Jacob Dineen

**Unexcuted**

**#**

**#HW 3 IST 687**

**####STEP 1**

**rm(list = ls(all = TRUE))#Clear Enviroment**

**x <- c(1,2,3,4,5,6,7,8,9,10,50) #Input Vector has (1,2,3,4,5,6,7,8,9,10,50), Stored as x**

**#Test Descriptive Statistics on x**

**mean (x)**

**median(x)**

**max(x)**

**min(x)**

**sd (x)**

**quantile(x, .05)**

**quantile(x, .95)**

**skewness(x)**

**#Create function returning descriptive statistics on 'x' vector. Rename Index Results per the Descriptive Test**

**printVecInfo <- function (x)**

**{**

**descriptive <- list( "mean" = mean(x), "median" = median(x), "min" = min(x),"maximum" = max(x), "stdev" = sd(x), "Quant.5" = quantile(x, .05),**

**"Quantile.95" = quantile(x, .95), "skewness" = skewness(x))**

**return(descriptive)**

**}**

**#Test Function**

**printVecInfo (x)**

**####STEP 2**

**# Create Variable with 50 Red, 50 Blue Marbles**

**#4**

**red<- "red"**

**v.red <- replicate(50, red)**

**blue <- "blue"**

**v.blue <- replicate(50, blue)**

**jar<-c(v.red, v.blue)**

**jar**

**#5**

**length(jar[grepl("red", jar)]) #Found numeric count of "reds" in dataset.**

**#Or**

**length(jar[jar=="red"]) #Found numeric count of "reds" in dataset.**

**#6**

**functjar <- function (jar)**

**{**

**v <-sample(jar, 10, replace =TRUE) #Sample 10 marbles**

**vnum <-length(v[grepl("red", v)])**

**print(vnum)**

**vperc <-paste(round(10\* vnum), "%", sep = "") #Convert to percentage format**

**print(vperc)**

**}**

**functjar(jar) #call function to see number of red marbles when sampling ten marbles in numeric/percent format**

**#7**

**#Function from above, minus the percentage conversion**

**functjar1<- function (jar)**

**{**

**v <-sample(jar, 10, replace =TRUE) #Sample 10 marbles**

**vnum <-length(v[grepl("red", v)])**

**}**

**replicate(20, mean(replicate(20,(functjar1(jar))))) #run function for finding 20 means of samples**

**rep1 <-replicate(20, mean(replicate(20,(functjar1(jar))))) #Store Vector**

**printVecInfo(rep1) #run descriptive stats on samples**

**hist(rep1) #histogram of samples**

**#8**

**replicate(20, mean(replicate(100,(functjar1(jar))))) #run function for finding 20 means of samples**

**rep2 <- replicate(20, mean(replicate(100,(functjar1(jar)))))#Store Vector**

**printVecInfo(rep2) #run descriptive stats on samples**

**hist(rep2) #histogram of samples**

**#9**

**replicate(100, mean(replicate(100,(functjar1(jar))))) #run function for finding 20 means of samples**

**rep3 <- replicate(100, mean(replicate(100,(functjar1(jar)))))#Store Vector**

**printVecInfo(rep3) #run descriptive stats on samples**

**hist(rep3) #histogram of samples**

**#Step 3**

**#10**

**tempvar <- (airquality)**

**#11**

**tempvar<- na.omit(tempvar)**

**tempvar**

**#12**

**#####Generate Desc. Stats and Histograms from each variable of the airquality dataset.**

**##Ozone**

**printVecInfo(tempvar$Ozone)**

**hist(tempvar$Ozone)**

**#Wind**

**printVecInfo(tempvar$Wind)**

**hist(tempvar$Wind)**

**#Temp**

**printVecInfo(tempvar$Temp)**

**hist(tempvar$Temp)**

**#Executed**

> #

> #HW 3 IST 687

>

>

> ####STEP 1

>

> rm(list = ls(all = TRUE))#Clear Enviroment

>

> x <- c(1,2,3,4,5,6,7,8,9,10,50) #Input Vector has (1,2,3,4,5,6,7,8,9,10,50), Stored as x

>

> #Test Descriptive Statistics on x

> mean (x)

[1] 9.545455

> median(x)

[1] 6

> max(x)

[1] 50

> min(x)

[1] 1

> sd (x)

[1] 13.72125

> quantile(x, .05)

5%

1.5

> quantile(x, .95)

95%

30

> skewness(x)

[1] 2.271318

attr(,"method")

[1] "moment"

>

> #Create function returning descriptive statistics on 'x' vector. Rename Index Results per the Descriptive Test

> printVecInfo <- function (x)

+ {

+ descriptive <- list( "mean" = mean(x), "median" = median(x), "min" = min(x),"maximum" = max(x), "stdev" = sd(x), "Quant.5" = quantile(x, .05),

+ "Quantile.95" = quantile(x, .95), "skewness" = skewness(x))

+ return(descriptive)

+ }

>

> #Test Function

> printVecInfo (x)

$mean

[1] 9.545455

$median

[1] 6

$min

[1] 1

$maximum

[1] 50

$stdev

[1] 13.72125

$Quant.5

5%

1.5

$Quantile.95

95%

30

$skewness

[1] 2.271318

attr(,"method")

[1] "moment"

>

>

> ####STEP 2

> # Create Variable with 50 Red, 50 Blue Marbles

> #4

> red<- "red"

> v.red <- replicate(50, red)

> blue <- "blue"

> v.blue <- replicate(50, blue)

> jar<-c(v.red, v.blue)

> jar

[1] "red" "red" "red" "red" "red" "red" "red" "red" "red" "red" "red" "red" "red" "red" "red" "red" "red"

[18] "red" "red" "red" "red" "red" "red" "red" "red" "red" "red" "red" "red" "red" "red" "red" "red" "red"

[35] "red" "red" "red" "red" "red" "red" "red" "red" "red" "red" "red" "red" "red" "red" "red" "red" "blue"

[52] "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue"

[69] "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue"

[86] "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue" "blue"

>

> #5

> length(jar[grepl("red", jar)]) #Found numeric count of "reds" in dataset.

[1] 50

> #Or

> length(jar[jar=="red"]) #Found numeric count of "reds" in dataset.

[1] 50

>

>

>

> #6

>

> functjar <- function (jar)

+ {

+ v <-sample(jar, 10, replace =TRUE) #Sample 10 marbles

+ vnum <-length(v[grepl("red", v)])

+ print(vnum)

+ vperc <-paste(round(10\* vnum), "%", sep = "") #Convert to percentage format

+ print(vperc)

+ }

>

> functjar(jar) #call function to see number of red marbles when sampling ten marbles in numeric/percent format

[1] 6

[1] "60%"

>

> #7

> #Function from above, minus the percentage conversion

>

> functjar1<- function (jar)

+ {

+ v <-sample(jar, 10, replace =TRUE) #Sample 10 marbles

+ vnum <-length(v[grepl("red", v)])

+ }

>

> replicate(20, mean(replicate(20,(functjar1(jar))))) #run function for finding 20 means of samples

[1] 5.05 6.00 4.85 4.90 5.00 5.65 4.70 5.15 4.75 5.10 5.05 4.90 5.60 5.10 4.50 4.30 5.00 4.40 5.25 5.40

>

>

> rep1 <-replicate(20, mean(replicate(20,(functjar1(jar))))) #Store Vector

> printVecInfo(rep1) #run descriptive stats on samples

$mean

[1] 4.9375

$median

[1] 4.9

$min

[1] 4.3

$maximum

[1] 5.55

$stdev

[1] 0.3508917

$Quant.5

5%

4.395

$Quantile.95

95%

5.5025

$skewness

[1] 0.02099437

attr(,"method")

[1] "moment"

> hist(rep1) #histogram of samples

>

> #8

> replicate(20, mean(replicate(100,(functjar1(jar))))) #run function for finding 20 means of samples

[1] 4.78 5.03 4.85 4.90 4.84 5.12 5.23 5.22 5.12 5.21 5.09 4.81 5.23 5.14 5.05 5.19 5.10 5.08 4.91 5.29

> rep2 <- replicate(20, mean(replicate(100,(functjar1(jar)))))#Store Vector

> printVecInfo(rep2) #run descriptive stats on samples

$mean

[1] 4.984

$median

[1] 5.01

$min

[1] 4.77

$maximum

[1] 5.23

$stdev

[1] 0.1439079

$Quant.5

5%

4.7795

$Quantile.95

95%

5.211

$skewness

[1] -0.01931109

attr(,"method")

[1] "moment"

> hist(rep2) #histogram of samples

>

> #9

> replicate(100, mean(replicate(100,(functjar1(jar))))) #run function for finding 20 means of samples

[1] 4.88 4.74 5.03 4.95 4.94 4.87 4.94 5.23 4.88 5.06 5.02 4.85 4.94 4.88 4.98 5.36 5.00 5.19 4.93 5.07 5.07 5.20 4.99

[24] 5.03 4.96 5.23 4.87 4.86 4.97 4.49 5.09 5.17 5.29 4.80 4.93 4.83 5.16 5.30 5.26 4.95 5.01 5.03 5.08 5.25 4.83 5.07

[47] 5.28 4.92 5.06 4.99 4.92 5.08 5.02 5.27 5.04 5.00 5.16 4.79 4.94 4.90 4.70 4.86 4.82 4.84 5.05 5.08 4.92 5.06 4.88

[70] 5.00 5.09 5.00 4.91 5.06 5.03 4.98 4.97 4.91 5.14 5.03 5.15 5.20 5.32 5.16 4.97 5.05 5.07 5.10 4.86 5.32 5.01 4.95

[93] 5.05 5.09 4.88 5.12 5.01 5.30 4.72 5.05

> rep3 <- replicate(100, mean(replicate(100,(functjar1(jar)))))#Store Vector

> printVecInfo(rep3) #run descriptive stats on samples

$mean

[1] 5.0187

$median

[1] 5.02

$min

[1] 4.67

$maximum

[1] 5.52

$stdev

[1] 0.1566548

$Quant.5

5%

4.7795

$Quantile.95

95%

5.271

$skewness

[1] 0.2794137

attr(,"method")

[1] "moment"

> hist(rep3) #histogram of samples

>

> #Step 3

>

> #10

> tempvar <- (airquality)

>

> #11

> tempvar<- na.omit(tempvar)

> tempvar

Ozone Solar.R Wind Temp Month Day

1 41 190 7.4 67 5 1

2 36 118 8.0 72 5 2

3 12 149 12.6 74 5 3

4 18 313 11.5 62 5 4

7 23 299 8.6 65 5 7

8 19 99 13.8 59 5 8

9 8 19 20.1 61 5 9

12 16 256 9.7 69 5 12

13 11 290 9.2 66 5 13

14 14 274 10.9 68 5 14

15 18 65 13.2 58 5 15

16 14 334 11.5 64 5 16

17 34 307 12.0 66 5 17

18 6 78 18.4 57 5 18

19 30 322 11.5 68 5 19

20 11 44 9.7 62 5 20

21 1 8 9.7 59 5 21

22 11 320 16.6 73 5 22

23 4 25 9.7 61 5 23

24 32 92 12.0 61 5 24

28 23 13 12.0 67 5 28

29 45 252 14.9 81 5 29

30 115 223 5.7 79 5 30

31 37 279 7.4 76 5 31

38 29 127 9.7 82 6 7

40 71 291 13.8 90 6 9

41 39 323 11.5 87 6 10

44 23 148 8.0 82 6 13

47 21 191 14.9 77 6 16

48 37 284 20.7 72 6 17

49 20 37 9.2 65 6 18

50 12 120 11.5 73 6 19

51 13 137 10.3 76 6 20

62 135 269 4.1 84 7 1

63 49 248 9.2 85 7 2

64 32 236 9.2 81 7 3

66 64 175 4.6 83 7 5

67 40 314 10.9 83 7 6

68 77 276 5.1 88 7 7

69 97 267 6.3 92 7 8

70 97 272 5.7 92 7 9

71 85 175 7.4 89 7 10

73 10 264 14.3 73 7 12

74 27 175 14.9 81 7 13

76 7 48 14.3 80 7 15

77 48 260 6.9 81 7 16

78 35 274 10.3 82 7 17

79 61 285 6.3 84 7 18

80 79 187 5.1 87 7 19

81 63 220 11.5 85 7 20

82 16 7 6.9 74 7 21

85 80 294 8.6 86 7 24

86 108 223 8.0 85 7 25

87 20 81 8.6 82 7 26

88 52 82 12.0 86 7 27

89 82 213 7.4 88 7 28

90 50 275 7.4 86 7 29

91 64 253 7.4 83 7 30

92 59 254 9.2 81 7 31

93 39 83 6.9 81 8 1

94 9 24 13.8 81 8 2

95 16 77 7.4 82 8 3

99 122 255 4.0 89 8 7

100 89 229 10.3 90 8 8

101 110 207 8.0 90 8 9

104 44 192 11.5 86 8 12

105 28 273 11.5 82 8 13

106 65 157 9.7 80 8 14

108 22 71 10.3 77 8 16

109 59 51 6.3 79 8 17

110 23 115 7.4 76 8 18

111 31 244 10.9 78 8 19

112 44 190 10.3 78 8 20

113 21 259 15.5 77 8 21

114 9 36 14.3 72 8 22

116 45 212 9.7 79 8 24

117 168 238 3.4 81 8 25

118 73 215 8.0 86 8 26

120 76 203 9.7 97 8 28

121 118 225 2.3 94 8 29

122 84 237 6.3 96 8 30

123 85 188 6.3 94 8 31

124 96 167 6.9 91 9 1

125 78 197 5.1 92 9 2

126 73 183 2.8 93 9 3

127 91 189 4.6 93 9 4

128 47 95 7.4 87 9 5

129 32 92 15.5 84 9 6

130 20 252 10.9 80 9 7

131 23 220 10.3 78 9 8

132 21 230 10.9 75 9 9

133 24 259 9.7 73 9 10

134 44 236 14.9 81 9 11

135 21 259 15.5 76 9 12

136 28 238 6.3 77 9 13

137 9 24 10.9 71 9 14

138 13 112 11.5 71 9 15

139 46 237 6.9 78 9 16

140 18 224 13.8 67 9 17

141 13 27 10.3 76 9 18

142 24 238 10.3 68 9 19

143 16 201 8.0 82 9 20

144 13 238 12.6 64 9 21

145 23 14 9.2 71 9 22

146 36 139 10.3 81 9 23

147 7 49 10.3 69 9 24

148 14 20 16.6 63 9 25

149 30 193 6.9 70 9 26

151 14 191 14.3 75 9 28

152 18 131 8.0 76 9 29

153 20 223 11.5 68 9 30

>

> #12

> #####Generate Desc. Stats and Histograms from each variable of the airquality dataset.

> ##Ozone

> printVecInfo(tempvar$Ozone)

$mean

[1] 42.0991

$median

[1] 31

$min

[1] 1

$maximum

[1] 168

$stdev

[1] 33.27597

$Quant.5

5%

8.5

$Quantile.95

95%

109

$skewness

[1] 1.231275

attr(,"method")

[1] "moment"

> hist(tempvar$Ozone)

>

> #Wind

> printVecInfo(tempvar$Wind)

$mean

[1] 9.93964

$median

[1] 9.7

$min

[1] 2.3

$maximum

[1] 20.7

$stdev

[1] 3.557713

$Quant.5

5%

4.6

$Quantile.95

95%

15.5

$skewness

[1] 0.449498

attr(,"method")

[1] "moment"

> hist(tempvar$Wind)

>

> #Temp

> printVecInfo(tempvar$Temp)

$mean

[1] 77.79279

$median

[1] 79

$min

[1] 57

$maximum

[1] 97

$stdev

[1] 9.529969

$Quant.5

5%

61

$Quantile.95

95%

92.5

$skewness

[1] -0.2220609

attr(,"method")

[1] "moment"

> hist(tempvar$Temp)