**#EXPLORE YOUR DATA**

-Read a CSV file

# Load readr

library(readr)

# Create bakeoff but skip first row

bakeoff <- read\_csv("bakeoff.csv", skip =1)

# Print bakeoff

bakeoff

-Assign Missing Value

# Load dplyr

library(dplyr)

# Filter rows where showstopper is UNKNOWN

bakeoff %>%

    filter(showstopper == "UNKNOWN")

# Edit to add list of missing values

bakeoff <- read\_csv("bakeoff.csv", skip = 1,

                    na = c("", "NA", "UNKNOWN"))

# Filter rows where showstopper is NA

bakeoff %>%

filter(is.na(showstopper))

-Know Your data

Function glimpse() untuk melihat semua data. Karena kao hanya pakai function print saja, jika kolomnya banyak ada yg tidak terlihat.

-use skim() : untuk melihat summary per variable(missing data, complete data, n, min\_length, max\_length, empty, n\_unique)

**\*library(skimr)**

**\*skim(nama\_table)**

-Summarize the data

# Load skimr

library(skimr)

# Edit to filter, group by, and skim

bakeoff %>%

  filter(!is.na(us\_season)) %>%

  group\_by(us\_season)  %>%

  skim()

-count with your data

**\*bakers %>%  
\*group\_by(series)**

**\*summarize(n=n())**

-count 2 variable

**\*bakers %>%  
\*group\_by(aired\_us,series)**

**\*mutate(prop\_bakers = n/sum(n))**

-count to rollup a level

**\*bakers %>%**

**\*count(aired\_us, series) %>%**

**\*count(aired\_us)**

-Count Specific :

# Count whether or not star baker

bakeoff %>%

  count(result == "SB")

result

# A tibble: 2 x 2

`result == "SB"` n

<lgl> <int>

1 FALSE 488

2 TRUE 61

-Count Episode by series

# Add second count by series

bakeoff %>%

  count(series, episode) %>%

  count(series)

-Count bakers by series

# Count the number of rows by series and baker

bakers\_by\_series <- bakeoff %>%

  count(series, baker)

# Print to view

bakers\_by\_series

# Count again by series

bakers\_by\_series %>%

  count(series)

# Count again by baker

bakers\_by\_series %>%

count(baker, sort =TRUE)

-Plot Counts

ggplot(bakeoff, aes(episode)) +

    geom\_bar() +

    facet\_wrap(~series)

**#TAME YOUR DATA**

-convert text to number : **parse\_number (“36 years”)**

-mengubah satu kolom dari character to number :

**\*library(readr)**

**\*bakers\_tame <- read\_csv(file=”bakers.csv”, col\_types = cols(age = col\_number()))**

**\*bakers\_tame %>% slice(1:4) -> melihat 4 row pertama**

-format date : **parse\_date(“14 August 2012”, format = “%d %B %Y”)**

-mengubah satu kolom menjadi format date yg kita inginkan :

**\*bakers <- read\_csv(“bakers.csv”, col\_types = cols(last\_date\_uk=col\_date(format = “%d %B %Y”)))**

-Cast column to a date

# Find format to parse uk\_airdate

parse\_date("17 August 2010", format = "%d %B %Y")

# Edit to cast uk\_airdate

desserts <- read\_csv("desserts.csv",

                     col\_types = cols(

                       uk\_airdate = col\_date(format = "%d %B %Y")

                     )

                    )

# Arrange by descending uk\_airdate

desserts %>%

  arrange(desc(uk\_airdate))

-cast a column to a number

# Edit code to fix the parsing error

desserts <- read\_csv("desserts.csv",

                      col\_types = cols(

                        uk\_airdate = col\_date(format = "%d %B %Y"),

                        technical = col\_number()

                      ),

                        na = c("", "NA", "N/A")

                     )

# View parsing problems

problems(desserts) -> melihat jika ada masalah dalam casting

-cast a column as a factor

# Cast result a factor

desserts <- read\_csv("desserts.csv",

                     na = c("", "NA", "N/A"),

                     col\_types = cols(

                       uk\_airdate = col\_date(format = "%d %B %Y"),

                       technical = col\_number(),

                       result = col\_factor(levels = NULL)

                     )

                    )

# Glimpse to view

glimpse(desserts)

-Recode Values :

**\*young\_bakers %>%**

**\*mutate(stu\_label = recode(student, ‘0’ = “other”, .default =”student”))**

-if we want to change from others to NA :

**\*young\_bakers %>%**

**\*mutate(stu\_label, recode(student,’0’ = NA\_character\_, .default = “student”))**

-Record Multiple Values :

**\*young\_bakers %>%**

**\*mutate(stu\_label = recode(student, ‘0’ = “other”, ‘2’=”law/med”, .default =”student”))**

-convert to NA only with na\_if :

**\*young\_bakers %>%**

**\*mutate(student = na\_if(student, 0))**

-Recode a character variable :

# Count rows grouping by nut variable

desserts %>%

    count(nut, sort = TRUE)

# Edit code to recode "no nut" as missing

desserts\_2 <- desserts %>%

  mutate(nut = recode(nut, "filbert" = "hazelnut",

                           "no nut" = NA\_character\_))

# Count rows again

desserts\_2 %>%

    count(nut, sort = TRUE)

-Recode a numeric variable

# Edit to recode tech\_win as factor

desserts <- desserts %>%

  mutate(tech\_win = recode\_factor(technical, `1` = 1,

                                  .default = 0))

# Count to compare values

desserts %>%

  count(technical == 1, tech\_win)