Statistical Computing and Simulation: Assignment 3

Deparment of Statistics, NCCU
葉佐晨　高崇哲
{112354016,112354020}@nccu.edu.tw

2024-04-25

Statistical Computing and Simulation

Assignment 3, Due April 26/2024

Question 1

Given the following data, use one of the orthogonalization methods introduced in class to perform regression analysis, including the parameter estimates and their standard errors. (You may use the functions of matrix computation built in S-Plus and R, but not the function `lm` or `glm`.) Compare your results with those from statistical software, such as SAS, SPSS, and Mintab.

為了估計迴歸方程式的係數以及標準差，我們採取的是QR Decomposition，需要先將X進行分解，再經過矩陣運算求解。使用y資料和預測y的預測值的殘差計算MSE，再和X的變異數矩陣進行計算，最後得到係數的標準差。計算係數和標準差後就可以進行t檢定，計算p-value以及判斷係數是否顯著。

下表是使用R語言內建的lm函數估計迴歸係數，我們將以這些估計值為基準，衡量QR方法估計的迴歸係數。

表 1: Coefficients of lm{stats}

	Estimate	Std. Error	t value	Pr(>|t|)
x1	0.7968	0.1662	4.7928	0.0001
x2	1.1114	0.4560	2.4373	0.0254
x3	-0.6250	0.0847	-7.3777	0.0000

下列數值是使用自定義的QR分解估計結果，我們可以看到估計的的數值相當接近內建函數`lm`的結果。

表 2: Coefficients of Mylm

	estimate	std.error	t-value	p-value
x1	0.7968	0.1649	4.8310	0.0000
x2	1.1114	0.4524	2.4567	0.0114
x3	-0.6250	0.0840	-7.4364	0.0000

Question 2

2. Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) both can be used to reduce the data dimensionality. Use the mortality data, 17 5-age groups for ages 0~4, 5~9, . . . , 80~84, in Taiwan area to demonstrate how these two methods work. The data of the years 1970-2000 are used as the "training" (insample) data and the years 2001-2005 are used as the "testing" (out-sample) data. You only need to perform one set of data, according to your gender.

```
## singular value of SVD:

##  [1] 4.14573508 1.02556144 0.49886046 0.45755014 0.35893998 0.35023988
##  [7] 0.26365928 0.24285279 0.22329255 0.18227517 0.14761967 0.13048893
## [13] 0.10500145 0.10125531 0.07893535 0.06278609 0.05604883

## Lee carter model with SVD, MAPE:

## [1] 0.1336392

## Information of PCA:

## Importance of components:
##                            PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     0.7569 0.18724 0.09108 0.08354 0.06553 0.06394 0.04814
## Proportion of Variance 0.8936 0.05468 0.01294 0.01088 0.00670 0.00638 0.00361
## Cumulative Proportion  0.8936 0.94828 0.96122 0.97211 0.97880 0.98518 0.98880
##                            PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.04434 0.04077 0.03328 0.02695 0.02382 0.01917 0.01849
## Proportion of Variance 0.00307 0.00259 0.00173 0.00113 0.00089 0.00057 0.00053
## Cumulative Proportion  0.99186 0.99446 0.99618 0.99732 0.99820 0.99877 0.99931
##                           PC15    PC16    PC17
## Standard deviation     0.01441 0.01146 0.01023
## Proportion of Variance 0.00032 0.00020 0.00016
## Cumulative Proportion  0.99963 0.99984 1.00000
```

```
## Lee carter model with PCA, MAPE:

## [1] 0.1931782
```

在進行SVD跟PCA之前，我們會先對資料取log,並減掉該年齡層的平均死亡率。

SVD: 進行SVD分解後，得到矩陣UPV'，U為時間成分，V為年齡成分，P為奇異值，選取第一個最大的奇異值4.145，與第一行的的時間成分去擬合迴歸模型，並預測2001~2005年，最後帶入平均死亡率和年齡成分取指數，就可以回推我們模型預測2001~2005年的死亡率，與真實值計算後，透過SVD處理的Lee carter模型，其MAPE值為0.133。

PCA: 透過prcomp函數進行PCA分解後，得到時間成分與年齡成分，並選取第一個變異比例最大的主成分，重複與SVD一樣的過程，回推得到我們模型預測2001~2005年的死亡率，透過PCA處理的Lee carter模型，其MAPE值為0.193。

已MAPE當作衡量指標，在Lee carter模型上SVD的表現較PCA要好。

Question 3

(a) Write a small program to perform the "Permutation test" and test your result on the correlation of DDT vs. eggshell thickness in class, and the following data: Check your answer with other correlation tests, such as regular Pearson and Spearman correlation coefficients.

下表是使用自定義的Permutation test進行檢定，表中呈現了總共取了多少個排列組合，p-value相當與有多少比例的組合高於樣本。

表 3: Correlation Test of MyPermTest

n	p-value
5040	0.0774

下列是使用R內建的Pearson's Correlation Test，可以看到p-value相較於Permutation Test較大。因此在小樣本的情況下，Permutation test可能會取得更好的結果。

```
##
##  Pearson's product-moment correlation
##
## data:  ddt and thick
## t = -1.5083, df = 5, p-value = 0.1919
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
```

```
##  -0.9234039  0.3348768
## sample estimates:
##        cor
## -0.5592018
```

(b) Simulate a set of two correlated normal distribution variables, with zero mean and variance 1. Let the correlation coefficient be 0.2 and 0.8. (Use Cholesky!) Then convert the data back to Uniform(0,1) and record only the first decimal number. (亦即只取小數第一位，0至9的整數) Suppose the sample size is 10. Apply the permutation test, Pearson and Spearman correlation coefficients, and records the p-values of these three methods. (10,000 simulation runs)

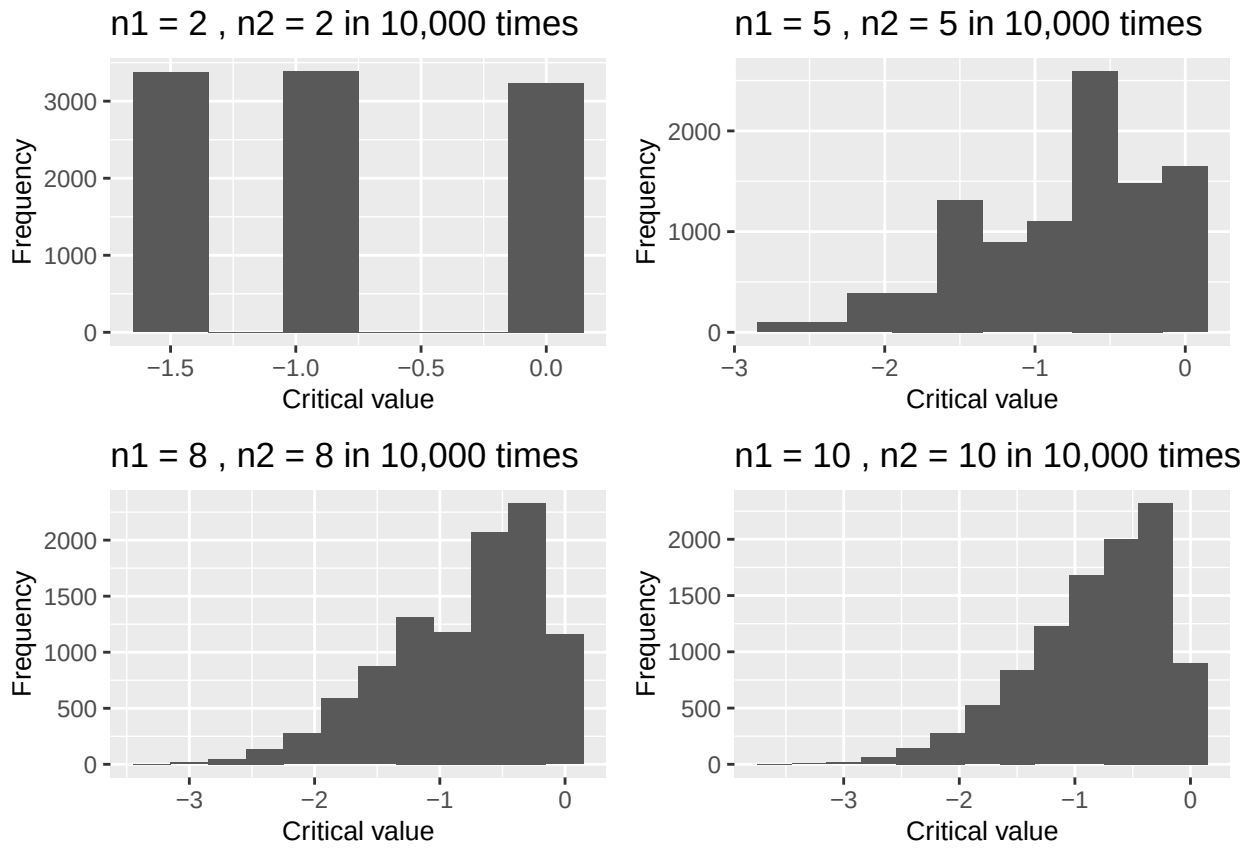
下表是模擬10000次抽樣，分別進行Pearson Correlation test, Spearson Correlation test, Permutation，檢定為顯著的次數，同時考慮了相關係數為0.2或0.8，樣本數為10。由於Permutation test若考慮全部可能的組合會造成計算上的負擔，因為組合數和樣本數呈階乘關係，因此這裡使用的是抽取可能的組合，用以估計p-value。

表 4: Reject Frequency Table of Correlation Test

	pearson	spearson	permutation
r=0.2	94	100	87
r=0.8	844	788	828

Question 4

4. Using simulation to construct critical values of the Mann-Whitney-Wilcoxon test in the case that $2 \leq n_1, n_2 \leq 10$, where n_1 and n_2 are the number of observations in two populations.(Note: The number of replications shall be at least 10,000.)



我們分別在期望值為1的指數分配，個別抽出了2, 5, 8和10個觀察值，並進行10,000次的模擬，進行Mann-Whitney-Wilcoxon 檢定，可以從結果的四張圖看出，當n1和n2越大，臨界值會越集中在-1~0之間。

Question 5

Similar to what Efron did in the Law school data example, compute the bootstrap simulation for 50, 100, . . . , 10,000 replications. But, instead of using the original 15 observations, we want to know if the number of observations plays an important role. Randomly select 10, 15, 20, and 25 observations and then see if the bootstrap variance converges as the number of replications increases. (Note: You also need to compare your results with that of population.)

首先我們直接計算母體的相關係數，用來和Bootstrap模擬比較，母體的相關係數是0.76。

```
## [1] "population correlation is 0.76"
```

Bootstrap方法是先在母體抽出一小部分作為樣本，再根據樣本重複抽取。我們會同時測試不同的樣本數以及重複次數，觀察標準差是否逐漸降低，也就是有收斂的傾向。本次會分別抽取10、15、20、25個樣本，再分別重複抽取50、100、1000、10000次，記錄其標準差以及平均數。

下表是重複次數和樣本數的標準差矩陣，我們可以觀察到隨著重複次數提高，標準差有逐漸降低的趨勢，但並不是絕對的下降，有蠻高的不確定性，若重複次數再提高的話可能會有更明顯的效果。

表 5: Std Deviation Matrix Given Replicate Time and Sample Observation

	obs=10	obs=15	obs=20	obs=25
rep=50	0.1316	0.1595	0.0683	0.1054
rep=100	0.3250	0.1328	0.0744	0.0814
rep=1000	0.0260	0.1116	0.0785	0.1027
rep=10000	0.1342	0.1750	0.0788	0.0453

下表是重複次數和樣本數的相關係數矩陣，可以看到在樣本數較高時，估計值較為接近，估計的差異只有大約0.2，而當樣本數只有10時，估計值的差異達到0.6，因此可以說明樣本數提高會得到較為穩定的估計。

表 6: Correlation Matrix Given Replicate Time and Sample Observation

	obs=10	obs=15	obs=20	obs=25
rep=50	0.8039	0.5810	0.7897	0.7382
rep=100	0.3570	0.8100	0.7783	0.6527
rep=1000	0.9667	0.8171	0.8018	0.6816
rep=10000	0.8227	0.5660	0.7383	0.8560

Question 6

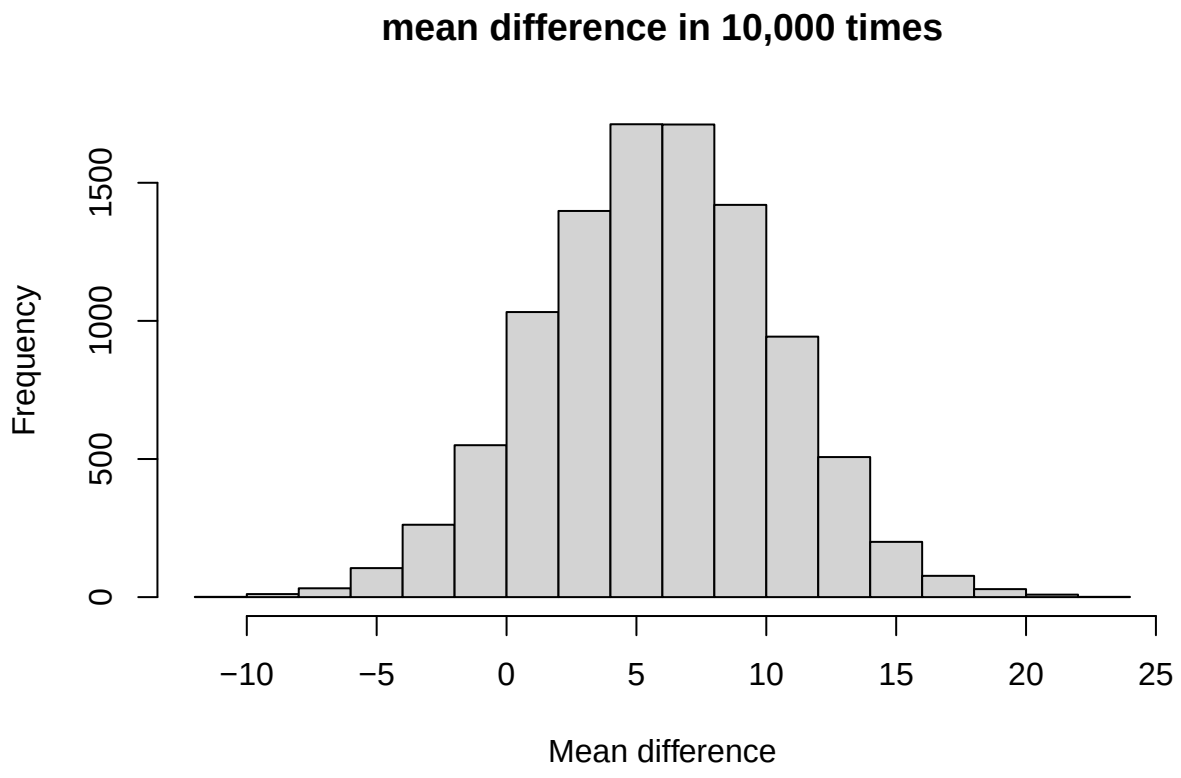
6. To compare teaching, twenty schoolchildren were divided into two groups: ten taught by conventional methods and ten taught by an entirely new approach. The following are the test results:

Category	1	2	3	4	5	6	7	8	9	10
Conventional	65	79	90	75	61	85	98	80	97	75
New	90	98	73	79	84	81	98	90	83	88

Are the two teaching methods equivalent in result? You need to use permutation test, (parametric and non-parametric) bootstrap, and parametric test, and then compare their differences in testing.

```
## Observed mean difference: 5.9
## p-value: 0.117
```

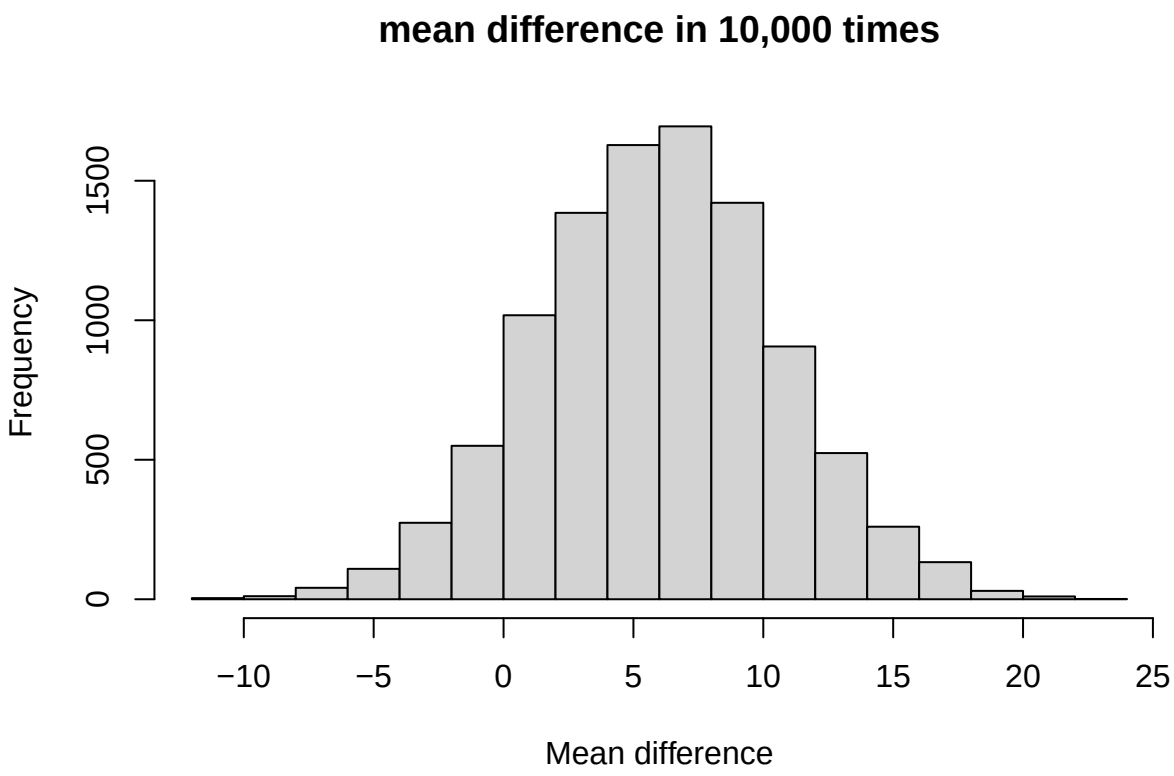
進行Permutation test時，我們先計算出new跟conventional的平均值差異，然後在將這兩組資料混和後重新隨機分類為兩類，重複模擬10,000次，並計算出隨機分為兩類，其平均值差異大於new跟conventional的平均值差異，作為p值，結果顯示，當$\alpha = 0.05$，不拒絕虛無假設，也就是我們沒有足夠證據顯示，new這組資料的平均值大於conventional。




```
## 95% Confidence Interval of Difference of Mean between new and convention:

## [1] -3.525361 15.325361
```

進行non-parametric bootstrap test時，我們會各自在new跟conventional這兩組資料，隨機抽出並放回各取出10個樣本，並計算這兩組樣本的平均值差異，重複模擬10,000次，從圖中可以看到平均值差異大多落在0~10之間，透過模擬結果，new跟conventional的平均值差異95%信賴區間包含0，也就是說，我們沒有足夠證據顯示，new這組資料的平均值大於conventional。




```
## 95% Confidence Interval of Difference of Mean between new and conventiol:

## [1]  0.4983528 20.0541149
```

進行parametric bootstrap test時，我們會各自在new跟conventional這兩組資料，計算期望值跟標準差，作為兩個常態母體的參數，並從這兩個常態分配隨機抽出各10個樣本，並計算這兩組樣本的平均值差異，重複模擬10,000次，從圖中可以看到平均值差異大多落在-2~12之間，透過模擬結果，new跟conventional的平均值差異95%信賴區間不包含0，也就是說，我們有足夠證據顯示，new這組資料的平均值大於conventional。

```
##
##  Paired t-test
##
## data:  new and conventional
## t = 1.2666, df = 9, p-value = 0.2371
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -4.637595 16.437595
## sample estimates:
```

```
## mean difference
##             5.9
```

從parametric test的結果我們可以看到，p值為0.2371，也就是說，當$\alpha = 0.05$，不拒絕虛無假設，我們沒有足夠證據顯示，new這組資料的平均值大於conventional。

從四個檢定的結果比較我們可以發現，只有parametric bootstrap test的結果顯示，new這組資料的平均值大於conventional是顯著的，但前提條件是這些資料需要滿足假設，從其他三個檢定結果推斷，數據可能不太符合假設。

Appendix

R code

question 1

```
stack <- read.csv("stack.csv", row.names = "Obs")
X <- subset(stack, select = -y) %>%
  as.matrix()
y <- as.matrix(subset(stack, select = y))
Mylm <- function(X, y){
  coeflab <- colnames(X)
  X <- as.matrix(X)
  y <- as.matrix(y)

  qrstr <- qr(X)
  Q <- qr.Q(qrstr)
  R <- qr.R(qrstr)

  b <- solve(R) %*% t(Q) %*% y
  mse <- sqrt(sum((y - X %*% b)^2) / (nrow(X) - ncol(X)))
  sd_b <- sqrt(diag(mse * solve(t(X) %*% X))) * 2

  t <- b / sd_b
  pval <- 1 - pt(abs(t), df = nrow(X))
  res <- cbind(b, sd_b, t, pval)
  colnames(res) <- c("estimate", "std.error", "t-value", "p-value")
  rownames(res) <- coeflab

  return(res)
```

```
}
sumy <- lm(y ~ 0 + x1 + x2 + x3, data = stack) %>%
  summary()

sumy[["coefficients"]] %>%
  kable(digits = 4, caption = "Coefficients of lm{stats}")
Mylm(X, y) %>%
  kable(digits = 4, caption = "Coefficients of Mylm")
```

question 2

```
###### 2. ##########

## data preprocess. ######
data <- round(read.csv('mortality_male_1970_2005.csv'), 2)
row.names(data) <- data[, 1]
data <- data[, -1]

train <- data[1:31, ]
test <- data[32:36, ]

train_matrix <- as.matrix(log(train))
test_matrix <- as.matrix(log(test))

alphax <- NULL
for (i in 1:17) {
  alphax[i] <- mean(train_matrix[,i])
}

for (i in 1:17) {
  train_matrix[, i] <- train_matrix[, i] - alphax[i]
}

## predictions and calculate MAPE. ######
PreMape <- function(time_component, age_component, singular_value) {
  train_x <- 1970:2000
  lm_model <- lm(time_component*singular_value ~ train_x)
  beta <- as.matrix(unlist(lm_model$coefficients))
```

```
  pre_y <- beta[1] + beta[2]*matrix(2001:2005, ncol = 1)
  lnmxt <- pre_y %*% matrix(age_component, ncol = 17) +
    matrix(rep(alphax, time = 5), ncol = 17, byrow = TRUE)

  pre_mxt <- exp(lnmxt)
  test_real <- exp(test_matrix)
  mape <- mean(as.vector(abs((pre_mxt - test_real) / test_real)))
  print(mape)
}

## SVD. ######
train_svd <- svd(train_matrix)

cat("singular value of SVD:\n")
train_svd$d

singular_value_1 <- train_svd$d[1]

timecompo_svd <- train_svd$u[, 1]
agecompo_svd <- train_svd$v[, 1]

cat("\n")
cat("Lee carter model with SVD, MAPE:\n")
PreMape(timecompo_svd, agecompo_svd, singular_value_1)

## PCA. ######
train_pca <- prcomp(train_matrix)

cat("\n")
cat("Information of PCA:\n")
summary(train_pca)

timecompo_pca <- train_pca$x[,1]
agecompo_pca <- train_pca$rotation[,1]

cat("\n")
cat("Lee carter model with PCA, MAPE:\n")
PreMape(timecompo_pca, agecompo_pca, train_pca$sdev[1])
```

question 3

```
ddt <- c(585, 1002, 472, 493, 408, 690, 291)
thick <- c(0.1, 0.2, 0.5, 1.0, 1.5, 2.0, 3.0)
MyPermTest <- function(x, y){
  s <- sum(x * y)
  per_y <- gtools::permutations(length(y), length(y), y)
  per_s <- apply(per_y, 1, function(y) sum(x * y))

  n <- length(per_s)
  n_g <- sum(per_s > s)
  n_l <- sum(per_s < s)

  p <- min(n_g, n_l) / n
  return(c(n = n, "p-value" = p))
}
MyPermTest(ddt, thick) %>%
  t %>%
  kable(digits = 4, caption = "Correlation Test of MyPermTest")
cor.test(ddt, thick)
CorTestRep <- function(n, size, r, alpha){
  # cat(paste0("simulation time: ", n, ", sample size: ", size, ", corelation = ", r,
  #            "\nthe count table of test reject no correlation is:\n"))
  sig <- matrix(c(1, r,
                  r, 1), ncol = 2)
  a <- chol(sig)

  rej <- replicate(n = n,{
    x <- matrix(rnorm(size * 2), ncol = 2) %*% a
    u <- floor((pnorm(x)*10))
    p <- cor.test(u[, 1], u[, 2])[["p.value"]] < alpha
    sp <- cor.test(u[, 1], u[, 2], method = "spearman")[["p.value"]] < alpha
    pm <- perm.cor.test(u[, 1], u[, 2], num.sim = 1000)[["p.value"]] < alpha
    return(c(pearson = p, spearson = sp, permutation = pm))
  })
  n_rej <- apply(rej, 1, sum)
  return(n_rej)
}
set.seed(SEED)
```

```
rbind(
  `r=0.2` = CorTestRep(n = 1000, size = 10, r = 0.2, alpha = 0.05),
  `r=0.8` = CorTestRep(n = 1000, size = 10, r = 0.8, alpha = 0.05)) %>%
  kable(caption = "Reject Frequency Table of Correlation Test")
```

question 4

```
###### 4. ##########
MWW <- function(x, y) {
  n_1 <- length(x)
  n_2 <- length(y)
  r_1 <- rank(c(x, y))[1:n_1]
  r_2 <- rank(c(x, y))[-(1:n_1)]
  w_1 <- sum(r_1)
  w_2 <- sum(r_2)
  u_1 <- n_1*n_2 + n_1*(n_1 + 1) / 2 - w_1
  u_2 <- n_1*n_2 + n_2*(n_2 + 1) / 2 - w_2
  u <- min(u_1, u_2)
  e_u <- n_1*n_2 / 2
  var_u <- n_1*n_2*(n_1 + n_2 + 1) / 12
  z = (u - e_u) / sqrt(var_u)
  return(z)
}

CriticalPlot <- function (n_1, n_2) {
  critical <- NULL
  for (i in 1:10000) {
    set.seed(SEED)
    critical <- c(critical, MWW(rexp(n_1, rate = 1), rexp(n_2, rate = 1)))
    SEED <- SEED + 1
    }
  SEED <- 123
  plot <- critical %>% data.frame(critical) %>%
    ggplot(aes(x = critical))+
    geom_histogram(binwidth = 0.3)+
    theme(axis.title = element_text(size = 10))+
    labs(title = paste("n1 =", n_1, ", n2 =", n_2, "in 10,000 times"), x = 'Critical value', y = '
  return(plot)
```

```
}

plot_1 <- CriticalPlot(2, 2)
plot_2 <- CriticalPlot(5, 5)
plot_3 <- CriticalPlot(8, 8)
plot_4 <- CriticalPlot(10, 10)

grid.arrange(plot_1, plot_2, plot_3, plot_4, ncol = 2)
```

question 5

```
data("law82")
pop.cor <- cor(law82$LSAT, law82$GPA)
reps <- c(50, 100, 1000, 10000)
obsers <- c(10, 15, 20, 25)
print(paste("population correlation is", round(pop.cor, 4)))
BootStrapCor <- function(n, size, val = "sd"){
  obs <- law82[sample(nrow(law82), size = size, replace = FALSE), c("LSAT", "GPA")]
  cor_b <- replicate(n = n, {
    x <- obs[sample(rownames(obs), size = size, replace = TRUE), ]
    cor(x$LSAT, x$GPA)
  })
  cor_sd <- sqrt(var(cor_b))
  corr <- mean(cor_b)
  # return(c(correlation = corr, cor_sd = cor_sd))
  if (val == "corr"){
    return(corr)
  }else{
    return(cor_sd)
  }
}
set.seed(SEED)
res <- outer(reps, obsers,
             Vectorize(function(x, y) BootStrapCor(n = x, size = y)))
rownames(res) <- paste0("rep=", reps)
colnames(res) <- paste0("obs=", obsers)
res %>%
  kable(digits = 4,
```

```
        caption = "Std Deviation Matrix Given Replicate Time and Sample Observation")
set.seed(SEED)
res_cor <- outer(reps, obsers,
                 Vectorize(function(x, y) BootStrapCor(n = x, size = y, val = "corr")))
rownames(res_cor) <- paste0("rep=", reps)
colnames(res_cor) <- paste0("obs=", obsers)

res_cor %>%
  kable(digits = 4,
        caption = "Correlation Matrix Given Replicate Time and Sample Observation")
```

question 6

```
###### 6. ##########

conventional <- c(65, 79, 90, 75, 61, 85, 98, 80, 97, 75)
new <- c(90, 98, 73, 79, 84, 81, 98, 90, 83, 88)

## permutation test. ######
PermuteTest <- function(conventional, new, n_permutations) {
  permuted_diffs <- numeric(n_permutations)
  all_data <- c(conventional, new)
  observed_diff <- mean(new) - mean(conventional)

  for (i in 1:n_permutations) {
    set.seed(SEED)
    permuted_labels <- sample(c(rep("conventional", 10), rep("new", 10)))
    permuted_diffs[i] <- mean(all_data[permuted_labels == "new"]) - mean(all_data[permuted_labels
    SEED <- SEED + 1
  }
  SEED <- 123
  p_value <- sum(permuted_diffs >= observed_diff) / n_permutations
  cat("Observed mean difference:", observed_diff, "\n")
  cat("p-value:", p_value, "\n")
}

PermuteTest(conventional, new, 10000)
## non-parametric bootstrap test. ######
```

```r
t_1 <- NULL
for (i in 1:10000) {
  set.seed(SEED)
  sample_con <- sample(conventional, size = 10, replace = T)
  sample_new <- sample(new, size = 10, replace = T)
  m_1 <- mean(sample_new) - mean(sample_con)
  t_1 <- c(t_1, m_1)
  SEED <- SEED + 1
}

SEED <- 123
hist(t_1, main = "mean difference in 10,000 times", xlab = "Mean difference")

cat("95% Confidence Interval of Difference of Mean between new and convention:\n")
c((mean(new) - mean(conventional)) - qt(0.975, 18)*sd(t_1),
  (mean(new) - mean(conventional)) + qt(0.975, 18)*sd(t_1))
## parametric bootstrap test. ######
t_2 <- NULL
for (i in 1:10000) {
  set.seed(SEED)
  rnorm_con <- rnorm(10, mean = mean(conventional), sd = sd(conventional))
  rnorm_new <- rnorm(10, mean = mean(new), sd = sd(new))
  m_2 <- mean(rnorm_new) - mean(rnorm_con)
  t_2 <- c(t_2, m_2)
  SEED <- SEED + 1
}

SEED <- 123
hist(t_2, main = "mean difference in 10,000 times", xlab = "Mean difference")

cat("95% Confidence Interval of Difference of Mean between new and conventiol: \n")
c((mean(rnorm_new) - mean(rnorm_con)) - qt(0.975, 18)*sd(t_2),
  (mean(rnorm_new) - mean(rnorm_con)) + qt(0.975, 18)*sd(t_2))
t.test(new, conventional, paired = T, alternative = "two.sided")
```