# Homework 2

张子乐,3230104237

## 数据预处理

sample.xls 中有 2 份工作表，“Sheet1” 和”Sheet2”，发现”Sheet1” 在删除 HDL-C 的缺失值后与”Sheet2” 完全一致。“Sheet1” 的 894 行中有 622 行缺少 HDL-C 值，这无法使用插补的方法，而 HDL-C 值又是判断代谢综合征的重要变量，因此只能将这些行直接删除。于是选取”Sheet2” 进行接下来的数据分析，对”Sheet2” 进行数据预处理，将字符串格式转为数值格式。观察到 smoke 列和 drunk 列有多种不同的回答，统一改为“是”“否”。

“Sheet2” 剩下的 272 行中，有 82 行含有各种不同的缺失值, 若直接删除会损失约 30% 的数据，这是难以接受的，于是使用多重插补的方法，对于连续型变量使用 Predictive Mean Matching(pmm)，对二分类变量使用 Logistic Regression Imputation(logreg)，而对于性别变量则直接删除含缺失值的行。

```
clean <- function(df){
  df <- df[-1, ]  # 删除中文描述的行
  #print(str(df)) # 查看每列的类型
  # 要转换成数值格式的列
  numeric_cols <- c("age", "sbp", "dbp", "weight", "height", "FPG", "TG", "HDL-C")
  df <- df |>
    mutate(
      across(all_of(numeric_cols), parse_number)
    )

  # 观察到 smoke 列有多种回答
  df <- df %>%
    mutate(
      smoke = case_when(
        smoke == " 是" ~ " 是",
        smoke %in% c(" 否", " 已戒烟", " 戒烟 3 年", " 戒烟 2 个月") ~ " 否",
        TRUE ~ NA_character_
      ) %>%
        factor(levels = c(" 否", " 是"))
    )
```

```r
  # 同样，drunk 列有“是”“否”“无”三种回答
  df <- df %>%
    mutate(
      drunk = case_when(
        drunk == " 是" ~ " 是",
        drunk %in% c(" 否", " 无") ~ " 否",
        TRUE ~ NA_character_
      ) %>%
        factor(levels = c(" 否", " 是"))
    )

  #df <- na.omit(df) # 删除含有缺失值的行
  df <- df %>% filter(!is.na(gender))

  meth <- rep("pmm", ncol(df))
  names(meth) <- names(df)  # 把列名赋给 meth
  meth["smoke"] <- "logreg"
  meth["drunk"] <- "logreg"
  df <- suppressWarnings(mice(df, m = 10, maxit = 50, method =meth, printFlag = FALSE,seed=42)) #
  df <- complete(df, 1) # 选取第一个
  #print(nrow(df)) # 打印剩下的行数
  print(" 验证缺失值: ")
  print(colSums(is.na(df))) # 验证是否还存在缺失值
  print(" 每列的类型: ")
  print(str(df)) # 再次查看每列的类型
  return(df)
}


df2 <- read_excel("sample.xls",sheet = "Sheet2")
df2 <- clean(df2) # 处理 Sheet2
```

```
## [1] "验证缺失值: "
##     No gender    age    sbp    dbp weight height  smoke  drunk    FPG     TG
##      0      0      0      0      0      0      0      0      0      0      0
##  HDL-C
##      0
## [1] "每列的类型: "
## 'data.frame':    270 obs. of  12 variables:
```

```
##  $ No    : chr  "201208090010" "201208090011" "201208090014" "201208090015" ...
##  $ gender: chr  "男" "女" "男" "女" ...
##  $ age   : num  56 27 44 29 64 43 59 64 27 44 ...
##  $ sbp   : num  95 112 131 98 141 100 107 137 128 127 ...
##  $ dbp   : num  60 58 92 68 81 69 57 80 87 81 ...
##  $ weight: num  66.3 54.5 74.9 67.3 59.9 59.5 47.8 72.1 72.7 58.5 ...
##  $ height: num  171 172 170 158 159 ...
##  $ smoke : Factor w/ 2 levels "否","是": 2 1 1 1 1 1 1 1 1 1 ...
##  $ drunk : Factor w/ 2 levels "否","是": 2 1 2 2 2 1 1 1 2 1 ...
##  $ FPG   : num  5.01 4.44 5.95 5.6 5.93 5.68 6.45 5.34 5.41 5.34 ...
##  $ TG    : num  0.74 0.75 0.66 0.75 1.76 5.03 0.84 0.83 0.97 0.98 ...
##  $ HDL-C : num  1.79 2.17 1.65 1.72 1.91 1.2 1.53 1.81 1.82 1.57 ...
## NULL
```

```
df = df2 # 不妨选 df2 作为接下来分析的对象

# 查看修改后 smoke 和 drunk 的结果
table(df$smoke, useNA = "always")
```

```
##
##   否   是 <NA>
##  186   84    0
```

```
table(df$drunk, useNA = "always")
```

```
##
##   否   是 <NA>
##  165  105    0
```

后续需要用到 BMI，因此计算 BMI 作为新的变量

```
df <- df %>%
  mutate(BMI = round(weight / (height / 100)^2, 2)) # BMI = 体重/（身高的平方）
print(head(df$BMI)) # 打印前几行 BMI
```

```
## [1] 22.67 18.42 25.77 26.96 23.69 21.21
```

接下来简单的观察异常值和数据分布形态，在后面的数据分析中再决定如何处理：

```r
numeric_vars <- c("age", "sbp", "dbp", "weight", "height", "FPG", "TG", "HDL-C","BMI")
# 描述性统计分析
df %>%
  select(all_of(numeric_vars)) %>%
  summary()
```

```
##       age              sbp              dbp             weight
##  Min.   : 1.00   Min.   : 79   Min.   : 53.00   Min.   : 36.30
##  1st Qu.:39.00   1st Qu.:107   1st Qu.: 66.25   1st Qu.: 56.10
##  Median :47.00   Median :119   Median : 77.00   Median : 66.05
##  Mean   :46.37   Mean   :119   Mean   : 76.14   Mean   : 66.33
##  3rd Qu.:54.00   3rd Qu.:129   3rd Qu.: 84.00   3rd Qu.: 74.90
##  Max.   :76.00   Max.   :173   Max.   :107.00   Max.   :164.50
##      height           FPG               TG             HDL-C
##  Min.   : 53.2   Min.   : 3.910   Min.   : 0.300   Min.   :0.600
##  1st Qu.:160.5   1st Qu.: 5.040   1st Qu.: 0.940   1st Qu.:1.373
##  Median :166.5   Median : 5.300   Median : 1.455   Median :1.550
##  Mean   :165.4   Mean   : 5.550   Mean   : 1.945   Mean   :1.611
##  3rd Qu.:171.5   3rd Qu.: 5.737   3rd Qu.: 2.518   3rd Qu.:1.810
##  Max.   :183.0   Max.   :10.830   Max.   :10.200   Max.   :2.980
##       BMI
##  Min.   : 15.92
##  1st Qu.: 21.22
##  Median : 23.82
##  Mean   : 25.96
##  3rd Qu.: 26.14
##  Max.   :581.22
```

观察到这里的 BMI 出现了异常值 581，我们找到那一行, 发现体重和身高严重不匹配，于是将其删除。

```r
max_bmi <- max(df$BMI, na.rm = TRUE)
max_bmi_row <- df[df$BMI == max_bmi, ]
print(max_bmi_row)
```

```
##               No gender age sbp dbp weight height smoke drunk  FPG   TG HDL-C
## 232 201208150158     女  56 108  64  164.5   53.2    否    否 5.17 1.28  1.94
##        BMI
## 232 581.22
```

```r
# 事实上这里直接选择将 50 作为 BMI 的合理阈值
df <- df[df$BMI <= 50, ]
# df %>%
#   select(all_of(numeric_vars)) %>%
#   summary()
```

## 第一题

### (1) 相关性

```r
X <- df[, c("BMI", "FPG", "sbp", "dbp", "TG", "HDL-C")]
cor_matrix <- cor(X)
round(cor_matrix, 3)
```

```
##          BMI    FPG    sbp    dbp     TG  HDL-C
## BMI    1.000  0.258  0.314  0.381  0.368 -0.294
## FPG    0.258  1.000  0.160  0.247  0.370 -0.176
## sbp    0.314  0.160  1.000  0.799  0.188 -0.080
## dbp    0.381  0.247  0.799  1.000  0.320 -0.111
## TG     0.368  0.370  0.188  0.320  1.000 -0.335
## HDL-C -0.294 -0.176 -0.080 -0.111 -0.335  1.000
```

```r
res <- rcorr(as.matrix(X))
cor_coef <- res$r                  # 相关系数
cor_p <- res$P                     # p 值
#print(round(cor_coef, 3))
print(round(cor_p,3))
```

```
##       BMI   FPG   sbp   dbp    TG HDL-C
## BMI    NA 0.000 0.000 0.000 0.000 0.000
## FPG     0    NA 0.009 0.000 0.000 0.004
## sbp     0 0.009    NA 0.000 0.002 0.190
## dbp     0 0.000 0.000    NA 0.000 0.068
## TG      0 0.000 0.002 0.000    NA 0.000
## HDL-C   0 0.004 0.190 0.068 0.000    NA
```

从结果来看，在 95% 置信水平下，BMI 与 FPG、sbp、dbp、TG 具有显著的正相关性，与 HDL-C 具有显著的负相关性；FPG 与 sbp、dbp、TG 具有显著的正相关关系，与 HDL-C 具有显著的负相关关系；sbp

与 dbp、TG 具有显著的正相关关系；dbp 与 TG 具有显著的正相关关系；TG 与 HDL-C 具有显著的负相关关系。

综上，SBP 和 DBP 的正相关性最强，相关系数为 0.799；HDL-C 与其它变量均为负相关关系，与 TG 的负相关关系最强，相关系数为-0.335。

**（2）分析患代谢综合症的比例有没有性别差异，与吸烟或喝酒是否有关？**

首先判断每个样本是否超重、高血糖、高血压、空腹血，从而判断是否患代谢综合症。结果为 269 个样本中有 34 人患有代谢综合征。

```
df$Overweight <- ifelse(df$BMI >= 25, 1, 0) # 超重
df$Hyperglycemia <- ifelse(df$FPG >= 6.1, 1, 0) # 高血糖
df$HTN <- ifelse(df$sbp >= 140| df$dbp >= 90, 1, 0) # 高血压
df$FBG <- ifelse(df$TG >= 1.7 |
                   (df$`HDL-C` < 0.9 & df$gender == " 男") |
                   (df$`HDL-C` < 1 & df$gender == " 女"), 1, 0) # 空腹血
df$sick <- ifelse(rowSums(cbind(df$Overweight,
                                df$Hyperglycemia,
                                df$HTN,
                                df$FBG)) >= 3, 1, 0) # 是否患代谢综合征，1 为患，0 为不患
table(df$sick)
```

```
##
##   0   1
## 235  34
```

判断患代谢综合征的比例有没有性别差异,34 名患者中仅 4 名为女性。

```
table(df$gender, df$sick)
```

```
##
##         0   1
##    男 141  30
##    女  94   4
```

```
contingency_table <- table(df$gender, df$sick)
chisq_test <- chisq.test(contingency_table) # 卡方检验
#fisher.test(contingency_table)
print(chisq_test)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  contingency_table
## X-squared = 9.0421, df = 1, p-value = 0.002638
```

原假设 H0：患代谢综合征的比例没有显著的性别差异。备择假设 H1：患代谢综合征的比例有显著的性别差异。

通过卡方检验，$p = 0.002638 < 0.05$，拒绝原假设，选择备择假设，即患代谢综合征的比例有显著的性别差异。

接下来分析患代谢综合征与吸烟或喝酒是否有关。

```
sick_smoke <- table(df$smoke, df$sick)
print(sick_smoke)
```

```
##
##        0   1
##   否 166  19
##   是  69  15
```

```
chisq_test <- chisq.test(sick_smoke)
print(chisq_test)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  sick_smoke
## X-squared = 2.3636, df = 1, p-value = 0.1242
```

原假设 H0：患代谢综合征的比例与是否吸烟没有显著关联。备择假设 H1：患代谢综合征的比例与是否吸烟有显著关联。

通过卡方检验，$p = 0.1242 > 0.05$，接受原假设，即患代谢综合征的比例与是否吸烟没有显著关联。

```
sick_drunk <- table(df$drunk, df$sick)
print(sick_drunk)
```

```
##
##        0   1
##   否 151  13
##   是  84  21
```

```r
chisq_test <- chisq.test(sick_drunk)
print(chisq_test)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  sick_drunk
## X-squared = 7.3924, df = 1, p-value = 0.00655
```

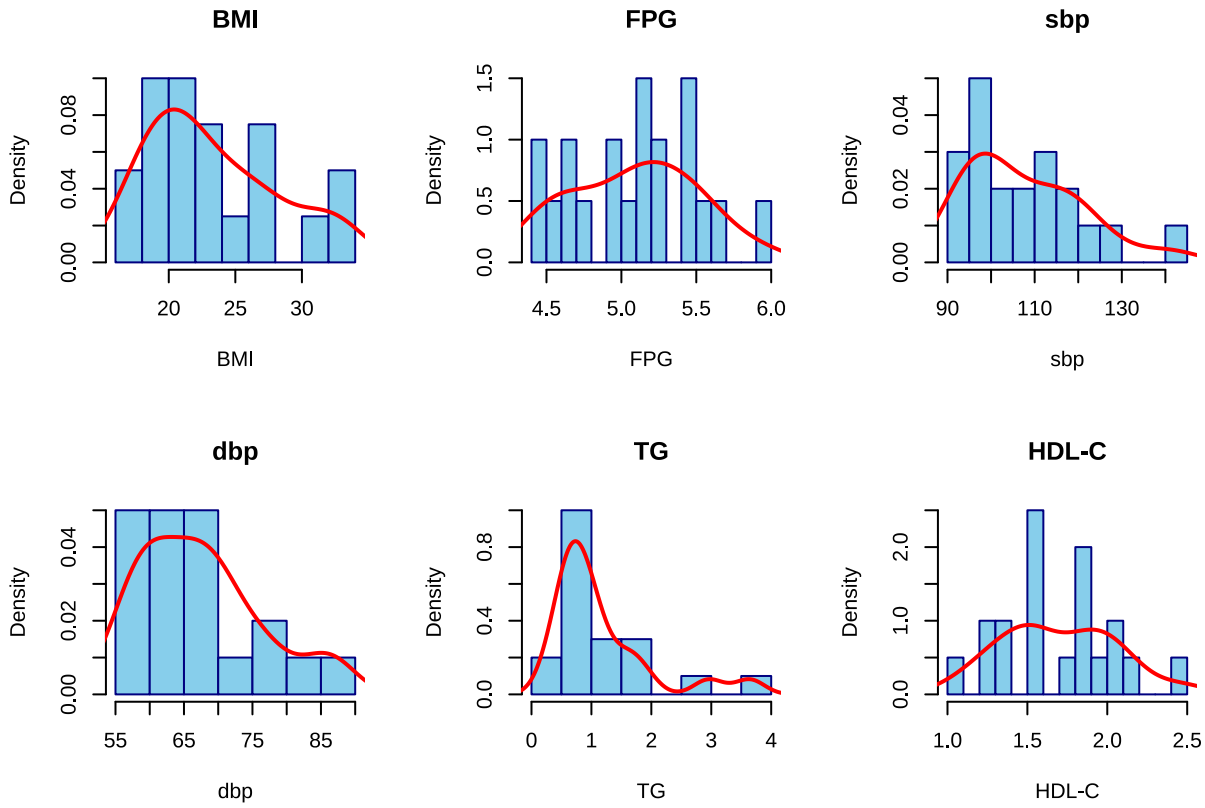
原假设 H0：患代谢综合征的比例与是否喝酒没有显著关联。备择假设 H1：患代谢综合征的比例与是否喝酒有显著关联。

通过卡方检验，$p = 0.00655 < 0.05$，拒绝原假设，选择备择假设，即患代谢综合征的比例与是否喝酒显著有关。

**（3）给出 20~30 年龄段，X 各个指标的分布情况**

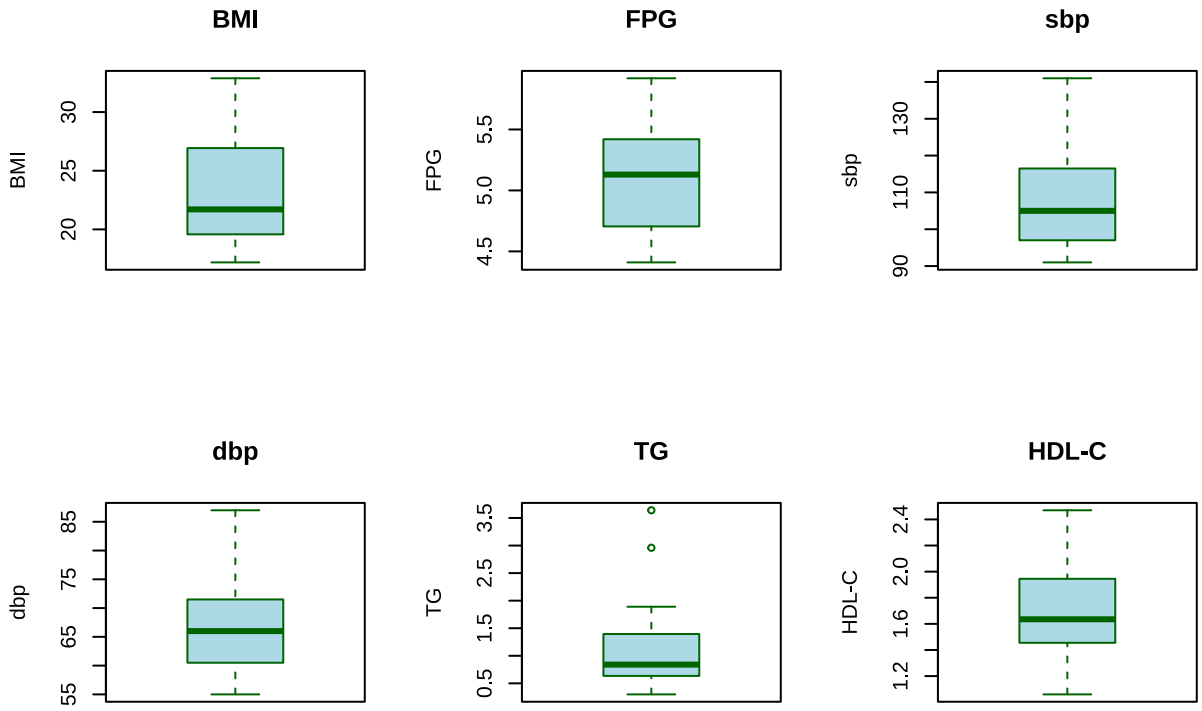
```r
df0 <- subset(df, age >= 20 & age <= 30)
X <- c("BMI", "FPG", "sbp", "dbp", "TG", "HDL-C")
df_X <- df0[,X]
#summary(df_X)

# 直方图
par(mfrow = c(2, 3))
for (var in X) {
  hist(df_X[[var]],
       breaks = 11,
       main = paste(var),
       xlab = var,
       col = "skyblue",
       border = "navy",
       freq = FALSE)
  lines(density(df_X[[var]], na.rm = TRUE), col = "red", lwd = 2)
}
```



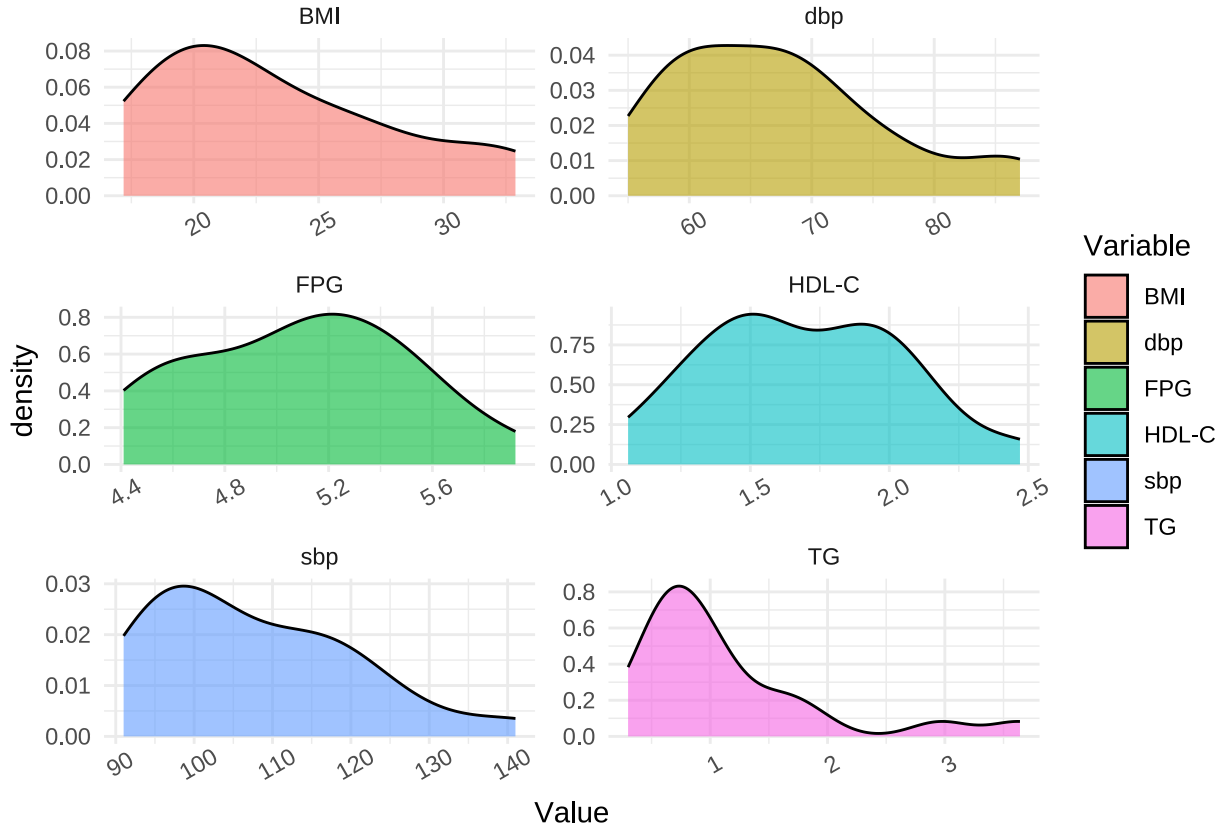

```r
par(mfrow = c(1, 1))

# 箱线图
par(mfrow = c(2, 3))
for (var in X) {
  boxplot(df_X[[var]], main = var, ylab = var, col = "lightblue", border = "darkgreen")
}
```




```r
par(mfrow = c(1, 1))

# 密度图
X_long <- df_X %>%
  pivot_longer(everything(), names_to = "Variable", values_to = "Value")
ggplot(X_long, aes(x = Value, fill = Variable)) +
  geom_density(alpha = 0.6) +
  facet_wrap(~ Variable, scales = "free", ncol = 2) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 30))
```




(4) 估计总体 $\mathbf{X}$

```
X <- df[, c("BMI", "FPG", "sbp", "dbp", "TG", "HDL-C")]
mean_vector <- colMeans(X, na.rm = TRUE) # 均值向量
cov_matrix <- cov(X, use = "complete.obs") # 协方差矩阵
cor_matrix <- cor(X, use = "complete.obs")# 相关矩阵
result <- list(均值向量 = round(mean_vector, 3), 协方差矩阵 = cov_matrix, 相关矩阵=cor_matrix)
print(result)
```

```
## $均值向量
##     BMI     FPG     sbp     dbp      TG   HDL-C
##  23.892   5.552 118.996  76.182   1.948   1.610
##
## $协方差矩阵
##               BMI         FPG          sbp          dbp         TG       HDL-C
## BMI    12.2358408  0.88530984  18.5085538  15.3548567  1.9214441 -0.34904626
## FPG     0.8853098  0.96189508   2.6427673   2.7961188  0.5409000 -0.05849166
## sbp    18.5085538  2.64276730 283.8395384 155.0342618  4.7278655 -0.45761139
```

```
## dbp    15.3548567  2.79611885 155.0342618 132.7316207  5.5048888 -0.43450480
## TG      1.9214441  0.54090000   4.7278655   5.5048888  2.2252074 -0.16942693
## HDL-C -0.3490463 -0.05849166  -0.4576114  -0.4345048 -0.1694269  0.11492013
##
## $相关矩阵
##               BMI        FPG          sbp        dbp          TG       HDL-C
## BMI     1.0000000  0.2580563  0.31406470  0.3810146  0.3682355 -0.29435270
## FPG     0.2580563  1.0000000  0.15994057  0.2474598  0.3697157 -0.17592682
## sbp     0.3140647  0.1599406  1.00000000  0.7987374  0.1881237 -0.08012395
## dbp     0.3810146  0.2474598  0.79873744  1.0000000  0.3203141 -0.11125239
## TG      0.3682355  0.3697157  0.18812366  0.3203141  1.0000000 -0.33504193
## HDL-C -0.2943527 -0.1759268 -0.08012395 -0.1112524 -0.3350419  1.00000000
```

## 第二题

```r
set.seed(42)

test <- function(true_params){
  nsim <- 500 # 模拟次数
  n_list <- c(50, 100, 500)    # 不同样本量

  # 误差项的分布
  distributions <- list(
    normal = function(n) rnorm(n, 0, 1),
    uniform = function(n) runif(n, -sqrt(3), sqrt(3)),
    t3 = function(n) rt(n, df = 3)
  )
  results <- data.frame()

  for (case in names(true_params)) {
    true_theta <- true_params[[case]]
    a_true <- true_theta[1]
    b_true <- true_theta[2]

    for (n in n_list) {
      for (distribution in names(distributions)) {
        func <- distributions[[distribution]]

        # 存储每次模拟的估计值
```

```r
a_hat <- numeric(nsim)
b_hat <- numeric(nsim)

for (s in 1:nsim) {

  Y <- numeric(n + 2)  # 包含 Y[-1], Y[0]
  Y[1:2] <- 0          # Y[-1] = Y[0] = 0

  # 生成误差项
  eps <- func(n)

  # 生成时间序列
  for (t in 3:(n+2)) {
    Y[t] <- a_true * Y[t-1] + b_true * Y[t-2] + eps[t-2]
  }

  Y_obs <- Y[3:(n+2)]           # Y1 ~ Yn
  Y_lag1 <- Y[2:(n+1)]          # Y0 ~ Yn-1
  Y_lag2 <- Y[1:n]              # Y-1 ~ Yn-2

  # 最小二乘估计: Y = X  +
  # 把 AR(2) 模型写成线性回归的形式, 即为 (a,b)
  X <- cbind(Y_lag1, Y_lag2)
  lm1 <-lm(Y_obs ~ X - 1)
  coef_est <- coef(lm1)
  a_hat[s] <- coef_est[1]
  b_hat[s] <- coef_est[2]

}

# 用均值作为 a 的估计值
mean_a <- mean(a_hat,na.rm = TRUE)
mean_b <- mean(b_hat,na.rm = TRUE)

# 计算 MSE
mse_a <- mean((a_hat - a_true)^2, na.rm = TRUE)
mse_b <- mean((b_hat - b_true)^2, na.rm = TRUE)

results <- rbind(results, data.frame(
```

```
          n = n,
          Distribution = distribution,
          Parameter = "a",
          True_Value = a_true,
          Estimation = mean_a,
          MSE = mse_a
        ))

        results <- rbind(results, data.frame(
          n = n,
          Distribution = distribution,
          Parameter = "b",
          True_Value = b_true,
          Estimation = mean_b,
          MSE = mse_b
        ))
      }
    }
  }
  print(results)
}

true_params1 <- list(case1 = c(a = 0.6, b = 0.3))
true_params2 <- list(case2 = c(a = 0.2, b = 0.3))

test(true_params1)
```

```
##      n Distribution Parameter True_Value Estimation         MSE
## a   50       normal         a        0.6  0.5808858 0.020515650
## b   50       normal         b        0.3  0.2742939 0.018726315
## a1  50      uniform         a        0.6  0.5952063 0.020277287
## b1  50      uniform         b        0.3  0.2577937 0.020568350
## a2  50           t3         a        0.6  0.5810219 0.020853852
## b2  50           t3         b        0.3  0.2716981 0.018973958
## a3 100       normal         a        0.6  0.5896103 0.008766209
## b3 100       normal         b        0.3  0.2864421 0.007894854
## a4 100      uniform         a        0.6  0.5958988 0.009289787
## b4 100      uniform         b        0.3  0.2821487 0.009656059
## a5 100           t3         a        0.6  0.5927909 0.008486000
```

```
## b5 100           t3         b        0.3  0.2853729 0.008392305
## a6 500       normal         a        0.6  0.6018178 0.001635586
## b6 500       normal         b        0.3  0.2944881 0.001952164
## a7 500      uniform         a        0.6  0.6011804 0.001785099
## b7 500      uniform         b        0.3  0.2930999 0.002004790
## a8 500           t3         a        0.6  0.5970449 0.001819230
## b8 500           t3         b        0.3  0.2960837 0.001634368
```

```
test(true_params2)
```

```
##      n Distribution Parameter True_Value Estimation         MSE
## a   50       normal         a        0.2  0.2056141 0.018995093
## b   50       normal         b        0.3  0.2577874 0.022294452
## a1  50      uniform         a        0.2  0.1967671 0.019934648
## b1  50      uniform         b        0.3  0.2570562 0.020216860
## a2  50           t3         a        0.2  0.1960710 0.014893926
## b2  50           t3         b        0.3  0.2690169 0.020640604
## a3 100       normal         a        0.2  0.1934229 0.008815068
## b3 100       normal         b        0.3  0.2829327 0.008375533
## a4 100      uniform         a        0.2  0.1992484 0.008894448
## b4 100      uniform         b        0.3  0.2822283 0.009393406
## a5 100           t3         a        0.2  0.2011868 0.009102115
## b5 100           t3         b        0.3  0.2798356 0.008256862
## a6 500       normal         a        0.2  0.2004436 0.001756640
## b6 500       normal         b        0.3  0.2946512 0.001869997
## a7 500      uniform         a        0.2  0.1975768 0.001795944
## b7 500      uniform         b        0.3  0.2980486 0.002048675
## a8 500           t3         a        0.2  0.2029806 0.001678056
## b8 500           t3         b        0.3  0.2968378 0.001820307
```

从结果可以看出，随着样本容量的增加，均方误差越来越小，估计量越来越接近真实值。以 a=0.6 为例，样本容量为 50 时 MSE 为 0.02 左右，样本容量为 100 时 MSE 为 0.01 左右，而当样本容量增加到 500，MSE 降低为 0.002 左右。而误差项取不同的分布对估计的 MSE 几乎没有显著的影响。