Summary Sheet

AUTHOR
Julia Gallucci

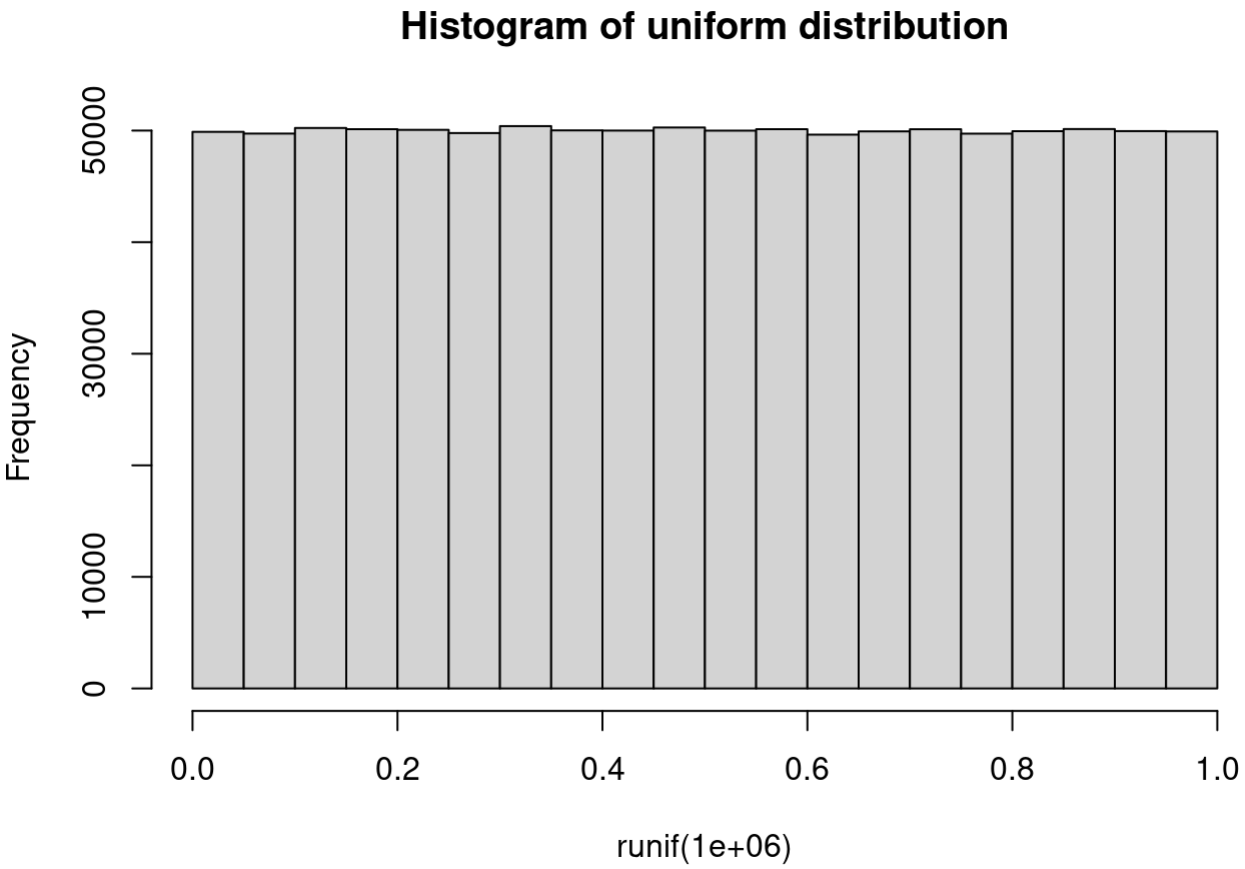
Class 5

Simulation

We can simulate random data from a uniform distribution using the `runif()` function. The `runif()` function generates random numbers from a uniform distribution with a specified **min and max.**

arguments = `n, min (default 0), max (default 1)`

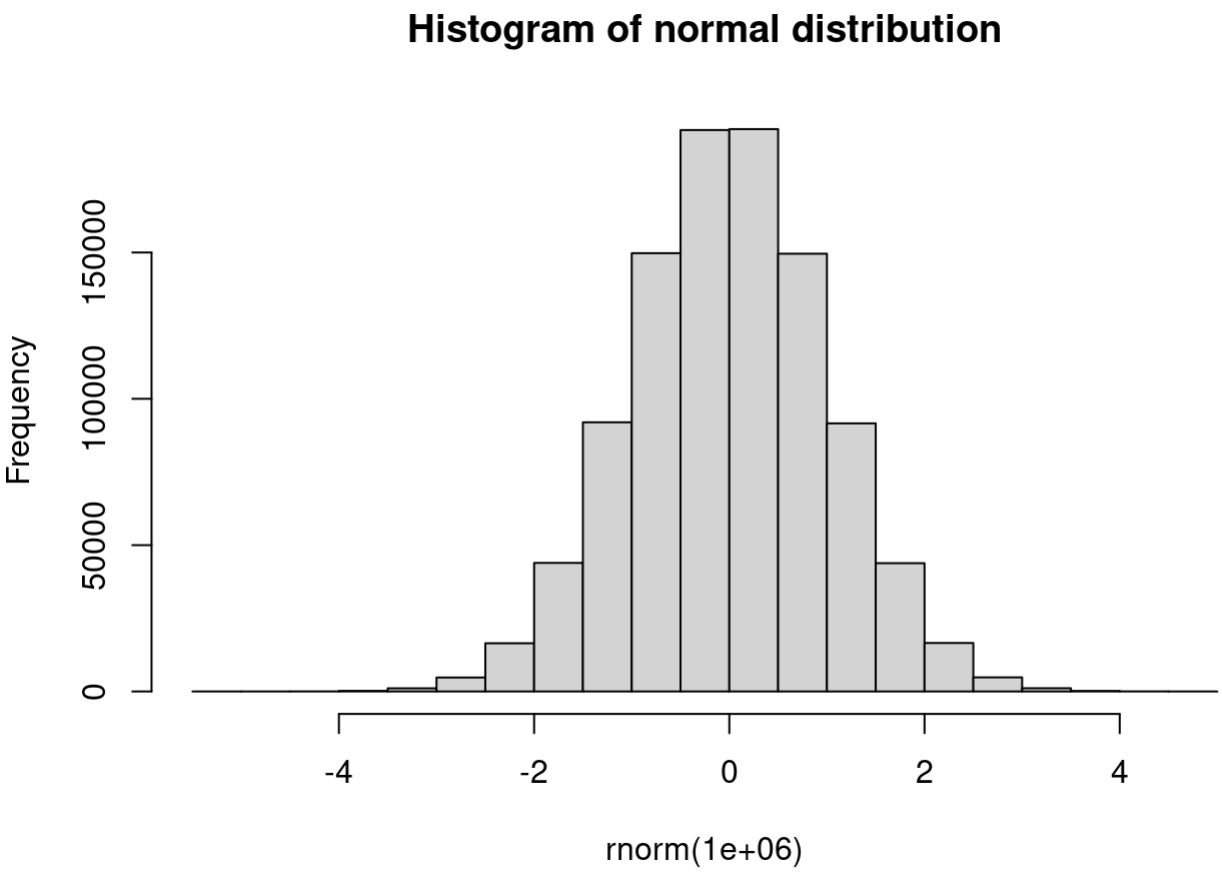
```
hist(runif(1000000), main = "Histogram of uniform distribution")
```



We can simulate random data from a normal distribution using the `rnorm()` function. The `rnorm()` function generates random numbers from a normal distribution with a specified **mean and standard deviation**.

arguments = `n, mean (default 0), sd (default 1)`

```r
hist(rnorm(1000000), main = "Histogram of normal distribution")
```



We set the seed using **`set.seed()`** to ensure reproducibility of the random data. By setting the seed to the same value, the random numbers generated will be the same each time the code is run.

`sample()` is used to generate random samples from a specified set of elements. It allows you to randomly select elements from a vector, shuffle the order of elements, or sample with replacement.

arguments = `vector, size, replace (default FALSE), prob (optional)`

For example,

```r
# Vector of specified elements
x <- c("a", "b", "c", "d", "e")

# Randomly select two elements from the vector
random_selection <- sample(x, 2)

# Print the random selection
print(random_selection)
```

```
[1] "c" "e"
```

For sizes greater than the length of the vector, replacement must be set to TRUE or else an error will occur.

```
random_selection_without_rep <- sample(x, 100, replace = FALSE)
```

```
Error in sample.int(length(x), size, replace, prob): cannot take a sample larger than the population when 'replace = FALSE'
```

Review concepts

Concept	Useful Functions
Data Reading	`read_*()` `* = csv, excel, sas, sav..`

Data Exploration
• Explore the dataset to gain an understanding of its structure, and variables. Can also get summary statistics of numeric variables
	`glimpse()`
	`head()`
	`str()`
	`summary()`

Data Cleaning
• Identify and handle missing or incomplete data.
• Address incorect data types (using explicit coercion)
• format data for consistency (i.e., all same naming convention)
	`as.*()`
	`* = numeric, factor, character..`
	`janitor::clean_names()`
	`is.na()`

Data Manipulation
• Perform data manipulations such as filtering, arranging, grouping and summarizing.
• Create new variables that might be useful for analysis.
	`filter()`
	`arrange()`
	`group_by()`
	`summarise()`
	Note: `%>%` to combine multiple functions

Data Wrangling	`pivot_wider()`

Concept	Useful Functions
• reshaping data from a long format to a wide format or vice versa. It's particularly useful when you want to change the structure of your data to better suit your analysis or visualization needs.	`pivot_longer()`