## HU Extension Assignment 01 E63 Big Data Analytics

## 

### Handed out: 01/30/2016 Due by 11:59PM on Friday, 02/05/2016

**Download and install the latest version of R and R Studio.**

**Problem 1.** Create a vector V with 8 elements (7,2,1,0,3,-1,-3,4).

> V <- c(7,2,1,0,3,-1,-3,4)

> V

[1] 7 2 1 0 3 -1 -3 4

// In the above command I have created a vector with 8 elements.

* Transform that vector into a rectangular matrix A of dimensions 4X2 (4- rows, 2-columns).

> A <- matrix(V, nrow=4)

// In the above command I constructed a matrix with 4 rows. And as ector V has 8 elements number of elements will be 2

> A

[,1] [,2]

[1,] 7 3

[2,] 2 -1

[3,] 1 -3

[4,] 0 4

* Create a matrix transpose to the above matrix A. Call that matrix AT.

> AT <- t(A)

// In the above command I used t() function that gives out transpose of input matrix.

> AT

[,1] [,2] [,3] [,4]

[1,] 7 2 1 0

[2,] 3 -1 -3 4

* Calculate matrix products: A\*AT and AT\*A. Present the results. What are the dimensions of those two product matrices.

> A %\*% AT

[,1] [,2] [,3] [,4]

[1,] 58 11 -2 12

[2,] 11 5 5 -4

[3,] -2 5 10 -12

[4,] 12 -4 -12 16

> AT %\*% A

[,1] [,2]

[1,] 54 16

[2,] 16 35

// Here I have used %\*% which does matrix multiplication

// matrix product A\*AT has the dimension 4 by 4 (4 rows, 4 columns)

// matrix product AT\*A has the dimension 2 by 2 (2 rows, 2 columns)

* Square matrixes sometimes have an inverse matrix. Try calculating inverse matrices (or matrixes, if you prefer) of above matrices (matrixes) A\*AT and AT\*A.

> solve(A %\*% AT)

Error in solve.default(A %\*% AT) :

Lapack routine dgesv: system is exactly singular: U[4,4] = 0

> det(A %\*% AT)

[1] 0

// In the above I have used solve() which gives inverse of a matrix

// inverse matrix of (A%\*AT) does not exist as determinant of that matrix is 0

> solve(AT %\*% A)

[,1] [,2]

[1,] 0.021419829 -0.009791922

[2,] -0.009791922 0.033047736

// Inverse matrix of (AT %\*% A) is above. It exists because determinant of that matrix is not 0

> det(AT %\*% A)

[1] 1634

* Extend the above vector V with the ninth number of value -2. Do it elegantly by concatenating two vectors (☺).

> V

[1] 7 2 1 0 3 -1 -3 4

> V <- c(V, c(-2))

> V

[1] 7 2 1 0 3 -1 -3 4 -2

// In the above command I have used concatenation of vector V and a vector with one element (i.e. -2)

* Transform that extended vector into a 3X3 matrix B.

> B <- matrix(V, nrow=3)

> B

[,1] [,2] [,3]

[1,] 7 0 -3

[2,] 2 3 4

[3,] 1 -1 -2

// In the above command, I have created a matrix B from vector V.

* Calculate the inverse matrix of matrix B. Call it Binv. Demonstrate that the product of B and Binv is the same as the product of Binv and B and is equal to what?

> Binv <- solve(B)

> Binv

[,1] [,2] [,3]

[1,] -2 3 9

[2,] 8 -11 -34

[3,] -5 7 21

// In the above commands I have used solve() function to calculate inverse of matrix

> round(B %\*% Binv)

[,1] [,2] [,3]

[1,] 1 0 0

[2,] 0 1 0

[3,] 0 0 1

> round(Binv %\*% B)

[,1] [,2] [,3]

[1,] 1 0 0

[2,] 0 1 0

[3,] 0 0 1

// In the above commands I have calculated matrix product (B \* Binv) and the matrix product (Binv \* B).

// So, both matrix products (B \* Binv) and (Binv \* B) are equal to Identity matrix, which is of dimension 3 rows by 3 columns.

* Determine the eigenvectors of matrixes B.

> eigen(B)

$values

[1] 6.854102 1.000000 0.145898

$vectors

[,1] [,2] [,3]

[1,] 0.86822600 0.1825742 0.2159107

[2,] 0.49436902 -0.9128709 -0.8426423

[3,] 0.04222416 0.3651484 0.4932914

// In the above command I have used eigen() command that gives eigen values and corresponding eigen vectors of the input matrix

* Construct a new matrix C which is made by using each eigenvector of matrix B as a column. Calculate the product of matrix C and matrix B and the product of matrix B and C. Is there any significance to the elements of the product matrixes.

> C <- eigen(B)$vectors

> class(C)

[1] "matrix"

//In the above commands I have created a matrix out of eigen vectors of matrix B

> C %\*% B

[,1] [,2] [,3]

[1,] 6.658641 0.3318118 -2.3062028

[2,] 0.792199 -1.8959704 -3.4493061

[3,] 1.519157 0.6021537 0.3473382

// Above command gives matrix product of matrices C and B

> B %\*% C

[,1] [,2] [,3]

[1,] 5.9509095 0.1825742 0.03150095

[2,] 3.3884557 -0.9128709 -0.12293986

[3,] 0.2894087 0.3651484 0.07197025

// Above command gives matrix product of matrices B and C

> det(C %\*% B)

[1] -0.1275499

> det(B %\*% C)

[1] -0.1275499

// Significance of elements of product matrices is that determinant of (C\*B) and (B\*C) is same ( -0.1275499)

* Transform matrix B into a matrix with names columns and named rows.

> dimnames(B) <- list(c("R1", "R2", "R3"), c("C1", "C2", "C3"))

> B

C1 C2 C3

R1 7 0 -3

R2 2 3 4

R3 1 -1 -2

// In the above commands I have used dimnames to set row and column names to matrix B

* Transformed that fully “named” matrix into a data.frame.

> B.data.frame <- data.frame(B)

> B.data.frame

C1 C2 C3

R1 7 0 -3

R2 2 3 4

R3 1 -1 -2

// In the above commands I have used data.frame() command to create data frame from the input matrix.

* Ask the object you just created what is its class().

> class(B.data.frame)

[1] "data.frame"

// So class of the object is data frame

**Problem 2.** Consider file 2006Data.csv upload to the class site in Assignment 01 folder. File represents actual measurement of power consumption in a country somewhere in a California. Import data contained in that file into a data frame. You are expected to Google and find a function that will let you perform that import. Create a scatter plot of power consumption vs. temperature and power consumption vs. hour of the day. Subsequently create a boxplot with power on the vertical axis and hour of the day on the horizontal axis. The objective is to present the distribution (variation) of power consumption for every hour of the day.

> setwd("~/work/big-data-analytics-harvard/week1-Jan-29-2016/assignment")

> power.consumption.dataframe <- read.delim("2006Data.csv", sep=",")

// The above command reads the csv file and sep parameter indicates that ‘,’ is the delimeter

Task: Create a scatter plot of power consumption vs. temperature.

> setwd("~/work/big-data-analytics-harvard/week1-Jan-29-2016/assignment")

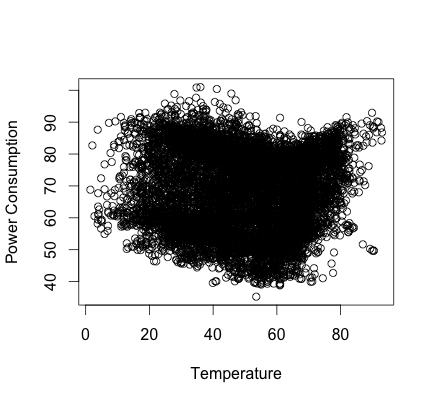
> power.consumption.dataframe <- read.delim("2006Data.csv", sep=",")

// The above command reads the csv file and sep parameter indicates that ‘,’ is the delimeter

> plot(power.consumption.dataframe$Temperature, power.consumption.dataframe$Power, xlab="Temperature", ylab="Power Consumption")

// Above command plots temperature on x axis and power on y axis. Giving us the below

// scatter plot of power consumption vs. temperature



Task: Create a scatter plot of power consumption vs. hour of the day

> setwd("~/work/big-data-analytics-harvard/week1-Jan-29-2016/assignment")

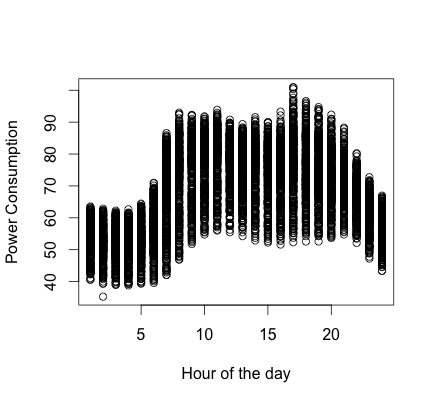
> power.consumption.dataframe <- read.delim("2006Data.csv", sep=",")

// The above command reads the csv file and sep parameter indicates that ‘,’ is the delimeter

> plot(power.consumption.dataframe$Hour, power.consumption.dataframe$Power, xlab="Hour of the day", ylab="Power Consumption")

// Above command plots Hour on x axis and Power on y axis giving us the below

// scatter plot of Power Consumption vs. Hour of the day



Task: create a boxplot with power on the vertical axis and hour of the day on the horizontal axis

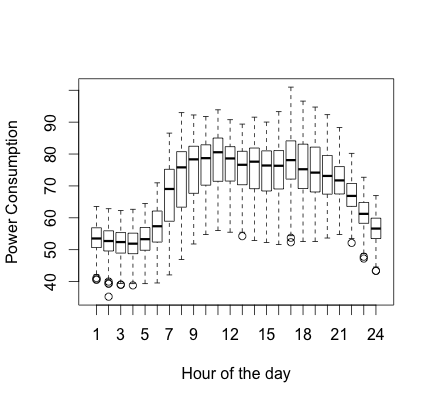
> setwd("~/work/big-data-analytics-harvard/week1-Jan-29-2016/assignment")

> power.consumption.dataframe <- read.delim("2006Data.csv", sep=",")

// The above command reads the csv file and sep parameter indicates that ‘,’ is the delimeter

> boxplot(Power ~ Hour, data=power.consumption.dataframe, ylab="Power Consumption", xlab="Hour of the day")

// Above command plots hour on x axis and power consumption on y axis giving us the below boxplot of Power vs. Hour of the day.



**Problem 3.** Separate temperature scale in a reasonable number of intervals: 50 or 100. Calculate average power consumption, minimum power consumption and maximum power consumptions for every interval. Present those three sets of values on a single scatter graph (perhaps in different colours). Calculate three covariance matrixes between temperature and each of those power indicators (min, average, max).

> setwd("~/work/big-data-analytics-harvard/week1-Jan-29-2016/assignment")

> power.consumption.dataframe <- read.delim("2006Data.csv", sep=",")

// The above command reads the csv file and sep parameter indicates that ‘,’ is the delimeter

Task: Separate temperature scale in a reasonable number of intervals: 50 or 100.

Solution:

> range(power.consumption.dataframe$Temperature)

[1] 1.475 93.000

// Above command gives range of values of Temperature in power consumption data

> temperature\_breaks <- seq(0,94,by=2)

// Above command creates breaks of the range of temperature values. Each break is 2 temperature units wide.

Task: Calculate average power consumption, minimum power consumption and maximum power consumptions for every interval.

Solutions:

> tapply(power.consumption.dataframe$Power, cut(power.consumption.dataframe$Temperature, temperature\_breaks, right=FALSE), mean)

[0,2) [2,4) [4,6) [6,8) [8,10) [10,12) [12,14) [14,16)

68.80810 67.59424 63.09021 66.48662 69.62228 68.63619 69.76366 69.15527

[16,18) [18,20) [20,22) [22,24) [24,26) [26,28) [28,30) [30,32)

69.53993 67.92169 70.16936 72.22699 69.99356 70.86809 70.39441 69.34062

[32,34) [34,36) [36,38) [38,40) [40,42) [42,44) [44,46) [46,48)

70.78043 68.56519 69.17780 69.64676 69.14286 67.05606 66.12145 68.14068

[48,50) [50,52) [52,54) [54,56) [56,58) [58,60) [60,62) [62,64)

66.21368 66.30797 65.02477 63.35977 64.55695 62.46747 61.67348 65.13010

[64,66) [66,68) [68,70) [70,72) [72,74) [74,76) [76,78) [78,80)

63.49158 63.66757 67.37782 70.24780 70.86237 73.31438 74.24855 76.49859

[80,82) [82,84) [84,86) [86,88) [88,90) [90,92) [92,94)

75.72409 72.56602 77.33655 75.89586 77.27179 79.46319 86.39373

// Above command cuts temperature into intervals specified by breaks variable. For each temperature interval, it takes all power values and calculates their average/mean value. This gives us temperature intervals and **average power** **consumption** for every interval.

> setwd("~/work/big-data-analytics-harvard/week1-Jan-29-2016/assignment")

> power.consumption.dataframe <- read.delim("2006Data.csv", sep=",")

> range(power.consumption.dataframe$Temperature)

[1] 1.475 93.000

> temperature\_breaks <- seq(0,94,by=2)

> tapply(power.consumption.dataframe$Power, cut(power.consumption.dataframe$Temperature, temperature\_breaks, right=FALSE), min)

[0,2) [2,4) [4,6) [6,8) [8,10) [10,12) [12,14) [14,16) [16,18) [18,20) [20,22)

68.8081 58.1601 54.9205 55.9044 60.4221 57.2195 50.2237 50.9279 49.8641 48.7796 46.3978

[22,24) [24,26) [26,28) [28,30) [30,32) [32,34) [34,36) [36,38) [38,40) [40,42) [42,44)

46.3474 47.6825 49.6092 45.9109 45.6151 45.3238 44.4238 46.4085 39.5643 40.0671 44.2881

[44,46) [46,48) [48,50) [50,52) [52,54) [54,56) [56,58) [58,60) [60,62) [62,64) [64,66)

41.2311 40.1007 40.8964 40.9541 35.2605 39.0662 39.3408 39.3739 38.8142 41.2297 41.4718

[66,68) [68,70) [70,72) [72,74) [74,76) [76,78) [78,80) [80,82) [82,84) [84,86) [86,88)

40.8431 42.7762 39.9362 48.8566 41.0651 42.7008 54.4724 56.8079 55.4768 56.9069 51.6770

[88,90) [90,92) [92,94)

49.7290 49.6333 84.2437

// Above command cuts temperature into intervals specified by breaks variable. For each temperature interval, it takes all power values and calculates their minimum value. This gives us all temperature intervals and the **minimum power** **consumption** for every interval.

> setwd("~/work/big-data-analytics-harvard/week1-Jan-29-2016/assignment")

> power.consumption.dataframe <- read.delim("2006Data.csv", sep=",")

> range(power.consumption.dataframe$Temperature)

[1] 1.475 93.000

> temperature\_breaks <- seq(0,94,by=2)

> tapply(power.consumption.dataframe$Power, cut(power.consumption.dataframe$Temperature, temperature\_breaks, right=FALSE), max)

[0,2) [2,4) [4,6) [6,8) [8,10) [10,12) [12,14) [14,16) [16,18) [18,20)

68.8081 87.6270 76.7704 89.7771 90.2336 91.6283 89.7539 90.9417 92.8717 92.9389

[20,22) [22,24) [24,26) [26,28) [28,30) [30,32) [32,34) [34,36) [36,38) [38,40)

91.5642 92.9723 95.8191 98.8383 96.7866 95.0473 96.6031 100.8437 100.9896 90.8336

[40,42) [42,44) [44,46) [46,48) [48,50) [50,52) [52,54) [54,56) [56,58) [58,60)

100.3930 95.7247 98.9550 96.8664 93.0825 91.7737 91.9065 86.5059 87.1483 87.9378

[60,62) [62,64) [64,66) [66,68) [68,70) [70,72) [72,74) [74,76) [76,78) [78,80)

91.3635 87.1151 85.5433 90.0549 88.1541 87.3791 85.8301 88.4328 87.2949 89.0667

[80,82) [82,84) [84,86) [86,88) [88,90) [90,92) [92,94)

91.5709 89.9356 88.1065 89.5421 92.9252 90.2133 88.1772

// Above command cuts temperature into intervals specified by breaks variable. For each temperature interval, it takes all power values and calculates their maximum value. This gives us all temperature intervals and the **maximum power consumption** for every interval.

Question: Present those three sets of values on a single scatter graph (perhaps in different colours)

> setwd("~/work/big-data-analytics-harvard/week1-Jan-29-2016/assignment")

> power.consumption.dataframe <- read.delim("2006Data.csv", sep=",")

> range(power.consumption.dataframe$Temperature)

[1] 1.475 93.000

> temperature\_breaks <- seq(0,94,by=2)

> power.consumption.matrix <- matrix(tapply(power.consumption.dataframe$Power, cut(power.consumption.dataframe$Temperature, temperature\_breaks, right=FALSE), mean))

//Above command creates a matrix with rows (one row for each temperature interval). And one column, that column being average of power in that temperature interval.

> power.consumption.matrix <- cbind(power.consumption.matrix, tapply(power.consumption.dataframe$Power, cut(power.consumption.dataframe$Temperature, temperature\_breaks, right=FALSE), min))

//Above command adds to the matrix a column, that column being minimum of power in that temperature interval.

> power.consumption.matrix <- cbind(power.consumption.matrix, tapply(power.consumption.dataframe$Power, cut(power.consumption.dataframe$Temperature, temperature\_breaks, right=FALSE), max))

//Above command adds to the matrix a column, that column being maximum of power in that temperature interval.

> colnames(power.consumption.matrix) <- c("average.power", "min.power", "max.power")

// Above command sets column names to power.consumption.matrix

//Below is the power consumption matrix. Each row corresponds to a temperature interval. There are the 3 columns (average,min,max) power consumption.

> power.consumption.matrix

average.power min.power max.power

[0,2) 68.80810 68.8081 68.8081

[2,4) 67.59424 58.1601 87.6270

[4,6) 63.09021 54.9205 76.7704

[6,8) 66.48662 55.9044 89.7771

[8,10) 69.62228 60.4221 90.2336

[10,12) 68.63619 57.2195 91.6283

[12,14) 69.76366 50.2237 89.7539

[14,16) 69.15527 50.9279 90.9417

[16,18) 69.53993 49.8641 92.8717

[18,20) 67.92169 48.7796 92.9389

[20,22) 70.16936 46.3978 91.5642

[22,24) 72.22699 46.3474 92.9723

[24,26) 69.99356 47.6825 95.8191

[26,28) 70.86809 49.6092 98.8383

[28,30) 70.39441 45.9109 96.7866

[30,32) 69.34062 45.6151 95.0473

[32,34) 70.78043 45.3238 96.6031

[34,36) 68.56519 44.4238 100.8437

[36,38) 69.17780 46.4085 100.9896

[38,40) 69.64676 39.5643 90.8336

[40,42) 69.14286 40.0671 100.3930

[42,44) 67.05606 44.2881 95.7247

[44,46) 66.12145 41.2311 98.9550

[46,48) 68.14068 40.1007 96.8664

[48,50) 66.21368 40.8964 93.0825

[50,52) 66.30797 40.9541 91.7737

[52,54) 65.02477 35.2605 91.9065

[54,56) 63.35977 39.0662 86.5059

[56,58) 64.55695 39.3408 87.1483

[58,60) 62.46747 39.3739 87.9378

[60,62) 61.67348 38.8142 91.3635

[62,64) 65.13010 41.2297 87.1151

[64,66) 63.49158 41.4718 85.5433

[66,68) 63.66757 40.8431 90.0549

[68,70) 67.37782 42.7762 88.1541

[70,72) 70.24780 39.9362 87.3791

[72,74) 70.86237 48.8566 85.8301

[74,76) 73.31438 41.0651 88.4328

[76,78) 74.24855 42.7008 87.2949

[78,80) 76.49859 54.4724 89.0667

[80,82) 75.72409 56.8079 91.5709

[82,84) 72.56602 55.4768 89.9356

[84,86) 77.33655 56.9069 88.1065

[86,88) 75.89586 51.6770 89.5421

[88,90) 77.27179 49.7290 92.9252

[90,92) 79.46319 49.6333 90.2133

[92,94) 86.39373 84.2437 88.1772

> install.packages("ggplot2")

// Above command installs ggplot2 library required for ggplot command tat I am going to use further.

> library("ggplot2")

// Above command loads the library ggplot2 into current session

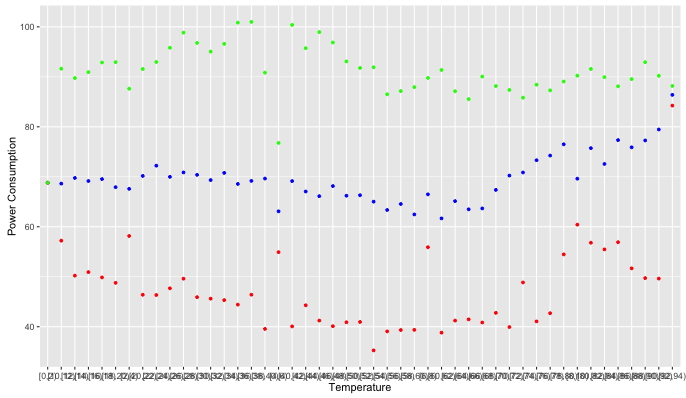
> power.consumption.metrics.dataframe <- data.frame(power.consumption.matrix)

// Above command creates a data frame from the input matrix

> ggplot(power.consumption.metrics.dataframe) + geom\_point(aes(x =rownames(power.consumption.metrics.dataframe), y = average.power), color="blue",size=1) + geom\_point(aes(x =rownames(power.consumption.metrics.dataframe), y = min.power), color="red",size=1) + geom\_point(aes(x =rownames(power.consumption.metrics.dataframe), y = max.power), color="green",size=1) + xlab("Temperature") + ylab("Power Consumption")

// Above command creates a scatter graph with temperature intervals on x axis and average power, min power, max power on y axis

// Note: green dots are max power, blue dots are average power, red dots are min power.



Task: Calculate covariance matrix between temperature and average power.

> setwd("~/work/big-data-analytics-harvard/week1-Jan-29-2016/assignment")

> power.consumption.dataframe <- read.delim("2006Data.csv", sep=",")

> range(power.consumption.dataframe$Temperature)

[1] 1.475 93.000

> temperature\_breaks <- seq(0,94,by=2)

> for(i in 1:47)

{

average.temperatures.vector[i] <- ((temperature\_breaks [i]+breaks[i+1])/2);

}

> average.power.consumption.matrix <- matrix(average.temperatures.vector)

> average.power.consumption.matrix <- cbind(average.power.consumption.matrix, tapply(power.consumption.dataframe$Power, cut(power.consumption.dataframe$Temperature, temperature\_breaks , right=FALSE), mean))

> colnames(average.power.consumption.matrix) <- c("average.temperature", "average.power.consumption")

//

// Below is a matrix. Each row corresponds to a temperature interval. First column is average temperature of the temperature interval., 2nd column is average power consumption in that interval.

> average.power.consumption.matrix

average.temperature average.power.consumption

[0,2) 1 68.80810

[2,4) 3 67.59424

[4,6) 5 63.09021

[6,8) 7 66.48662

[8,10) 9 69.62228

[10,12) 11 68.63619

[12,14) 13 69.76366

[14,16) 15 69.15527

[16,18) 17 69.53993

[18,20) 19 67.92169

[20,22) 21 70.16936

[22,24) 23 72.22699

[24,26) 25 69.99356

[26,28) 27 70.86809

[28,30) 29 70.39441

[30,32) 31 69.34062

[32,34) 33 70.78043

[34,36) 35 68.56519

[36,38) 37 69.17780

[38,40) 39 69.64676

[40,42) 41 69.14286

[42,44) 43 67.05606

[44,46) 45 66.12145

[46,48) 47 68.14068

[48,50) 49 66.21368

[50,52) 51 66.30797

[52,54) 53 65.02477

[54,56) 55 63.35977

[56,58) 57 64.55695

[58,60) 59 62.46747

[60,62) 61 61.67348

[62,64) 63 65.13010

[64,66) 65 63.49158

[66,68) 67 63.66757

[68,70) 69 67.37782

[70,72) 71 70.24780

[72,74) 73 70.86237

[74,76) 75 73.31438

[76,78) 77 74.24855

[78,80) 79 76.49859

[80,82) 81 75.72409

[82,84) 83 72.56602

[84,86) 85 77.33655

[86,88) 87 75.89586

[88,90) 89 77.27179

[90,92) 91 79.46319

[92,94) 93 86.39373

> var(average.power.consumption.matrix)

average.temperature average.power.consumption

average.temperature 752.00000 59.13762

average.power.consumption 59.13762 23.51404

// Above command gives **co-variance matrix of average temperature and average power consumption.**

// The numbers indicate that variance of average temperature within an interval is 752, variance of average power consumption is 23.5

// And co-variance between average temperature and average power is 59.13

Task: Calculate covariance matrix between temperature and min power.

> setwd("~/work/big-data-analytics-harvard/week1-Jan-29-2016/assignment")

> power.consumption.dataframe <- read.delim("2006Data.csv", sep=",")

> range(power.consumption.dataframe$Temperature)

[1] 1.475 93.000

> temperature\_breaks <- seq(0,94,by=2)

> for(i in 1:47)

{

average.temperatures.vector[i] <- ((temperature\_breaks [i]+breaks[i+1])/2);

}

> min.power.consumption.matrix <- matrix(average.temperatures.vector)

> min.power.consumption.matrix <- cbind(min.power.consumption.matrix, tapply(power.consumption.dataframe$Power, cut(power.consumption.dataframe$Temperature, temperature\_breaks , right=FALSE), min))

> colnames(min.power.consumption.matrix) <- c("average.temperature", "min.power.consumption")

> min.power.consumption.matrix

average.temperature min.power.consumption

[0,2) 1 68.8081

[2,4) 3 58.1601

[4,6) 5 54.9205

[6,8) 7 55.9044

[8,10) 9 60.4221

[10,12) 11 57.2195

[12,14) 13 50.2237

[14,16) 15 50.9279

[16,18) 17 49.8641

[18,20) 19 48.7796

[20,22) 21 46.3978

[22,24) 23 46.3474

[24,26) 25 47.6825

[26,28) 27 49.6092

[28,30) 29 45.9109

[30,32) 31 45.6151

[32,34) 33 45.3238

[34,36) 35 44.4238

[36,38) 37 46.4085

[38,40) 39 39.5643

[40,42) 41 40.0671

[42,44) 43 44.2881

[44,46) 45 41.2311

[46,48) 47 40.1007

[48,50) 49 40.8964

[50,52) 51 40.9541

[52,54) 53 35.2605

[54,56) 55 39.0662

[56,58) 57 39.3408

[58,60) 59 39.3739

[60,62) 61 38.8142

[62,64) 63 41.2297

[64,66) 65 41.4718

[66,68) 67 40.8431

[68,70) 69 42.7762

[70,72) 71 39.9362

[72,74) 73 48.8566

[74,76) 75 41.0651

[76,78) 77 42.7008

[78,80) 79 54.4724

[80,82) 81 56.8079

[82,84) 83 55.4768

[84,86) 85 56.9069

[86,88) 87 51.6770

[88,90) 89 49.7290

[90,92) 91 49.6333

[92,94) 93 84.2437

> var(min.power.consumption.matrix)

average.temperature min.power.consumption

average.temperature 752.00000 -24.66792

min.power.consumption -24.66792 79.81808

// Above command gives **co-variance matrix of average temperature and min power consumption.**

// The numbers indicate that variance of average temperature within an interval is 752, variance of min power consumption is 79.81

// And co-variance between average temperature and min power is -24.66

Task: Calculate covariance matrix between temperature and max power.

> setwd("~/work/big-data-analytics-harvard/week1-Jan-29-2016/assignment")

> power.consumption.dataframe <- read.delim("2006Data.csv", sep=",")

> range(power.consumption.dataframe$Temperature)

[1] 1.475 93.000

> temperature\_breaks <- seq(0,94,by=2)

> for(i in 1:47)

{

average.temperatures.vector[i] <- ((temperature\_breaks [i]+breaks[i+1])/2);

}

> max.power.consumption.matrix <- matrix(average.temperatures.vector)

> max.power.consumption.matrix <- cbind(max.power.consumption.matrix, tapply(power.consumption.dataframe$Power, cut(power.consumption.dataframe$Temperature, temperature\_breaks , right=FALSE), max))

> colnames(max.power.consumption.matrix) <- c("average.temperature", "max.power.consumption")

> max.power.consumption.matrix

average.temperature max.power.consumption

[0,2) 1 68.8081

[2,4) 3 87.6270

[4,6) 5 76.7704

[6,8) 7 89.7771

[8,10) 9 90.2336

[10,12) 11 91.6283

[12,14) 13 89.7539

[14,16) 15 90.9417

[16,18) 17 92.8717

[18,20) 19 92.9389

[20,22) 21 91.5642

[22,24) 23 92.9723

[24,26) 25 95.8191

[26,28) 27 98.8383

[28,30) 29 96.7866

[30,32) 31 95.0473

[32,34) 33 96.6031

[34,36) 35 100.8437

[36,38) 37 100.9896

[38,40) 39 90.8336

[40,42) 41 100.3930

[42,44) 43 95.7247

[44,46) 45 98.9550

[46,48) 47 96.8664

[48,50) 49 93.0825

[50,52) 51 91.7737

[52,54) 53 91.9065

[54,56) 55 86.5059

[56,58) 57 87.1483

[58,60) 59 87.9378

[60,62) 61 91.3635

[62,64) 63 87.1151

[64,66) 65 85.5433

[66,68) 67 90.0549

[68,70) 69 88.1541

[70,72) 71 87.3791

[72,74) 73 85.8301

[74,76) 75 88.4328

[76,78) 77 87.2949

[78,80) 79 89.0667

[80,82) 81 91.5709

[82,84) 83 89.9356

[84,86) 85 88.1065

[86,88) 87 89.5421

[88,90) 89 92.9252

[90,92) 91 90.2133

[92,94) 93 88.1772

> var(max.power.consumption.matrix)

average.temperature max.power.consumption

average.temperature 752.000000 -4.265874

max.power.consumption -4.265874 32.274647

// Above command gives **co-variance matrix of average temperature and max power consumption.**

// The numbers indicate that variance of average temperature within an interval is 752, variance of max power consumption is 32.27

// And co-variance between average temperature and max power is -4.26

So to summarize

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Variance of | | | Co-variance between | | |
| Average power | Min power | Max power | Average power and Temperature | Min power and Temperature | Max power and Temperature |
| 23.51 | 79.81 | 32.27 | 59.13 | -24.66 | -4.26 |

This means that average power varies the list.

Max Power varies more than that.

Min Power varies the most.

Co-variance between Max power and Temperature is the least and is negative.

Co-variance between Min power and Temperature is more than that and is negative.

Co-variance between Average power and Temperature is more than that.

**SUBMISSION INSTRUCTIONS:**

Your main submission should be an MS Word document containing your code, results produced by that code and brief textual descriptions of what you did and why. Typically, you just copy your code and results from the R console and past them into the Word document. Start with this text of homework assignment as the template. Please add any other files that you might have used or generated.

Please read detailed submission and grading instructions on class site.