statistics project

Nguyen Ngoc Duy

Part 1: Simulation Exercise Instructions

1.1 Simulations:

```
#install.packages("tinytex")
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

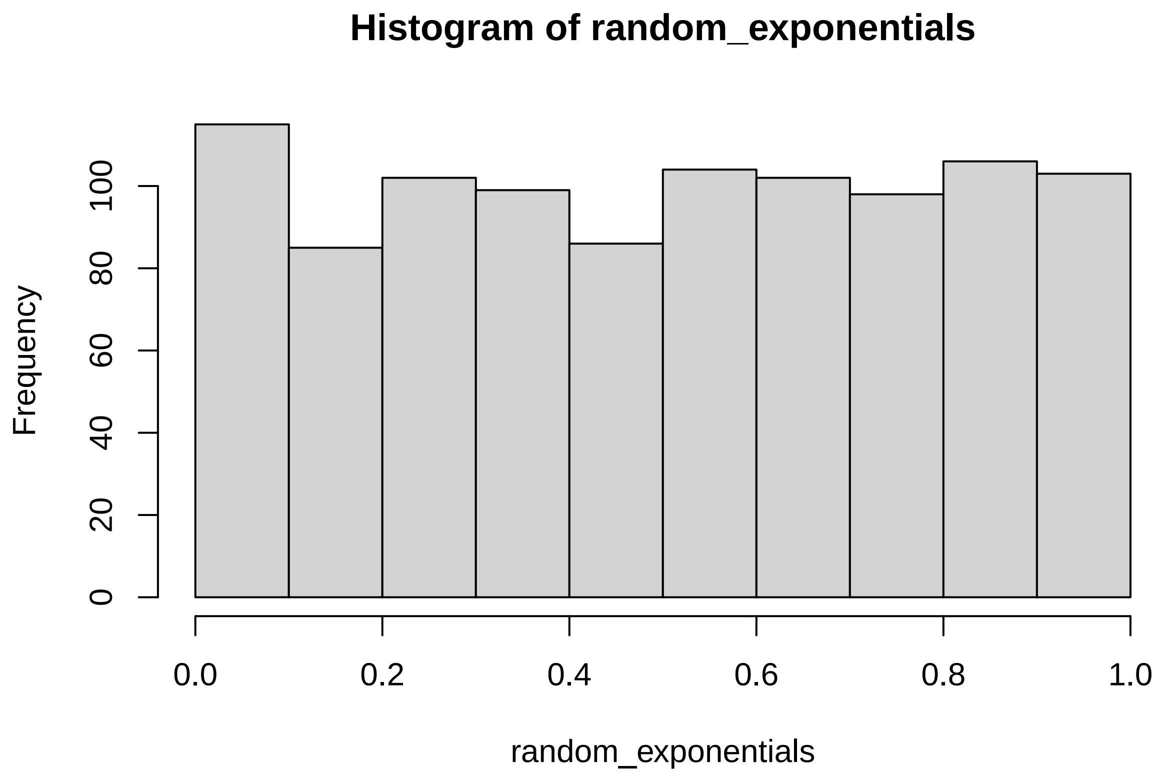
```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
### As a motivating example, compare the distribution of 1000 random uniforms
#hist(runif(1000))
#and the distribution of 1000 averages of 40 random uniforms
#mns = NULL
#for (i in 1 : 1000) mns = c(mns, mean(runif(40)))
#hist(mns)
```

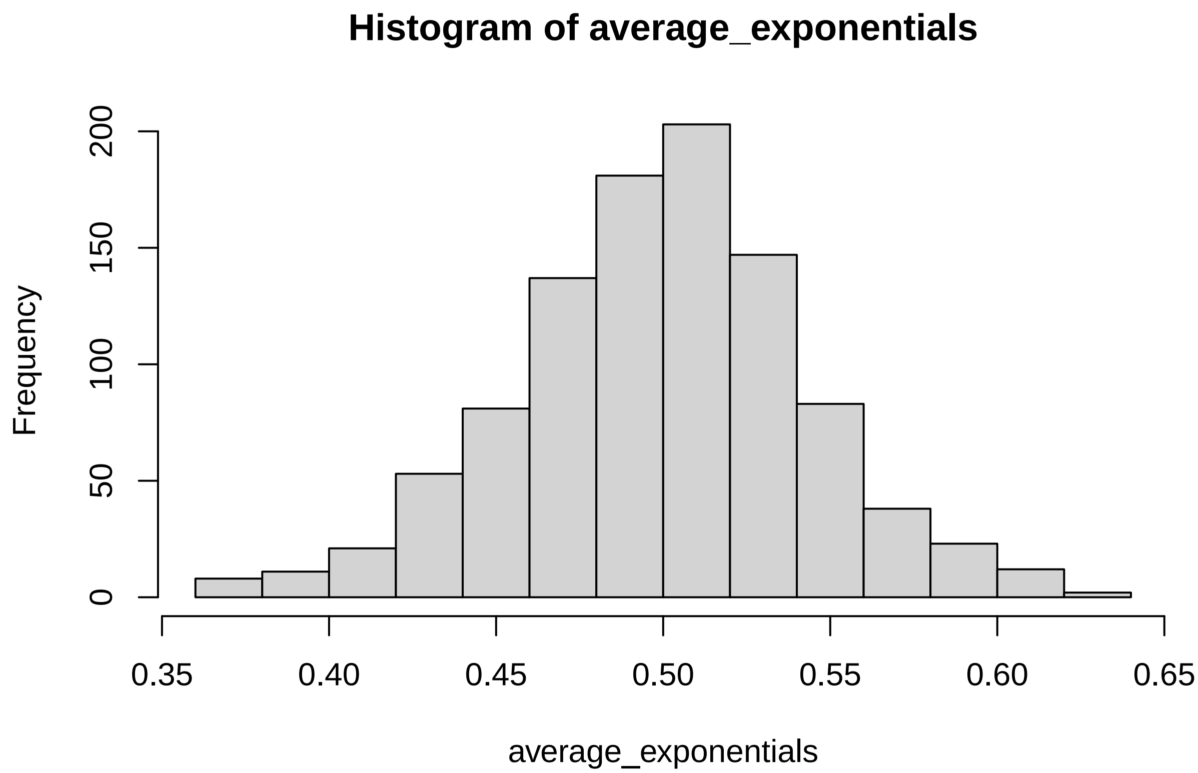
Distribution of a large collection of random exponentials

```
lambda <- 0.2
n <- 1000
random_exponentials <- runif(rexp(n,lambda))
hist(random_exponentials)
```



Distribution of a large collection of averages of 40 exponentials.

```r
mns = NULL
n=40
for (i in 1 : 1000) mns = c(mns, mean(runif(rexp(n,lambda) )))
average_exponentials <- mns
hist(average_exponentials)
```




```r
data_combine <- cbind(random_exponentials,average_exponentials)
```

```r
library(reshape2)
```

```
##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

```r
data_combine_melt <- melt(data_combine)
head(data_combine_melt)
```

```
##   Var1                Var2     value
## 1    1 random_exponentials 0.7025296
## 2    2 random_exponentials 0.6728588
## 3    3 random_exponentials 0.7038738
## 4    4 random_exponentials 0.6509497
## 5    5 random_exponentials 0.8261549
## 6    6 random_exponentials 0.5441878
```

1.2 Show the sample mean and compare it to the theoretical mean of the distribution.

```
population_mean <- mean(random_exponentials)
sample_mean <- mean(average_exponentials)
sample_mean
```

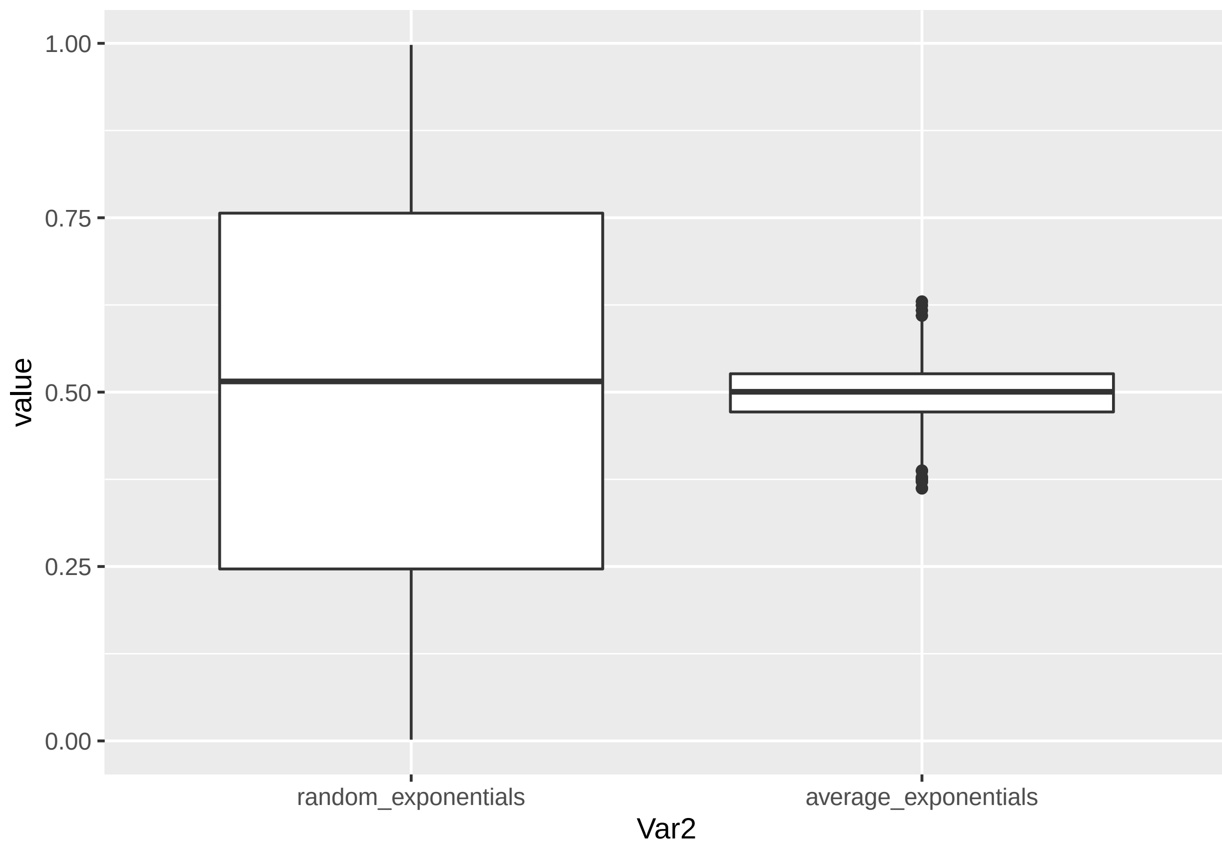
```
## [1] 0.4988868
```

```
population_mean
```

```
## [1] 0.5017484
```

Sample_mean and population_mean is nearly the same

```
data_combine_melt %>%
  ggplot(aes(x=Var2,y=value)) +
  geom_boxplot()
```



1.3. Show how variable the sample is (via variance) and compare it to the theoretical variance of the distribution.

```r
population_var <- var(random_exponentials)
sample_var <- var(average_exponentials)
sample_var
```

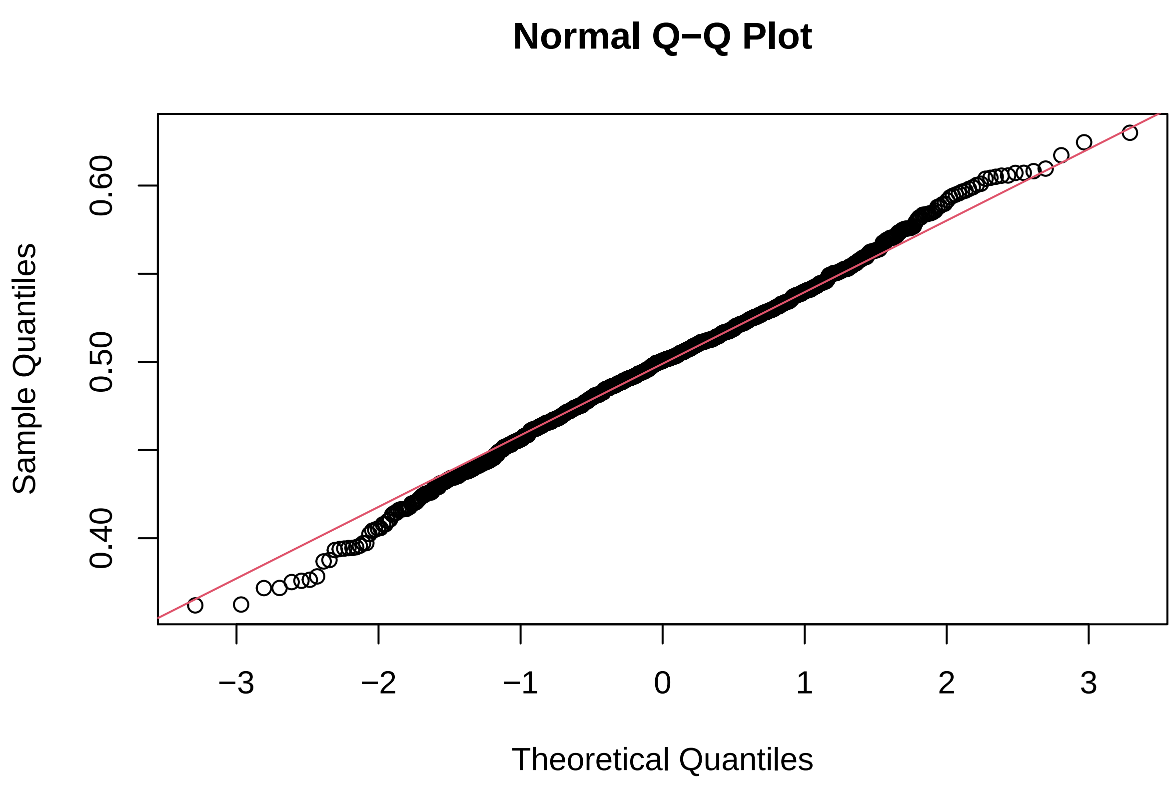
```
## [1] 0.001906171
```

```r
population_var
```

```
## [1] 0.0850905
```

Large different between sample and population variance ### 1.4. Show that the distribution is approximately normal.

```r
qqnorm(average_exponentials)
qqline(average_exponentials, col = 2)
```




```r
# Test normal distribution
shapiro.test(average_exponentials)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  average_exponentials
## W = 0.99659, p-value = 0.02883
```

From the output, the p-value > 0.05 implying that the distribution of the data are not significantly different from normal distribution. In other words, we can assume the normality.