# An Advanced Introduction to R

Kazuharu Yanagimoto

January 13, 2023

# Project Based Workflow

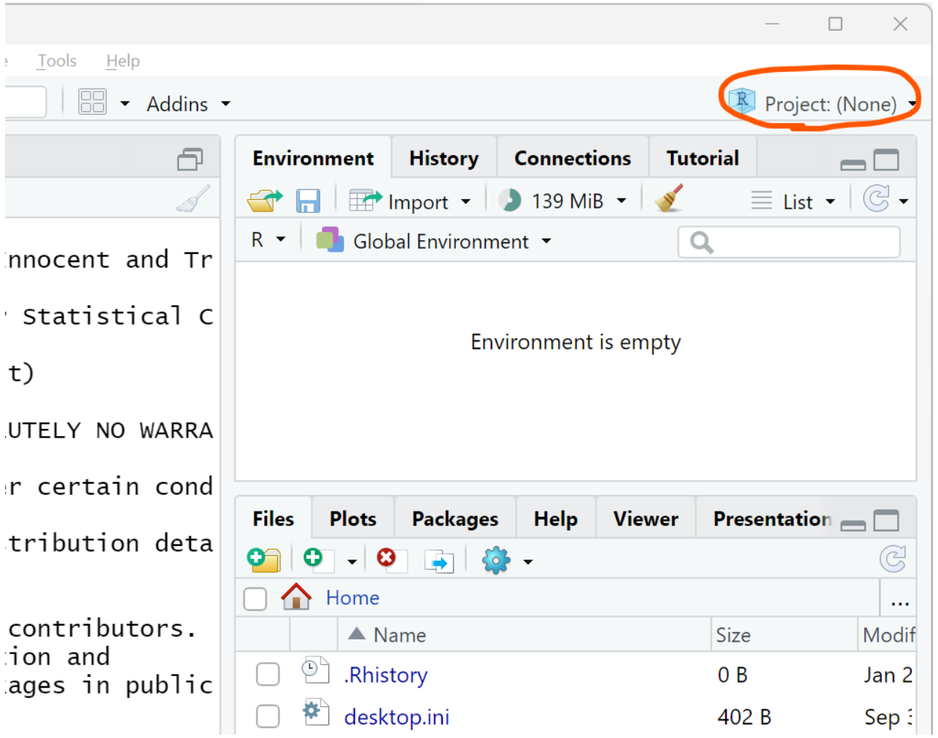
Q. Why Don't Your Codes Work on My Computer?

A. Conflicts in *Path* or *Package Version*

A. You don't use `here` and `renv` under **R projct**

# R Project

Have you ever click this button?




## You should ALWAYS use R Project!

# Why Do We Need to Use R Project?

## **Path** Manager


## **Package** Manager

# Always Use `here` for Paths

The function `here::here()` treats the proejct directory as the root directory.

```
here::here()
```

```
[1] "/home/rstudio/workshop-r-2022"
```

You should always specify the path by `here::here()`

```
data <- readr::read_csv(
  here::here("data/tiny.csv")
)
```

It works in Windows, Mac, Linux (of course, in a Docker environment)

# Remember...

> If the first line of your R script is
> `setwd("C:\Users\jenny\path\that\only\I\have")`
>
> I* will come into your office and SET YOUR COMPUTER ON FIRE 🔥.
>
> –Bryan (2018)

# `renv` Is Smarter than Us

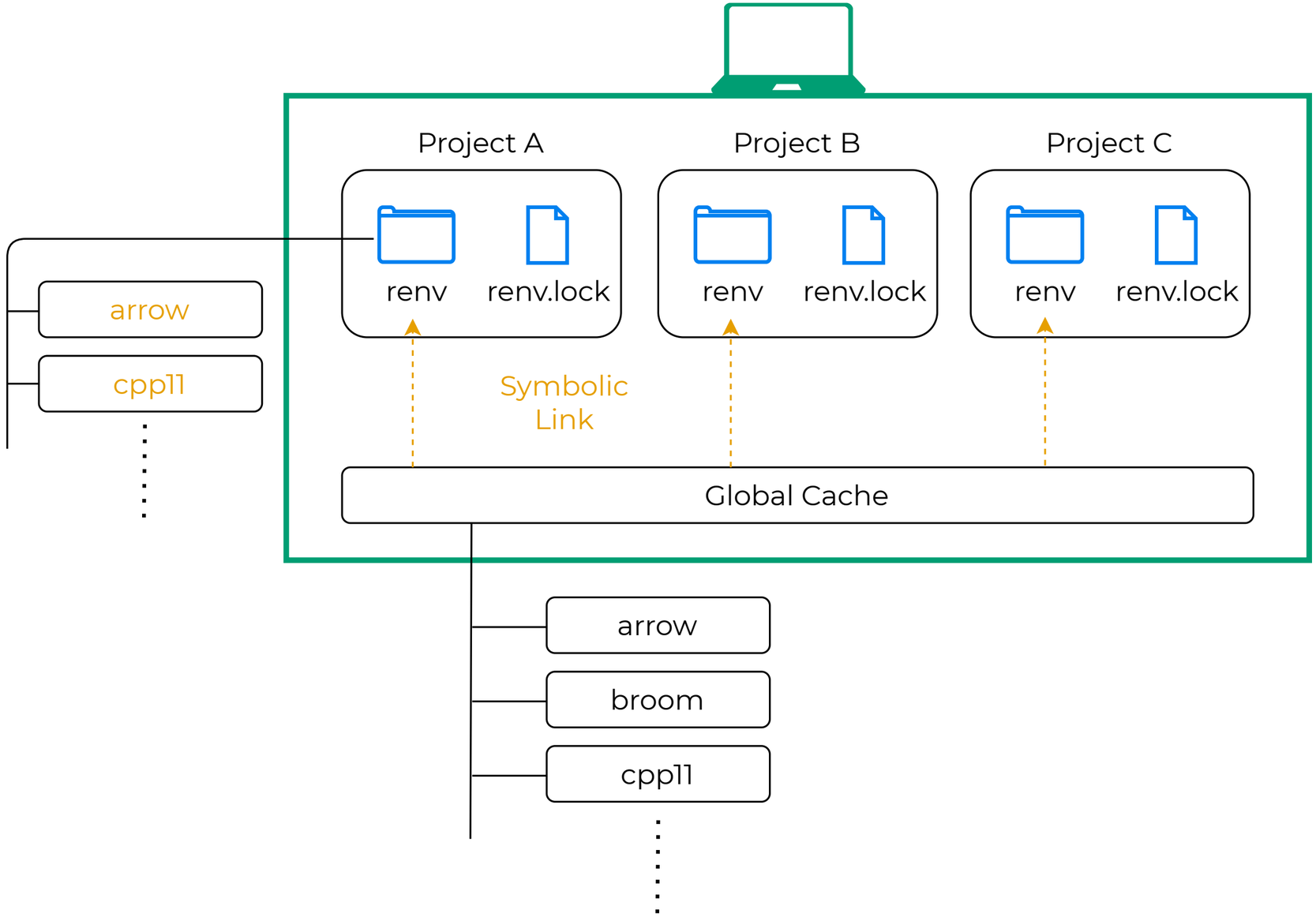
- Init the environment with `renv::init()`. It creates `renv/` and `renv.lock` file
- At some point, you can record your package and its version information with `renv::snapshot()`
- Your collaborater can install the packages just by `renv::restore()`

renv.lock

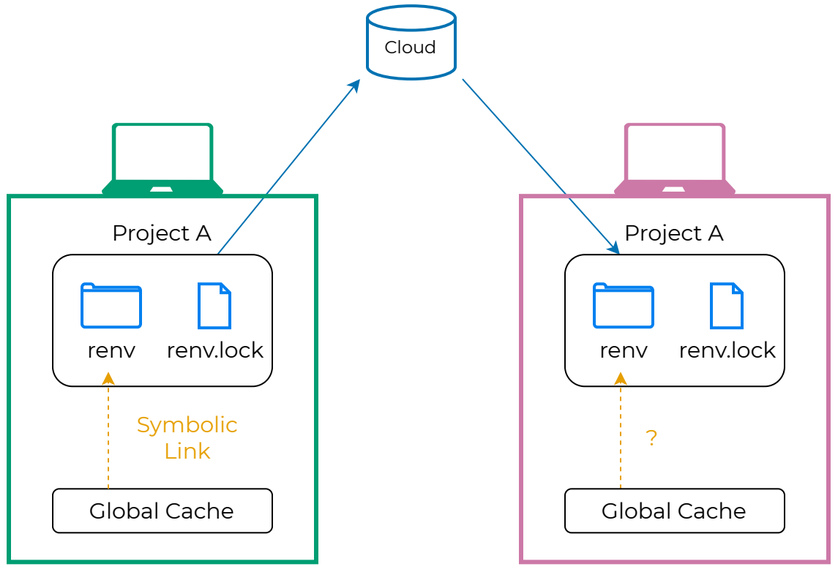
```
{
  "R": {
    "Version": "4.2.2",
    "Repositories": [
      {
        "Name": "CRAN",
        "URL": "https://packagemanager.posi
      }
    ]
  },
  "Packages": {
    "DBI": {
      "Package": "DBI",
      "Version": "1.1.3",
      "Source": "Repository",
      "Repository": "RSPM",
      "Hash": "b2866e62bab9378c3cc9476a195
      "Requirements": []
```

But Dropbox might ruin...

# (Advanced) How renv Works in Background



# (Advanced) `renv` with Cloud Storage




## Problem

- `renv.lock` is necessary and sufficient
- `renv` folder should not be shared (broken symbolic link)
- Need to sync-ignore (e.g. Dropbox)
- Packages in `renv` are *git-ignored* by default

# (Advanced) Docker

Problems `renv` can solve are only packages. They may come from differences in

- R versions ⇒ Always use the latest version of R
- Non-R dependencies (e.g., geospatial packages) ⇒ Docker can solve
- OS (only Windows binary produces bugs...) ⇒ Docker can solve

## Docker

- A virtual machine. Write a blueprint (Dockerfile) including information of OS (Linux), Application (R and others), and Packages
- If you work on Docker, others can *perfectly* replicate your environment

# Handson

1. Clone (or download) the course repositiory
2. Open the course project (`workshop-r-2022.Rproj`)
3. Run `renv::restore()` in R console
4. Confirm you can run any file in `code/`

> ⚠ Warning
>
> Please make sure if you are using the latest R version 4.2.2 (2022-10-31).

# Cleaning Strategy

# Fundamental Theorem of Readability

> ⓘ Fundamental Theorem of Readability (Boswell and Foucher 2011)
>
> Code should be written to minimize the time it would take for someone else to understand it.

$$\text{Code} := \arg\min_{c \in \mathcal{C}} \mathbb{E}_i \left[ R_i(c) \right]$$

where

- $\mathcal{C}$: Set of codes that work
- $i$: A potential reader including yourself at a different time point
- $R_i(c)$: Time taken by person $i$ to understand code $c$

# Naming

For readability, you need to name variables **informatively** and **non-misleadingly**

| | 🦁 Good | 🤦 Bad |
|---|---|---|
| Bool | is_female, has_kids | female, no_kids |
| Category | industry8, emp3 | industry, emp_status |
| Bins | age_bin5, wage_bin10 | age, wage |

# Naming

For readability, you need to name variables **informatively** and **non-misleadingly**

| | 🦁 Good | 🤦 Bad |
|---|---|---|
| Bool | is_female, has_kids | female, no_kids |
| Category | industry8, emp3 | industry, emp_status |
| Bins | age_bin5, wage_bin10 | age, wage |

## Boolean

- `is_*`, `has_*`, `should_*` indicates the type boolean.
- Starting with `not_*`/`no_*` increases a step of recognition

# Naming

For readability, you need to name variables **informatively** and **non-misleadingly**

| | 🦁 Good | 🤦 Bad |
|---|---|---|
| Bool | is_female, has_kids | female, no_kids |
| Category | industry8, emp3 | industry, emp_status |
| Bins | age_bin5, wage_bin10 | age, wage |

## Categorical

- Attached number indicates if it is *categorical* and its *number*

# Naming

For readability, you need to name variables **informatively** and **non-misleadingly**

| | 🦁 Good | 🤦 Bad |
|---|---|---|
| Bool | is_female, has_kids | female, no_kids |
| Category | industry8, emp3 | industry, emp_status |
| Bins | age_bin5, wage_bin10 | age, wage |

**Bins** of continuous variables

- Need to avoid the confusion with its continuous variable
- Attached number shows the width of the bin

# Rename at Once

```r
raw <- read_delim(here("data/raw/accident_bike/txt/year=2022/file.txt"),
         delim = ";", show_col_types = FALSE)
```

```
Rows: 42,547
Columns: 5
$ num_expediente <dbl> 2.022e+04, 2.022e+04, 2.022e+05, 2.022e+05, 2.022e+05, …
$ fecha          <chr> "01/01/2022", "01/01/2022", "01/01/2022", "01/01/2022",…
$ hora           <time> 01:30:00, 01:30:00, 00:30:00, 00:30:00, 00:30:00, 01:5…
$ localizacion   <chr> "AVDA. ALBUFERA, 19", "AVDA. ALBUFERA, 19", "PLAZA. CAN…
$ numero         <chr> "19", "19", "2", "2", "2", "53", "53", "728", "728", "+…
```

```r
code <- read_csv(here("data/translate/accident_bike.csv"),
                 show_col_types = FALSE)
renamed <- raw |>
  rename_at(vars(code$spanish), ~code$english)
```

```
Rows: 42,547
Columns: 5
$ id_1922    <dbl> 2.022e+04, 2.022e+04, 2.022e+05, 2.022e+05, 2.022e+05, 2.02…
$ date       <chr> "01/01/2022", "01/01/2022", "01/01/2022", "01/01/2022", "01…
$ hms        <time> 01:30:00, 01:30:00, 00:30:00, 00:30:00, 00:30:00, 01:50:00…
$ street     <chr> "AVDA. ALBUFERA, 19", "AVDA. ALBUFERA, 19", "PLAZA. CANOVAS…
$ num_street <chr> "19", "19", "2", "2", "2", "53", "53", "728", "728", "+0050…
```

| spanish | english |
|---|---|
| num_expediente | id_1922 |
| fecha | date |
| hora | hms |
| localizacion | street |
| numero | num_street |
| cod_distrito | code_district |
| distrito | district |
| tipo_accidente | type_accident |
| estado_meteorológico | weather |
| tipo_vehiculo | type_vehicle |
| tipo_persona | type_person |
| rango_edad | age_c |
| sexo | gender |
| cod_lesividad | code_injury8 |
| lesividad | injury8 |
| coordenada_x_utm | coord_x |
| coordenada_y_utm | coord_y |
| positiva_alcohol | positive_alcohol |
| positiva_droga | positive_drug |

# Type: Date & Time

`lubridate` provides strong date-parsering functions.

```
lubridate::ymd("2021/08/31")
```

```
[1] "2021-08-31"
```

```
lubridate::mdy("Sep. 10, 19")
```

```
[1] "2019-09-10"
```

```
lubridate::dmy_hm("02/04/1999 16:00", tz="America/New_York")
```

```
[1] "1999-04-02 16:00:00 EST"
```

```r
renamed |> select(date, hms) |> head()
```

```
# A tibble: 6 × 2
  date       hms
  <chr>      <time>
1 01/01/2022 01:30
2 01/01/2022 01:30
3 01/01/2022 00:30
4 01/01/2022 00:30
5 01/01/2022 00:30
6 01/01/2022 01:50
```

```r
renamed |>
  mutate(time = lubridate::dmy_hms(str_c(date, hms), tz = "Europe/Madrid")) |>
  select(date, hms, time) |>
  head()
```

```
# A tibble: 6 × 3
  date       hms    time
  <chr>      <time> <dttm>
1 01/01/2022 01:30  2022-01-01 01:30:00
2 01/01/2022 01:30  2022-01-01 01:30:00
3 01/01/2022 00:30  2022-01-01 00:30:00
4 01/01/2022 00:30  2022-01-01 00:30:00
5 01/01/2022 00:30  2022-01-01 00:30:00
6 01/01/2022 01:50  2022-01-01 01:50:00
```

# Type: Categorical Variables

```
renamed |>
  mutate(
    type_person = recode_factor(type_person,
        "Conductor" = "Driver",
        "Pasajero" = "Passenger",
        "Peatón" = "Pedestrian",
        "NULL"= NULL)) |>
  janitor::tabyl(type_person)
```

```
type_person     n    percent
     Driver 34567 0.81244271
  Passenger  6503 0.15284274
 Pedestrian  1477 0.03471455
```

`recode_factor()` finishes:

1. Define as *factor* variables
2. Order *factor* variable
3. Rename & Translate (labels in plots & tables)
4. Handle `NA` values (next slide)

# Handle NA Values

Some datasets include NA values as string format

```
1 unique(renamed$weather) # "Se desconoce" is also essentially NA
```

```
[1] "Despejado"      "NULL"           "Se desconoce"   "Lluvia débil"
[5] "Nublado"        "LLuvia intensa" "Granizando"     "Nevando"
```

## Solution 1: Define NA values when you load

```
1 sol1 <- read_delim(here("data/raw/accident_bike/txt/year=2019/file.txt"),
2                    delim = ";", show_col_types = FALSE,
3                    na = c("", "NA", "NULL", "Se desconoce", "Desconocido")) |>
4         rename(weather = "estado_meteorológico")
5
6 unique(sol1$weather)
```

```
[1] "Despejado"      NA               "Lluvia débil"   "Nublado"
[5] "LLuvia intensa" "Granizando"     "Nevando"
```

Cannot use when specific numbers as NA values (9, 99,...)

# Solution2: na_if()

```
renamed |>
  mutate(
    weather_old = weather,# Presentation Purpose
    weather = na_if(weather, "Se desconoce"),
    weather = na_if(weather, "NULL"),
    ) |>
  select(weather_old, weather) |>
  head()
```

```
# A tibble: 6 × 2
  weather_old weather
  <chr>       <chr>
1 Despejado   Despejado
2 Despejado   Despejado
3 NULL        <NA>
4 NULL        <NA>
5 NULL        <NA>
6 Despejado   Despejado
```

Works for any case. But need to write for each `NA` value.

# Soltion 3: Recode as NULL

```
renamed |>
  mutate(
    weather_spanish = weather,# Presentation Purpose
    weather = recode_factor(weather,
        "Despejado" = "sunny",
        "Nublado" = "cloud",
        "Lluvia débil" = "soft rain",
        "Lluvia intensa" = "hard rain",
        "LLuvia intensa" = "hard rain",
        "Nevando" = "snow",
        "Granizando" = "hail",
        "Se desconoce" = NULL,
        "NULL" = NULL)) |>
  select(weather_spanish, weather) |>
  head()
```

```
# A tibble: 6 × 2
  weather_spanish weather
  <chr>           <fct>
1 Despejado       sunny
2 Despejado       sunny
3 NULL            <NA>
4 NULL            <NA>
5 NULL            <NA>
6 Despejado       sunny
```

Only works for categorical variables. But practically useful.

# Parquet Format

| | Speed | Size | Keep Type | Multi-Language |
|---|---|---|---|---|
| csv, tsv | ✗ | ✗ | ✗ | All |
| rds, RData | ✗ | ✔ | ✔ | ✗ |
| parquet | ✔ | ✔ | ✔ | Python, Julia, MATLAB, Stata,... |

You can find a benchmark in Kastrun (2022)

# arrow::read_parquet()

You can load parquet data as **column-information only**

```r
df <- arrow::read_parquet(
  here("data/cleaned/accident_bike.parquet"),
  as_data_frame = TRUE)

df
```

```
# A tibble: 168,574 × 23
   id_1922    date  hms   street num_s…¹ code_…² distr…³
type_…⁴ weather type_…⁵
   <chr>      <chr> <chr> <chr>  <chr>     <int> <chr>
<chr>   <fct>   <chr>
 1 2018S0178… 04/0… 9:10… CALL.… 1             1 Centro
Colisi… sunny   Motoci…
 2 2018S0178… 04/0… 9:10… CALL.… 1             1 Centro
Colisi… sunny   Turismo
 3 2019S0000… 01/0… 3:45… PASEO… 168          11 Caraba…
Alcance <NA>    Furgon…
 4 2019S0000… 01/0… 3:45… PASEO… 168          11 Caraba…
Alcance <NA>    Turismo
 5 2019S0000… 01/0… 3:45… PASEO… 168          11 Caraba…
Alcance <NA>    Turismo
```

```r
info <- arrow::read_parquet(
  here("data/cleaned/accident_bike.parquet"),
  as_data_frame = FALSE)

info
```

```
Table
168574 rows x 23 columns
$id_1922 <string>
$date <string>
$hms <string>
$street <string>
$num_street <string>
$code_district <int32>
$district <string>
$type_accident <string>
$weather <dictionary<values=string, indices=int32>>
$type_vehicle <string>
$type_person <dictionary<values=string, indices=int32>>
$age_c <dictionary<values=string, indices=int32>>
$gender <dictionary<values=string, indices=int32>>
```

# Release Parquet on Memory

```
 1  info |>
 2    collect()
```

```
# A tibble: 168,574 × 23
   id_1922    date  hms   street num_s…¹ code_…² distr…³
type_…⁴ weather type_…⁵
   <chr>      <chr> <chr> <chr>  <chr>     <int> <chr>
<chr>   <fct>   <chr>
 1 2018S0178… 04/0… 9:10… CALL.… 1             1 Centro
Colisi… sunny   Motoci…
 2 2018S0178… 04/0… 9:10… CALL.… 1             1 Centro
Colisi… sunny   Turismo
 3 2019S0000… 01/0… 3:45… PASEO… 168          11 Caraba…
Alcance <NA>    Furgon…
 4 2019S0000… 01/0… 3:45… PASEO… 168          11 Caraba…
Alcance <NA>    Turismo
 5 2019S0000… 01/0… 3:45… PASEO… 168          11 Caraba…
Alcance <NA>    Turismo
```

```
 1  info |>
 2    filter(is_hospitalized) |>
 3    select(time, gender, age_c, positive_alcohol) |>
 4    collect()
```

```
# A tibble: 8,724 × 4
   time                gender age_c positive_alcohol
   <dttm>              <fct>  <fct> <lgl>
 1 2019-01-01 03:50:00 Men    21-24 FALSE
 2 2019-01-01 08:05:00 Women  60-64 FALSE
 3 2019-01-01 22:15:00 Men    35-39 FALSE
 4 2019-01-01 12:29:00 Men    55-59 FALSE
 5 2019-01-02 15:00:00 Men    60-64 FALSE
 6 2019-01-02 15:00:00 Women  50-54 FALSE
 7 2019-01-02 20:45:00 Men    70-74 FALSE
 8 2019-01-03 00:42:00 Men    35-39 FALSE
 9 2019-01-03 10:30:00 Men    15-17 FALSE
10 2019-01-03 13:25:00 Men    30-34 FALSE
# … with 8,714 more rows
```

- `dplyr::collect()` releases the loaded parquet data on memory
- You can load them after `select()` or `filter()`
- Also, `group_by()` and `summarize()` are available
- Quite useful for **large datasets**

# Parquet with Partitioned Dataset

```
data/raw/accident_bike/parquet/
├── year=2019
│   └── part-0.parquet
├── year=2020
│   └── part-0.parquet
├── year=2021
│   └── part-0.parquet
└── year=2022
    └── part-0.parquet
```

```
info <- open_dataset(
          here("data/raw/accident_bike/parquet"))
info
```

```
FileSystemDataset with 4 Parquet files
num_expediente: string
fecha: string
hora: string
localizacion: string
numero: string
cod_distrito: int32
distrito: string
tipo_accidente: string
estado_meteorológico: string
tipo_vehiculo: string
tipo_persona: string
rango_edad: string
sexo: string
cod_lesividad: string
```

- Given this structure, `arrow::open_dataset()` loads them as one parquet file
- A Partitioning variable (`year`) becomes a new variable
- For more instructions, you can refer to Mock (2022)

# Cleaning Workflow

## 1. Naming

- Put **informative** and **non-misleading** names
- If necessary, translate the variable names
- You can use a correspondence table and rename variables at once

## 2. Determine Types

- *Date*: `lubridate` parsing functions
- *Categorical*: `recode_factor()`
- *NA-values*: `na_if()` and `recode_factor()`

## 3. Export

- Parquet format is better than any other data format
- Parquet makes it easy to handle large datasets

# Tips in Plots

# Data-ink Ratio

**Data-ink Ratio Principle (Tufte 2001)**

Maximize the data-ink ratio in a plot:

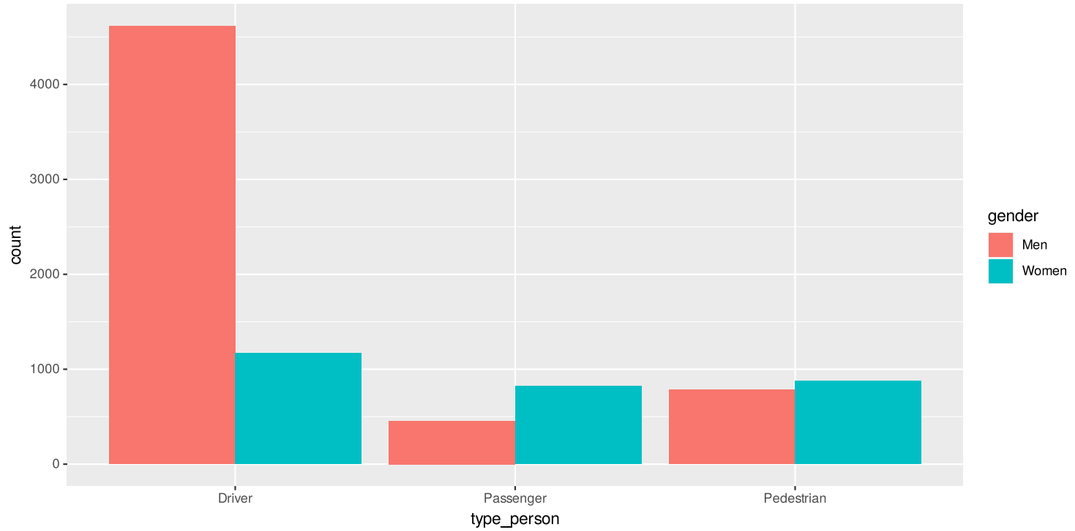
$$\text{Data-ink ratio} := \frac{\text{Data-ink}}{\text{Total ink used to print in the graphic}}$$

**Collolary**

Omit all the proportions of a graphic that can be erased without losing information

# Maximize Data-ink Ratio

```
accident_bike |>
  ggplot(aes(x = type_person, fill = gender)) +
  geom_bar(position = "dodge")
```



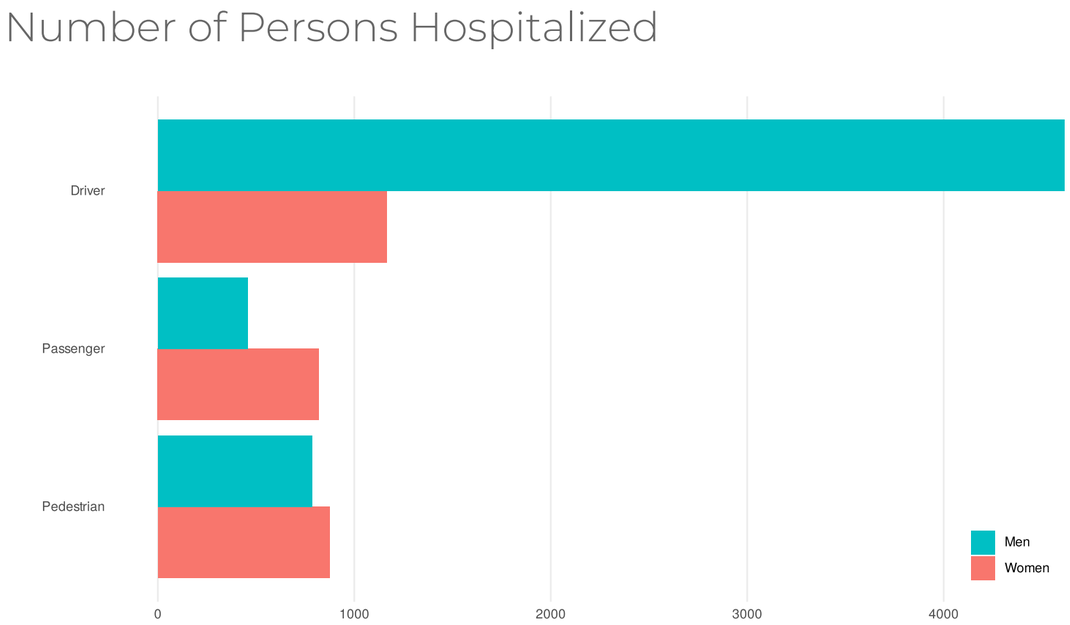
# Maximize Data-ink Ratio

```
accident_bike |>
  ggplot(aes(x = type_person, fill = gender)) +
  geom_bar(position = "dodge") +
  labs(x = NULL, y = NULL, fill = NULL) +
  theme_minimal() +
  theme(panel.grid.minor = element_blank(),
        panel.grid.major.x = element_blank())
```


- Omit axis label. The title of the plot can tell them
- Omit legend label. The label “gender” does not add any information
- Omit background grids

# More Readability: Order Bar Plot

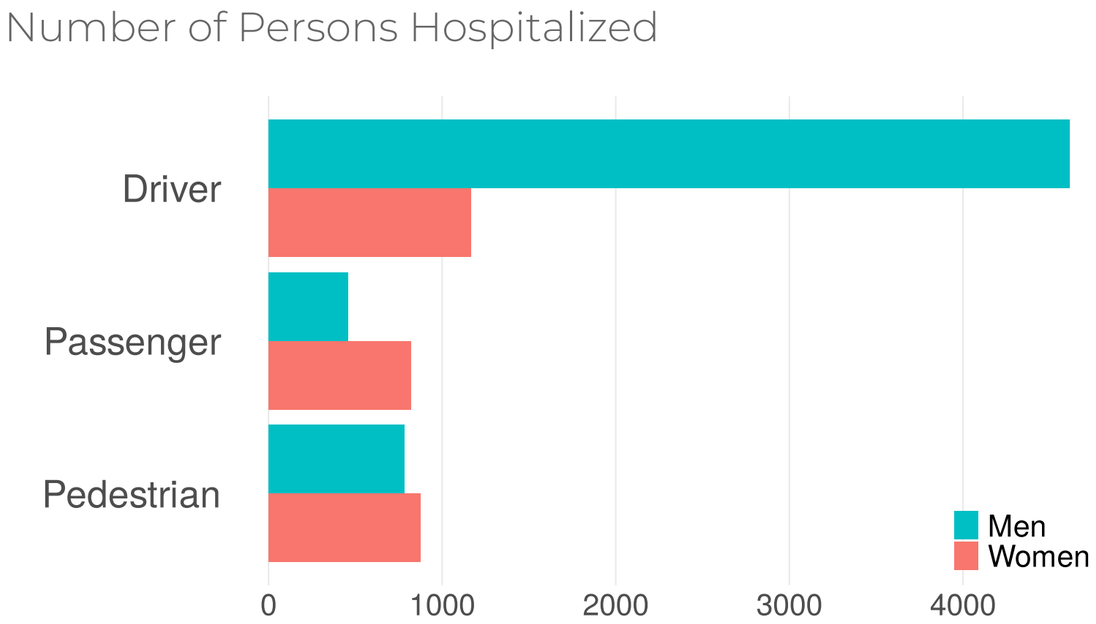
```
accident_bike |>
  ggplot(aes(x = fct_rev(type_person),
             fill = fct_rev(gender))) +
  geom_bar(position = "dodge") +
  coord_flip() +
  labs(x = NULL, y = NULL, fill = NULL) +
  theme_minimal() +
  theme(panel.grid.minor = element_blank(),
        panel.grid.major.y = element_blank(),
        legend.position = c(0.9, 0.1)) +
  guides(fill = guide_legend(reverse = TRUE))
```



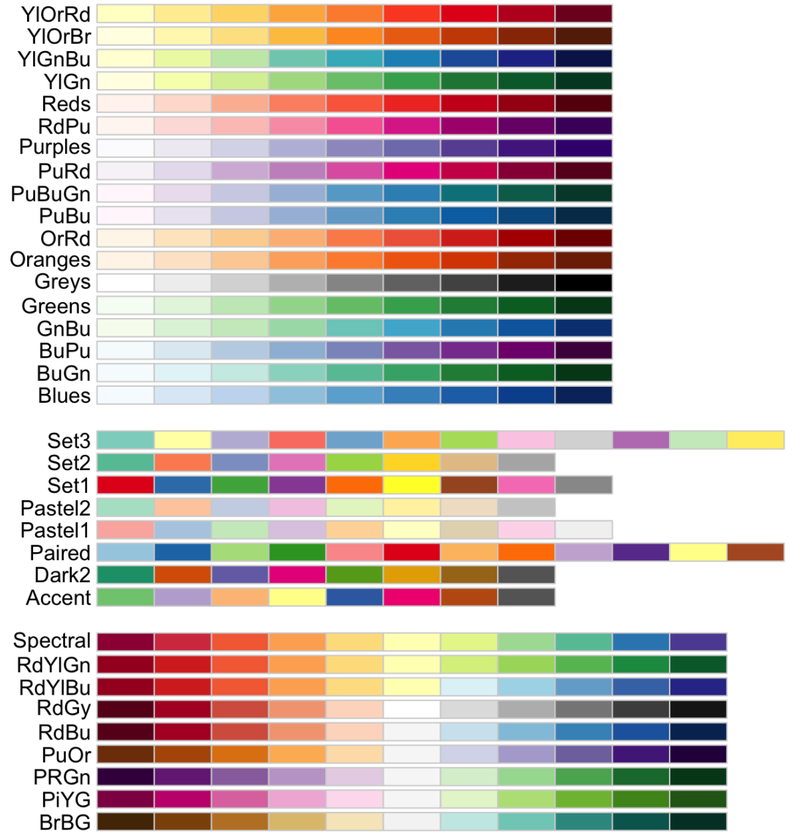

- Coord flipped. Reorder the factor variables
- Put legends inside the plot to make the plot bigger

# More Readability: Increase Font Size

```
accident_bike |>
  ggplot(aes(x = fct_rev(type_person),
             fill = fct_rev(gender))) +
  geom_bar(position = "dodge") +
  coord_flip() +
  labs(x = NULL, y = NULL, fill = NULL) +
  theme_minimal() +
  theme(panel.grid.minor = element_blank(),
        panel.grid.major.y = element_blank(),
        legend.position = c(0.9, 0.1),
        axis.text.x = element_text(size = 20),
        axis.text.y = element_text(size = 25),
        legend.text = element_text(size = 20)) +
  guides(fill = guide_legend(reverse = TRUE))
```

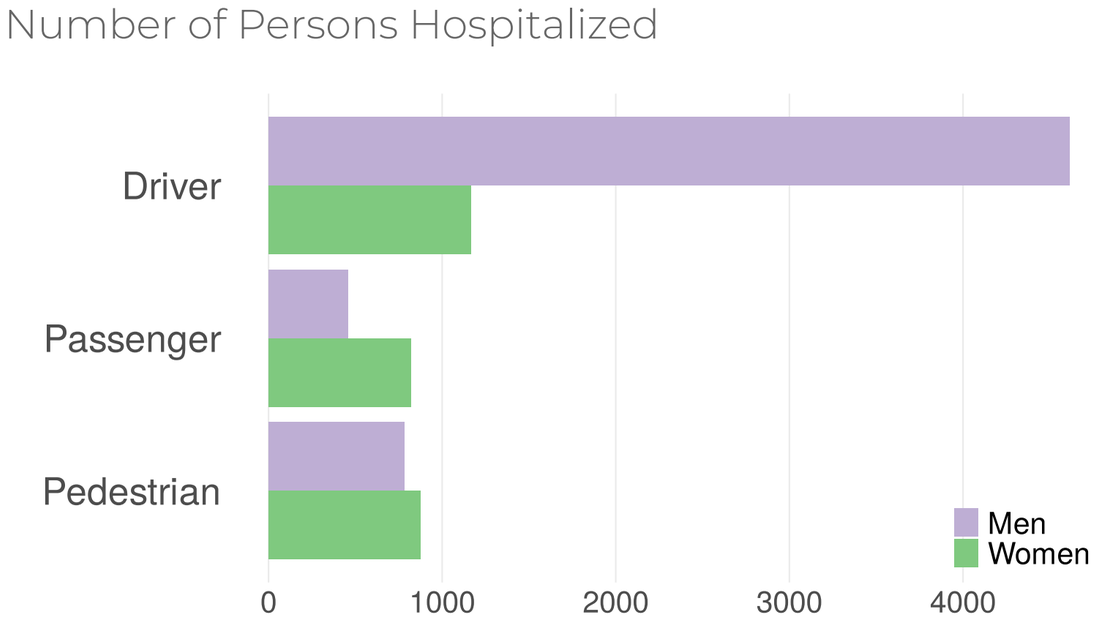


# R Color Brewer's Palettes



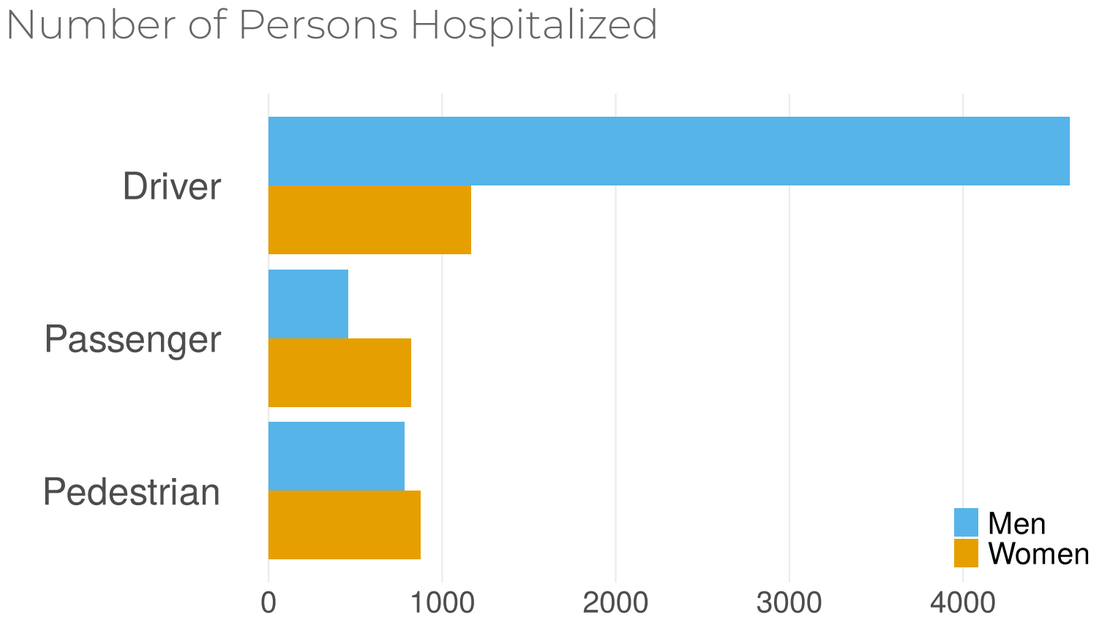
# R Color Brewer's Palettes

```r
accident_bike |>
  ggplot(aes(x = fct_rev(type_person),
             fill = fct_rev(gender))) +
  geom_bar(position = "dodge") +
  coord_flip() +
  labs(x = NULL, y = NULL, fill = NULL) +
  scale_fill_brewer(palette = "Accent") +
  theme_minimal() +
  theme(panel.grid.minor = element_blank(),
        panel.grid.major.y = element_blank(),
        legend.position = c(0.9, 0.1),
        axis.text.x = element_text(size = 20),
        axis.text.y = element_text(size = 25),
        legend.text = element_text(size = 20)) +
  guides(fill = guide_legend(reverse = TRUE))
```



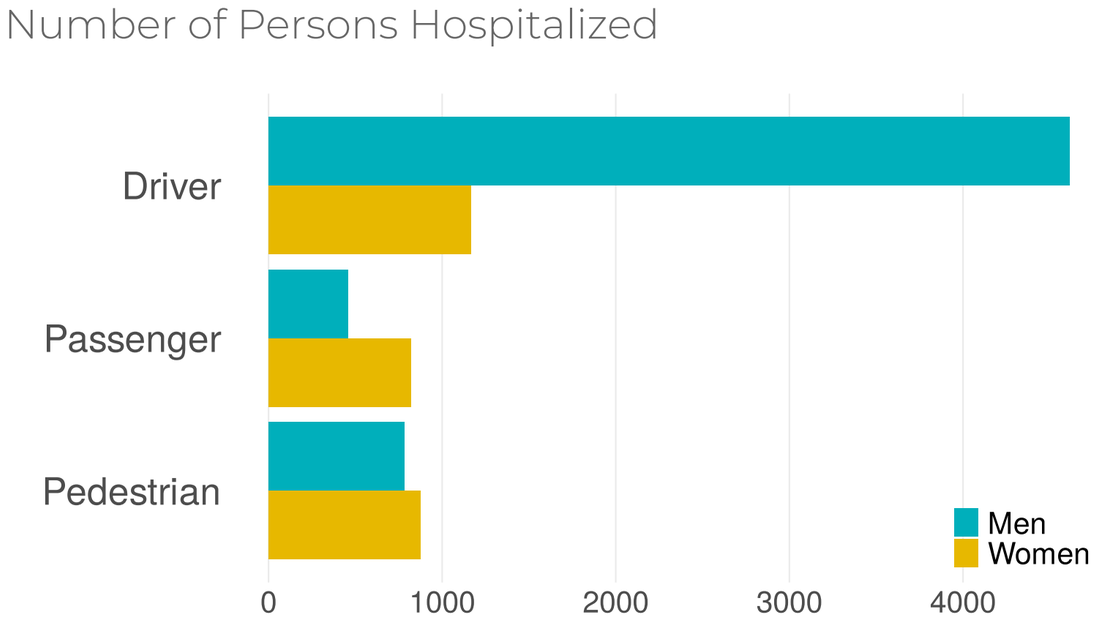
# Color-Safe Pallette: Okabe-Ito Palette

```r
accident_bike |>
  ggplot(aes(x = fct_rev(type_person),
             fill = fct_rev(gender))) +
  geom_bar(position = "dodge") +
  coord_flip() +
  labs(x = NULL, y = NULL, fill = NULL) +
  see::scale_fill_okabeito() +
  theme_minimal() +
  theme(panel.grid.minor = element_blank(),
        panel.grid.major.y = element_blank(),
        legend.position = c(0.9, 0.1),
        axis.text.x = element_text(size = 20),
        axis.text.y = element_text(size = 25),
        legend.text = element_text(size = 20)) +
  guides(fill = guide_legend(reverse = TRUE))
```



# Custom Palette

```r
accident_bike |>
  ggplot(aes(x = fct_rev(type_person),
             fill = fct_rev(gender))) +
  geom_bar(position = "dodge") +
  coord_flip() +
  labs(x = NULL, y = NULL, fill = NULL) +
  scale_fill_manual(values = c("#E7B800", "#00AFBB")) +
  theme_minimal() +
  theme(panel.grid.minor = element_blank(),
        panel.grid.major.y = element_blank(),
        legend.position = c(0.9, 0.1),
        axis.text.x = element_text(size = 20),
        axis.text.y = element_text(size = 25),
        legend.text = element_text(size = 20)) +
  guides(fill = guide_legend(reverse = TRUE))
```

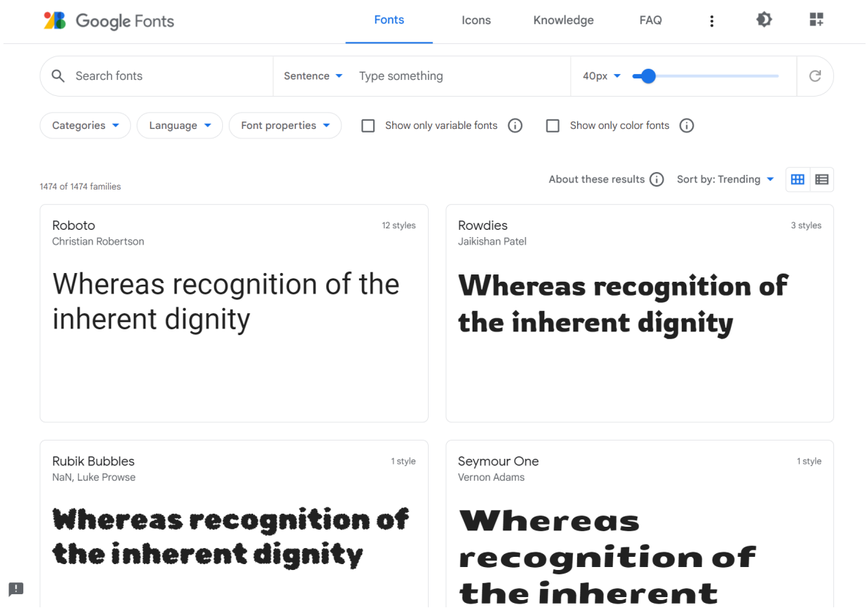


# Fonts

## Goolge Fonts

- You can download well-designed free fonts
- My recommendation: **Condensed** fonts

Roboto Condensed, Fira Sans Condensed, IBM Plex Sans Condensed,...



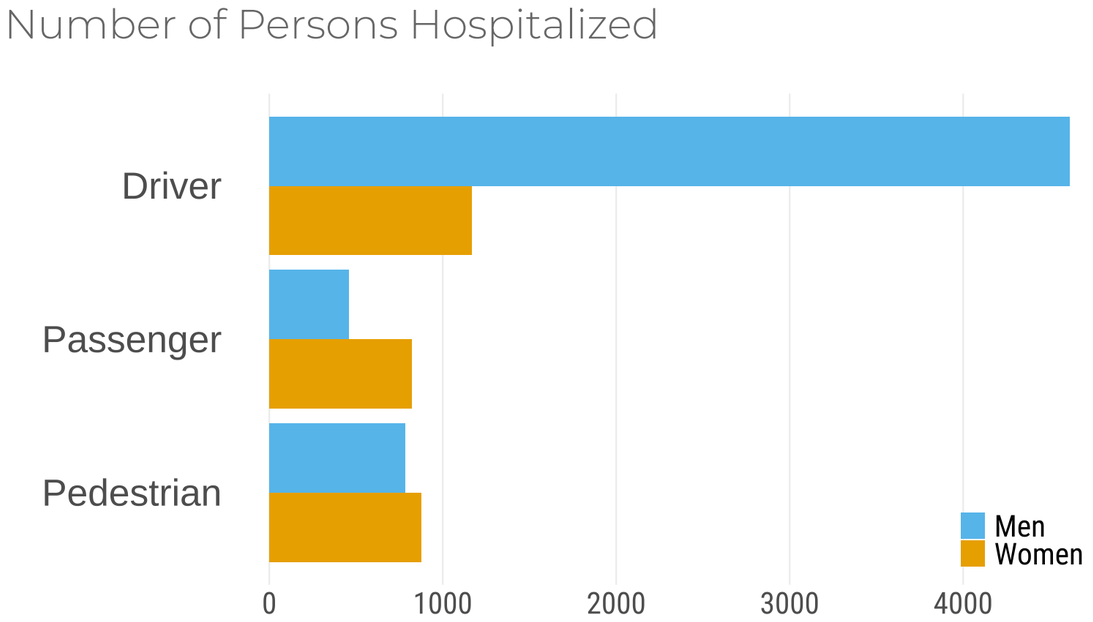

## showtext

- Your collaborators need to download the fonts
- `font_add_google()` and `showtext_auto()` automatically solve the problem

# Roboto Condensed

```
library(showtext)
font_base  <- "Roboto Condensed"
font_light <- "Roboto Condensed Light 300"
font_add_google(font_base, font_light)
showtext_auto()

accident_bike |>
  ggplot(aes(x = fct_rev(type_person), fill = fct_rev(g
  geom_bar(position = "dodge") +
  coord_flip() +
  labs(x = NULL, y = NULL, fill = NULL) +
  see::scale_fill_okabeito() +
  theme_minimal() +
  theme(panel.grid.minor = element_blank(),
        panel.grid.major.y = element_blank(),
        legend.position = c(0.9, 0.1),
        axis.text.x = element_text(size = 20, family =
        axis.text.y = element_text(size = 25, family =
        legend.text = element_text(size = 20, family =
```



# Global Options

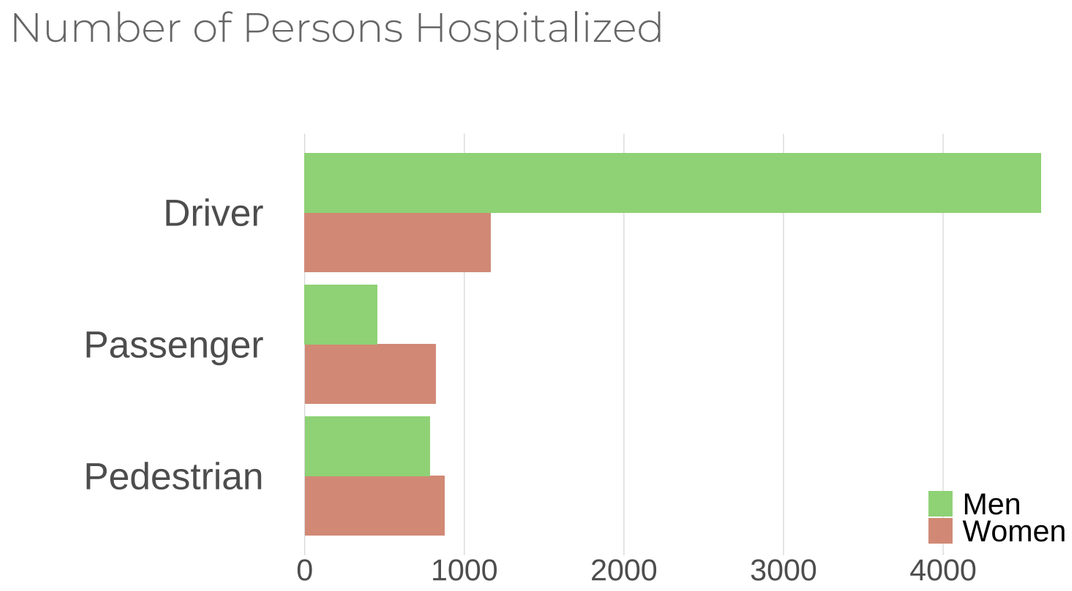
Don’t worry. You can set the default theme before plotting. (e.g. Scherer (2021))

```
theme_set(theme_minimal(base_size = 12, base_family = "Roboto Condensed"))
theme_update(
  axis.ticks = element_line(color = "grey92"),
  axis.ticks.length = unit(.5, "lines"),
  panel.grid.minor = element_blank(),
  legend.title = element_text(size = 12),
  legend.text = element_text(color = "grey30"),
  plot.title = element_text(size = 18, face = "bold"),
  plot.subtitle = element_text(size = 12, color = "grey30"),
  plot.caption = element_text(size = 9, margin = margin(t = 15))
)
```

Alternatively, create a custom theme and color palette (e.g. Heiss (2021))

# Third-party Themes: hrbrthemes

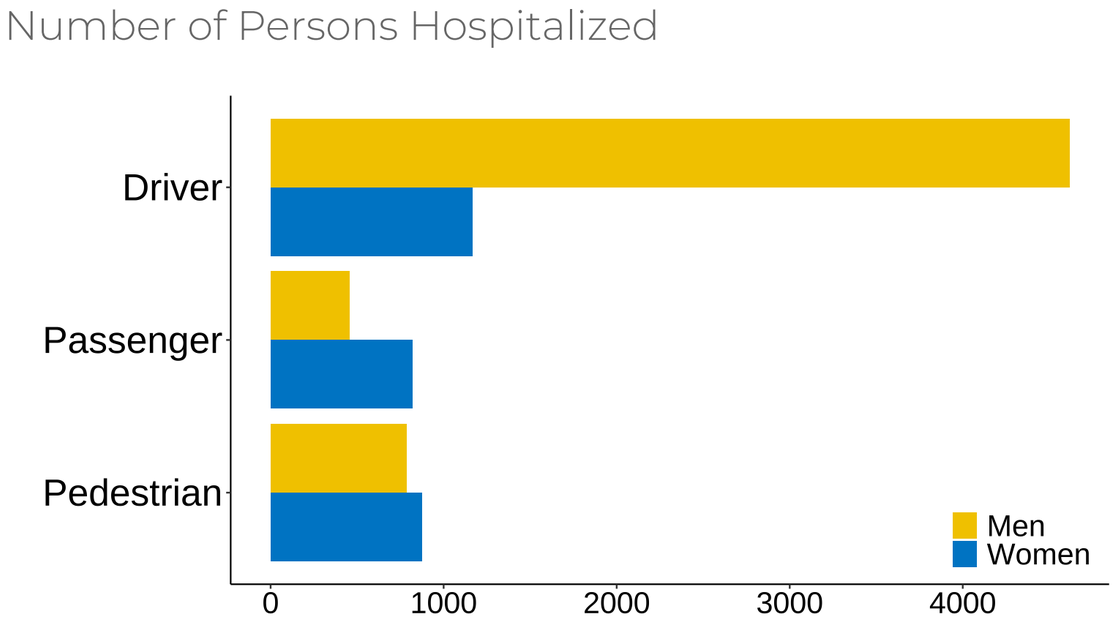
```
accident_bike |>
  ggplot(aes(x = fct_rev(type_person),
         fill = fct_rev(gender))) +
  geom_bar(position = "dodge") +
  coord_flip() +
  labs(x = NULL, y = NULL, fill = NULL) +
  hrbrthemes::scale_fill_ipsum() +
  hrbrthemes::theme_ipsum_rc() +
  theme(panel.grid.minor = element_blank(),
        panel.grid.major.y = element_blank(),
        legend.position = c(0.9, 0.1),
        axis.text.x = element_text(size = 20),
        axis.text.y = element_text(size = 25),
        legend.text = element_text(size = 20)) +
  guides(fill = guide_legend(reverse = TRUE))
```



# Third-party Themes:: ggpubr & ggsci Plaette

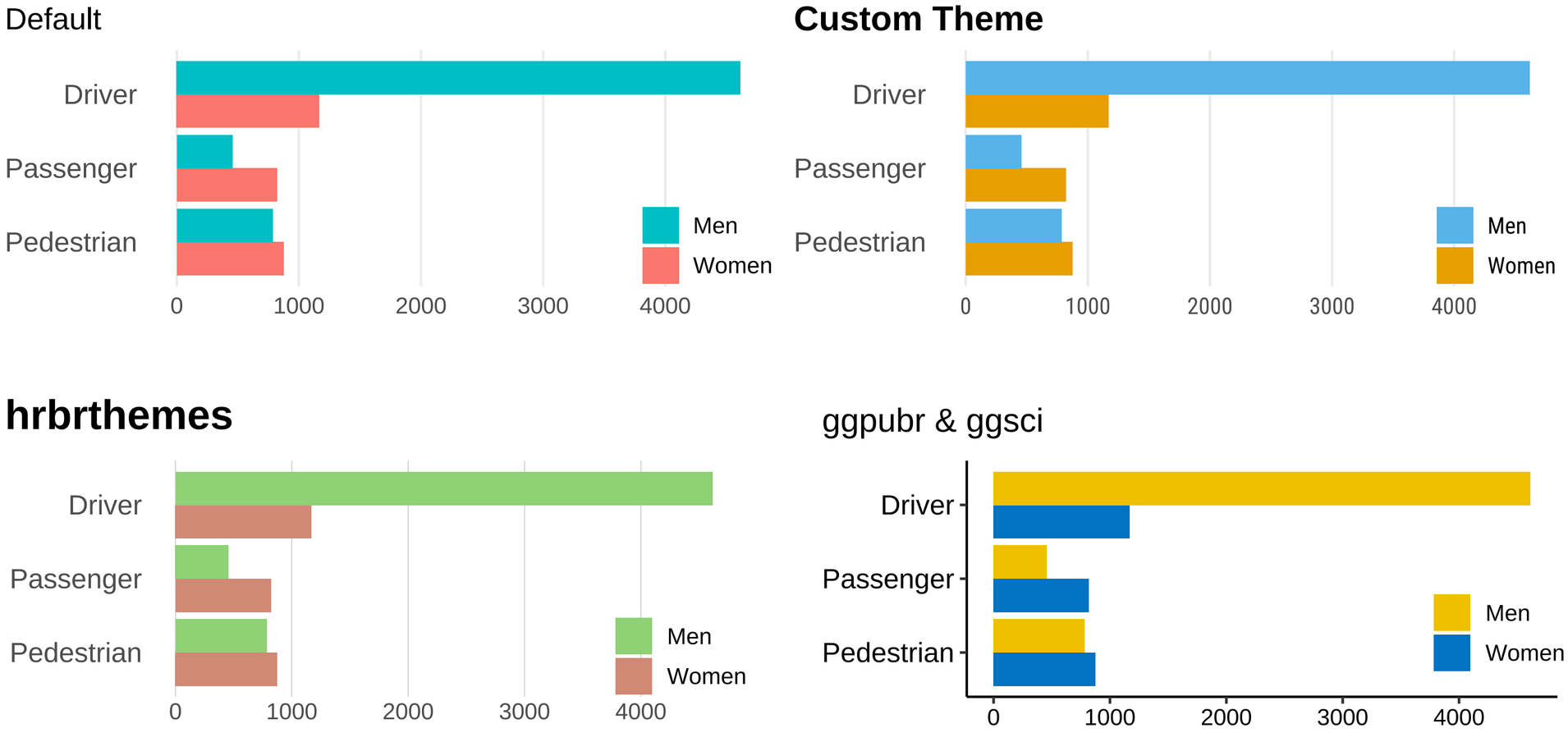
```
p <- accident_bike |>
  ggplot(aes(x = fct_rev(type_person),
         fill = fct_rev(gender))) +
  geom_bar(position = "dodge") +
  coord_flip() +
  labs(x = NULL, y = NULL, fill = NULL) +
  ggpubr::theme_pubr() +
  theme(panel.grid.minor = element_blank(),
        panel.grid.major.y = element_blank(),
        legend.position = c(0.9, 0.1),
        axis.text.x = element_text(size = 20),
        axis.text.y = element_text(size = 25),
        legend.text = element_text(size = 20)) +
  guides(fill = guide_legend(reverse = TRUE))

ggpubr::set_palette(p, "jco") # choose one of ggsci pal
```



# Patchwork

```
1 library(patchwork)
2
3 (p_default + p_custom) / (p_hrbrthemes + p_ggpubr)
```



# Takeaway

## Maximize Data-ink Ratio

- Omit all the unnecessary elements in a plot

## Colors & Fonts

- Color Palette: `RColorBrewer`, Okabe-Ito, `ggsci`
- Fonts: Google Fonts with `showtext`. Especially, *condensed* fonts.
- Ready-made Themes: `hrbrthemes`, `ggpubr`

## Further Readings (Online Books)

- “Data Visualization: A Practical Introduction” Healy (2018)
- “Fundamentals of Data Visualization” Wilke (2019)

# Automated Table Creation

# kableExtra: Example

```
1 tab
```

```
# A tibble: 6 × 9
# Groups:   weather [6]
  weather    n_Men_2019 n_Men_2…¹ n_Men…² n_Men…³ n_Wom…⁴ n_Wom…⁵
n_Wom…⁶ n_Wom…⁷
  <fct>           <int>     <int>   <int>   <int>   <int>   <int>
<int>   <int>
1 sunny           24399     14969   19208   19420   11971    6958
9417    9298
2 cloud            1159      1190    1325    1633     555     554
630     774
3 soft rain        2126      1198    1281    1408    1068     542
605     716
4 hard rain         386       202     386     352     222      96
210     179
5 snow                2         2     124       5      NA      NA
```

```
1  library(kableExtra)
2  options(knitr.kable.NA = '')
3
4  ktb <- tab |>
5    kbl(format = "latex", booktabs = TRUE,
6        col.names = c(" ", 2019:2022, 2019:2022)) |>
7    add_header_above(c(" ", "Men" = 4, "Women" = 4)) |>
8    pack_rows(index = c("Good" = 2, "Bad" = 4))
9
10 ktb |>
11   save_kable(here("output/tex/kableextra/tb_accident_bike.tex"))
```

Table 1: Number of Persons Involved in Traffic Accidents

| | Men | | | | Women | | | |
|---|---|---|---|---|---|---|---|---|
| | 2019 | 2020 | 2021 | 2022 | 2019 | 2020 | 2021 | 2022 |
| **Good** | | | | | | | | |
| sunny | 24399 | 14969 | 19208 | 19420 | 11971 | 6958 | 9417 | 9298 |
| cloud | 1159 | 1190 | 1325 | 1633 | 555 | 554 | 630 | 774 |
| **Bad** | | | | | | | | |
| soft rain | 2126 | 1198 | 1281 | 1408 | 1068 | 542 | 605 | 716 |
| hard rain | 386 | 202 | 386 | 352 | 222 | 96 | 210 | 179 |
| snow | 2 | 2 | 124 | 5 | | | 38 | 1 |
| hail | 11 | 5 | 6 | 4 | 3 | 3 | 1 | 2 |

- `booktabs = TRUE` for `booktabs` package in LaTeX
- You can specify the column names by `col.names`
- You can pack columns and rows by `add_header_above()` and `pack_rows()`
- `save_kable()` saves in a tex file if the file name ends with ".tex"

# kableExtra

## Dataframe (`tibble`) to Table

- Create a tibble table by `dplyr::group_by` & `dpyr::summarize` and `janitor::tabyl()`
- For regression tables, you can use `modelsummary` (next slide)

## Pack Columns and Rows

- As far as I know, Python, Julia, and Stata do not allow us to pack them easily

## More Complicated Tables

- You can refer to Hao Zhu's document
- If a table contains a mathematical expression, use `escape=FALSE`. See a discussion in stacoverflow

# modelsummary

Given the following regression results,

```
library(fixest) # for faster regression with fixed effect

models <- list(
    "(1)" = feglm(is_hospitalized ~ type_person + positive_alcohol + positive_drug | age_c + gender,
                  family = binomial(logit), data = data),
    "(2)" = feglm(is_hospitalized ~ type_person + positive_alcohol + positive_drug | age_c + gender + type_vehicle,
                  family = binomial(logit), data = data),
    "(3)" = feglm(is_hospitalized ~ type_person + positive_alcohol + positive_drug | age_c + gender + type_vehicle +
                  family = binomial(logit), data = data),
    "(4)" = feglm(is_died ~ type_person + positive_alcohol + positive_drug | age_c + gender,
                  family = binomial(logit), data = data),
    "(5)" = feglm(is_died ~ type_person + positive_alcohol + positive_drug | age_c + gender + type_vehicle,
                  family = binomial(logit), data = data),
    "(6)" = feglm(is_died ~ type_person + positive_alcohol + positive_drug | age_c + gender + type_vehicle + weather,
                  family = binomial(logit), data = data)
)
```

# modelsummary: Init

```
1 modelsummary(models)
```

| | (1) | (2) | (3) | (4) | (5) | (6) |
|---|---|---|---|---|---|---|
| type_personPassenger | 0.049 | 0.530 | 0.507 | −1.781 | −1.575 | −1.565 |
| | (0.104) | (0.071) | (0.070) | (0.759) | (0.783) | (0.784) |
| type_personPedestrian | 2.124 | 2.402 | 2.323 | 2.280 | 2.418 | 2.422 |
| | (0.115) | (0.066) | (0.064) | (0.301) | (0.287) | (0.285) |
| positive_alcoholTRUE | −0.077 | 0.310 | 0.353 | −13.710 | −13.455 | −13.492 |
| | (0.088) | (0.095) | (0.093) | (0.053) | (0.064) | (0.063) |
| Num.Obs. | 149918 | 149831 | 134006 | 90852 | 89300 | 86330 |
| R2 | 0.055 | 0.171 | 0.165 | 0.107 | 0.145 | 0.148 |
| R2 Adj. | 0.054 | 0.170 | 0.163 | 0.086 | 0.113 | 0.112 |
| R2 Within | 0.047 | 0.054 | 0.052 | 0.073 | 0.076 | 0.076 |
| R2 Within Adj. | 0.047 | 0.054 | 0.052 | 0.070 | 0.072 | 0.073 |
| AIC | 62871.0 | 55210.6 | 53565.4 | 1601.9 | 1552.2 | 1534.5 |
| BIC | 63079.3 | 55696.5 | 54085.1 | 1780.8 | 1824.8 | 1834.2 |
| RMSE | 0.23 | 0.22 | 0.23 | 0.04 | 0.04 | 0.04 |
| Std.Errors | by: age_c | by: age_c | by: age_c | by: age_c | by: age_c | by: age_c |
| FE: age_c | X | X | X | X | X | X |
| FE: gender | X | X | X | X | X | X |
| FE: type_vehicle | | X | X | | X | X |
| FE: weather | | | X | | | X |

# modelsummary: Modify Coefficients

```
cm  <-  c(
    "type_personPassenger" = "Passenger",
    "type_personPedestrian" = "Pedestrian",
    "positive_alcoholTRUE" = "Positive Alcohol"
)

modelsummary(models,
  coef_map = cm
)
```

| | (1) | (2) | (3) | (4) | (5) | (6) |
|---|---|---|---|---|---|---|
| Passenger | 0.049 | 0.530 | 0.507 | −1.781 | −1.575 | −1.565 |
| | (0.104) | (0.071) | (0.070) | (0.759) | (0.783) | (0.784) |
| Pedestrian | 2.124 | 2.402 | 2.323 | 2.280 | 2.418 | 2.422 |
| | (0.115) | (0.066) | (0.064) | (0.301) | (0.287) | (0.285) |
| Positive Alcohol | −0.077 | 0.310 | 0.353 | −13.710 | −13.455 | −13.492 |
| | (0.088) | (0.095) | (0.093) | (0.053) | (0.064) | (0.063) |
| Num.Obs. | 149918 | 149831 | 134006 | 90852 | 89300 | 86330 |
| R2 | 0.055 | 0.171 | 0.165 | 0.107 | 0.145 | 0.148 |
| R2 Adj. | 0.054 | 0.170 | 0.163 | 0.086 | 0.113 | 0.112 |
| R2 Within | 0.047 | 0.054 | 0.052 | 0.073 | 0.076 | 0.076 |
| R2 Within Adj. | 0.047 | 0.054 | 0.052 | 0.070 | 0.072 | 0.073 |
| AIC | 62871.0 | 55210.6 | 53565.4 | 1601.9 | 1552.2 | 1534.5 |
| BIC | 63079.3 | 55696.5 | 54085.1 | 1780.8 | 1824.8 | 1834.2 |
| RMSE | 0.23 | 0.22 | 0.23 | 0.04 | 0.04 | 0.04 |
| Std.Errors | by: age_c | by: age_c | by: age_c | by: age_c | by: age_c | by: age_c |
| FE: age_c | X | X | X | X | X | X |
| FE: gender | X | X | X | X | X | X |
| FE: type_vehicle | | X | X | | X | X |
| FE: weather | | | X | | | X |

# modelsummary: Modify Statitics

```
cm  <-  c(
    "type_personPassenger" = "Passenger",
    "type_personPedestrian" = "Pedestrian",
    "positive_alcoholTRUE" = "Positive Alcohol"
)

gm <- tibble(
    raw = c("nobs", "FE: age_c", "FE: gender", "FE: type_vehicle",
    clean = c("Observations", "FE: Age Group", "FE: Gender", "FE: T
    fmt = c(0, 0, 0, 0, 0)
)

modelsummary(models,
  coef_map = cm,
  gof_map = gm
)
```

| | (1) | (2) | (3) | (4) | (5) | (6) |
|---|---|---|---|---|---|---|
| Passenger | 0.049 | 0.530 | 0.507 | −1.781 | −1.575 | −1.565 |
| | (0.104) | (0.071) | (0.070) | (0.759) | (0.783) | (0.784) |
| Pedestrian | 2.124 | 2.402 | 2.323 | 2.280 | 2.418 | 2.422 |
| | (0.115) | (0.066) | (0.064) | (0.301) | (0.287) | (0.285) |
| Positive Alcohol | −0.077 | 0.310 | 0.353 | −13.710 | −13.455 | −13.492 |
| | (0.088) | (0.095) | (0.093) | (0.053) | (0.064) | (0.063) |
| Observations | 149918 | 149831 | 134006 | 90852 | 89300 | 86330 |
| FE: Age Group | X | X | X | X | X | X |
| FE: Gender | X | X | X | X | X | X |
| FE: Type of Vehicle | | X | X | | X | X |
| FE: Weather | | | X | | | X |

# modelsummary: Stars & Headers

```
code-line-numbers="7,16"
cm  <-  c(
    "type_personPassenger" = "Passenger",
    "type_personPedestrian" = "Pedestrian",
    "positive_alcoholTRUE" = "Positive Alcohol"
)

gm <- tibble(
    raw = c("nobs", "FE: age_c", "FE: gender", "FE: type_vehicle",
    clean = c("Observations", "FE: Age Group", "FE: Gender", "FE: T
    fmt = c(0, 0, 0, 0, 0)
)

modelsummary(models,
  stars = c("+" = .1, "*" = .05, "**" = .01),
  coef_map = cm,
  gof_map = gm) |>
  add_header_above(c(" ", "Hospitalization" = 3, "Died within 24 ho
```

| | Hospitalization | | | Died within 24 hours | | |
|---|---|---|---|---|---|---|
| | (1) | (2) | (3) | (4) | (5) | (6) |
| Passenger | 0.049 | 0.530** | 0.507** | −1.781* | −1.575+ | −1.565+ |
| | (0.104) | (0.071) | (0.070) | (0.759) | (0.783) | (0.784) |
| Pedestrian | 2.124** | 2.402** | 2.323** | 2.280** | 2.418** | 2.422** |
| | (0.115) | (0.066) | (0.064) | (0.301) | (0.287) | (0.285) |
| Positive Alcohol | −0.077 | 0.310** | 0.353** | −13.710** | −13.455** | −13.492** |
| | (0.088) | (0.095) | (0.093) | (0.053) | (0.064) | (0.063) |
| Observations | 149918 | 149831 | 134006 | 90852 | 89300 | 86330 |
| FE: Age Group | X | X | X | X | X | X |
| FE: Gender | X | X | X | X | X | X |
| FE: Type of Vehicle | | X | X | | X | X |
| FE: Weather | | | X | | | X |

+ p < 0.1, * p < 0.05, ** p < 0.01

# modelsummary: Export to LaTeX

```
cm  <-  c(
    "type_personPassenger" = "Passenger",
    "type_personPedestrian" = "Pedestrian",
    "positive_alcoholTRUE" = "Positive Alcohol"
)

gm <- tibble(
    raw = c("nobs", "FE: age_c", "FE: gender", "FE: type_vehicle",
    clean = c("Observations", "FE: Age Group", "FE: Gender", "FE: T
    fmt = c(0, 0, 0, 0, 0)
)

modelsummary(models,
  output = "latex_tabular",
  stars = c("+" = .1, "*" = .05, "**" = .01),
  coef_map = cm,
  gof_map = gm) |>
  add_header_above(c(" ", "Hospitalization" = 3, "Died within 24 ho
```

Table 2: Logit Regression of Hospitalization and Death within 24 Hours

| | Hospitalization | | | Died within 24 hours | | |
|---|---|---|---|---|---|---|
| | (1) | (2) | (3) | (4) | (5) | (6) |
| Passenger | 0.049 | 0.530** | 0.507** | −1.781* | −1.575+ | −1.565+ |
| | (0.104) | (0.071) | (0.070) | (0.759) | (0.783) | (0.784) |
| Pedestrian | 2.124** | 2.402** | 2.323** | 2.280** | 2.418** | 2.422** |
| | (0.115) | (0.066) | (0.064) | (0.301) | (0.287) | (0.285) |
| Positive Alcohol | −0.077 | 0.310** | 0.353** | −13.710** | −13.455** | −13.492** |
| | (0.088) | (0.095) | (0.093) | (0.053) | (0.064) | (0.063) |
| Observations | 149 918 | 149 831 | 134 006 | 90 852 | 89 300 | 86 330 |
| FE: Age Group | X | X | X | X | X | X |
| FE: Gender | X | X | X | X | X | X |
| FE: Type of Vehicle | | X | X | | X | X |
| FE: Weather | | | X | | | X |

Notes: Passenger and pedestrian's coefficients are normalized by driver. **p<.01; *p<.05; +p<.1. Standard errors are clustered by age group.

`output = "latex_tabular"` produces a tex file not containing `table` tag

# Takeaway

## `kableExtra` & `modelsummary`

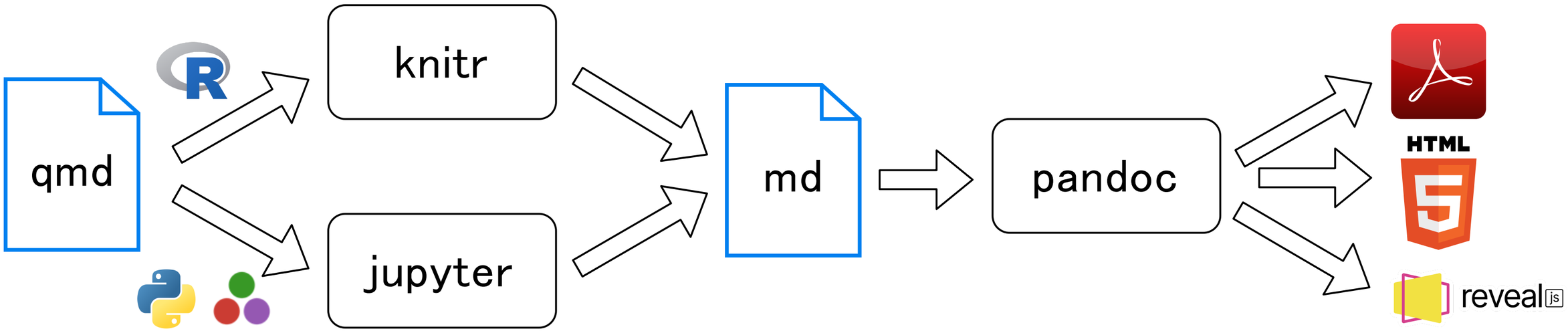
- You can quickly export tibble (dataframe) as latex table by `kableExtra`
- `modelsummary` produces kableExtra object from regression results
- You can see the latex table in `output/tex/` and the compiled results in `code/thesis/`

## Further Readings

- Official Document modelsummary and Zhu (2021)
- gt is a great alternative to `kableExtra`. I use `gt` tables in my slides

# Quarto

# What Is Quarto (.qmd)?




I use Quarto for

- **Reporting**: Easy to show the progress to supervisor/coauthors
- **Presentation**: Reveal.js produces reasonably beautiful slides

# Quarto (Markdown) Is Easy-version of LaTeX!

## Quarto (Markdown)

### Headings

```
# Heading 1
## Heading 2
### Heading 3
```

### Bullet points

```
- item 1
- item 2
- item 3
```

### Enumerate

```
1. item 1
1. item 2
1. item 3
```

## LaTeX

### Headings

```
\section{Heading 1}
\subsection{Heading 2}
\subsubsection{Heading 3}
```

### Bullet points

```
\begin{itemize}
  \item item 1
  \item item 2
  \item item 3
\end{itemize}
```

### Enumerate

```
\begin{enumerate}
  \item item 1
  \item item 2
  \item item 3
\end{enumerate}
```

# Quarto (Markdown) Is Easy-version of $\LaTeX$!

## Quarto (Markdown)

### Text Formatting

```
**bold letters**
_italic letters_
$f_n(x)$
```

### Display Math

```
$$
\begin{aligned}
u(x) &= \frac{c^{1 - \gamma}}{1 - \gamma} \\
u'(x) &= c^{1- \gamma}
\end{aligned}
$$
```

### Cross References

```
@bib_tex_key
@fig-label_fig
@tbl-label_tbl
```

## $\LaTeX$

```
\textbf{bold letters}
\textit{italic letters}
$f_n(x)$
```

```
\begin{align*}
u(x) &= \frac{c^{1 - \gamma}}{1 - \gamma} \\
u'(x) &= c^{1- \gamma}
\end{align*}
```

```
\cite(bib_tex_key)
\ref{fig:label_fig}
\ref{tbl:label_tbl}
```

# Quarto Presentation

## Quarto (Reveal.js)

```
## First Slide

Blah, Blah, Blah

## Second Slide

Yeah, Yeah, Yeah
```

## LaTeX (Beamer)

```
\begin{frame}{First Slide}

Blah, Blah, Blah

\end{frame}

\begin{frame}{Secon Slide}

Yeah, Yeah, Yeah

\end{frame}
```

# Quarto Presentation: Fragments

## Quarto (Reveal.js)

### Pause

```
First fragment

. . .

Second fragment
```

## LaTeX (Beamer)

```
First fragment

\pause

Second fragment
```

### Incremental List

```
::: {.incremental}

- 1st element
- 2nd element
- 3rd element

:::
```

```
\begin{itemize}[<+->]
    \item 1st element
    \item 2nd element
    \item 3rd element
\end{itemize}
```

For more complicated examples, see Tom Mock's this part of the slides

# Why Do I Use Quarto?

## Reports

- Analysis, Results, and Interpretation are done in one file
- Easy to communicate with supervisor/coauthors

## Presentations

- I prefer its design to Beamer. Highly customizable
- Same effort as Beamer slides. The syntax is almost the same
- For more reasons and techniques, read my blog

# References


Boswell, Dustin, and Trevor Foucher. 2011. *The Art of Readable Code*. 1st ed. Theory in Practice. Sebastopol, Calif: O'Reilly.

Bryan, Jenny. 2018. "Zen And The aRt Of Workflow Maintenance." Part of 47 JAIIO. https://github.com/jennybc/zen-art-workflow.

Healy, Kieran. 2018. *Data Visualization: A Practical Introduction*. 1st edition. Princeton, NJ: Princeton University Press. https://socviz.co/.

Heiss, Andrew. 2021. "Who Cares About Crackdowns? Exploring the Role of Trust in Individual Philanthropy." https://github.com/andrewheiss/who-cares-about-crackdown/blob/ad6312957de927674a5da2437a2f993e52f53d88/R/graphics.R.

Kastrun, Tomaz. 2022. "Comparing Performances of CSV to RDS, Parquet, and Feather File Formats in R R-Bloggers." R-bloggers. *R-Bloggers*. https://www.r-bloggers.com/2022/05/comparing-performances-of-csv-to-rds-parquet-and-feather-file-formats-in-r/.

Mock, Tom. 2022. "Outrageously Efficient Exploratory Data Analysis with Apache Arrow and Dplyr." Voltron Data. https://jthomasmock.github.io/arrow-dplyr/.

Scherer, C'edric. 2021. "Ggplot Wizardry: My Favorite Tricks and Secrets for Beautiful Plots in R." Online. https://www.cedricscherer.com/slides/useR-2021_ggplot-wizardry-extended.pdf.

Tufte, Edward R. 2001. *The Visual Display of Quantitative Information*. Cheshire, Conn.

Wilke, Claus O. 2019. *Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures*. Sebastopol, CA. https://clauswilke.com/dataviz/.

Zhu, Hao. 2021. "Create Awesome LaTeX Table with Knitr::kable and kableExtra," February. https://cran.r-project.org/web/packages/kableExtra/vignettes/awesome_table_in_pdf.pdf.