R Recitation – 14 October Worksheet

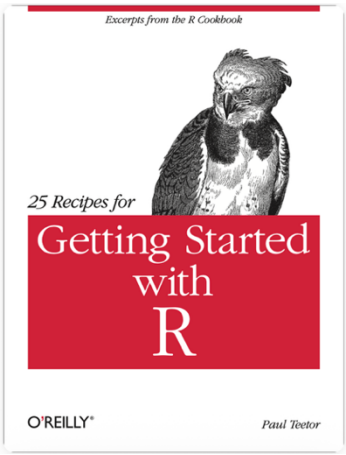
Learning goals

By the end of this worksheet, you will be able to:

- Navigate **RStudio** (Source, Console, Environment, Files/Plots/Packages/Help) and understand how code and narrative text are combined in **R Markdown**.
- **Import datasets** stored as CSV, TSV/TXT, and XLSX files using idiomatic functions.
- Install and load essential libraries: **readr**, **readxl**, **dplyr**, **tidyr**, **ggplot2**, **moments**.
- Compute and report **descriptive statistics** for a numeric column: *`n`, `mean`, `median`, `mode`, `minimum`, `maximum`, `interquartile range (IQR)`, `variance`, `standard deviation`*; optionally read `skewness` and (excess) `kurtosis` as shape indicators.
- Produce and annotate a **base R histogram** (with bin control and axis/title labels).

Dataset used in examples: `child_iq.csv` with columns 'ppvt' child IQ at 3, 'momage' mother's age and 'educ_cat' as education. Place the file in the same folder as your Rmd.

References:



Pages: 13-22

+

R Markdown Tutorial for Beginners:
https://www.datacamp.com/tutorial/r-markdown-tutorial

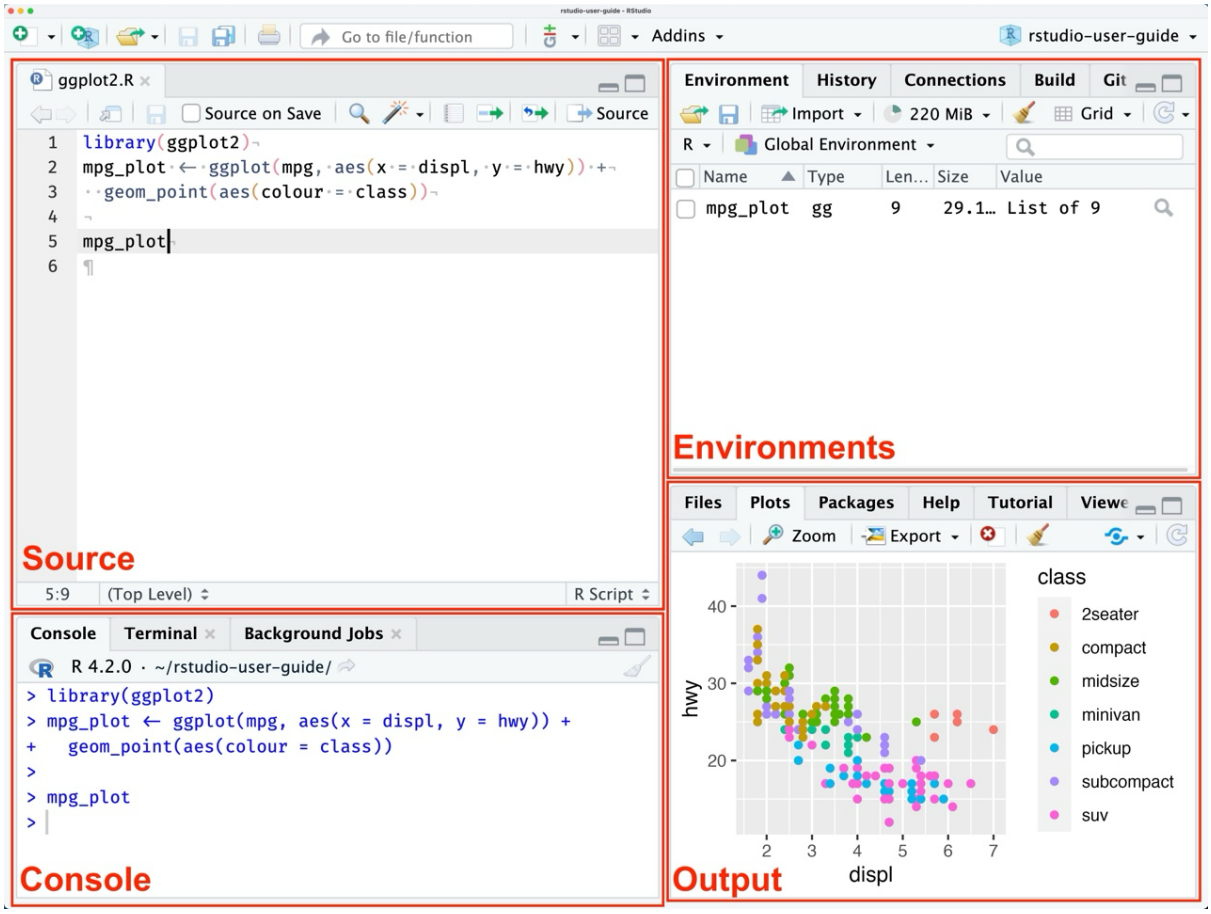
R for Data Science - R Markdown:
https://r4ds.had.co.nz/r-markdown.html

RStudio overview

RStudio presents four main panes:

- the **Source** editor (where you write scripts and R Markdown),
- the **Console** (where R executes commands),
- the **Environment/History** pane (showing objects in memory and recent commands),
- the **Files/Plots/Packages/Help/Viewer** pane (for file navigation, plots, packages, and documentation).

You should recognise where outputs appear (Console/Plots) and how objects are listed (Environment).



R Markdown basics

R Markdown blends explanation and computation. Narrative text is written normally; executable code goes inside **chunks**. A minimal YAML header tells RStudio how to render the document:

```
---
title: "COGS536 Recitation Document"
author: "Your Name"
output: html_document
---
```

Create a code chunk by clicking *Insert Chunk* (green C-plus icon) or typing three backticks:

````
```{r}
# This is an R code chunk
1 + 1
```
````

Inline code injects computed values into text, e.g., `` The dataset has `r mean(5+7)` rows. ``

Use # to add comments inside chunks.

Setup: packages

Before using functions from external packages, install them once (per machine), then load them in each new session.

```
# Install (only once; comment out after the first run)
install.packages("readr")
install.packages("readxl")
install.packages("dplyr")
install.packages("tidyr")
install.packages("ggplot2")
install.packages("moments")
```

Load (every session)

```
library(readr)
library(readxl)
library(dplyr)
library(tidyr)
library(ggplot2)
library(moments)
```

Importing data (CSV, TSV/TXT, XLSX)

The **readr** package provides fast, friendly functions for delimited text files.

CSV (comma-separated values)

```
# Recommended (readr)
IQ_child_data <- read_csv("child_iq.csv")
```

TSV (tab-separated values)

```
iq_tsv <- read_tsv("child_iq.tsv")
```

TXT (generic delimited text)

```
iq_txt <- read.table("child_iq.txt", header = TRUE, sep = "\t")
```

XLSX (Excel)

```
iq_xlsx <- read_excel("child_iq.xlsx", sheet = 1)
```

After import, inspect structure to confirm column names and types:

```
str(IQ_child_data)
```

Selecting columns and making a working vector

Analyses often begin by isolating a single numeric column. Here we extract iq as a simple vector; the three lines below are equivalent.

```r
{r}
# Three common ways to pull a column as a vector
x <- IQ_child_data$ppvt
# x <- dplyr::pull(IQ_child_data, ppvt)
# x <- IQ_child_data[[“ppvt”]]
# Simple range checks
length(x) # sample size n
min(x, na.rm = TRUE) # minimum
max(x, na.rm = TRUE) # maximum
```

Note on missing values: Most summary functions accept `na.rm = TRUE` to ignore `NA`s. Without this, results may be `NA`.

Descriptive statistics

This section computes the core descriptive measures for the `ppvt` vector.

```{r}
# Main summaries
n <- length(x)
mean_v <- mean(x, na.rm = TRUE)
median_v <- median(x, na.rm = TRUE)
mode_v <- mode(x)
range_v <- range(x, na.rm = TRUE) # c(min, max)
iqr_v <- IQR(x, na.rm = TRUE) # interquartile range
var_v <- var(x, na.rm = TRUE)
sd_v <- sd(x, na.rm = TRUE)
```

```{r}
# Optional shape diagnostics
skew_v <- skewness(x, na.rm = TRUE)
exkurt_v <- kurtosis(x, na.rm = TRUE) - 3 # excess kurtosis
```

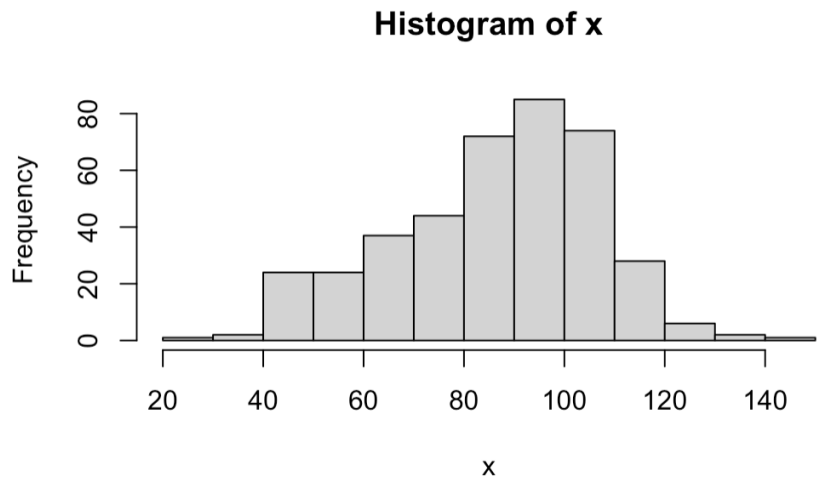
```
# Nicely formatted table for the knitted report
summary_tbl <- tibble(statistic =
c("n","mean","median","mode","min","max","IQR","variance","sd","skewne
ss","excess_kurtosis"), value = c(n, mean_v, median_v, mode_v,
range_v[1],range_v[2], iqr_v, var_v, sd_v, skew_v, exkurt_v))
knitr::kable(summary_tbl, digits = 3)
```

Interpretation hints: Compare **`mean`** and **`median`** to sense skew. A large **`IQR`** indicates spread within the central 50% of the data. **`Variance`** and **`sd`** grow with dispersion. **`Skewness`** (≈ 0 is symmetric) and **`excess kurtosis`** (≈ 0 is normal-like) are optional shape descriptors.

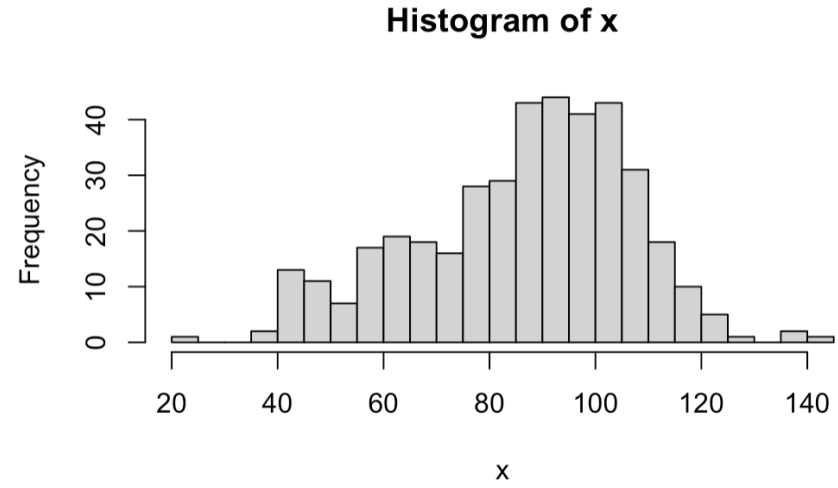
Base R histogram

A histogram visualises the distribution of a numeric variable by counting how many observations fall into each bin.

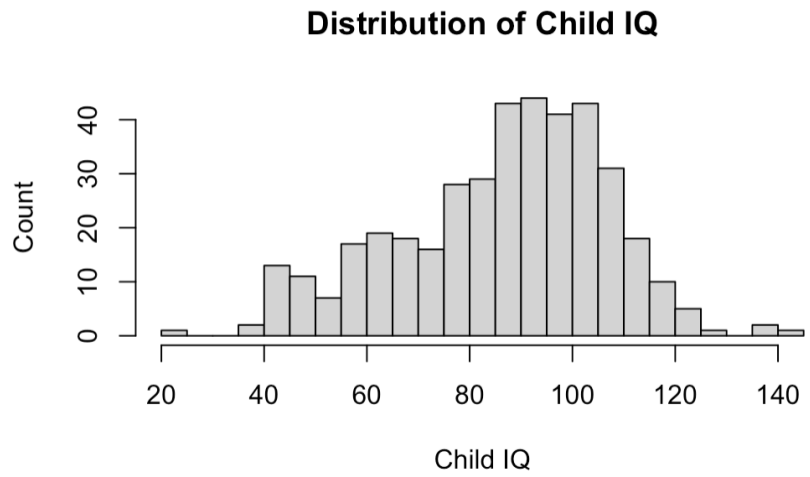
```{r}
# Quick default histogram
hist(x)
```




```{r}
# Control the number of bins
hist(x, breaks = 30)
```



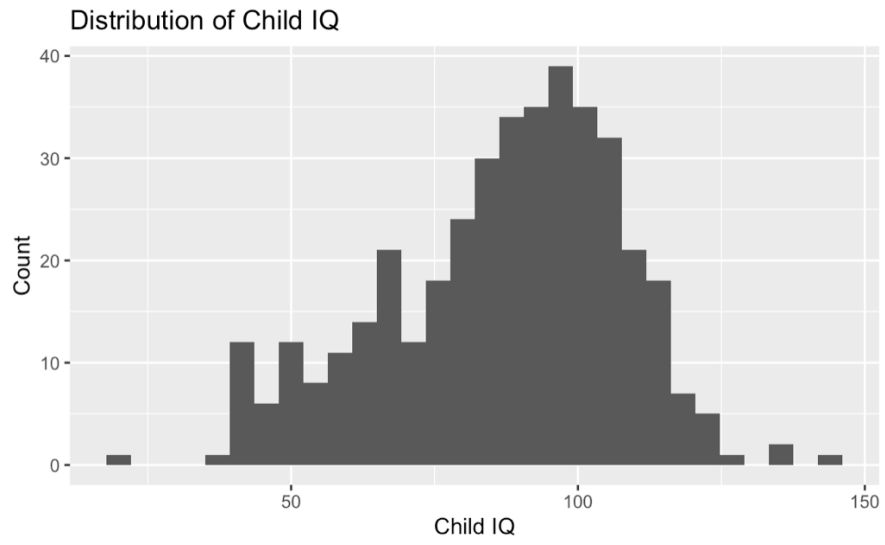
```r
# Add labels and a title
hist(x,
     breaks = 30,
     xlab = "Child IQ",
     ylab = "Count",
     main = "Distribution of Child IQ")
```



(Optional) A **ggplot2** version provides polished defaults and layering:

```
ggplot(IQ_child_data, aes(x = ppvt)) + geom_histogram(bins = 30) + labs(x = "Child IQ", y = "Count", title = "Distribution of Child IQ")
```

```r
ggplot(IQ_child_data, aes(x = ppvt)) + geom_histogram(bins = 30) + labs(x = "Child IQ", y = "Count", title = "Distribution of Child IQ")
```



Short R Markdown write-up

Students should add a short paragraph interpreting the numerical and graphical summaries. For example: report *n*, *mean*, *median*, *IQR*, and *sd* for `ppvt`; note whether the distribution appears symmetric or skewed using **skewness**; and embed the histogram with 30 bins and custom axis labels.

Mini-exercises

1. Filter to `mother_age >= 30` and recompute all statistics for `ppvt`.
2. Compare histograms across two `mother_edu` groups (e.g., `<=12` vs `>12` years).
3. Create a one-sentence narrative using inline code to insert the computed values.

Troubleshooting

- **"No such file or directory"**: Confirm `child_iq.csv` is in the same folder as your `Rmd`. Check your working directory with `getwd()` or use the **Files** pane to set it.
- **Encoding issues**: If Turkish characters appear garbled, try `read_csv(..., locale = locale(encoding = "UTF-8"))`.
- **Package not found**: Ensure `install.packages(...)` has been run at least once, then `library(...)` each new session.

Command recap

```
# Import and inspect
IQ_child_data <- readr::read_csv("child_iq.csv")
str(IQ_child_data)

# Column selection
x <- IQ_child_data$iq

# Descriptives
n <- length(x); mean(x, na.rm=TRUE); median(x, na.rm=TRUE)
IQR(x, na.rm=TRUE); var(x, na.rm=TRUE); sd(x, na.rm=TRUE)

# Histogram
hist(x)
hist(x, breaks=30, xlab="Child IQ", ylab="Count", main="Distribution
of child iq")
```