STAT 3675Q Homework 6

Due date: **Thursday, October 9, at noon**

Zeshi Feng

Note:

- Ensure that your code is fully visible in the PDF and not cropped. If needed, break the code into multiple lines to fit.
- It is recommended to write descriptive answers outside of R code chunks (i.e., as text in the main body), while comments within the code chunks can be reserved for brief code annotations.
- In all homework questions, include a written explanation of any output to earn full credit.

Question 1 [30 points]

a. Create a vector of length 10. Apply any 5 of the following functions to the vector:

abs, sqrt, ceiling, floor, round, trunc, cos, sin, tan, acos, asin, atan, log, log10, exp

Answer:

```r
v <- c(-1,-2,-3,4,5,6,7,8,9,10)
abs(v)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```r
sqrt(v)
```

```
## Warning in sqrt(v): NaNs produced
```

```
##  [1]      NaN      NaN      NaN 2.000000 2.236068 2.449490 2.645751 2.828427
##  [9] 3.000000 3.162278
```

```r
ceiling(v)
```

```
##  [1] -1 -2 -3  4  5  6  7  8  9 10
```

```r
floor(v)
```

```
##  [1] -1 -2 -3  4  5  6  7  8  9 10
```

b. Set the random seed to 1111. Create a vector containing 100 numbers drawn randomly from the standard uniform distribution. Compute the sample mean, median, and standard deviation.

Answer:

```
set.seed(1111)
v <- runif(100)
mean_v <- mean(v)
median_v <- median(v)
sd_v <- sd(v)

mean_v
```

```
## [1] 0.5314407
```

```
median_v
```

```
## [1] 0.553795
```

```
sd_v
```

```
## [1] 0.3051975
```

c. Redo part b for the normal distribution with mean 1 and standard deviation 0.5.

Answer:

```
set.seed(1111)
v <- rnorm(100, mean = 1, sd = 0.5)
mean_v <- mean(v)
median_v <- median(v)
sd_v <- sd(v)

mean_v
```

```
## [1] 1.120754
```

```
median_v
```

```
## [1] 1.089917
```

```
sd_v
```

```
## [1] 0.5518796
```

d. Redo part b for the exponential distribution with rate 2.

Answer:

```
set.seed(1111)
v <- rexp(100, rate=2)
```

```r
mean_v <- mean(v)
median_v <- median(v)
sd_v <- sd(v)

mean_v
```

```
## [1] 0.4973352
```

```r
median_v
```

```
## [1] 0.3760153
```

```r
sd_v
```

```
## [1] 0.5233821
```

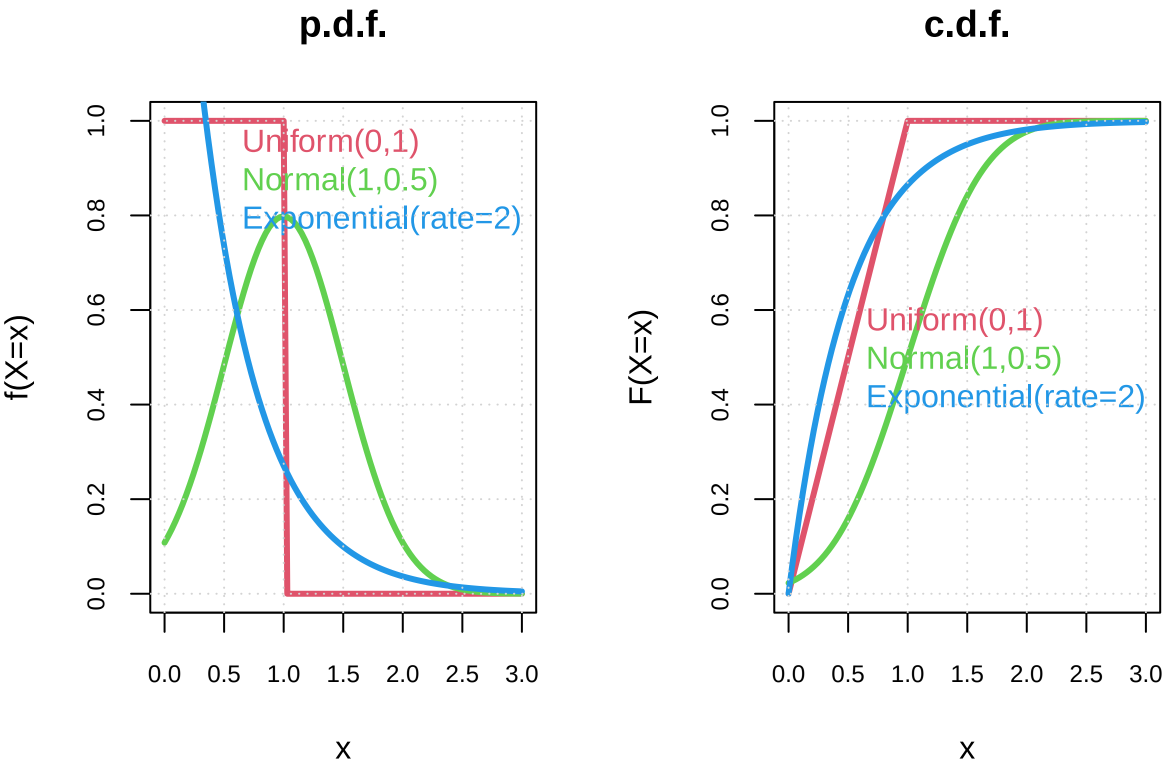
e. Reproduce the plot on page 36 of Lecture 6 Part 1. Hint: Refer to the code on page 33. You do not need to use exactly the same graphical settings (such as line width and font size).

Answer:

```r
x <- seq(0,3, length=100)
par(mfrow=c(1,2))

plot(x, dunif(x), type="l", col=2, lwd=3,
     xlab="x", ylab="f(X=x)", main="p.d.f.",cex.axis=0.75)
lines(x, dnorm(x, mean=1, sd=0.5), col=3, lwd=3)
lines(x, dexp(x, rate=2), col=4, lwd=3)
legend("topright",legend=c("Uniform(0,1)","Normal(1,0.5)","Exponential(rate=2)"),text.co
grid()

plot(x,punif(x, 0, 1),col=2,type='l',lwd=3,xlab="x",
ylab="F(X=x)",main="c.d.f.",cex.axis=0.75)
lines(x, pnorm(x, mean=1, sd=0.5), col=3, lwd=3)
lines(x, pexp(x, rate=2), col=4, lwd=3)
legend("right",legend=c("Uniform(0,1)","Normal(1,0.5)","Exponential(rate=2)"),text.col=c
grid()
```



Question 2 [55 points]

a. If `x <- "Good Morning! "`, find out the number of characters in x.

Answer:

```
x <- "Good Morning! "
nchar(x)
```

```
## [1] 14
```

b. Consider the character vector `x2 <- c ("Nature's", "Best ")`, how many characters are there in x2?

Answer:

```
x2 <- c ("Nature's", "Best ")
nchar(x2)
```

```
## [1] 8 5
```

c. If `fname <- "James"` and `lname <- "Bond"`, write some R code that will produce the output “James Bond”.

Answer:

```
fname <- "James"
lname <- "Bond"
```

```
fullname <- paste(fname, lname)
fullname
```

```
## [1] "James Bond"
```

d. If `m <- "Capital of America is Washington"` then extract the string "Capital of America" from the character vector m.

Answer:

```
m <- "Capital of America is Washington"
result <- substr(m, 1, 19)
result
```

```
## [1] "Capital of America "
```

e. Write some R code to replace the first occurrence of the word "failed" with "failure" in the string "Success is not final, failed is not fatal".

Answer:

```
m <- "Success is not final, failed is not fatal"
result <- sub("failed", "failure", m)
result
```

```
## [1] "Success is not final, failure is not fatal"
```

f. Explain the difference between the functions `sub` and `gsub` using an example. You can define a character string, e.g., "England is Beautiful. England is not part of EU", and try replacing "England" with "UK".

Answer:

```
text <- "England is Beautiful. England is not part of EU"
sub_result <- sub("England", "UK", text)
gsub_result <- gsub("England", "UK", text)

sub_result
```

```
## [1] "UK is Beautiful. England is not part of EU"
```

```
gsub_result
```

```
## [1] "UK is Beautiful. UK is not part of EU"
```

"sub_result only change the first 'England'; gsub_result change all 'England' to 'UK' "

g. Read the lines of the file via `readLines()`

```
abt <- readLines(
  'https://raw.githubusercontent.com/matloff/fasteR/master/data/aboutR.txt')
```

and then check the structure of `abt` using `str()`.

Answer:

```
str(abt)
```

```
##  chr [1:70] "" "What is R?" "" "Introduction to R" "" ...
```

h. Create one long vector to connect (paste) all components of abt. Use the collapse argument.

Answer:

```
abt_long <- paste(abt, collapse = " ")
abt_long
```

```
## [1] " What is R?  Introduction to R     R is a language and environment for statistic
```

i. Find the total number of characters in the above long vector. Then extract the substring from position 288 to 336.

Answer:

```
substr(abt_long, 288, 336)
```

```
## [1] "nd colleagues. R can be considered as a    differ"
```

j. Break the long vector into individual words using `strsplit()`, and save it as y. What is the class of y? Check the structure of y using `str()` and explain what `y[[1]][2]` give in output.

Answer:

```
y <- strsplit(abt_long, " ")
class(y)
```

```
## [1] "list"
```

```
str(y)
```

```
## List of 1
##  $ : chr [1:722] "" "What" "is" "R?" ...
```

```
y[[1]][2]
```

```
## [1] "What"
```

"accesses the first element of the list (the character vector of words). accesses the second word in that vector."

k. First unlist y using `unlist()` so that it becomes a character vector. Note that the `strsplit()` function in part j treated the excess blanks as words, resulting in a lot of `""`. How to fix this problem? Hint: Extract the subvector of y whose entries are not `""`.

Answer:

```r
y <- strsplit(abt_long, "\\W+")
y_clean <- unlist(y)
y_clean <- y_clean[y_clean != ""]

head(y_clean, 10)
```

```
##  [1] "What"         "is"           "R"            "Introduction" "to"
##  [6] "R"            "R"            "is"           "a"            "language"
```

Question 3 [15 points]

The file `shortstory.txt` contains the short story "A Perfect Day for Bananafish", by J.D. Salinger.

a. Use the function `readLines` to read the content of the file into an object named `txt` in R.

Answer:

```r
txt <- readLines("shortstory.txt")
str(txt)
```

```
##  chr [1:451] "There were ninety-seven New York advertising men in the hotel, and, the
```

b. Explain what the following code does.

Note: Once the data is imported, you may convert `eval=FALSE` to `eval=TRUE` (or simply delete `eval=FALSE`), so that the R code chunks can be evaluated when this Rmd script is knitted.

```r
txt <- paste(txt, collapse = "") # line 1
txt <- tolower(txt) # line 2
txt <- unlist(strsplit(txt,"")) # line 3
```

Answer: 1st line merge all lines into one long string, line 2 Convert all text to lowercase, 3rd line split into individual characters