**IMS Notes**

**1.1 + 1.2**

Summary statistic – single number summarizing data from a sample

Case (observational unit) – formal name for a row

Variable – formal name for a column

**Try to use a tidy data frame (allows new cases to be added as rows or new variables as columns)**

Discrete variable – can only take numerical values with jumps

Levels – possible values of categorical variables

Ordinal variable – categorical variable with special ordering

Diagram

Description automatically generated

Scatterplots – studies relationship between 2 numerical variables

**Association does not imply causation**

**2.1**

Simple Random Sample – each case in the population has an equal chance of being included and the cases in the sample are not related to each other

Convenience Sample – individuals who are easily accessible are more likely to be included in the sample

Sampling Methods:

1. Simple Random Sampling - each case in the population has an equal chance of being included in the final sample and knowing that a case is included in a sample does not provide useful information about which other cases are included
2. Stratified Sampling - The population is divided into groups called strata. The strata are chosen so that similar cases are grouped together, then a second sampling method, usually simple random sampling, is employed within each stratum
   1. Useful when the cases in each stratum are very similar with respect to the outcome of interest
3. Cluster Sample - we break up the population into many groups, called **clusters**. Then we sample a fixed number of clusters and include all observations from each of those clusters in the sample
4. Multistage Sample - collect a random sample within each selected cluster

**2.2 + 2.3**

Experiments – studies where the researchers assign treatments to cases

Randomized Experiment – when the assignment includes randomization

Principles of experimental design

1. Controlling – control other differences in the group
2. Randomization
   1. Confounding variable – variable associated with both the explanatory and response variables
3. Replication
   1. Replication Crisis - ongoing methodological crisis in which past findings from scientific studies in several disciplines have failed to be replicated
   2. **Pseudoreplication** occurs when individual observations under different treatments are heavily dependent on each other
4. Blocking – grouping individuals based on a variable

Reducing bias in human experiments

Control Groups and Treatment Groups

Blind study – patients are uninformed of their treatment

Double blind setup – researchers and patients do not know who is treated

Observational Studies

Observational data – data where no treatment has been explicitly applied

2 Forms

1. Prospective study – identifies individuals and collects info as events unfold
2. Retrospective studies – collect data after events have taken place

**R4DS Notes**

**Ch 3**

Grammar of graphics – coherent system for describing and building graphs

*ggplot2 package*

To import: *library(tidyverse)*

To create a *ggplot*

*ggplot(data = mpg) +*

*geom\_point(mapping = aes(x = displ, y = hwy))*

**Template** to use *ggplot*: *ggplot(data = <DATA>) +*

*<GEOM\_FUNCTION>(mapping = aes(<MAPPINGS>))*

Plot into facets: ggplot(data = mpg) +

geom\_point(mapping = aes(x = displ, y = hwy)) +

facet\_wrap(~ class, nrow = 2)

Plot into facet grids: *ggplot(data = mpg) +*

*geom\_point(mapping = aes(x = displ, y = hwy)) +*

*facet\_grid(drv ~ cyl)*

Display multiple geoms: *ggplot(data = mpg) +*

*geom\_point(mapping = aes(x = displ, y = hwy)) +*

*geom\_smooth(mapping = aes(x = displ, y = hwy))*

Updated *ggplot* **template**: *ggplot(data = <DATA>) +*

*<GEOM\_FUNCTION>(*

*mapping = aes(<MAPPINGS>),*

*stat = <STAT>,*

*position = <POSITION>*

*) +*

*<COORDINATE\_FUNCTION> +*

*<FACET\_FUNCTION>*

Data frame – rectangular collection of variables (columns) and observations (rows)

Can add a third variable to a 2D scatterplot by mapping the *ggplot* to an aesthetic

Aesthetic – visual property of the objects in your plot (i.e size, shape, or color of points on plot)

To map an aesthetic variable, associate the name of the aesthetic to the name of the variable

*Alpha* aesthetic controls transparency of points

*Shape* aesthetic controls shape of points

*Color* aesthetic controls color of points

Facets – subplots that each display one subset of data

Geom – geometrical object that a plot uses to represent data

Scatterplots use *geom\_point*

*Geom\_smooth*

*Geom\_bar*

*Stat\_count*

*Stat\_summary*

Using the *colour* and *fill* aesthetic

3 options to avoid stacked bar chart: “identity”, “dodge”, “fill”

*position = "identity"* will place each object exactly where it falls in the context of the graph

*position = "fill"* works like stacking, but makes each set of stacked bars the same height. This makes it easier to compare proportions across groups

*position = "dodge"* places overlapping objects directly beside one another. This makes it easier to compare individual values.

*Position = “jitter”* adds a small amount of random noise to each point. This spreads the points out

Coordinate Systems

*coord\_flip()*switches the x and y axes. This is useful (for example), if you want horizontal boxplots.

*coord\_quickmap()*sets the aspect ratio correctly for maps.

*coord\_polar()*uses polar coordinates

**Ch 4**

Assignment statement syntax: *object\_name <- value*

Calling functions: *function\_name(arg1 = val1, arg2 = val2, … )*

**Ch 5**

Tibble – data frames that are slightly tweaked to work better in tidyverse

Types of variables: int, dbl, chr, dttm (date-times), lgl (logical vectors), fctr (factors), date

[filter()](https://rdrr.io/r/stats/filter.html) allows you to subset observations based on their values. The first argument is the name of the data frame. The second and subsequent arguments are the expressions that filter the data frame

x %in% y. This will select every row where x is one of the values in y

*is.na(x)* will help determine if a value is missing

arrange() works similarly to [filter()](https://rdrr.io/r/stats/filter.html) except that instead of selecting rows, it changes their order. It takes a data frame and a set of column names (or more complicated expressions) to order by. If you provide more than one column name, each additional column will be used to break ties in the values of preceding columns

*desc()* re-orders column in descending order

select() allows you to rapidly zoom in on a useful subset using operations based on the names of the variables

* Starts\_with(“…”)
* Ends\_with(“…”)
* Contains(“…”)
* matches("(.)\\1"): selects variables that match a regular expression. This one matches any variables that contain repeated characters. You’ll learn more about regular expressions in [strings](https://r4ds.had.co.nz/strings.html#strings)
* num\_range("x", 1:3): matches x1, x2 and x3

select() can be used to rename variables, but it’s rarely useful because it drops all of the variables not explicitly mentioned. Instead, use rename(), which is a variant of select() that keeps all the variables that aren’t explicitly mentioned

mutate() always adds new columns at the end of your dataset so we’ll start by creating a narrower dataset so we can see the new variables. Remember that when you’re in RStudio, the easiest way to see all the columns is [View()](https://rdrr.io/r/utils/View.html)

summarise(). It collapses a data frame to a single row