Package Management

`install.packages()`

installs a package

```
install.packages("tidyverse")
```

base R · Week 1

`library()`

loads a package

```
library(tidyverse)
```

base R · Week 1

Data Import/Export

`read_csv()`

reads a csv file

```
data <- read_csv("data.csv")
```

{readr} Week 1

`download.file()`

downloads a file from the Internet

```
download.file("url", destfile = "my_path/my_file.csv")
```

Base R Weeks 6-8

`read_excel()`

reads an Excel file into R

```
read_excel("data.xlsx", sheet = 1)
```

{readxl} Weeks 6-8

`write_csv()`

writes a dataframe to a CSV file

```
write_csv(data, "output.csv")
```

{readr} Weeks 6-8

`read_rds()`

reads an RDS file into R

```
data <- read_rds("data.rds")
```

{readr} Weeks 9-11

`write_rds()`

writes a data frame to an RDS file

```
data |> write_rds("data.rds")
```

{readr} Weeks 9-11

Data Exploration

`glimpse()`

shows a summary of a data frame

```
glimpse(penguins)
```

{dplyr} Week 1

`skim()`

summary statistics about variables in data frames

```
skim(penguins)
```

{skimr} Week 1

`tbl_summary()`

calculates descriptive statistics

```
tbl_summary(penguins)
```

{gtsummary} Week 1

`makeDataReport()`

makes a data overview report

```
makeDataReport(penguins)
```

{dataReporter} Week 1

`scan_data()`

makes a data overview report

```
scan_data(penguins)
```

{pointblank} Week 1

`is.nan()`

checks if a value is NaN (Not a Number)

```
is.nan(c(1, NaN, 3))
```

base R Weeks 9-11

`is.na()`

checks if a value is NA (Not Available)

```
is.na(c(1, NA, 3))
```

base R Weeks 9-11

Data Wrangling and Analysis

R for the **Rest of Us**

`select()`

keep or drop columns

```
starwars |> select(species, starships)
```

{dplyr} Week 2

`mutate()`

create, modify, and delete columns

```
starwars |> mutate(body_ratio = mass / height)
```

{dplyr} Week 2

`filter()`

keep or drop rows that match a condition

```
starwars |> filter(eye_color == “yellow”)
```

{dplyr} Week 2

`summarize()`

summarize a variable

```
starwars |> summarize(avg_height = mean(height))
```

{dplyr} Week 2

`group_by()`

group by one or more variables

```
starwars |> group_by(species) |> summarize(avg_height = mean(height))
```

{dplyr} Week 2

`arrange()`

order the rows of a data frame

```
starwars |> arrange(desc(height))
```

{dplyr} Week 2

`pivot_longer()`

converts wide data into long format

```
pivot_longer(data, cols = c(var1, var2), names_to = “variable”, values_to = “value”)
```

{tidyr} Weeks 6-8

`pivot_wider()`

converts long data into wide format

```
pivot_wider(data, names_from = “variable”, values_from = “value”)
```

{tidyr} Weeks 6-8

`separate_wider_delim()`

separates one column into multiple columns

```
separate_wider_delim(data, col = “variable”, into = c(“var1”, “var2”), sep = “_“)
```

{tidyr} Weeks 6-8

`separate_longer_delim()`

splits one column into multiple rows by a delimiter

```
separate_longer_delim(data, col = “tags”, delim = “,”)
```

{tidyr} Weeks 6-8

`count()`

counts instances of unique values in a column

```
data |> count(category)
```

{dplyr} Weeks 6-8

`distinct()`

returns unique rows in a dataset

```
data |> distinct()
```

{dplyr} Weeks 6-8

Data Wrangling and Analysis

R for the Rest of Us

parse_number()

extracts numbers from character strings

```
parse_number("$1,234.56")
```

{readr} Weeks 6-8

as.numeric()

converts a column to numeric type

```
as.numeric(c("1", "2", "3"))
```

{base} Weeks 6-8

case_when()

performs conditional value assignment

```
mutate(data, category =
case_when(value > 10 ~ "High",
.default ~ "Low"))
```

{dplyr} Weeks 6-8

case_match()

maps values to new categories

```
mutate(data, category =
case_match(x, "A" ~ "Alpha", "B"
~ "Beta"))
```

{dplyr} Weeks 6-8

na_if()

replaces a specific value with NA

```
mutate(data, var = na_if(var,
"Unknown"))
```

{dplyr} Weeks 6-8

contains()

selects columns that contain a string

```
select(data, contains("score"))
```

{dplyr} Weeks 6-8

starts_with()

selects columns that start with a string

```
select(data, starts_with("age"))
```

{dplyr} Weeks 6-8

str_detect()

checks if a string contains a pattern

```
filter(data, str_detect(name,
"John"))
```

{stringr} Weeks 6-8

str_replace()

replaces text patterns in a string

```
mutate(data, name =
str_replace(name, "Mr.", ""))
```

{stringr} Weeks 6-8

bind_rows()

binds multiple dataframes by rows

```
bind_rows(df1, df2)
```

{dplyr} Weeks 6-8

inner_join()

joins two datasets, keeping only matching rows

```
inner_join(df1, df2,
join_by("id"))
```

{dplyr} Weeks 6-8

left_join()

joins two datasets, keeping all rows from the left

```
left_join(df1, df2,
join_by("id"))
```

{dplyr} Weeks 6-8

Data Wrangling and Analysis

R for the **Rest of Us**

right_join()

joins two datasets, keeping all rows from the right

```
right_join(df1, df2,
join_by("id"))
```

{dplyr} Weeks 6-8

full_join()

joins two datasets, keeping all rows from both

```
full_join(df1, df2,
join_by("id"))
```

{dplyr} Weeks 6-8

ungroup()

removes grouping structure from a dataset

```
data |> ungroup()
```

{dplyr} Weeks 9-11

fct_reorder()

reorders a factor based on another variable

```
data |> mutate(category =
fct_reorder(category, value))
```

{forcats} Weeks 9-11

slice_max()

selects rows with the highest values of a column

```
data |> slice_max(order_by =
value, n = 5)
```

{dplyr} Weeks 9-11

pull()

extracts a single column as a vector

```
vector <- data |>
pull(column_name)
```

{dplyr} Weeks 9-11

fct_relevel()

relevels a factor, moving specified levels first

```
data |>mutate(category =
fct_relevel(category, "High"))
```

{forcats} Weeks 9-11

lag()

shifts values down by one or more rows

```
data |> mutate(prev_value =
lag(value))
```

{dplyr} Weeks 9-11

rename()

renames columns in a dataset

```
data |> rename(new_name =
old_name)
```

{dplyr} Weeks 9-11

drop_na()

removes rows with missing values

```
drop_na(data, column_name)
```

{tidyr} Weeks 9-11

str_glue()

creates formatted strings using variables

```
data |> mutate(full_name =
str_glue("My name is
{first_name} {last_name}"))
```

{glue} Weeks 9-11

Data Visualization

R for the Rest of Us

`ggplot()`

initializes a ggplot object

```
ggplot(data = starwars, aes(x = height, y = mass))
```

{ggplot2} Week 3

`aes()`

defines aesthetic mappings

```
aes(x = height, y = mass, color = gender)
```

{ggplot2} Week 3

`geom_point()`

creates a scatterplot

```
ggplot(starwars, aes(x = height, y = mass)) + geom_point()
```

{ggplot2} Week 3

`geom_bar()`

creates a bar chart

```
ggplot(starwars, aes(x = species)) + geom_bar()
```

{ggplot2} Week 3

`geom_histogram()`

creates a histogram

```
ggplot(starwars, aes(x = height)) +
geom_histogram(binwidth = 10)
```

{ggplot2} Week 3

`geom_boxplot()`

creates a boxplot

```
ggplot(starwars, aes(x = gender, y = height)) + geom_boxplot()
```

{ggplot2} Week 3

`facet_wrap()`

splits the plot into multiple panels

```
ggplot(starwars, aes(x = height, y = mass)) + geom_point() + facet_wrap(vars(species))
```

{ggplot2} Week 3

`facet_grid()`

creates a grid of plots

```
ggplot(starwars, aes(x = height, y = mass)) + geom_point() + facet_grid(rows = vars(gender), cols = vars(species))
```

{ggplot2} Week 3

`labs()`

adds labels to the plot

```
ggplot(starwars, aes(x = height, y = mass)) + geom_point() + labs(title = “Height vs Mass in Star Wars”)
```

{ggplot2} Week 3

`theme()`

customizes plot appearance

```
ggplot(starwars, aes(x = height, y = mass)) + geom_point() + theme_minimal()
```

{ggplot2} Week 3

`scale_color_manual()`

manually sets color scale

```
ggplot(starwars, aes(x = height, y = mass, color = gender)) + geom_point() + scale_color_manual(values = c(“blue”, “pink”, “purple”))
```

{ggplot2} Week 3

`ggsave()`

saves plots to a file

```
ggsave(filename = “plots/starwars-plot.pdf”, height = 8, width = 11, units = “in”)
```

{ggplot2} Week 3

Data Visualization

`geom_line()`

adds a line plot layer to ggplot

```
ggplot(data, aes(x, y)) +
geom_line()
```

{ggplot2} Weeks 9-11

`geom_text()`

adds text labels to a ggplot

```
ggplot(data, aes(x, y, label =
label)) + geom_text()
```

{ggplot2} Weeks 9-11

`geom_text_repel()`

adds text labels that avoid overlapping

```
ggplot(data, aes(x, y, label =
label)) + geom_text_repel()
```

{ggrepel} Weeks 9-11

`scale_y_continuous()`

adjusts the y-axis scale

```
ggplot(data, aes(x, y)) +
scale_y_continuous(labels =
scales::percent)
```

{ggplot2} Weeks 9-11

`percent_format()`

formats numbers as percentages

```
ggplot(data, aes(x, y)) +
scale_y_continuous(labels =
percent_format())
```

{scales} Weeks 9-11

`annotate()`

adds text or shapes to a ggplot

```
ggplot(data, aes(x, y)) +
annotate("text", x = 5, y = 10,
label = "Note")
```

{ggplot2} Weeks 9-11

Quarto

R for the Rest of Us

format: html

sets document output format to HTML

```
format: html
```

Quarto · Week 4

format: pdf

sets document output format to PDF

```
format: pdf
```

Quarto · Week 4

format: docx

sets document output format to Word

```
format: docx
```

Quarto · Week 4

toc: true

adds a table of contents

```
toc: true
```

Quarto · Week 4

 # Heading

creates a heading

```
# Heading
```

Quarto · Week 4

bold

formats text as bold

```
**bold**
```

Quarto · Week 4

italic

formats text as italic

```
*italic*
```

Quarto · Week 4

Link

creates a hyperlink

```
[Google](https://www.google.com)
```

Quarto · Week 4

List item

creates an unordered list

```
- Item 1
- Item 2
- Item 3
```

Quarto · Week 4

1. Ordered item

creates an ordered list

```
1. First
2. Second
3. Third
```

Quarto · Week 4

Blockquote

creates a blockquote

```
> This is a blockquote
```

Quarto · Week 4

Inline code

formats text as code

```
The top response was `r
top_response`
```

Quarto · Week 4

Quarto

R for the **Rest of Us**

Code block

creates a chunk of R code in a Quarto document

```
```{r}
library(tidyverse)
```
```

Quarto · Week 4

Horizontal line

creates a horizontal line

```
---
```

Quarto · Week 4

Image

inserts an image

```
![Quarto Logo](quarto.png)
```

Quarto · Week 4

#| echo: false

code chunk option to hide code but shows output

```
echo: false
```

Quarto · Week 4

#| eval: false

code chunk option to prevent code execution

```
#| eval: false
```

Quarto · Week 4

#| warning: false

code chunk option to hide warnings

```
#| warning: false
```

Quarto · Week 4

#| message: false

code chunk option to hide messages

```
#| message: false
```

Quarto · Week 4

#| cache:

code chunk option to cache code output

```
#| cache: true
```

Quarto · Week 4

#| fig-align: 'center'

code chunk option to align figure in the center

```
#| fig-align: "center"
```

Quarto · Week 4

#| fig-cap: 'Caption'

code chunk option to add a caption to a figure

```
#| fig-cap: "Scatterplot"
```

Quarto · Week 4

#| fig-width:

code chunk option to set figure width

```
#| fig-width: 6
```

Quarto · Week 4

#| fig-height:

code chunk option to set figure height

```
#| fig-height: 4
```

Quarto · Week 4

Quarto

R for the Rest of Us

`quarto_render()`

renders a Quarto document to the specified format

```
quarto_render("report.qmd", output_format = "html")
```

{quarto} Week 12

`tibble()`

creates a modern data frame

```
tibble(name = c("Alice", "Bob"), age = c(25, 30))
```

{tibble} Week 12

`pwalk()`

iterates over multiple lists, applying a function

```
pwalk(list(names, ages), ~ print(paste(.x, "is", .y, "years old")))
```

{purrr} Week 12

Tables

R for the Rest of Us

`flextable()`

creates a customizable table

```
flextable(data)
```

{flextable} Week 12

`gt()`

creates a gt table for formatted output

```
gt(data)
```

{gt} Week 12

`cols_label()`

renames columns in a gt table

```
data |> gt() |>
cols_label(column_name = “New Label”)
```

{gt} Week 12

`cols_width()`

sets column widths in a gt table

```
data |> gt() |>
cols_width(vars(column1) ~ px(100), vars(column2) ~ px(200))
```

{gt} Week 12

`cols_align()`

aligns columns in a gt table

```
data |> gt() |> cols_align(align = “center”, columns = vars(column_name))
```

{gt} Week 12

`tab_caption()`

adds a caption below a gt table

```
data |> gt() |>
tab_caption(“This is a table caption.”)
```

{gt} Week 12

`opt_interactive()`

makes the table interactive (sortable, searchable)

```
data |> gt() |>
opt_interactive()
```

{gt} Week 12