Unit 1: Tidyverse, reading/wrangling & Visualization

R and the Tidyverse(Ch.1)

Identify the different types of data analysis questions and categorize a question into the correct type.

Question type	Description	Example
Descriptive	A question that asks about summarized characteristics of a data set without interpretation (i.e., report a fact).	How many people live in each province and territory in Canada?
Exploratory	A question that asks if there are patterns, trends, or relationships within a single data set. Often used to propose hypotheses for future study.	Does political party voting change with indicators of wealth in a set of data collected on 2,000 people living in Canada?
Predictive	A question that asks about predicting measurements or labels for individuals (people or things). The focus is on what things predict some outcome, but not what causes the outcome.	What political party will someone vote for in the next Canadian election?
Inferential	A question that looks for patterns, trends, or relationships in a single data set **and** also asks for quantification of how applicable these findings are to the wider population.	Does political party voting change with indicators of wealth for all people living in Canada?
Causal	A question that asks about whether changing one factor will lead to a change in another factor, on average, in the wider population.	Does wealth lead to voting for a certain political party in Canadian elections?
Mechanistic	A question that asks about the underlying mechanism of the observed patterns, trends, or relationships (i.e., how does it happen?)	How does wealth lead to voting for a certain political party in Canadian elections?

- We only focus on the first 4
- Through summarization(helps with description questions), visualization (answers descriptive and exploratory questions), classification(predictive questions), regression(predicts a quantitative value for a new observation), clustering(exploratory questions), and estimation(inferential questions.)

Load the tidyverse package into R.

Rows are considered to be "observations,"
and columns as "variables,"

1. Load tidy verse

library(tidyverse)

dplyr: This package provides a set of functions for data manipulation tasks, such as filtering rows, selecting columns, arranging data, summarizing data, and joining multiple datasets.

ggplot2: ggplot2 is a powerful data visualization package that allows you to create high-quality, customizable graphs and plots. It follows the Grammar of Graphics approach, providing a structured and consistent way to build visualizations.

tidyr: tidyr provides functions for reshaping and tidying up data. It helps in transforming data between wide and long formats, dealing with missing values, and creating new variables based on existing ones.

readr: readr focuses on efficient and consistent methods for reading and writing data files, including CSV, TSV, and delimited text files. It offers faster alternatives to the base R functions for reading data.

purrr: purrr provides a set of functions for working with functional programming concepts in R. It enables iteration, mapping functions over data structures, and applying functions to groups of data.

tibble: tibble is an enhanced version of R's data frame, providing improved printing, subsetting, and other functionalities. It is designed to work well with the tidyverse packages and integrates seamlessly with other tools.

Read tabular data with read_csv.

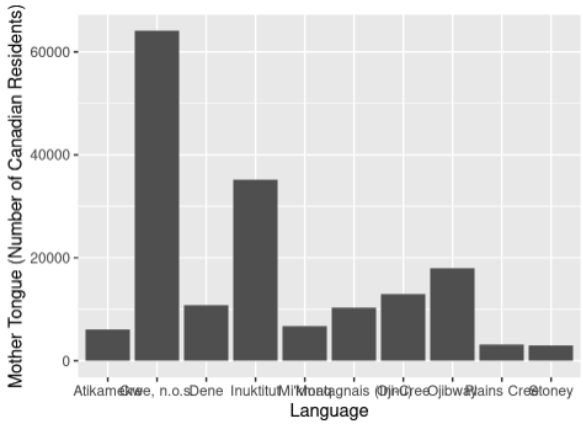
```
can_lang <- read_csv("data/can_lang.csv")
```

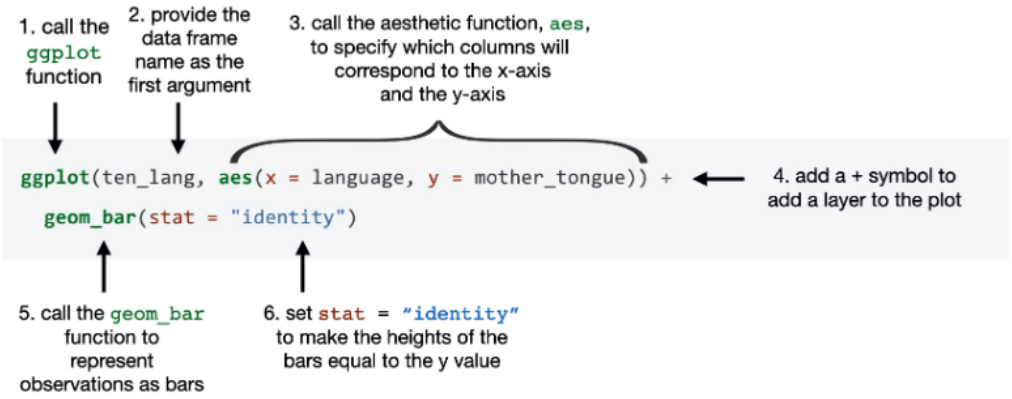
Create and organize subsets of tabular data using filter, select, arrange, and slice.

- ☐ Filter: obtain the subset of rows with desired values from a data frame.
- ☐ Select: obtain columns.
- ☐ Arrange: allows us to order the rows of a data frame by the values of a particular column. Arrange naturally goes ascending; use “desc()” to show descending
- ☐ Slice:which selects rows according to their row number

Visualize data with a ggplot bar plot..

```
ggplot(ten_lang, aes(x = language, y = mother_tongue)) +
  geom_bar(stat = "identity") +
  xlab("Language") +
  ylab("Mother Tongue (Number of Canadian Residents)")
```





Chapter 2

Define the following:

1. absolute file path : longer path that is specific to your computer
2. relative file path : shorter path, usable in other systems

```
happy_data <- read_csv("data/happiness_report.csv")
```

3. Uniform Resource Locator (URL):

- To read a excel from an URL

```
download.file("https://s3.amazonaws.com/happiness-report/2018/WHR2018Chapter2OnlineData.xls",
              destfile = "data/WHR2018Chapter2OnlineData.xls")
happy_df <- read_excel(path = "data/WHR2018Chapter2OnlineData.xls", sheet = 1)
happy_df
```

-

4. Read data into R using a relative path and a URL.:Uniform Resource Locator (URL)

```
url <- "https://raw.githubusercontent.com/UBC-DSCI/data/main/can_lang.csv"
canlang_data <- read_csv(url)
```

5. Compare and contrast the following functions:
   Read_csv : comma
   Read_tsv :tab
   Read_csv2 : ;
   Read_delim : , or tab

Read_excel : .xlsx

```
library(readxl)

canlang_data <- read_excel("data/can_lang.xlsx")
canlang_data
```

European countries).

Table 2.1: Summary of `read_*` functions

Data File Type	R Function	R Package
Comma (`,`) separated files	`read_csv`	`readr`
Tab (`\t`) separated files	`read_tsv`	`readr`
Semicolon (`;`) separated files	`read_csv2`	`readr`
Various formats (`.csv` , `.tsv`)	`read_delim`	`readr`
Excel files (`.xlsx`)	`read_excel`	`readxl`

> **Note:** `readr` is a part of the `tidyverse` package so we did not need to load this package separately since we loaded `tidyverse` .

Connect to a database using the DBI package's dbConnect function.

```
library(DBI)

conn_lang_data <- dbConnect(RSQLite::SQLite(), "data/can_lang.db")
```

> get the name of the tables from the data base

```
tables <- dbListTables(conn_lang_data)
tables
```

>To reference a table in the database (so that we can perform operations like selecting columns and filtering rows), we use the tbl function from the dbplyr package

```
library(dbplyr)

lang_db <- tbl(conn_lang_data, "lang")
lang_db
```

> R does not collect all the data, to collect everything its better to filter first efore you collect () to retrieve the entire data

Chapter 3

Define the term "tidy data."

1. Each variable(a characteristic, number, or quantity that can be measured. (Y, column) forms a column
2. each observation (all of the measurements for a given entity. (X, row)forms a row
3. each cell should be a single measurement./value (a single measurement of a single variable for a given entity.)

Discuss the advantages of storing data in a tidy data format.

- Helps others understand the data
- Helps you manipulate and visualize the data

Define what vectors, lists, and data frames are in R, and describe how they relate to each other.

Vector:a way of storing alist of entries ex. 1, 2, 3
numbers <- c("one", "two", "three")
Variable: a vector with a name ex. Numbers
List: elements in a list can be different

Describe the common types of data in R and their uses.

Table 3.1: Basic data types in R

Data type	Abbreviation	Description	Example
character	chr	letters or numbers surrounded by quotes	"1" , "Hello world!"
double	dbl	numbers with decimals values	1.2333
integer	int	numbers that do not contain decimals	1L, 20L (where "L" tells R to store as an integer)
logical	lgl	either true or false	`TRUE` , `FALSE`
factor	fct	used to represent data with a limited number of values (usually categories)	a `color` variable with levels `red` , `green` and `orange`

Recall and use the following functions for their intended data-wrangling tasks:

- [ ] c
- [ ] filter
- [ ] group_by
- [ ] select
- [ ] map
- [ ] mutate
- [ ] pull
- [ ] pivot_longer
- [ ] pivot_wider
- [ ] rowwise
- [ ] separate
- [ ] summarize

Recall and use the following operators for their intended data wrangling tasks:
==
%in%
!
&
|
|> and %>%

Chapter 4

Describe when to use the following kinds of visualizations to answer specific questions using a data set:

- scatter plots: visualize the relationship between two quantitative variables
- line plots: visualize trends with respect to an independent, ordered quantity (e.g., time)
- bar plots: visualize comparisons of amounts
- histogram plots: visualize the distribution of one quantitative variable (i.e., all its possible values and how often they occur)

Given a data set and a question, interpret the graph based on these questions

1. **Establihs a setting, scope, pose the question you are trying to answer / visualize.**
2. **Direction:** if the y variable tends to increase when the x variable increases, then y has a **positive** relationship with x. If y tends to decrease when x increases, then y has a **negative** relationship with x. If y does not meaningfully increase or decrease as x increases, then y has **little or no** relationship with x.
3. **Strength:** if the y variable *reliably* increases, decreases, or stays flat as x increases, then the relationship is **strong**. Otherwise, the relationship is **weak**. Intuitively, the relationship is strong when the scatter points are close together and look more like a "line" or "curve" than a "cloud."
4. **Shape:** if you can draw a straight line roughly through the data points, the relationship is **linear**. Otherwise, it is **nonlinear**.
5. **trends (lines):** Does a line describe the trend well? If so, the trend is linear, and if not, the trend is nonlinear. Is the trend increasing, decreasing, or neither? Is there a periodic oscillation (wiggle) in the trend? Is the trend noisy (does the line "jump around" a lot) or smooth?
6. **distributions (**scatters, histograms): How spread out are the data? Where are they centered, roughly? Are there any obvious "clusters" or "subgroups," which would be visible as multiple bumps in the histogram?
7. **distributions of two variables** (scatters): Is there a clear / strong relationship between the variables (points fall in a distinct pattern), a weak one (points fall in a pattern but there is some noise), or no discernible relationship (the data are too noisy to make any conclusion)?
8. **amounts (bars):** How large are the bars relative to one another? Are there patterns in different groups of bars?
9.

Given a visualization and a question, evaluate the effectiveness of the visualization and suggest improvements to better answer the question.

- Does it answer the question
- Is there a legend or label
- Are text, lines and symbols big enough and easy to read
- Colour scheme makes sense
- No redundant
- Minimize noice
- No overpotting
- Not overly zoomed in or out

Define the three key aspects of ggplot objects:

- aesthetic mappings: which tells ggplot how the columns in the data frame map to properties of the visualization. Colour to the data frame
- geometric objects: which specifies how the mapped data should be displayed
- Scales: limits to the x and y

Use the `ggplot2` package in R to create and refine the above visualizations using:

- geometric objects: `geom_point`, `geom_line`, `geom_histogram`, `geom_bar`, `geom_vline`, `geom_hline`
- scales: `xlim`, `ylim`
- aesthetic mappings: `x`, `y`, `fill`, `color`, `shape`
- labeling: `xlab`, `ylab`, `labs`
- font control and legend positioning: `theme`
- subplots: `facet_grid`

```
ggplot(can_lang, aes(x = most_at_home_percent,
                     y = mother_tongue_percent,
                     color = category,
                     shape = category)) +
  geom_point() +
  xlab("Language spoken most at home \n (percentage of Canadian residents)") +
  ylab("Mother tongue \n (percentage of Canadian residents)") +
  theme(text = element_text(size = 12),
        legend.position = "top",
```

```
      legend.direction = "vertical") +
scale_x_log10(labels = comma) +
scale_y_log10(labels = comma) +
scale_color_brewer(palette = "Set2")
```

Describe the difference in raster and vector output formats.

Raster images are represented as a 2-D grid of square pixels, each with its own color. Raster images are often compressed before storing so they take up less space

- JPEG, PNG, BMP, TIFF

Vector images are represented as a collection of mathematical objects (lines, surfaces, shapes, curves).

- A vector image takes space and time to load corresponding to how complex the image is, since the computer has to draw all the elements each time it is displayed
- SVG, EPS

☐ Use `ggsave` to save visualizations in `.png` and `.svg` format.

```
ggsave("img/faithful_plot.svg", faithful_plot)
```

Table 4.1: File sizes of the scatter plot of the Old Faithful data set when saved as differe

Image type	File type	Image size
Raster	PNG	0.15 MB
Raster	JPG	0.42 MB
Raster	BMP	3.15 MB
Raster	TIFF	9.44 MB
Vector	SVG	0.03 MB