Họ và tên: Vũ Tiến Đạt

MSV: 23000111

------------------------------------------------------

# Câu 1:

> install.packages("MASS")

Error in install.packages : Updating loaded packages

> library(MASS)

>

> data(hills)

> View(hills)

> head(hills)

dist climb time

Greenmantle 2.5 650 16.083

Carnethy 6.0 2500 48.350

Craig Dunain 6.0 900 33.650

Ben Rha 7.5 800 45.600

Ben Lomond 8.0 3070 62.267

Goatfell 8.0 2866 73.217

>

> #a. Tìm giá trị trung bình và độ lệch tiêu chuẩn của độ dài đường đua,

> #độ cao đạt được và thời gian chạy của người chiến thắng

> ?colMeans

> print("Giá trị trung bình")

[1] "Giá trị trung bình"

> colMeans(hills)

dist climb time

7.528571 1815.314286 57.875714

> apply(hills, 2, mean)

dist climb time

7.528571 1815.314286 57.875714

> print("Độ lệch tiêu chuẩn")

[1] "Độ lệch tiêu chuẩn"

> # a <- hills$dist

> # sd(a)

> ?apply

> apply(hills, 2, sd)

dist climb time

5.523936 1619.150536 50.040716

>

> #b Các biến trong bộ dữ liệu có tuân theo phân phối chuẩn hay không ?

> par(mfrow=c(3,2))

> hist(hills$dist, main = "Histogram dist", xlab = "Distance")

> qqnorm(hills$dist)

> qqline(hills$dist, col = "red")

>

> hist(hills$climb, main = "Histogram climb", xlab = "Climb")

> qqnorm(hills$climb)

> qqline(hills$climb, col = "red")

>

> hist(hills$time, main = "Histogram time", xlab = "time")

>

> qqnorm(hills$time)

> qqline(hills$time, col = "red")

>

> dev.off()

null device

1

>

> # Kiểm định Shapiro-Wilk

> shapiro.test(hills$dist)

Shapiro-Wilk normality test

data: hills$dist

W = 0.74904, p-value = 2.362e-06

> print("Do p-value < 0.05 nên bác bỏ H0 => Dist không tuân theo phân phối chuẩn")

[1] "Do p-value < 0.05 nên bác bỏ H0 => Dist không tuân theo phân phối chuẩn"

> shapiro.test(hills$climb)

Shapiro-Wilk normality test

data: hills$climb

W = 0.79245, p-value = 1.456e-05

> print("Do p-value < 0.05 nên bác bỏ H0 => Climb không tuân theo phân phối chuẩn")

[1] "Do p-value < 0.05 nên bác bỏ H0 => Climb không tuân theo phân phối chuẩn"

> shapiro.test(hills$time)

Shapiro-Wilk normality test

data: hills$time

W = 0.73673, p-value = 1.458e-06

> print("Do p-value < 0.05 nên bác bỏ H0 => Time không tuân theo phân phối chuẩn")

[1] "Do p-value < 0.05 nên bác bỏ H0 => Time không tuân theo phân phối chuẩn"

>

>

> # Dùng hàm Pairs để vẽ ma trận biểu đồ tán xạ cho bộ dữ liệu.

> pairs(hills, main = "Scatterplot matrix of hills dataset")

>

> print("Nhận xét: dist ~ time, climb ~ time")

[1] "Nhận xét: dist ~ time, climb ~ time"

>

> model <- lm(time ~ dist + climb, data = hills)

> resid\_sq <- residuals(model)^2

>

> shapiro.test(resid\_sq)

Shapiro-Wilk normality test

data: resid\_sq

W = 0.27219, p-value = 5.258e-12

> qqnorm(resid\_sq); qqline(resid\_sq, col="red")

> print("Do p-value < 0.05 nên bác bỏ H0 => bình phương hiệu số không tuân theo phân phối chuẩn")

[1] "Do p-value < 0.05 nên bác bỏ H0 => bình phương hiệu số không tuân theo phân phối chuẩn"

>

> print("------------------------")

[1] "------------------------"

> #Câu 2:

> head(iris)

Sepal.Length Sepal.Width Petal.Length Petal.Width Species

1 5.1 3.5 1.4 0.2 setosa

2 4.9 3.0 1.4 0.2 setosa

3 4.7 3.2 1.3 0.2 setosa

4 4.6 3.1 1.5 0.2 setosa

5 5.0 3.6 1.4 0.2 setosa

6 5.4 3.9 1.7 0.4 setosa

> #a. Tìm giá trị trung bình và độ lệch tiêu chuẩn của độ dài, độ rộng cánh hoa và đài hoa

> # Giá trị trung bình

> colMeans(iris[,1:4])

Sepal.Length Sepal.Width Petal.Length Petal.Width

5.843333 3.057333 3.758000 1.199333

> # Độ lệch chuẩn

> apply(iris[,1:4], 2, sd)

Sepal.Length Sepal.Width Petal.Length Petal.Width

0.8280661 0.4358663 1.7652982 0.7622377

>

> #b. Các biến trong bộ dữ liệu có tuân theo phân phối chuẩn không?

> shapiro.test(iris$Sepal.Length)

Shapiro-Wilk normality test

data: iris$Sepal.Length

W = 0.97609, p-value = 0.01018

> print("Do p-value < 0.05 nên bác bỏ H0 => Độ dài cánh hoa không tuân theo phân phối chuẩn")

[1] "Do p-value < 0.05 nên bác bỏ H0 => Độ dài cánh hoa không tuân theo phân phối chuẩn"

> shapiro.test(iris$Sepal.Width)

Shapiro-Wilk normality test

data: iris$Sepal.Width

W = 0.98492, p-value = 0.1012

> print("Do p-value < 0.05 nên bác bỏ H0 => Độ rộng cánh hoa không tuân theo phân phối chuẩn")

[1] "Do p-value < 0.05 nên bác bỏ H0 => Độ rộng cánh hoa không tuân theo phân phối chuẩn"

> shapiro.test(iris$Petal.Length)

Shapiro-Wilk normality test

data: iris$Petal.Length

W = 0.87627, p-value = 7.412e-10

> print("Do p-value < 0.05 nên bác bỏ H0 => Độ dài đài hoa không tuân theo phân phối chuẩn")

[1] "Do p-value < 0.05 nên bác bỏ H0 => Độ dài đài hoa không tuân theo phân phối chuẩn"

> shapiro.test(iris$Petal.Width)

Shapiro-Wilk normality test

data: iris$Petal.Width

W = 0.90183, p-value = 1.68e-08

> print("Do p-value < 0.05 nên bác bỏ H0 => Độ rộng đài hoa không tuân theo phân phối chuẩn")

[1] "Do p-value < 0.05 nên bác bỏ H0 => Độ rộng đài hoa không tuân theo phân phối chuẩn"

>

> #c. Vẽ đồ thị hàm mật độ và hàm phân phối của các biến có phân phối chuẩn trên cùng

> #một khung hình. Đặt tên đồ thị, tên và độ dài các trục phù hợp.

> print("Không biến nào có phân phối chuẩn")

[1] "Không biến nào có phân phối chuẩn"

> par(mfrow=c(1,2))

> plot(density(iris$Sepal.Length),

+ main="Hàm mật độ",

+ xlab="Sepal.Length", ylab="Density")

>

> plot(ecdf(iris$Sepal.Length),

+ main="Hàm phân phối",

+ xlab="Sepal.Length", ylab="F(x)")

>

> #d. Với mức ý nghĩa 5%, có thể nói độ rộng trung bình của đài hoa nhỏ hơn 3.5 không?

> t.test(iris$Sepal.Width, mu = 3.5, alternative = "less")

One Sample t-test

data: iris$Sepal.Width

t = -12.439, df = 149, p-value < 2.2e-16

alternative hypothesis: true mean is less than 3.5

95 percent confidence interval:

-Inf 3.116237

sample estimates:

mean of x

3.057333

>

> print("Do p-value < 0.05 nên bác bỏ H0 -> Độ rộng trung bình của đài hoa nhỏ hơn 3.5")

[1] "Do p-value < 0.05 nên bác bỏ H0 -> Độ rộng trung bình của đài hoa nhỏ hơn 3.5"

>

> #e. Tìm a, biết xác suất để độ rộng đài hoa nhỏ hơn a là 70%.

> a <- quantile(iris$Sepal.Width, 0.7)

> a

70%

3.2

>

> #f. Độ rộng trung bình của đài hoa của loài setosa và versicolor có thực sự khác nhau

> # không? Mức ý nghĩa 1%.

>

> setosa <- iris$Sepal.Width[iris$Species=="setosa"]

> versicolor <- iris$Sepal.Width[iris$Species=="versicolor"]

>

> t.test(setosa, versicolor, alternative = "two.sided", var.equal = TRUE, conf.level = 0.99)

Two Sample t-test

data: setosa and versicolor

t = 9.455, df = 98, p-value = 1.845e-15

alternative hypothesis: true difference in means is not equal to 0

99 percent confidence interval:

0.475184 0.840816

sample estimates:

mean of x mean of y

3.428 2.770

> print("Do p-value < 0.05 => bác bỏ H0, Độ rộng trung bình đài hoa của setosa và versicolor là khác nhau với mức ý nghĩa 1%")

[1] "Do p-value < 0.05 => bác bỏ H0, Độ rộng trung bình đài hoa của setosa và versicolor là khác nhau với mức ý nghĩa 1%"

>

> print("-------------------------")

[1] "-------------------------"

> # Câu 3:

> set.seed(12)

> n <- 20

> m <- 30

> kv <- 158

> dlc <- sqrt(8)

>

> # Sinh 30 mẫu ngẫu nhiên,

> data <- replicate(m, rnorm(n, mean = kv, sd = dlc), simplify = FALSE)

> data

[[1]]

[1] 153.8123 162.4609 155.2939 155.3978 152.3498 157.2298 157.1081 156.2230 157.6989 159.2106 155.8003 154.3403 155.7951

[14] 158.0338 157.5689 156.0103 161.3627 158.9631 159.4339 157.1704

[[2]]

[1] 158.6326 163.6772 160.8623 157.1445 155.1002 157.2437 157.4368 158.3709 158.4124 159.0241 159.9063 163.8606 156.4697

[14] 154.9722 156.9465 156.6278 158.7772 156.6437 160.2574 155.1590

[[3]]

[1] 158.2969 154.7304 159.6352 153.4869 157.1274 159.2713 155.2365 158.5374 160.0689 156.6067 157.8793 157.6813 159.2921

[14] 163.7144 155.0276 160.0779 159.5252 154.2827 157.2928 158.8887

[[4]]

[1] 159.1499 160.8126 160.4205 158.5576 160.3598 160.3951 163.5270 151.9210 160.7467 161.2387 156.5139 158.7080 156.7855

[14] 157.4838 157.7078 156.2072 154.4049 156.9140 159.4616 157.4966

[[5]]

[1] 158.0120 154.3964 157.4283 161.2936 157.9339 160.5375 157.5001 161.1500 156.4673 155.2751 159.0648 155.2149 160.5387

[14] 158.3656 160.9238 157.0319 159.2792 156.0350 157.3240 155.1509

[[6]]

[1] 163.4496 158.1455 163.0927 156.4402 158.3028 160.2320 157.3091 159.0866 154.6929 162.2943 158.2064 156.5075 156.5164

[14] 160.1126 154.6556 154.1683 157.3996 157.7469 158.7270 161.4062

[[7]]

[1] 157.5078 159.9522 159.1692 153.9540 154.3240 161.9693 157.3796 164.0316 152.8142 157.6874 160.1658 158.9150 155.9200

[14] 161.2406 157.2130 160.9892 159.0395 160.3971 154.4282 159.4941

[[8]]

[1] 161.0813 156.0216 162.0193 162.0702 154.3363 155.4627 152.6690 158.0778 155.4334 153.3400 158.7137 159.8167 157.0187

[14] 156.5426 159.1262 160.6971 158.7359 152.2013 158.8145 159.0237

[[9]]

[1] 153.2292 161.5275 152.9931 158.6303 156.4146 156.5628 155.3446 162.6263 156.1536 160.3737 156.2701 156.9486 156.5427

[14] 157.0333 159.9344 160.2581 155.7378 158.6495 162.7901 155.8264

[[10]]

[1] 153.6353 154.2648 156.0885 156.9620 156.5465 155.2868 157.6559 159.8836 161.5036 161.3487 162.2380 157.1823 163.4410

[14] 159.4666 159.3480 158.8559 160.3228 159.1104 153.9258 155.5424

[[11]]

[1] 149.5530 157.8570 161.6756 154.1777 155.3136 162.7690 158.6910 154.0739 160.5092 158.0797 158.7894 157.7863 158.3480

[14] 155.0827 158.2537 161.7391 157.1298 162.1820 156.6720 155.5510

[[12]]

[1] 160.8264 154.9117 160.4144 156.2264 159.1899 156.8803 156.8976 159.0849 162.7033 154.8091 157.6293 156.8209 159.8452

[14] 153.3393 156.0865 154.3076 156.8534 161.8441 156.1297 155.4132

[[13]]

[1] 158.3628 157.3624 159.8953 159.5211 157.8437 159.4202 156.1891 163.0993 160.8248 160.4826 156.7135 157.7500 158.1122

[14] 159.5473 157.3617 159.1134 158.3495 158.4562 154.3943 154.6704

[[14]]

[1] 158.3582 162.9248 156.3419 156.7408 157.7541 158.7915 158.3027 157.9830 162.9331 157.8264 150.8317 159.1116 159.8888

[14] 157.4967 156.8792 156.5290 160.4853 156.4745 161.8586 162.7135

[[15]]

[1] 157.3312 159.4017 155.5165 157.5701 161.2167 151.5426 162.6743 161.2804 155.5055 160.6737 158.2762 149.6913 157.9427

[14] 159.1033 152.5046 156.1950 157.1298 160.0640 155.3003 160.4269

[[16]]

[1] 153.8100 155.3686 155.1246 157.4013 162.6750 160.0514 157.3431 159.6532 164.8319 158.0585 158.9542 158.4495 160.6181

[14] 160.3746 154.9059 157.4319 154.9556 158.7352 155.7665 152.3351

[[17]]

[1] 159.6751 156.8423 153.8726 158.4246 155.8324 159.8016 158.2031 159.4727 159.9931 162.4141 161.1491 156.3904 155.4027

[14] 158.0001 157.6955 152.4709 157.1125 156.9679 157.2388 159.4715

[[18]]

[1] 156.8249 155.1587 158.4716 158.7978 161.5172 157.4282 153.5703 158.6319 154.4362 162.4394 157.1378 153.7825 158.6581

[14] 159.6000 157.0768 160.6119 154.7749 155.9121 155.0147 159.4092

[[19]]

[1] 155.5567 155.7784 159.4000 157.5608 159.7708 155.3461 157.9126 158.6195 158.3805 160.3921 157.4091 158.0945 160.2976

[14] 164.7913 159.8269 159.5318 159.9870 154.7805 158.0175 154.4356

[[20]]

[1] 154.4613 156.7587 158.4802 156.2301 162.6127 153.6271 157.4744 160.9014 157.7309 157.9647 157.9157 165.3379 161.7797

[14] 153.7153 161.3727 161.1456 157.8115 155.7222 155.8312 157.2956

[[21]]

[1] 158.2464 154.1302 154.0683 158.8634 158.4395 158.6066 157.4902 156.2956 162.9215 161.0685 161.7008 155.2857 159.3597

[14] 153.5708 161.3907 160.8315 158.0792 157.1749 161.0179 157.2527

[[22]]

[1] 159.4985 155.3366 153.7229 158.6897 157.6956 157.1382 155.3104 158.6286 152.5293 155.1377 159.7801 158.5266 158.8609

[14] 158.9541 158.9823 154.5571 156.0694 152.9154 157.8007 157.5776

[[23]]

[1] 156.6791 159.8493 156.2678 161.2867 158.2808 160.0558 160.0180 159.3349 157.2955 159.4225 160.0857 151.7970 156.6589

[14] 158.6079 159.6993 158.2796 159.8470 156.5242 157.7659 158.8023

[[24]]

[1] 154.6486 160.3897 156.8830 157.5146 162.2118 158.8924 156.8520 158.4173 156.5195 158.8859 158.9335 156.8068 158.7755

[14] 156.9959 161.0299 156.2855 165.5448 163.6647 152.3219 156.3869

[[25]]

[1] 158.7897 157.5801 163.6326 154.9205 160.4262 157.4187 161.0853 160.1525 154.0243 157.8423 158.4190 159.7600 151.4202

[14] 156.5221 156.2925 156.5581 159.8885 152.0091 158.1294 156.7740

[[26]]

[1] 153.6179 160.4648 158.1380 158.4816 158.2112 158.8133 156.7722 157.5888 155.2934 151.1673 158.8584 154.2811 157.4718

[14] 157.2661 156.1123 160.3149 151.5241 158.1277 159.0381 158.1209

[[27]]

[1] 155.8986 157.3156 154.1083 154.1651 159.2576 161.4003 159.2828 159.2181 157.6380 154.4087 161.0499 154.7610 156.5248

[14] 157.2016 157.2170 159.0149 152.4558 158.5623 158.0923 158.2078

[[28]]

[1] 158.9007 157.3136 160.4345 158.4544 155.7163 157.2534 150.9491 160.1226 154.8641 155.0237 158.6481 157.3807 160.6958

[14] 154.4575 161.4414 160.7501 158.0721 155.0402 161.6952 158.9015

[[29]]

[1] 156.2246 157.2987 155.8851 158.4420 155.5859 160.2673 158.9601 156.7279 154.5139 156.0715 158.5928 163.9892 162.6179

[14] 158.2796 157.8841 157.2402 156.9738 156.1300 162.0482 155.1535

[[30]]

[1] 152.8128 160.1195 158.8996 157.1861 157.6582 160.0529 157.6021 156.8432 154.3804 157.2780 159.1453 154.5603 161.7591

[14] 157.9137 156.4900 156.7070 163.0594 157.4445 154.3159 157.7982

>

> # Mẫu thứ bao nhiêu có khoảng tin cậy rộng nhất/hẹp nhất?

> ktc <- function(x, alpha = 0.02) {

+ xbar <- mean(x)

+ s <- sd(x)

+ n <- length(x)

+ error <- qt(1 - alpha/2, df = n-1) \* s / sqrt(n)

+ c(lower = xbar - error, upper = xbar + error, mean = xbar)

+ }

>

> cis <- t(sapply(data, ktc))

> print(cis)

lower upper mean

[1,] 155.6709 158.4555 157.0632

[2,] 156.8565 159.6960 158.2763

[3,] 156.4371 159.2289 157.8330

[4,] 156.9405 159.9408 158.4406

[5,] 156.7136 159.1787 157.9462

[6,] 156.8945 159.9547 158.4246

[7,] 156.6594 159.9998 158.3296

[8,] 155.8607 159.2595 157.5601

[9,] 156.0965 159.2882 157.6923

[10,] 156.5022 159.7587 158.1304

[11,] 155.8724 159.5510 157.7117

[12,] 156.0293 158.9920 157.5107

[13,] 157.2133 159.5336 158.3735

[14,] 156.8941 160.1285 158.5113

[15,] 155.5215 159.4132 157.4673

[16,] 156.0997 159.5847 157.8422

[17,] 156.4624 159.1807 157.8215

[18,] 156.0090 158.9164 157.4627

[19,] 156.9144 159.6745 158.2945

[20,] 156.4418 159.9752 158.2085

[21,] 156.7520 159.8274 158.2897

[22,] 155.5988 158.1724 156.8856

[23,] 157.1357 159.5201 158.3279

[24,] 156.6593 160.1368 158.3980

[25,] 155.8865 159.2781 157.5823

[26,] 155.5059 158.4605 156.9832

[27,] 155.9272 158.6509 157.2890

[28,] 156.2177 159.3938 157.8058

[29,] 156.4885 159.4001 157.9443

[30,] 156.1749 159.0278 157.6013

>

> widths <- cis[, "upper"] - cis[, "lower"]

> cat("Mẫu rộng nhất:", which.max(widths), "\n")

Mẫu rộng nhất: 15

> cat("Mẫu hẹp nhất:", which.min(widths), "\n")

Mẫu hẹp nhất: 13

>

>

> # c. Mẫu có mean lớn nhất

> max\_idx <- which.max(cis[, "mean"])

> cat("Mẫu thứ", max\_idx, "có giá trị trung bình lớn nhất\n")

Mẫu thứ 14 có giá trị trung bình lớn nhất

>

> #d.

> # Vẽ các khoảng tin cậy ở ý (i) thỏa mãn các khoảng đó chứa giá trị trung bình của mẫu

> # gồm toàn bộ các quan sát trong 30 mẫu ngẫu nhiên được sinh ban đầu. Ký hiệu hai

> # đầu mút mỗi khoảng bằng chấm tròn đỏ, độ dài khoảng có màu xanh lá cây; đặt tên

> # đồ thị là "Khoảng tin cậy cho GTTB của các mẫu ngẫu nhiên", trục Ox là "Khoảng

> # tin cậy cho GTTB", trục Oy là "Các mẫu ngẫu nhiên".

>

> plot(NULL, xlim = range(cis[, c("lower","upper")]), ylim = c(1, m),

+ xlab = "Khoảng tin cậy cho GTTB", ylab = "Các mẫu ngẫu nhiên",

+ main = "Khoảng tin cậy cho GTTB của các mẫu ngẫu nhiên")

>

> for (i in 1:m) {

+ segments(cis[i, "lower"], i, cis[i, "upper"], i, col = "green", lwd = 2)

+ points(cis[i, c("lower","upper")], c(i,i), col = "red", pch = 19)

+ }

>

> print("-------------------")

[1] "-------------------"