114-1-程式設計與統計軟體 期中考 答案卷

系級: _____ 學號: _____ 姓名:

注意事項:

1. 本答案卷上請務必於 Console 內複制「執行後的程式碼、結果及圖形」，於本答案卷貼上(Courier New, 10 點字，白底黑字)，不是只有程式碼，不是只有結果或報表。貼圖時，請注意圖形之大小，應適中; 圖內之文字數字應可識別。
2. 作答完請將此 word 檔存檔，檔名為「姓名-StatSoft-Midterm.docx」(更改成自己「姓名」)並上傳。
3. 帳號: statsoft，密碼: 1141.
4. 如果上傳網站出現空白頁，請將滑鼠移至「網址列」後，按「Enter」即可。若再不行，請清除快取 Cache 再試一次。若再不行，請換其它瀏覽器(IE/Edge/Firefox/Chrome)
5. 答案卷常見錯誤填答格式提醒 (請務必避免)(已事先於課程 FB 公告):
   https://hmwu.idv.tw/web/teaching/114-1/StatSoftware/R-WrongFormat_remark.pdf

---

```
# ex1
> #ex1
> dept_grade <- scan(what = "character", nmax = 1, quiet = TRUE)
1: 統計一
> student_id <- scan(what = "character", nmax = 1, quiet = TRUE)
1: 123456789
> student_name <- scan(what = "character", nmax = 1, quiet = TRUE)
1: 王小明
> cat("本人(", dept_grade, ",", student_id, ",", student_name, ")恪遵各項考試規則,若如違反,願受校方最嚴厲處罰,謹誓。\n", sep = "")
本人(統計一,123456789,王小明)恪遵各項考試規則,若如違反,願受校方最嚴厲處罰,謹誓。

#ex2
> set.seed(123)
> exp <- gl(2,10,labels = c('placebo' , 'treatment'))
> sample(exp)
 [1] treatment treatment treatment placebo   placebo   placebo   placebo
 [8] treatment placebo   placebo   treatment placebo   treatment treatment
[15] treatment placebo   treatment placebo   placebo   treatment
Levels: placebo treatment

# ex3
(a)
> print_triangle <- function(n) {
```

```
+   i <- 1
+   while (i <= n) {
+     cat(1:i, "\n")
+     i <- i + 1
+   }
+ }
>
> print_triangle(4)
1
1 2
1 2 3
1 2 3 4
(b)
> print_triangle(7)
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7

#ex4
(a)
> correct_answers <- c("B", "D", "B", "D", "D", "A", "C", "D", "C", "B")
> answers_df <- read.csv("./1141-StatSoft-Midterm/data/answer.csv" , skip = 1)
> print(head(answers_df, 5))
  Student V1 V2 V3 V4 V5 V6 V7 V8 V9 V10
1      s1  C  D  D  A  D  A  B  C  C   B
2      s2  B  D  B  D  D  A  C  D  B   B
3      s3  B  A  A  B  D  A  C  B  C   B
4      s4  B  D  B  A  B  C  C  D  C   B
5      s5  B  D  D  D  A  C  C  D  A   B
> print(tail(answers_df, 5))
    Student V1 V2 V3 V4 V5 V6 V7 V8 V9 V10
188    s188  D  D  A  D  D  C  A  D  D   A
189    s189  B  D  B  D  D  A  C  D  C   B
190    s190  B  D  B  D  D  D  C  D  A   B
191    s191  B  A  B  D  D  A  C  C  D   B
192    s192  B  D  B  D  D  A  C  D  C   B
```

(B)

```
> student_answers <- c("A", "D", "B", "D", "B", "A", "B", "D", "C", "B")
> correct_flags <- (student_answers == correct_answers)
> total_score <- sum(correct_flags) * 10
> cat("該學生所得之總分為:" , total_score , "\n")
該學生所得之總分為: 70
```

(C)

```
> calculate_score <- function(row_answers) {
+   correct_flags <- (row_answers == correct_answers)
+   score <- sum(correct_flags) * 10
+   return(score)
+ }
>
> student_answer_data <- answers_df[, 2:11]
> all_scores <- apply(student_answer_data, 1, calculate_score)
> answers_df$score <- all_scores
> score.table <- table(answers_df$score)
> print(score.table)

  0  10  20  30  40  50  60  70  80  90 100
  3  10   9  11  19  23  28  40  30  12   7
```

(d)

```
> score_80th_percentile <- quantile(answers_df$score, probs = 0.80)
> score_20th_percentile <- quantile(answers_df$score, probs = 0.20)
>
> high_group_df <- answers_df[answers_df$score >= score_80th_percentile, ]
>
> low_group_df <- answers_df[answers_df$score <= score_20th_percentile, ]
>
> high_group_count <- nrow(high_group_df)
> low_group_count <- nrow(low_group_df)
>
> cat("高分組人數:", high_group_count, "\n")
高分組人數: 49
> cat("低分組人數:", low_group_count, "\n")
低分組人數: 52
```

(E)

```
> high_group_answers <- high_group_df[, 2:11]
> low_group_answers <- low_group_df[, 2:11]
> P_H <- numeric(10)
> P_L <- numeric(10)
> for (i in 1:10) {
+  correct_ans <- correct_answers[i]
+  P_H[i] <- sum(high_group_answers[, i] == correct_ans) / high_group_count
+  P_L[i] <- sum(low_group_answers[, i] == correct_ans) / low_group_count
+ }
> cat("高分組每題答對百分比 (P_H):" ,'\n' ,  P_H , '\n')
高分組每題答對百分比 (P_H):
 0.877551 0.877551 0.7755102 0.7959184 0.877551 0.8163265 0.9591837 0.755102 0.8571429
0.9387755
> cat("低分組每題答對百分比 (P_L):" ,'\n' ,  P_L , '\n')
低分組每題答對百分比 (P_L):
 0.3076923 0.2692308 0.2884615 0.3076923 0.2884615 0.1923077 0.2115385 0.3269231
0.1730769 0.2692308

(F)
> P <- (P_H + P_L) / 2
> D <- P_H - P_L
>
> cat("每題難度 (P):" ,'\n' ,  P , '\n')
每題難度 (P):
 0.5926217 0.5733909 0.5319859 0.5518053 0.5830063 0.5043171 0.5853611 0.5410126
0.5151099 0.6040031
> cat("每題難度 (D):" ,'\n' ,  D , '\n')
每題難度 (D):
 0.5698587 0.6083203 0.4870487 0.4882261 0.5890895 0.6240188 0.7476452 0.428179
0.6840659 0.6695447

#EX5
(a)
> my_lm <- function(x, y) {
+   n <- length(x)
+   sum_xy <- sum(x * y)
+   sum_x <- sum(x)
+   sum_y <- sum(y)
+   sum_x_sq <- sum(x^2)
+   numerator_b1 <- (n * sum_xy) - (sum_x * sum_y)
```

```
+   denominator_b1 <- (n * sum_x_sq) - (sum_x^2)
+   beta_1_hat <- numerator_b1 / denominator_b1
+   y_bar <- mean(y)
+   x_bar <- mean(x)
+   beta_0_hat <- y_bar - (beta_1_hat * x_bar)
+   result <- list(intercept = beta_0_hat, slope = beta_1_hat)
+   return(result)
+ }
>
(B)
> x <- iris[, 1]
> y <- iris[, 2]
>
> print(my_lm(x, y))
$intercept
[1] 3.418947

$slope
[1] -0.0618848
> builtin_lm_result <- lm(y ~ x)
>
> cat("my_lm() intercept:", my_lm_result$intercept, "\n")
my_lm() intercept: 3.418947
> cat("lm() intercept:   ", coef(builtin_lm_result)[1], "\n")
lm() intercept:   3.418947
> cat("my_lm() slope:    ", my_lm_result$slope, "\n")
my_lm() slope:    -0.0618848
> cat("lm() slope:      ", coef(builtin_lm_result)[2], "\n")
lm() slope:       -0.0618848



#EX6
(a)
> bag <- rep(c("white", "red"), times = c(6, 4))
> set.seed(123456)
> draw_1 <- sample(bag, 3, replace = FALSE)
> cat("實驗一次的抽球結果:" , draw_1 , "\n")
實驗一次的抽球結果: red white white
> counts_1 <- table(draw_1)
> print(counts_1)
```

```
draw_1
  red white
    1    2
#(b)
> n_reps_10 <- 10
> draw_results_df <- data.frame(白球 = integer(n_reps_10),
+                           紅球 = integer(n_reps_10),
+                           row.names = 1:n_reps_10)
> for (i in 1:n_reps_10) {
+   draw_i <- sample(bag, 3, replace = FALSE)
+   counts_i <- table(factor(draw_i, levels = c("white", "red")))
+   draw_results_df$白球[i] <- counts_i["white"]
+   draw_results_df$紅球[i] <- counts_i["red"]
+ }
> draw_results_df
   白球 紅球
1    1   2
2    1   2
3    2   1
4    2   1
5    2   1
6    2   1
7    3   0
8    2   1
9    1   2
10   1   2

(C)
> n_sims_1000 <- 1000
> all_results <- replicate(n_sims_1000, {
+   draw <- sample(bag, 3, replace = FALSE)
+   n_white <- sum(draw == "white")
+   n_white == 2
+ })
> prob_estimate <- mean(all_results)
> print(prob_estimate)
[1] 0.493
```