Code Summary

Created	@March 12, 2024 4:42 PM
Class	ADS
Reviewed	☐

```
#数据生成
num <- seq(0, 50) #1-50的数，可以用来作为横坐标
date = sample(365,26,replace = TRUE)
normative_scores <- runif(26, min = 0, max = 100)
#数据获取
data <- read.csv("Chicago2013.csv")
list = data[ ,1]
data <- read.table("Week_7_Tests_PGP3.txt", header = TRUE, se
print(nrow(data[data$Country=="USA",]))
# 计算四分位数
quartiles <- quantile(age_data, probs = c(0.25, 0.5, 0.75))
#数据处理
head(WNV)：
这行代码使用head函数来显示数据框"WNV"的前几行（通常是前6行）。这可以帮助
tail(WNV)：
这行代码使用tail函数来显示数据框"WNV"的最后几行（通常是最后6行）。这可以
nrow(WNV)：
这一行代码使用nrow函数来计算数据框"WNV"中的行数，即数据框中的观测值数量。
attributes(WNV)：
这行代码使用attributes函数来显示数据框"WNV"的属性。这通常包括数据框的列
table(WNV)：
这行代码使用table函数来创建一个频数表，显示数据框"WNV"中每个不同值的出现
str(WNV)：
这行代码使用str函数来显示数据框"WNV"的结构。它会列出数据框中的列名、数据
#数据清洗重整
library(tidyr)
data_spread = spread(data_cleaned, measured, value,
       fill = NA, convert = FALSE,
       drop = TRUE, sep = NULL)
```

```
df <- df %>%
  gather(key = "time.point", value = "ELISA.od", elisa.od, el

library(dplyr)
library(tidyr)
WNV <- WNV %>% rename("YEAR" = "SEASON.YEAR")
anyNA(WNV)
anyDuplicated(WNV)
WNV <- WNV %>% drop_na()
WNV$TEST.DATE <- as.POSIXct(WNV$TEST.DATE, format = "%m/%d/%Y

#正态分布
heights_men <- rnorm(num_men, mean=mean_height_men, sd=sd_hei
population <- rnorm(1e6,100,5)
popmean <- round(mean(population),1)
p = pnorm(i, 50 ,10)
#计算生日重复循环
repeat {
  pb=1
  for (i in 1:(count-1)) {
    pb <- pb * ((365 - i) / 365)
  }
  s=1-pb
  prob = c(prob,s)
  count <- count+1

  if(count > n) {
    break
  }
}
#取样循环
cnt = 0
mean2 = c()
sd2 = c()
while (cnt<1000) {
  a = sample(population,100)
  b = mean(a)
  c = sd (a)
```

```
  mean2 = c(mean2,b)
  sd2 = c(sd2,c)
  cnt =cnt+1
}
hist(sd2)

#i循环
c=c()
for (i in 1:100000){
  a = sample(list,10)
  b=mean(a)
  c=c(c,b)
}

for (i in 1:num_simulations) {
  normative_scores <- runif(26, min = 0, max = 100)
  mean_normative_score <- mean(normative_scores)
    mean_list[i] = mean_normative_score
    if (mean_normative_score < 40) {
    count_mean_lt_40 <- count_mean_lt_40 + 1
  }
}

#双循环嵌套
population <- rnorm(1e6,100,5)
cnt2 = 5
list = c()
sd = c()
while (cnt2<=100) {
  mean = c()
  cnt =0
  while (cnt<100) {
    a = sample(population,cnt2)
    b = mean(a)
    mean = c(mean,b)
    cnt =cnt+1
  }
  c = sd (mean)
```

```
  sd = c(sd,c)
  list = c(list,cnt2)
  cnt2 =cnt2+1
}
plot(list,sd)
#更多取样循环详见week8的project有五个问题采用数据生成并循环看结果


#t检验

#双样本
data1 = data("ToothGrowth")
str(ToothGrowth)
t.test(len ~ supp, data = ToothGrowth)

t_test_dose_0.5_1.0 <- t.test(len ~ dose, data = subset(Tooth
print(t_test_dose_0.5_1.0)

#paired
blood_pressure_data <- read.table("blood_pressure.txt", heade
treatment_effect <- t.test(blood_pressure_data$bp_before, blo
print(treatment_effect)


#画图
boxplot(heights_men, main="men_heights",ylab = "Height") #箱线
hist(z,breaks=0.5:20) #直方图设置断点


morning_scores <- data$Score[data$Time == "Morning"]
afternoon_scores <- data$Score[data$Time == "Afternoon"]
# 创建条形图
barplot(c(mean(morning_scores),mean(afternoon_scores)), names
# 创建两个箱线图
boxplot(morning_scores, afternoon_scores,
        names = c("Morning", "Afternoon"),
        xlab = "Time", ylab = "Score",
        main = "Morning vs Afternoon Scores")
```

```
#不同年龄的成绩图
time_20to30 = c()
time_30to40 = c()
time_40to50 = c()
time_50to60 = c()
for (i in 1:170) {
  if (data2[i,4]>=20&&data2[i,4]<30) {
    time_20to30 = c(time_20to30,data2[i,5])
  }
  if (data2[i,4]>=30&&data2[i,4]<40) {
    time_30to40 = c(time_30to40,data2[i,5])
  }
  if (data2[i,4]>=40&&data2[i,4]<50) {
    time_40to50 = c(time_40to50,data2[i,5])
  }
  if (data2[i,4]>=50&&data2[i,4]<60) {
    time_50to60 = c(time_50to60,data2[i,5])
  } #循环还可以使用这个语句：data_under20=data2[data2$Age < 30&da
}
mean_list = c(mean(time_20to30),mean(time_30to40),mean(time_4
barplot(mean_list,names.arg = c("20~30","30~40","40~50","50~6
```

```
# 计算数据的距离矩阵
# Calculate the distance
matrixd <- dist(data[, c("age_norm", "hours_norm")], method
= "euclidean")
```

```
# power计算
# 1
count1 = 0
for (i in 1:100000) {
  sample1 = rnorm(10, 130, 30)
  sample2 = rnorm(10, 117, 27)
  t1 = t.test(sample1, sample2,alternative = "greater")
  if (t1$p.value <= 0.05 ){
    count1 = count1 + 1
  }
```

```
}

power1 = count1 / 100000

# 2

delta <- 130 * 0.1
sd <- 30
power <- 0.8
sig.level <- 0.05
result <- power.t.test(delta = delta, sd = sd, sig.level =
sig.level, power = power, type = "two.sample", alternative
= "one.sided")
result$n

# 3
delta <- 13
sd <- 30
power <- 0.8
sig.level <- 0.05
result <- power.t.test(delta = delta, sd = sd, sig.level =
sig.level, power = power, type = "paired", alternative = "o
ne.sided")
result$n

# 4
sig.levels <- seq(0.01, 0.05, by = 0.01)
a = c()
for (i in sig.levels){
  result <- power.t.test(delta = delta, sd = sd, sig.level
= sig.level, power = power, type = "one.sample", alternativ
e = "one. Sided")
}
```