(CA2 EPS) به نام خدای مهربان 810198523 ملیکا عباسی سرحلی

بخش (I) :

```
countries-data <- read.csv("countries.csv", header = T, na.strings = c("", "<NA>"))
```
(1)

(2) راه‌حل‌ها برای missing values : ۱- قدم اول می‌توانند تشخیص آن‌ها باشد توسط is.na( ) که با توجه به وکتور بودن یا نبودن متغیری که به آن می‌دهیم به ازای مقادیر گم‌شده T برمی‌گرداند و به ازای بقیه مقادیر F برمی‌گرداند

۲- می‌توان همین کاری که در سؤال ۱ انجام دادیم این مقادیر را recode کنیم به NA

۳- اگر عمل ریاضی‌واری (Arithmetic) بخواهیم انجام می‌دادیم مثل میانگین گرفتن می‌توان mean(x, na.rm = TRUE) استفاده کرد یعنی مقادیر گم‌شده ها را در حساب کردن آن حذف می‌کند وگرنه میانگین NA می‌شد

۴- می‌توان وکتوری از مقادیر گم‌شده با توجه به ردیفشان گرفت با استفاده از [!complete.cases( )] ← اسم data ← اسم data

۵- می‌توان مقادیر گم‌شده را به کلی حذف کرد با تابع ( na.omit ← اسم data یا na.exclude( )

۶- شاید بتوان در بعضی موارد برای مقادیر عددی که در بعضی مسائل خاص هستند به جای NA ، از 0 استفاده کرد یعنی با استفاده از تابع is.na و درهم‌ریختن ابتدا ماتریس را با آرگومان replace = TRUE سپس دادن این کار را انجام داد

مهم ★ ۷- راه حل بعدی این است که اگر دو وکتور از داده‌ها را باهم وقتی روی نمودار می‌کشیم و حالت شبیه زیر ۴۵ درجه اول می‌شوند یا به زبان دیگر همبستگی بالایی و نزدیک به ۱ دارند، خطی شده و می‌توان به جای داده‌های گم‌شده یکی مقدار متناظر با وکتور دیگر با آن مقدار را گذاشت

③ ماتریس همبستگی: $\rho(x,y) = cov\left(\frac{x-\mu_x}{\sigma_x}, \frac{y-\mu_y}{\sigma_y}\right)$ $\qquad -1 \le \rho(x,y) \le 1$

ماتریس کوواریانس: $cov(x,y) = E[XY] - E[X]E[Y]$

دستور درست کردن ماتریس همبستگی در R، cor() است اما در قسمت خواسته شده اگر use را برابر با pairwise.complete.obs نگذاریم هر بار یکی از مقادیر که دو به دو cor می‌کند NA باشد جواب NA می‌شود با این کار یکی از روش‌های handle کردن missing values را به کار بردم و کل بیشترین‌ها رسیدند. درضمن ـــ[sapply(ـــ, is.numeric)] را به کار بردم تا فقط اعداد را با هم همبستگی‌شان را بگیرد.

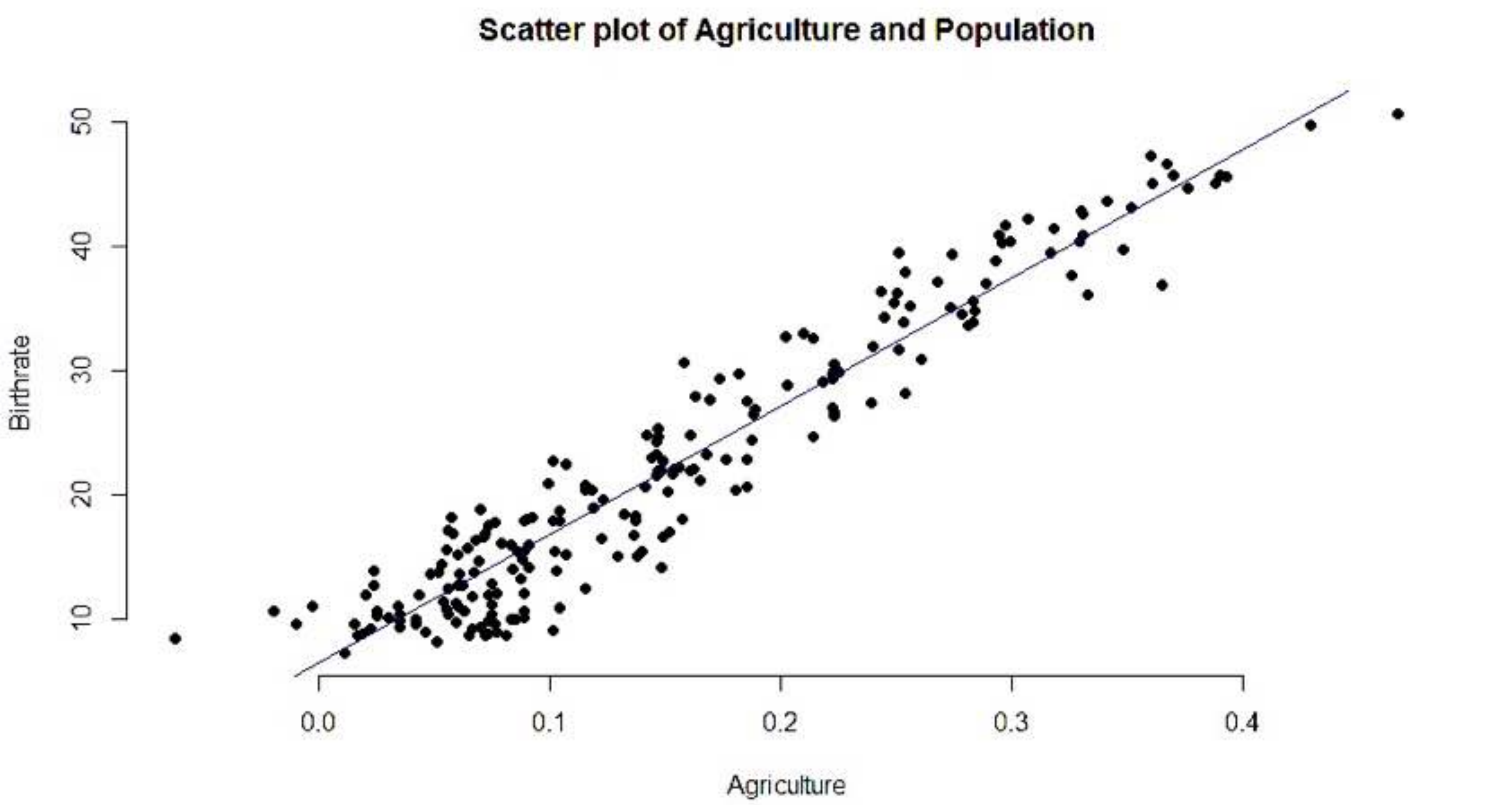
اسم dataframe

Correlation_of_Birthrate_and_agriculture

$= \boxed{0.9582602}$

④ رسم شده است چون همبستگی آن‌ها نزدیک به ۱ می‌باشد به صورت نمودار ربع اول در آمده.
و باز از pairwise.complete.obs استفاده کردم تا حاصل خیلی‌ها NA نشود

⑤ چون به صورت تقریبا $y = x$ هستند می‌توان با استفاده از اینکه خط هستند و ضریب همبستگی بالایی دارند به جای مقادیر گمشده مقادیر متناظر آن یکی را به جای آن مقدار گذاشت.


Scatter plot of Agriculture and Population
Birthrate
10
20
30
40
50
0.0
0.1
0.2
0.3
0.4
Agriculture

```
#CA_2_EPS_MELIKA_MINAEI_BIDGOLI_810198523
# FirstPart
----------------------------------------------------------------
-----------------------------------
#1
countries_data <- read.csv("countries.csv", header = T,
na.strings = c("", "<NA>"))

#3
main_correlation <- cor(x =
countries_data[sapply(countries_data,is.numeric)],
         y = countries_data[sapply(countries_data,is.numeric)],
use = "pairwise.complete.obs")

#4
correlaion_of_birthrate_and_agriculture <- cor(x =
countries_data$Birthrate,
                                          y = countries_data$Agriculture,
                                          use = "pairwise.complete.obs")
#print(correlaion_of_birthdate_and_agriculture)
y = countries_data$Birthrate
x = countries_data$Agriculture
plot(x, y,
    main = "Scatter plot of Agriculture and Population",
    xlab = "Agriculture", ylab = "Birthrate",
    pch = 19, frame = FALSE)
abline(lm(y ~ x), col = "blue")

#5
na_indexes <- which(is.na(x))
number = length(na_indexes)
for(i in 1 : number) {
  x[na_indexes[i]] = y[na_indexes[i]]
}
```

تمرین (II)

(1) uniform random variables :

```
U = runif(n, min = 0, max = 1)
```

تعداد $n = 1$ برای ساخت یک متغیر تصادفی با توزیع یکنواخت استاندارد

با استفاده از runif در واقع pseudo-random می‌سازد نه به طور کامل تصادفی و عدد را بین min و max انتخاب می‌کند

(۲) Bernoulli random variables :

```
P = 0.6
U = runif(1, min = 0, max = 1)
X = (U < p)
```

→ با این کار می‌توان برنولی که شکست یا موفقیت است را شبیه سازی کرد چرا که

با این کار بازه $[0, 1]$ را به دو بخش $[0.6, 1]$ و $[0, 0.4)$ تقسیم می‌کند

حال من این جا $[0, 0.4)$ را موفقیت در نظر گرفتم که TRUE بدهد

ولی می‌توانستم برعکس در نظر بگیرم یعنی علامت کوچکتر را بزرگتر می‌کردم (ذکر شده در کامنت)

و اگر شکست باشد یعنی صدق نکند FALSE می‌دهد

(۳) ابتدا Binomial random variables : اگر $X_i$ ها برنولی باشند و داشته باشیم $X_1 + X_2 + \dots + X_n$ با احتمال p

هرکدام باشد می‌توان آن را با $Bin(n, p)$ نشان داد سپس صرفاً در آخر $X = sum(U < p)$ باشد و در runif

یا $X = sum(U > p)$

n را برابر ۱۰ که صورت گفته بذاریم یعنی ۱۰ تا از این متغیر درست کرده

حال برای ساختن ۱۰۰ نمونه از این متغیر

↓

این متغیر برایم ابهام داشت که $n = 1$ یا $n = 10$ باشد (؟)

سپس آن را در یک لوپ ۱۰۰ بار انداختم و هر بار مقدار متغیر تصادفی را در یک وکتور ریختم تا بعد به طریق mean( )

و var( ) ، میانگین و واریانس آن ها را بدست آورم.

حدت اینکه میانگین تقریباً ۵ یا ۶ کمی بالا و پایین و واریانس تقریباً ۲ یا ۳ است باز هم کمی بالا یا پایین.

برای توزیع Binomial داشتیم : میانگین یا $np = E[X]$ که در این جا $E[X] = np = 10 \times 0/6 = \boxed{6}$

و برای واریانس : $var(x) = npq$ ← $var(x) = 10 \times 0/6 \times 0/4 = \boxed{2/4}$

پس هم خوانی دارد و درست است

```
# SecondPart
------------------------------------------------------------------------------------------------------------------
#1
Uniform_random_variable = runif(1, min = 0, max = 1)

#2
p = 0.6
Uniform_random_variable = runif(1, min = 0, max = 1)
# Depending on which range is considered a win ">" and "<"
changes!
#Bernoulli_distribution = (Uniform_random_variable > p)
Bernoulli_distribution = (Uniform_random_variable < p)

#3/1
p = 0.6
n = 10
Uniform_random_variable = runif(n, min = 0, max = 1)
# Depending on which range is considered a win ">" and "<"
changes!
#Binomial_variable = sum(Uniform_random_variable > p)
Binomial_variable = sum(Uniform_random_variable < p)

#3/2
number = 100
vector_of_random_vars <- c()
for(i in 1 : number) {
  p = 0.6
  # It was not vivid for me to use 1 or 10 for runif()
  #n = 1
  n = 10
  Uniform_random_variable = runif(n, min = 0, max = 1)
  # Depending on which range is considered a win ">" and "<"
changes!
  #Binomial_variable = sum(Uniform_random_variable > p)
  Binomial_variable = sum(Uniform_random_variable < p)
  vector_of_random_vars[i] <- Binomial_variable
}
means = mean(vector_of_random_vars)
variance = var(vector_of_random_vars)
```

کنش III

① درستی رابطه $U \sim Uniform[0,1]$ و $F_X(x)$ تابع توزیع جمعی $X$ ، داریم $X = F^{-1}(U)$ اثبات :

$$P(X \le x) \xrightarrow{X=F^{-1}(U)} p(F^{-1}(U) \le x) \xrightarrow{\text{گرفتن } F} p(U \le F(x)) = F_X(x)$$

تابع افزایشی

پس $F_{(x)}$ CDF است

② اثبات اینکه $X \sim exp(\lambda)$ خواهیم داشت : $X = \frac{-1}{\lambda} \ln(U)$

با فرض $\lambda = 1$

$F(x) = 1 - e^{-x} \quad x > 0$

$U \sim Uniform(0,1)$

$$X = F^{-1}(U) = -\ln(1-U) \Rightarrow X \sim F \xrightarrow{1-U \sim Uniform(0,1)} X = -\ln(U)$$

$U \quad 1-U$

$0 \quad\quad 1$

اما اگر بخواهیم $\lambda$ را هم در رابطه به کار ببریم $X = \frac{-1}{\lambda}\ln(U)$ می شود

$p(X \le x) = p(-\ln(1-U)/\lambda \le x) =$

روش دیگر این است که :

$$p(\ln((1-U)^{-1}) \le \lambda x) = p(\frac{1}{1-U} \le e^{\lambda x}) = p(1-U \ge e^{-\lambda x}) = p(U \le 1-e^{-\lambda x})$$

$= 1 - e^{-\lambda x}$ ★

$$p(-\ln(1-U) \le x) = p(-\ln(U) \le x) = p(U^{-1} \le e^{\lambda x}) = p(U \ge e^{-\lambda x}) = 1 - p(U \le e^{-\lambda x}) = 1 - e^{-\lambda x} \text{ ★★}$$

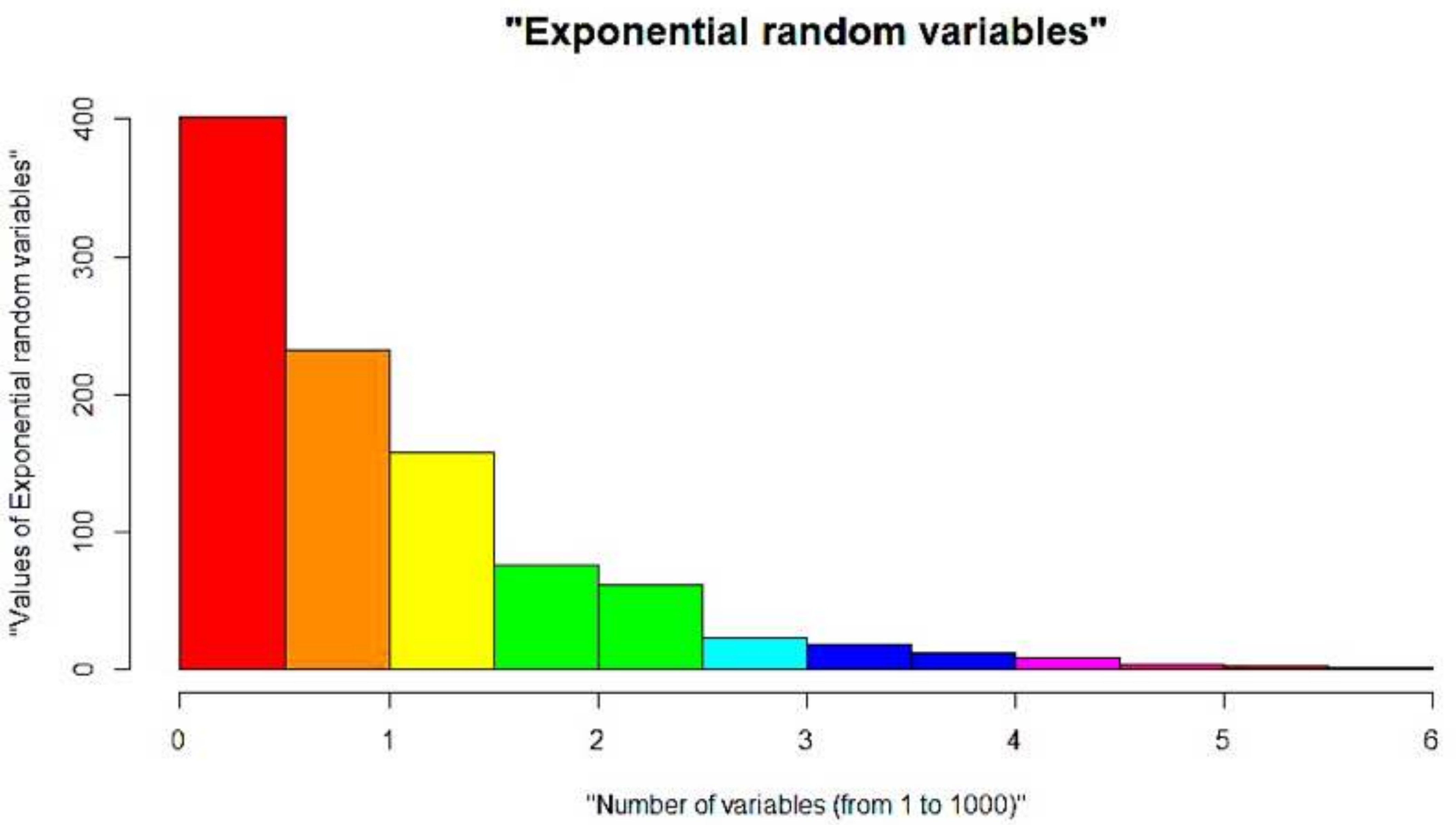
$\xrightarrow{\text{طبق ★ و ★★}} U = 1 - U$ پس درست بوده است

یک لوپ ۱۰۰۰ تایی زدم و هر بار یک وکتور ۱۰۰۰ عضوی از متغیر تصادفی را درست کرده و در آن ریختم

سپس با استفاده از plot و hist مقدارهای آن را کشیدم

در hist به طور واضح نشان داده شده که به صورت تابع نمایی ln ای چون کاهنده است ، مقادیر تغییر کنند

سپس تبدیل را نیز plot ، barplot را کاشت کردم


"Exponential random variables"
"Values of Exponential random variables"
400
300
200
100
0
0
1
2
3
4
5
6
"Number of variables (from 1 to 1000)"

```
# ThirdPart
-----------------------------------------------------------------------
------------------------------------
#1
Uniform_random_variable = runif(n, min = 0, max = 1)
lambda = 1
Exponential_variable = (-1)/lambda *
log(Uniform_random_variable)

#2
number = 1000
vector_of_random_vars <- c()
for(i in 1 : number) {
  Uniform_random_variable = runif(n, min = 0, max = 1)
  lambda = 1
  Exponential_variable = (-1)/lambda *
log(Uniform_random_variable)
  vector_of_random_vars[i] <- Exponential_variable
}
hist(vector_of_random_vars, col = rainbow(10),
     xlab = "\"Number of variables (from 1 to 1000)\"", main =
"\"Exponential random variables\"",
     ylab = "\"Values of Exponential random variables\"",
cex.main = 1.5)



#plot(vector_of_random_vars, xlab = "\"Number of variables
(from 1 to 1000)\"",
#     ylab = "\"Values of Exponential random variables\"", col =
rainbow(10))
#barplot(vector_of_random_vars,
#        main = "\"Exponential random variables\"",
#        xlab = "\"Number of variables (from 1 to 1000)\"",
#        col = (rainbow(3)),
#        beside = T,
#        args.legend = list(title = "Countries", x = "topright", inset
= c(-0.01, 0.02), cex = 0.5))
```