay^{(1)}) = A\mu_1 = (1, 2)(4, 3)' = 10$.

(c) $COV(y^{(1)}) = \Sigma_{11} = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}$.

(d) $COV(Ay^{(1)}) = A\Sigma_{11}A' = (1, 2) \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix} (1, 2)' = 7$.

(e) $E(By^{(2)}) = B\mu_2 = \begin{pmatrix} 1 & -2 \\ 2 & -1 \end{pmatrix} (2, 1)' = (0, 3)'$.

(f) $COV(By^{(2)}) = B\Sigma_{22}B' = \begin{pmatrix} 1 & -2 \\ 2 & -1 \end{pmatrix} \begin{pmatrix} 9 & -2 \\ -2 & 4 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & -1 \end{pmatrix} = \begin{pmatrix} 33 & 36 \\ 36 & 48 \end{pmatrix}$.

(g) $COV(y^{(1)}, y^{(2)}) = \Sigma_{12} = \begin{pmatrix} 2 & 2 \\ 1 & 0 \end{pmatrix}$.

(h) $COV(Ay^{(1)}, By^{(2)}) = A\Sigma_{12}B' = (1, 2) \begin{pmatrix} 2 & 2 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & -1 \end{pmatrix} = (0, 6)$.

4 Variables measured in milliequivalents per 100g:

```
y <- y[,-1]
z1 <- y[,1] + y[,2] + y[,3]
z2 <- 2 * y[,1] - 3 * y[,2] + 2 * y[,3]
z3 <- - y[,1] - 2 * y[,2] - 3 * y[,3]
z <- data.frame(z1,z2,z3)
```

(a) The sample mean vector and the sample covariance matrix can be calculated by:

```
> mean_vector <- colMeans(z)
> mean_vector
```

```
     z1      z2      z3
 38.369  40.838 -51.727
> cov_matrix <- cov(z)
> cov_matrix
          z1        z2        z3
z1  323.6376  19.2526 -460.9770
z2   19.2526 588.6710  104.0717
z3 -460.9770 104.0717  686.2697
```

(b) The sample correlation matrix can be calculated by:

```
> D <- diag(1/sqrt(diag(cov_matrix)))
> corr_matrix <- D %*% cov_matrix %*% D
> corr_matrix
            [,1]       [,2]       [,3]
[1,]  1.00000000 0.04410862 -0.9781430
[2,]  0.04410862 1.00000000  0.1637378
[3,] -0.97814302 0.16373782  1.0000000
```

(c) The generalized variance and total variance of z are respectively:

```
> det(cov_matrix)
[1] 45995.55
> sum(diag(cov_matrix))
[1] 1598.578
```

(d) The spectral decomposition of S_z is given by:

```
> cov_matrix_eig <- eigen(cov_matrix)
> cov_matrix_eig #The spectral decomposition
eigen() decomposition
$values
[1] 1.013775e+03 5.847259e+02 7.759291e-02

$vectors
           [,1]        [,2]       [,3]
[1,] -0.5433288  0.20391455  0.8143787
[2,]  0.1763352  0.97613268 -0.1267711
[3,]  0.8207921 -0.07472522  0.5663183
```

and the square root matrix of S_z is:

```
> cov_matrix_eig$vectors %*% diag(sqrt(cov_matrix_eig$values)) %*%
↪  t(cov_matrix_eig$vectors)
           [,1]      [,2]       [,3]
```

```
[1,]  10.589534  1.733925 -14.439283
[2,]   1.733925 24.035112   2.824513
[3,] -14.439283  2.824513  21.674846
```

We can also obtain the square root matrix of S_z by Cholesky decomposition,

```
> chol(cov_matrix)
         z1        z2         z3
z1 17.98993  1.070187 -25.624168
z2  0.00000 24.238929   5.424925
z3  0.00000  0.000000   0.491830
```

The spectral decomposition of R_z is given by:

```
> corr_matrix_eig <- eigen(corr_matrix)
> corr_matrix_eig #The spectral decomposition
eigen() decomposition
$values
[1] 1.9854859438 1.0143393778 0.0001746784

$vectors
            [,1]       [,2]       [,3]
[1,] -0.69986611 0.16435410  0.6951080
[2,]  0.08647836 0.98550551 -0.1459465
[3,]  0.70901969 0.04203123  0.7039350
```

and the square root matrix of R_z is:

```
> corr_matrix_eig$vectors %*% diag(sqrt(corr_matrix_eig$values)) %*%
↪  t(corr_matrix_eig$vectors)
            [,1]       [,2]       [,3]
[1,]  0.72377273 0.07650653 -0.6857841
[2,]  0.07650653 0.98897895  0.1267572
[3,] -0.68578407 0.12675720  0.7166818
```

We can also obtain the square root matrix of R_z by Cholesky decomposition,

```
> chol(corr_matrix)
     [,1]       [,2]        [,3]
[1,]    1 0.04410862 -0.97814302
[2,]    0 0.99902674  0.20708391
[3,]    0 0.00000000  0.01877447
```

5 Los Angeles area air-pollution variables

(a) The scatterplot matrix for air−pollution variables can be given by:

```
pairs(y, main = "Pairwise Scatter Plot Matrix")
ggpairs(y)
```

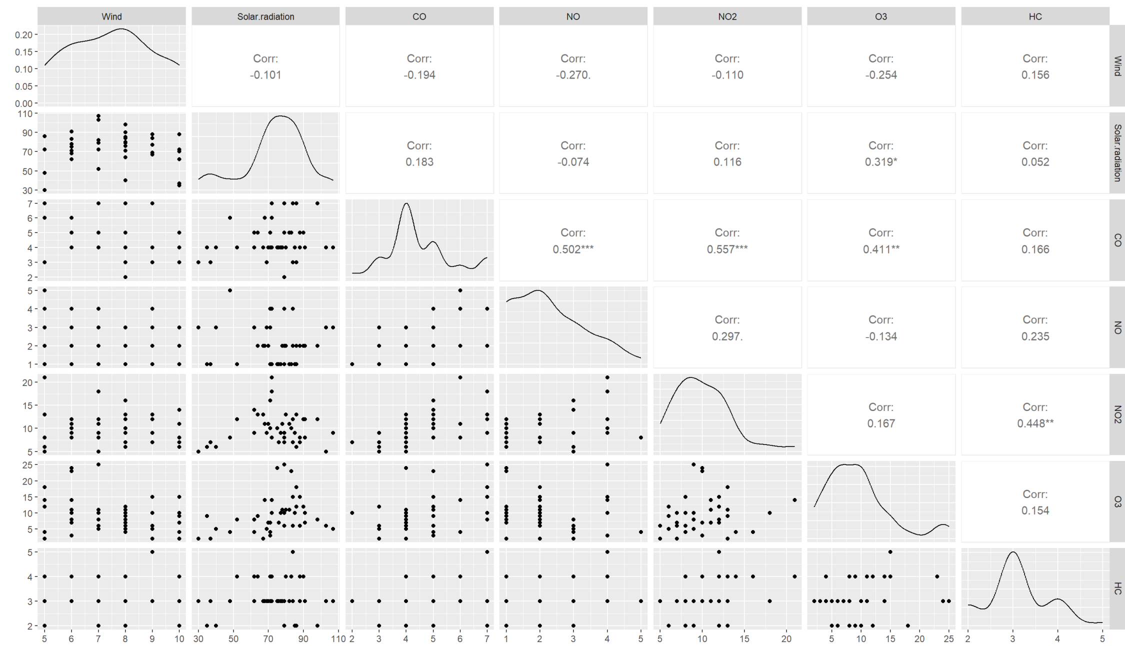

图 1: Pairwise Scatter Plot Matrix

We can see that, some of the variables are discrete since they locates on the grid. Also, we can see that there's a week relationship appeared between gases (CO, NO, etc.) and other variables.

(b) The sample mean vector, sample covariance matrix and sample correlation matrix can be given by:

```
> mean_vector <- colMeans(y)
> mean_vector
           Wind Solar.radiation              CO              NO             NO2
       7.500000       73.857143        4.547619        2.190476       10.047619
             O3              HC
       9.404762        3.095238

> cov_matrix <- cov(y)
> cov_matrix
                      Wind Solar.radiation         CO         NO        NO2
Wind             2.5000000      -2.7804878 -0.3780488 -0.4634146 -0.5853659
Solar.radiation -2.7804878     300.5156794  3.9094077 -1.3867596  6.7630662
```

```
CO              -0.3780488       3.9094077  1.5220674  0.6736353  2.3147503
NO              -0.4634146      -1.3867596  0.6736353  1.1823461  1.0882695
NO2             -0.5853659       6.7630662  2.3147503  1.0882695 11.3635308
O3              -2.2317073      30.7909408  2.8217189 -0.8106852  3.1265970
HC               0.1707317       0.6236934  0.1416957  0.1765389  1.0441347
                        O3         HC
Wind            -2.2317073 0.1707317
Solar.radiation 30.7909408 0.6236934
CO               2.8217189 0.1416957
NO              -0.8106852 0.1765389
NO2              3.1265970 1.0441347
O3              30.9785134 0.5946574
HC               0.5946574 0.4785134
```

```
> cor_matrix <- cor(y)
> cor_matrix
                      Wind Solar.radiation         CO          NO        NO2
Wind             1.0000000     -0.10144191 -0.1938032 -0.26954261 -0.1098249
Solar.radiation -0.1014419      1.00000000  0.1827934 -0.07356907  0.1157320
CO              -0.1938032      0.18279338  1.0000000  0.50215246  0.5565838
NO              -0.2695426     -0.07356907  0.5021525  1.00000000  0.2968981
NO2             -0.1098249      0.11573199  0.5565838  0.29689814  1.0000000
O3              -0.2535928      0.31912373  0.4109288 -0.13395214  0.1666422
HC               0.1560979      0.05201044  0.1660323  0.23470432  0.4477678
                        O3         HC
Wind            -0.2535928 0.15609793
Solar.radiation  0.3191237 0.05201044
CO               0.4109288 0.16603235
NO              -0.1339521 0.23470432
NO2              0.1666422 0.44776780
O3               1.0000000 0.15445056
HC               0.1544506 1.00000000
```

The majority of variable pairs exhibited weak correlations, with only a few exceptions, such as CO and NO, and CO and NO2, which demonstrated moderate correlations. There were no instances of strong correlation observed. This observation is consistent with the patterns indicated in the pairwise scatterplots.

(c) The Eucleadian distance matrix can be computed by:

```
y_subset <- data.matrix(y[1:5, ])
euclidean_dist_matrix <- as.matrix(dist(y_subset, method = "euclidean"))
```

```
> euclidean_dist_matrix
          1         2         3         4         5
1  0.000000 10.535654  9.486833 13.304135  9.110434
2 10.535654  0.000000  5.744563 21.771541 16.852300
3  9.486833  5.744563  0.000000 18.083141 13.076697
4 13.304135 21.771541 18.083141  0.000000  7.211103
5  9.110434 16.852300 13.076697  7.211103  0.000000
```

The Mahalanobis/statistical distance matrix can be computed by:

```
inv_cov_matrix <- (solve(cov_matrix))
n <- nrow(y_subset)
mahalanobis_dist_matrix <-  matrix(as.double(1:25), nrow = 5, ncol = 5)
for (i in 1:5) {
  for (j in 1:5) {
    diff_vec <- as.double(y_subset[i, ] - y_subset[j, ])
    mahalanobis_dist_matrix[i,j] <- sqrt(matrix(diff_vec, nrow = 1) %*%
     ↪  inv_cov_matrix %*% matrix(diff_vec, ncol = 1))
  }
}
> mahalanobis_dist_matrix
         [,1]     [,2]     [,3]     [,4]     [,5]
[1,] 0.000000 4.221941 4.518621 4.694563 4.097358
[2,] 4.221941 0.000000 1.626539 3.811112 2.063497
[3,] 4.518621 1.626539 0.000000 3.402224 2.099450
[4,] 4.694563 3.811112 3.402224 0.000000 3.313883
[5,] 4.097358 2.063497 2.099450 3.313883 0.000000
```

Relative to Euclidean distance, the Mahalanobis distance offers a refined measure of distance by accounting for the variance within certain variables and the correlation among certain variable pairs, thus excluding these factors from its computation.

(d) We have that:

```
> det(cov_matrix) # generalized sample variance
[1] 35307.53
> sum(diag(cov_matrix)) # total sample variance
[1] 348.5407
```

(e) The spectral decomposition is given by:

```
> cov_matrix_eig <- eigen(cov_matrix)
> cov_matrix_eig #The spectral decomposition
eigen() decomposition
```

```
$values
[1] 304.2578640  28.2761046  11.4644830   2.5243296   1.2795247   0.5287288
[7]   0.2096157

$vectors
             [,1]        [,2]        [,3]          [,4]          [,5]         [,6]
[1,] -0.010039244  0.07622439  0.03087761  0.9203045748  0.3423859285  0.011779079
[2,]  0.993199405  0.11615518  0.00659069 -0.0002118679  0.0022391022  0.003353218
[3,]  0.014062314 -0.09956775 -0.18282641 -0.1382922410  0.6500776063 -0.563893916
[4,] -0.004710175  0.01320423 -0.13021553 -0.3277842624  0.6431560485  0.497513370
[5,]  0.024255644 -0.15038113 -0.95526318  0.1023719020 -0.2065840405 -0.009009299
[6,]  0.112429558 -0.97335904  0.16981025  0.0632480276 -0.0002935726  0.051067254
[7,]  0.002340785 -0.02382046 -0.08519558  0.1095073458  0.0619613872  0.657012233
             [,7]
[1,] -0.169729925
[2,] -0.001781987
[3,]  0.443577538
[4,] -0.462855916
[5,] -0.105029951
[6,] -0.066992404
[7,]  0.738019426
```

and the Cholesky decomposition is:

```
> chol(cov_matrix) #Cholesky decomposition
                    Wind Solar.radiation        CO         NO         NO2
Wind            1.581139       -1.758535 -0.239099 -0.2930891 -0.37021787
Solar.radiation 0.000000       17.245963  0.202305 -0.1102964  0.35440324
CO              0.000000        0.000000  1.193303  0.5244867  1.80552154
NO              0.000000        0.000000  0.000000  0.8995517  0.07990644
NO2             0.000000        0.000000  0.000000  0.0000000  2.79903104
O3              0.000000        0.000000  0.000000  0.0000000  0.00000000
HC              0.000000        0.000000  0.000000  0.0000000  0.00000000
                        O3         HC
Wind            -1.4114556 0.10798021
Solar.radiation  1.6414767 0.04717512
CO               1.8035341 0.13238040
NO              -2.2113774 0.16003329
NO2             -0.3777415 0.29138247
O3               4.2433768 0.21087886
HC               0.0000000 0.54048060
```

The spectral decomposition breaks down the sample covariance matrix into two orthogonal matrices and a diagonal matrix. In contrast, the Cholesky decomposition splits it into an upper triangular matrix and its transpose.

(f) We can draw a 3-D scatter plot by using the package "plotly",

```
plot_ly(x = ~y$Wind, y = ~y$O3, z = ~y$Solar.radiation, type = 'scatter3d', mode =
↪  'markers',
        marker = list(size = 5, color = y$O3, colorscale = c('Blues','Reds'),
        ↪  opacity = 0.8)) %>%
  layout(title = '3D Scatter Plot',
         scene = list(xaxis = list(title = 'x-Wind'),
                      yaxis = list(title = 'y-O3'),
                      zaxis = list(title = 'z-Solar.radiation')))
```

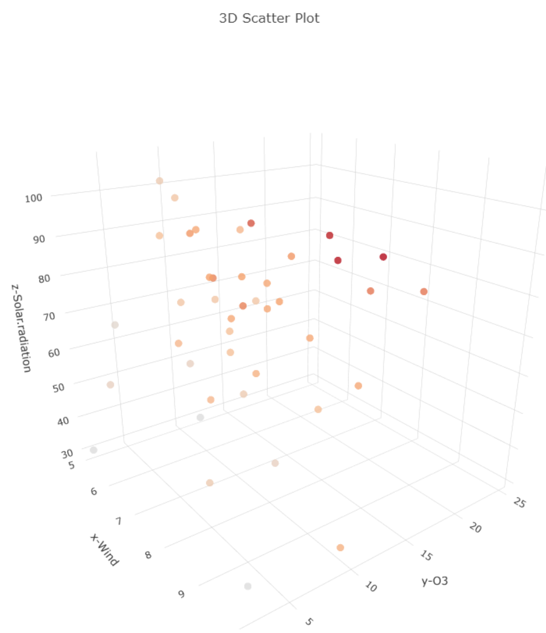

图 2: 3D Scatter Plot

6 Plot the scatterplots between X and Y under the respective settings

(a) X and Y are positively correlated;

```
X <- rnorm(n, mean = 50, sd = 10)
Y <- 0.5*X + rnorm(n, mean = 0, sd = 5) # Positive correlation
plot(X, Y, main = "Positively Correlated X and Y", xlab = "X", ylab = "Y", pch = 19)
```

(b) X and Y are negatively correlated;

```
Y <- -0.5*X + rnorm(n, mean = 0, sd = 5) # Negative correlation
plot(X, Y, main = "Negatively Correlated X and Y", xlab = "X", ylab = "Y", pch = 19)
```

(c) X and Y are perfectly positive-correlated;

```
Y <- 2*X # Perfect positive correlation
plot(X, Y, main = "Perfectly Positive-Correlated X and Y", xlab = "X", ylab = "Y",
↪  pch = 19, ylim = range(Y))
```

(d) X and Y are uncorrelated;

```
Y <- rnorm(n, mean = 50, sd = 10) # Uncorrelated
plot(X, Y, main = "Uncorrelated X and Y", xlab = "X", ylab = "Y", pch = 19)
```

(e) X and Y are nonlinearly correlated.

```
Y <- X^2 + rnorm(n, mean = 0, sd = 100) # Nonlinear correlation
plot(X, Y, main = "Nonlinearly Correlated X and Y", xlab = "X", ylab = "Y", pch =
↪  19)
# Ensure positive values for X
X <- runif(n, min = 1, max = 100) # Uniform distribution for positive values
# Logarithmic relationship
Y <- 20*log(X) + rnorm(n, mean = 0, sd = 10) # Adding some noise
# Plot
plot(X, Y, main = "Nonlinearly (Logarithmic) Correlated X and Y", xlab = "X", ylab =
↪  "Y", pch = 19)
```

The plots are illustrated below:

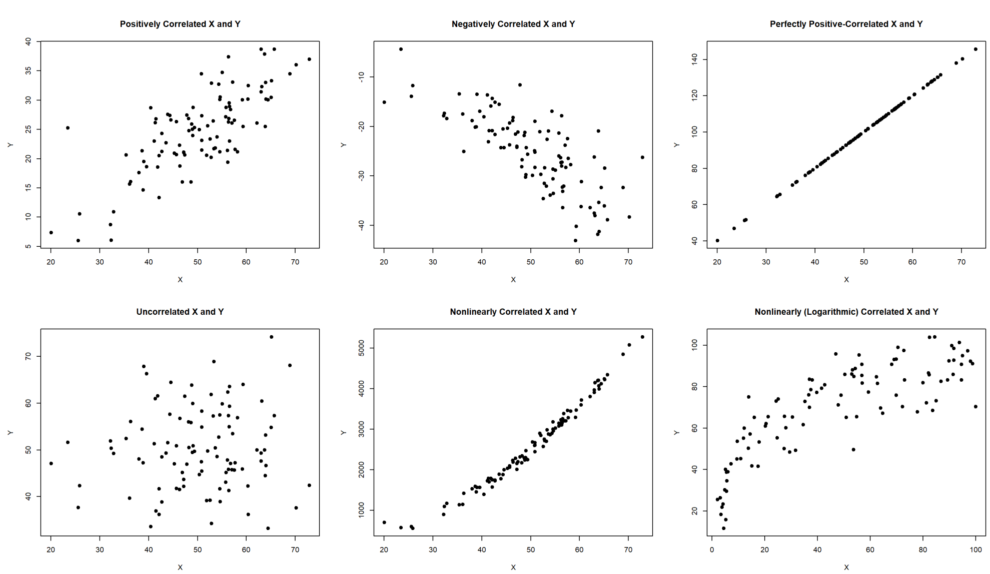

图 3: Scatterplots between X and Y