SCGI138

AY24/25 SEM 2

miimi & ba3

R Coding

```
rep(a, b) # replicate item a by b times
seq(from=a,to=b,by=c)
matrix(v,nrows=a,ncols=b,byrow=T)
df <- data.frame(m); names(df) = c(...)
; row.names(df)= c(...)
df[a,b:c]; merge(df1, df2, by="id"); df$abc;
df[order(val),];df[rev(order(val)),] # asc, desc
```

Basic Functions

- `length()`:returns the length of a vector `length(c(...))`
- `sort()`: sorts the elements of a vector `sort(c(1,2,3,4),decreasing=TRUE)`
- `rev()`: Reverses the vector elements. `rev(c(1,2,3,4))`
- `order(),rank()`: The first function returns ranking indices, while the second returns element ranks.
- `unique()`:this function removes the duplicates of a vector.
- `duplicated()`: Indicates previously encountered elements.
- `cut()`:divide continuous data
- `breaks()`:used to define the intervals

Operations on Matrices and Data.Frames

- `dim()`: size of the matrix or data.frame
- `nrow()`: number of rows
- `ncol()`: number of columns
- `dimnames()`: names of rows and columns (as a list)
- `names(),colnames()`: names of columns
- `rownames()`: names of rows

Merging Tables

- `cbind()` to merge columns
- `rbind()` to merge rows

```
e.g.
X1<-data.frame(Id=1:4,
               GENDER=c("M", "F", "F", "M"),
               Weight=c(75,68,48,72))
X2<-data.frame(Id=1:4,
               GENDER=c("M", "F", "F", "M"),
               Height=c(182,165,160,178))
cbind(X1,X2)
merge(X1,X2)
```

Basic Descriptive Statistics

- `summary(x)`summary statistics of the numbers in a vector x.
- `mean(x)`mean of the numbers in a vector x.
- `sd(x)` estimated sd of the numbers in a vector x.
- `var(x)` variance of the numbers in a vector x.
- `min(x)`minimum of the numbers in a vector x.
- `max(x)` maximum of the numbers in a vector x.
- `median(x)` median of the numbers in a vector x.
- `length(x)` number of the element in a vector x.
- `range(x)` minimum and maximum of the numbers.
- `quantile(x)` quantile of the numbers in a vector x.
- `fivenum(x)` five summary of the numbers in a vector x.

The barplot function

```
barplot(x, y,type, main, xlab, ylab, pch,
col, las, bty, bg, cex, ...)
e.g.#InvestigateTheFrequencyOfImprovement
barplot(t2, main = "Distribution of
Improvement", xlab = "Improvement",
ylab = "No. of Patients")
e.g.#CompareTheFrequencyOfImprovementByGroup
t4 = t(t3)
barplot(t4,
main = "Distribution of Improvement",
xlab = "Improvement",
ylab = "No. of Patients",
beside=FALSE,
legend = rownames(t4))
e.g.#Simple Pie Chart
x = c(10, 30, 60, 10, 50)
labels=c("one","two","three","four","five")
pie(x, labels, main="Pie Chart")
#Recall Global Superstore
ex25 = superstore %>%
filter(Region == "Southeast Asia")
freq = aggregate(data = ex25,
City ~ Country,
function(x) length(unique(x)))
library(RColorBrewer)
myPalette=brewer.pal(length(freq[,2]),"Set3")
pie(freq[,2],labels=freq[,1],
main = "Simple Pie Chart of Number of City
by Country in Southeast Asia",
col=myPalette)
```

Histogram

```
hist(v,main,xlab,xlim,ylim,breaks,col,border)
```

- `v` is a vector containing numeric values used in histogram.
- `main` indicates title of the chart.
- `col` is used to set color of the bars.
- `border` is used to set border color of each bar.
- `xlab` is used to give description of x-axis.
- `xlim` is used to specify the range of values on the x-axis.
- `ylim` is used to specify the range of values on the y-axis.
- `breaks` is used to mention the width of each bar.

One-population inferences

```
mean.battery = 60 #begin e.g.1
sd.battery = 20
N = 25
error = qt(.025,N-1,lower.tail=FALSE)*sd.
battery/sqrt(N)
lower = mean.battery - error
upper = mean.battery + error
ci95.battery = c(lower,upper)
ci95.battery #end e.g.1
#begin e.g.2
batt = c(82, 64, 68, 44, 54, 47, 50, 85)
t.test(batt,conf.level = 0.95) #end e.g.2
```

binom.test() and prop.test()

```
binom.test(x,n,p=0.5,alternative="two.sided")
prop.test(x,n,p= NULL ,alternative=
"two.sided", correct = TRUE)
e.g.#OutOf5000People,275WatchedBigBangTheory
>rating <- prop. test(x = 275, n = 5000)
> rating
```

- `x`: the number of of successes
- `n`: the total number of trials
- `p`: the probability to test against.

TWO-population inferences

t.test (stats) Student's t-Test Description Performs one and two sample t-tests on vectors of data.Usage

```
t.test(x, ...)
## Default S3 method:
t.test(x, y = NULL,
alternative = c("two.sided", "less",
"greater"),
mu = 0, paired = FALSE, var.equal = FALSE,
conf. level = 0.95, ...)
## S3 method for class 'formula'
t.test(formula, data, subset, na.action,..)
E.g.#Choose Between two Machines
machine1 <-c(24.58, 22.09, 23.70, 18.89)
machine2 <- c(21.61, 19.06, 20.72, 15.77)
var.test(machine1,machine2,alternative=
"two.sided")
t.test(machine1,machine2,var.equal = TRUE,
alternative="two.sided", mu=0)
E.g.2
before = c(198,201,210,185,204,156)
after = c(194,203,200,183,200,153)
t.test(before,after,alternative =
"less",paired=TRUE)
```

From Mock

```
input1 : m1=matrix(1:8,nrow=2)
output1 : [,1] [,2] [,3] [,4]
    [1,]    1    3    5    7
    [2,]    2    4    6    8
input2 : v1 = c(1,3,4,8,12)
         v2 = seq(2,8,2) #Creates sequence
         #from 2 to 8 in steps of 2
         v1 + v2
output2 : [1]  3  7  10  16  14
input3:cbind(c("S","C","M","A"),seq(0,15,5))
Output3:  [,1] [,2]
     [1,] "S"  "0"
     [2,] "C"  "5"
     [3,] "M"  "10"
     [4,] "A"  "15"
input4: matrix(data=c(1,2,3,4,5,6),nrow=3,
byrow=TRUE)
output4: [,1] [,2]
    [1,]    1    2
    [2,]    3    4
    [3,]    5    6
input5: credit <- data.frame(
id = c("cc2020", "cc5243", "cc0921"),
inc = c(50, 38, 25),
card_fam = c("Platinum", "Gold", "Silver"),
lim = c(120000, 60000, 45000))
#ObtainTheAverageMonthlyIncome
average_income<-mean(credit$inc)
cat("Average:",average_income, "\n")
#ObtainTheCustomerIDOfTheGoldCreditCardHold
gold_ids <- credit$id[credit$card_fam ==
"Gold"]
cat("Gold card holders:", gold_ids, "\n")
#4.Create Frequency Table
data <- read_excel("DailyFoodNutrition.xlsx
freq_table <- table(data$variable_name)
#5
library(readxl) #import data set
table(student$final_grade) #like 4.
table(student$final_grade, student$partici
pation_label) #like above but 2 data column
bins = seq(min(student$engagement_score),
max(student$engagement_score),by = 3)
quiz_score = cut(student$'quiz_score (%)'
,breaks=seq(50,100,by=10))
```

Section 2: Excel

- Construct and interpret a 95% confidence interval for the mean shopping time.
- Confidence Interval (CI): CI = Mean $\pm$95% Confidence

Excel

Sales Analysis Formulas

Smith's sales are in cells B2-B9 and Total Sales are in cell B10. C2-C9 = percentages

Use the formula: (B2 / $B$10) * 100 to calculate the percentage, then drag it down.

Total Sales: = SUM(B2:B9)

Average Sales: = AVERAGE(B2:B9)

Highest Sales: = MAX(B2:B9) **Lowest Sales:** = MIN(B2:B9)

Total Number of Entries: = COUNT(B2:B9)

Number Formatting Settings

Percentage: Select the Percentage column (C2:C9) → Click *Format Cells* → Choose *Percentage* (0 or 2 decimal places).

Currency (₿): Select the Sales column (B2:B9) and the values below (B10:B14) → Click *Format Cells* → Choose *Currency*.

Chart

- Select the data you want (Sale Name and Percentage), highlighting all cells in the row.
- Data analysis → descriptive statistics Or Home → options → Add-Ins → go → analysis ToolPak → descriptive statistics Insert → Pie Chart.

Descriptive Statistics

- **Input range:** Column or row, e.g., `$A$2:$A$12` (A2 to A12).
- **Output range:** Click the cell where you want to place the output.
- In the last box, check the option to find the **kth smallest or largest value** and specify the rank.

Create pivot

Insert → pivot table → selected table (click to choose) → choose new or exit worksheet

PivotTable Fields: Select the data, then drag it into the desired area (the 4 boxes below).