CIS_HW4

2024-09-22

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

###1. Uniform random number generators in R

```
LCG <- function(n, a, c, m, seed) {
  # Initialize state to seed
  state <- seed
  # Initialize an empty vector to store random numbers
  random_numbers <- numeric(n)

  for (i in 1:n) {
    # Update the state using the LCG formula
    state <- (a * state + c) %% m
    # Normalize the result and store it in the vector
    random_numbers[i] <- state / m
  }

  # Return the generated random numbers
  return(random_numbers)
}

# Example usage
a <- 1103515245
c <- 12345
m <- 2^31
seed <- 5400
random_numbers <- LCG(n = 10, a = a, c = c, m = m, seed = seed)
print(random_numbers)
```

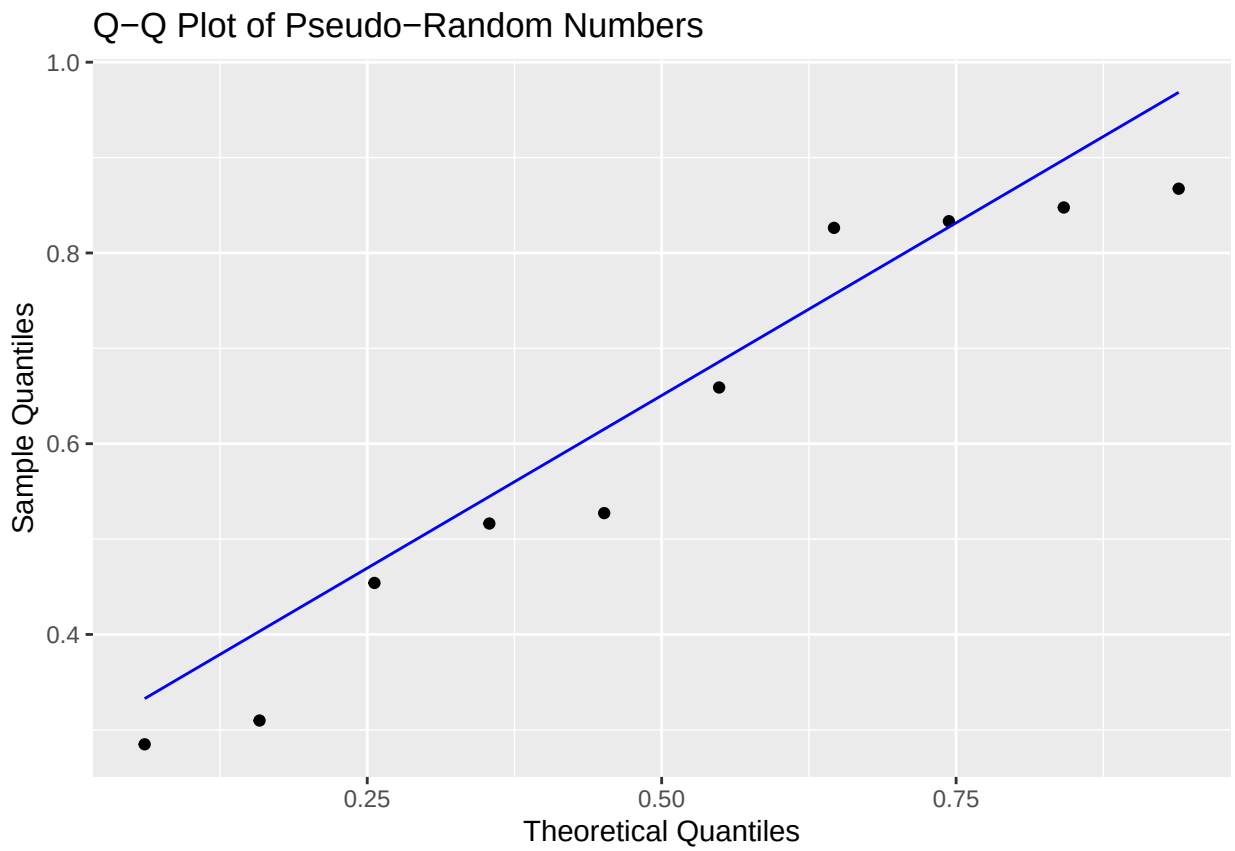
```
##  [1] 0.8673853 0.8263298 0.8333236 0.3097559 0.4540139 0.2848278 0.8476917
##  [8] 0.5272413 0.6589490 0.5163056
```

```
# Compare with the built-in runif function
set.seed(seed)
print(runif(10))
```

```
##  [1] 0.96715156 0.85109119 0.90646319 0.40919220 0.84486948 0.25170803
##  [7] 0.24688742 0.03665934 0.94193288 0.54458507
```

###2. Q-Q plot

```r
# Assuming the LCG function is defined here

# Generate 10 pseudo-random numbers
a <- 1103515245
c <- 12345
m <- 2^31
seed <- 5400
random_numbers <- LCG(n = 10, a = a, c = c, m = m, seed = seed)

# Load necessary library for Q-Q plot
# Install it if you haven't already: install.packages("ggplot2")
library(ggplot2)

# Create a Q-Q plot
qq_plot <- ggplot(data.frame(x = random_numbers), aes(sample = x)) +
  stat_qq(distribution = qunif, dparams = list(min = 0, max = 1)) +
  stat_qq_line(distribution = qunif, dparams = list(min = 0, max = 1), color = "blue") +
  ggtitle("Q-Q Plot of Pseudo-Random Numbers") +
  xlab("Theoretical Quantiles") +
  ylab("Sample Quantiles")

# Print the Q-Q plot
print(qq_plot)
```

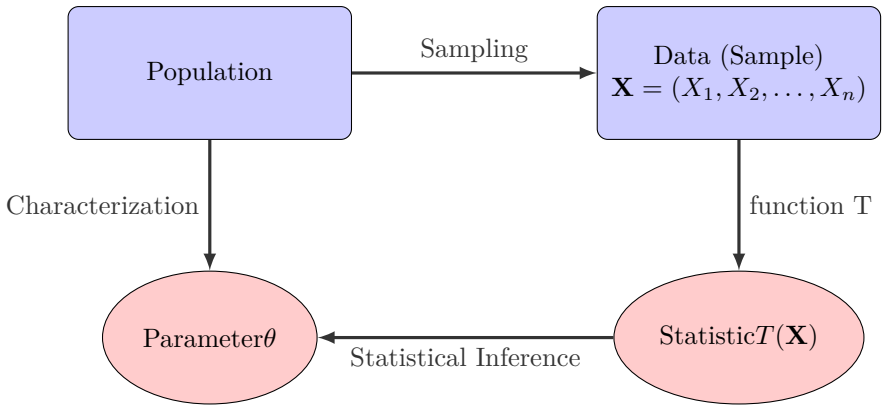


3. LaTeX & Tikz library I

Name	PDF
Gamma	$p(y \mid \alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} y^{\alpha-1} e^{-\beta y}$

Table 1: A univariate continuous density.

4. LaTeX & Tikz library II



###5. (Coding practice) Regrouping problem

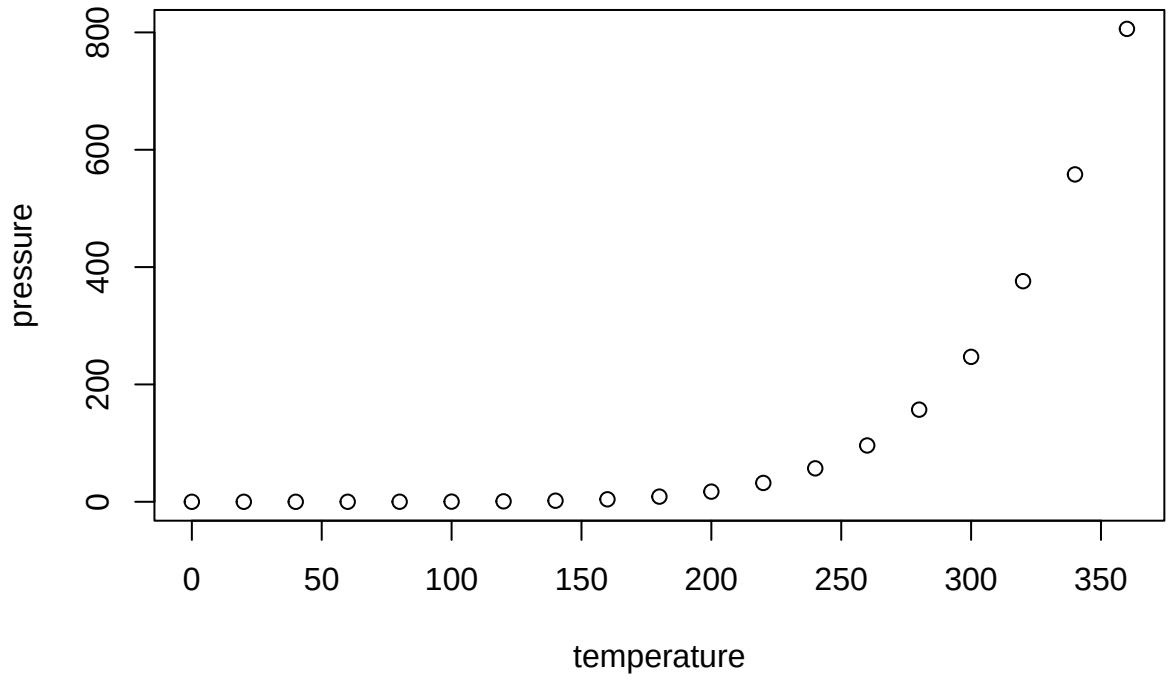
```r
regroup <- function(old_grp) {
  num_groups <- length(old_grp)
  group_size <- length(old_grp[[1]])

  # Create a vector of all numbers
  all_numbers <- unlist(old_grp)

  # Shuffle all numbers randomly
  shuffled_numbers <- sample(all_numbers)

  # Initialize a new list to store new groups
  new_groups <- vector("list", num_groups)

  # Create the first group without any exclusions
  new_groups[[1]] <- shuffled_numbers[1:group_size]
  shuffled_numbers <- shuffled_numbers[-(1:group_size)]

  # For each subsequent group, select numbers ensuring no old partners are in the same group
  for (i in 2:num_groups) {
    # Identify which old group members can't be in this group
    excluded <- unlist(old_grp[sapply(old_grp, function(x) any(x %in% new_groups[[i - 1]]))])

    # Select new members for the current group, excluding those from the same old group
    new_group_members <- shuffled_numbers[!shuffled_numbers %in% excluded][1:group_size]

    # Store the new group
    new_groups[[i]] <- new_group_members

    # Remove selected members from the shuffled list
    shuffled_numbers <- shuffled_numbers[!shuffled_numbers %in% new_group_members]
  }

  return(new_groups)
}

# Example usage
old_grp <- split(1:30, rep(1:6, rep(5, 6)))
new_groups <- regroup(old_grp)
print(new_groups)
```

```
## [[1]]
## 51 34 25 13 32
## 21 14 10  3 12
##
## [[2]]
## 61 64 45 65 63
## 26 29 20 30 28
##
## [[3]]
## 54 55 24 12 23
## 24 25  9  2  8
##
## [[4]]
```

```
## 62 33 44 31 42
## 27 13 19 11 17
##
## [[5]]
## 11 21 53 52 15
##  1  6 23 22  5
##
## [[6]]
##   43   41   35 <NA> <NA>
##   18   16   15   NA   NA
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.