

A photograph of a person's hands interacting with a laptop. The person is wearing a red and blue plaid shirt. Their right hand is pointing their index finger towards the laptop screen, which displays a QR code. A smartphone lies on top of an open book on the wooden desk in front of the laptop.

# An Advanced Introduction to

Kazuharu Yanagimoto  
January 13, 2023

# Project Based Workflow

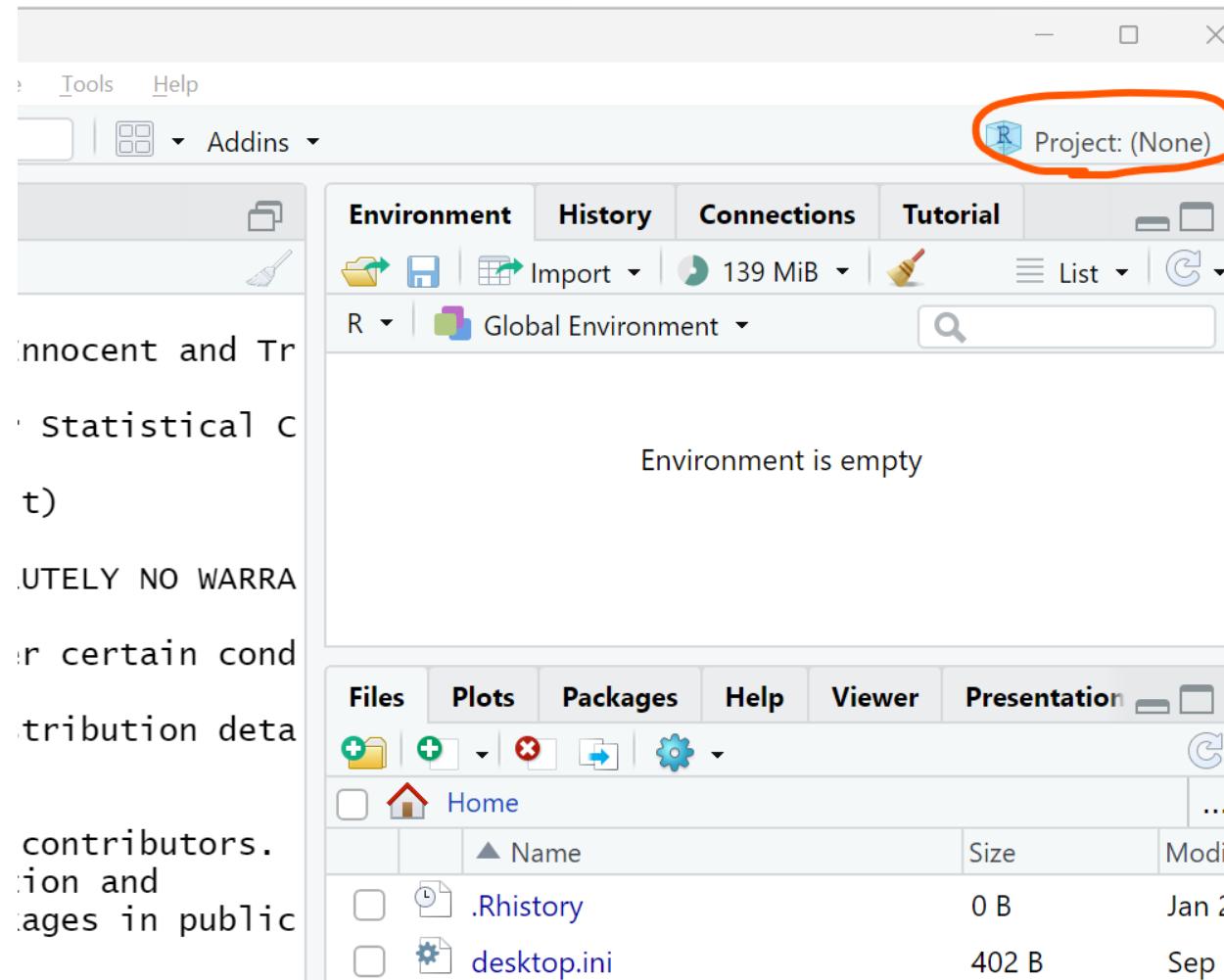
**Q. Why Don't Your Codes Work on My Computer?**

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

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

# 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 project directory as the root directory.

```
1 here::here()  
[1] "/home/rstudio/workshop-r-2022"
```

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

```
1 data <- readr::read_csv(  
2   here::here("data/tiny.csv")  
3 )
```

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)

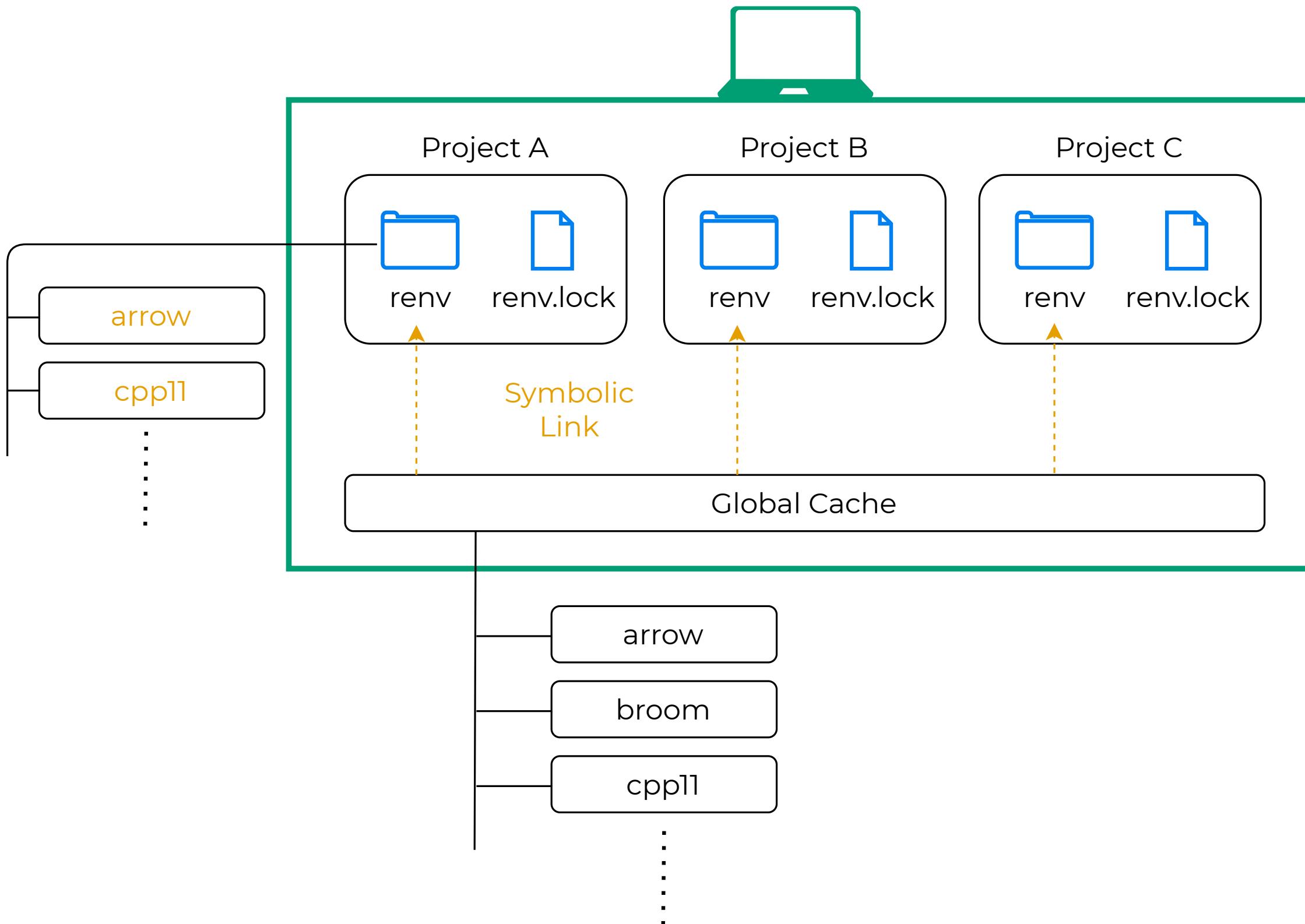
# renv Is Smarter than Us

- 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()`

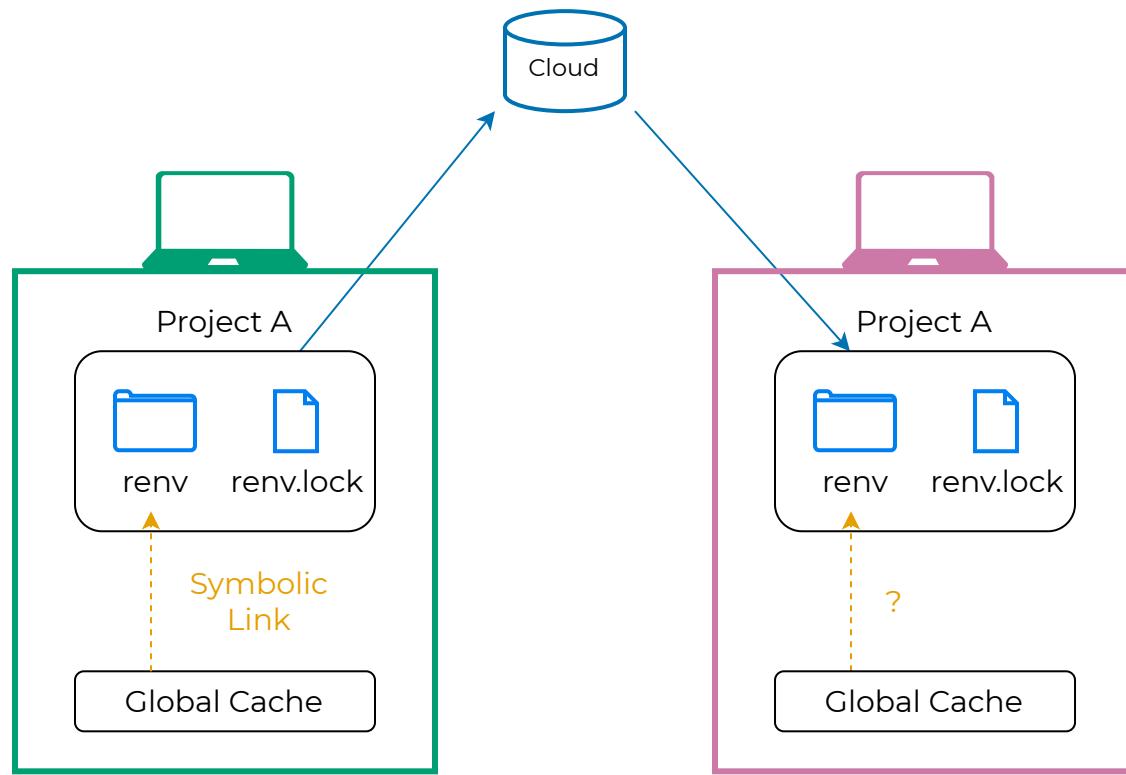
```
renv.lock
1  {
2    "R": {
3      "Version": "4.2.2",
4      "Repositories": [
5        {
6          "Name": "CRAN",
7          "URL": "https://packagemanager.posit.org"
8        }
9      ],
10     },
11   "Packages": {
12     "DBI": {
13       "Package": "DBI",
14       "Version": "1.1.3",
15       "Source": "Repository",
16       "Repository": "RSPM",
17       "Hash": "b2866e62bab9378c3cc9476a1954",
18       "Requirements": []
19     }
20   }
```

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 repository
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

(i) 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.

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

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

```
1 raw <- read_delim(here("data/raw/accident_bike/txt/year=2022/file.txt"),  
2   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", "+...
```

```
1 code <- read_csv(here("data/translate/accident_bike.csv"),  
2   show_col_types = FALSE)  
3 renamed <- raw |>  
4   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-parsing functions.

```
1 lubridate::ymd("2021/08/31")
[1] "2021-08-31"

1 lubridate::mdy("Sep. 10, 19")
[1] "2019-09-10"

1 lubridate::dmy_hm("02/04/1999 16:00", tz="America/New_York")
[1] "1999-04-02 16:00:00 EST"
```

```
1 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
```

```
1 renamed |>
2   mutate(time = lubridate::dmy_hms(str_c(date, hms), tz = "Europe/Madrid")) |>
3   select(date, hms, time) |>
4   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

```
1 renamed |>
2   mutate(
3     type_person = recode_factor(type_person,
4       "Conductor" = "Driver",
5       "Pasajero" = "Passenger",
6       "Peatón" = "Pedestrian",
7       "NULL"= NULL)) |>
8   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()

```
1 renamed |>
2   mutate(
3     weather_old = weather,# Presentation Purpose
4     weather = na_if(weather, "Se desconoce"),
5     weather = na_if(weather, "NULL"),
6   ) |>
7   select(weather_old, weather) |>
8   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

```
1 renamed |>
2   mutate(
3     weather_spanish = weather,# Presentation Purpose
4     weather = recode_factor(weather,
5       "Despejado" = "sunny",
6       "Nublado" = "cloud",
7       "Lluvia débil" = "soft rain",
8       "Lluvia intensa" = "hard rain",
9       "LLuvia intensa" = "hard rain",
10      "Nevando" = "snow",
11      "Granizando" = "hail",
12      "Se desconoce" = NULL,
13      "NULL" = NULL)) |>
14   select(weather_spanish, weather) |>
15   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**

```
1 info <- arrow::read_parquet(  
2   here("data/cleaned/accident_bike.parquet"),  
3   as_data_frame = FALSE)  
4  
5 info
```

---

table  
68574 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
  <chr>        <chr>    <chr>  <chr>    <chr>        <int> <chr>
  type_...⁴ weather type_...⁵
  <chr>        <fct>    <chr>
  1 2020S0195... 01/0... 4:30... AVDA.... 57          13  Puente...
  Colisi... sunny  Turismo
  2 2020S0195... 01/0... 4:30... AVDA.... 57          13  Puente...
  Colisi... sunny  Turismo
  3 2020S0195... 01/0... 4:30... AVDA.... 57          13  Puente...
  Colisi... sunny  Turismo
  4 2020S0195... 01/0... 4:30... AVDA.... 57          13  Puente...
  Colisi... sunny  Turismo
  5 2020S0195... 01/0... 4:30... AVDA.... 57          13  Puente...
  Colisi... sunny  Turismo
  6 2021S0000... 01/0... 0:50... AVDA.... 1
```

```
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 2021-01-01 00:50:00 Women  35-39 FALSE
2 2021-01-01 00:50:00 Men   15-17 FALSE
3 2021-01-01 00:50:00 Men   <5      FALSE
4 2021-01-01 00:30:00 Men   40-44 TRUE
5 2021-01-01 00:30:00 Women 45-49 TRUE
6 2021-01-01 22:30:00 Men   40-44 FALSE
7 2021-01-01 11:38:00 Men   45-49 FALSE
8 2021-01-02 17:15:00 Men   10-14 FALSE
9 2021-01-02 21:30:00 Men   25-29 FALSE
10 2021-01-02 23:10:00 Men  18-20 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

```
1 data/raw/accident_bike/parquet/
2   └── year=2019
3     └── part-0.parquet
4   └── year=2020
5     └── part-0.parquet
6   └── year=2021
7     └── part-0.parquet
8   └── year=2022
9     └── part-0.parquet
```

```
1 info <- open_dataset(
2   here("data/raw/accident_bike/parquet"))
3 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:

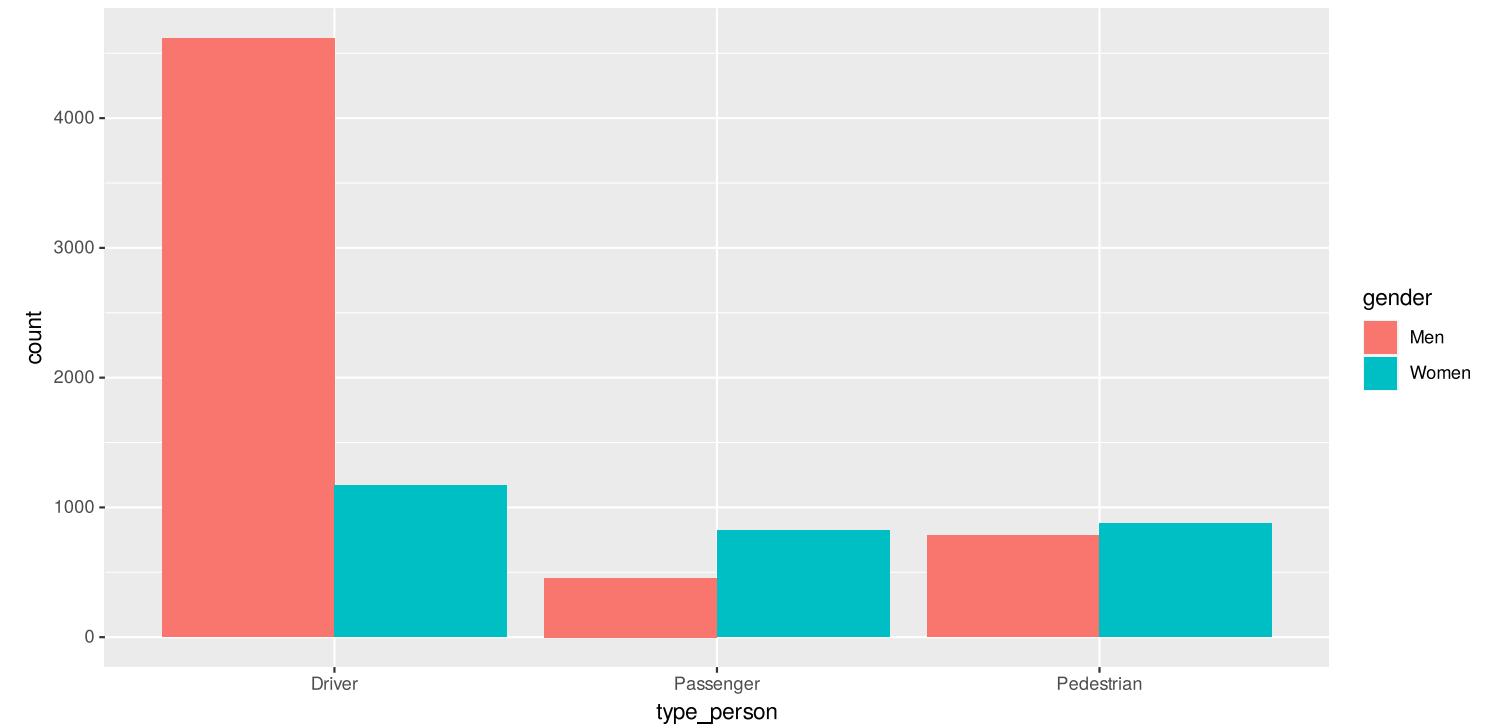
$$\text{Data-ink ratio} := \frac{\text{Data-ink}}{\text{Total ink used to print in the graphic}}$$

## Collorary

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

# Maximize Data-ink Ratio

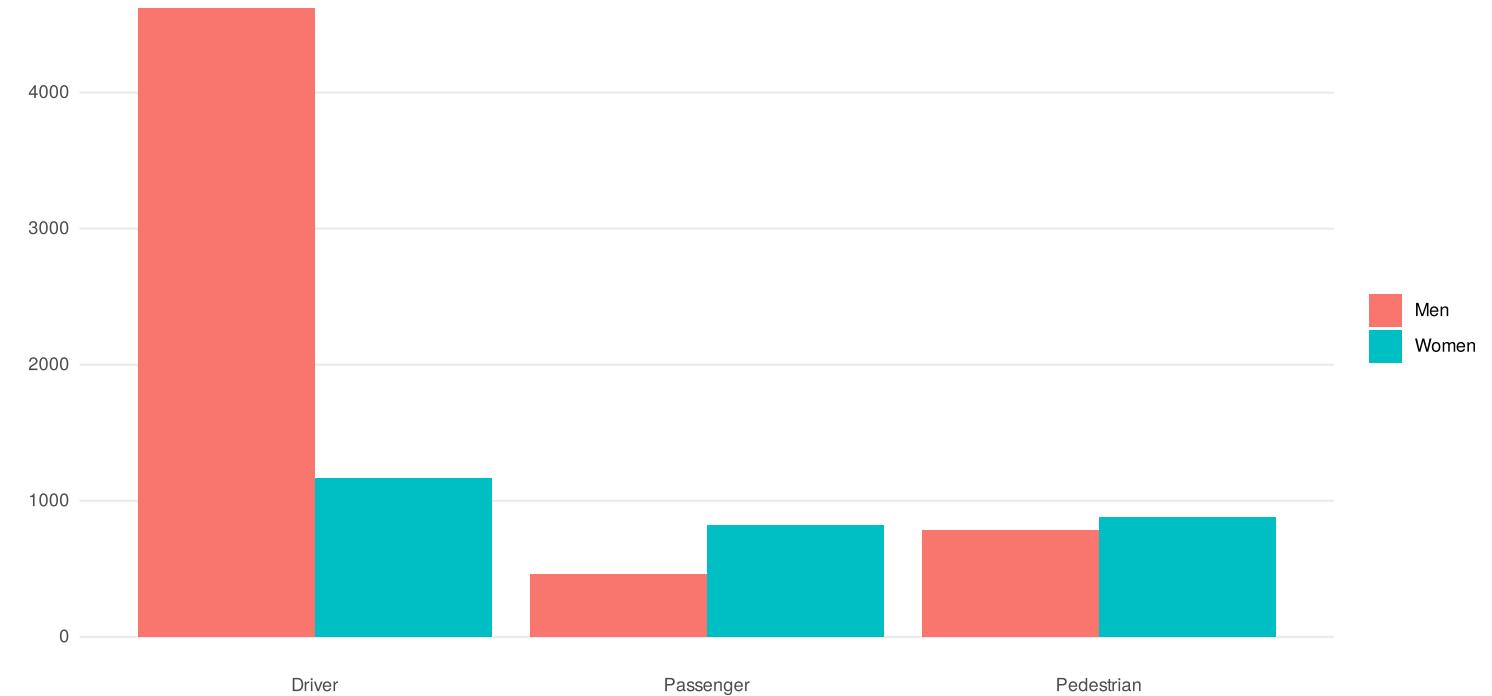
```
1 accident_bike |>  
2 ggplot(aes(x = type_person, fill = gender)) +  
3 geom_bar(position = "dodge")
```



# Maximize Data-ink Ratio

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

Number of Persons Hospitalized

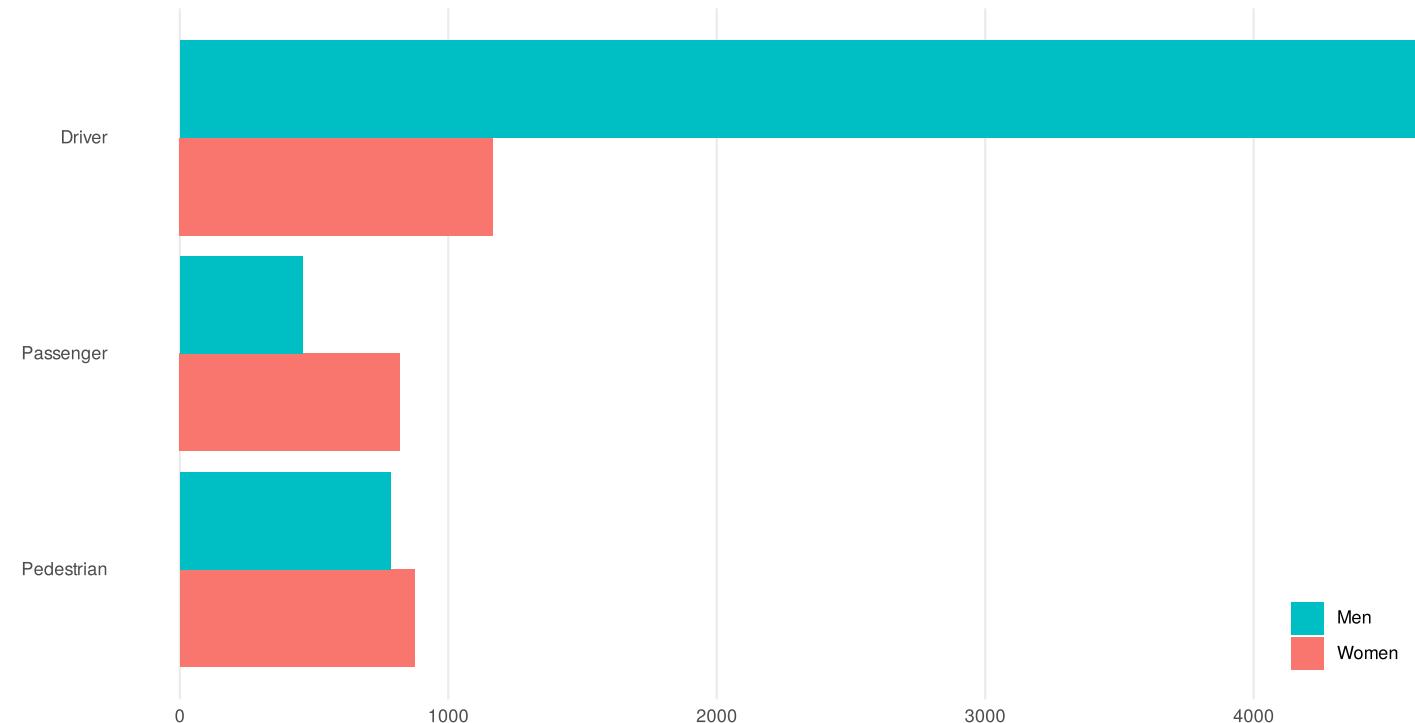


- Omit axis label. The title of the plot can tell them
- Omit legend label. The label “gender” does not add any information
- Omit background grids

# More Readability: Order Bar Plot

```
1 accident_bike |>
2   ggplot(aes(x = fct_rev(type_person),
3               fill = fct_rev(gender))) +
4   geom_bar(position = "dodge") +
5   coord_flip() +
6   labs(x = NULL, y = NULL, fill = NULL) +
7   theme_minimal() +
8   theme(panel.grid.minor = element_blank(),
9         panel.grid.major.y = element_blank(),
10        legend.position = c(0.9, 0.1)) +
11   guides(fill = guide_legend(reverse = TRUE))
```

Number of Persons Hospitalized

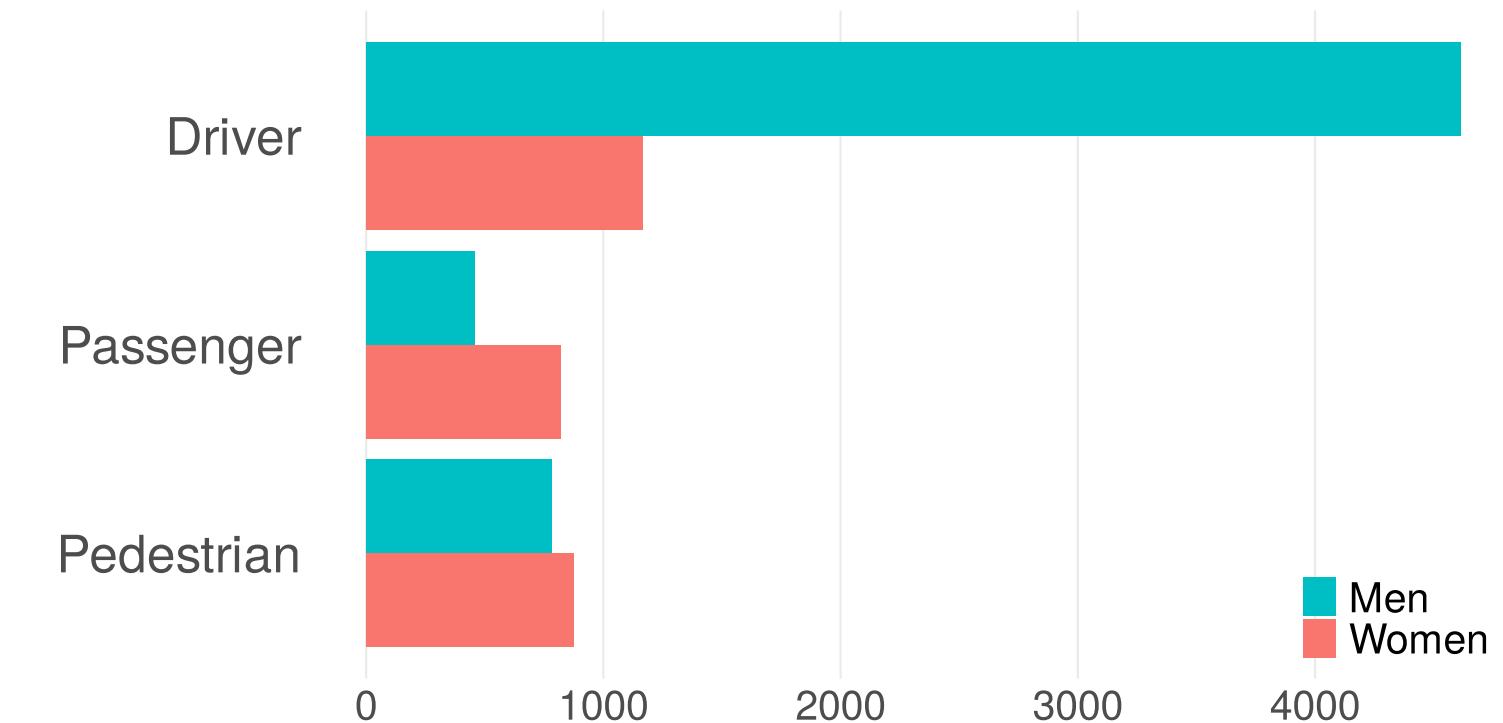


- Coord flipped. Reorder the factor variables
- Put legends inside the plot to make the plot bigger

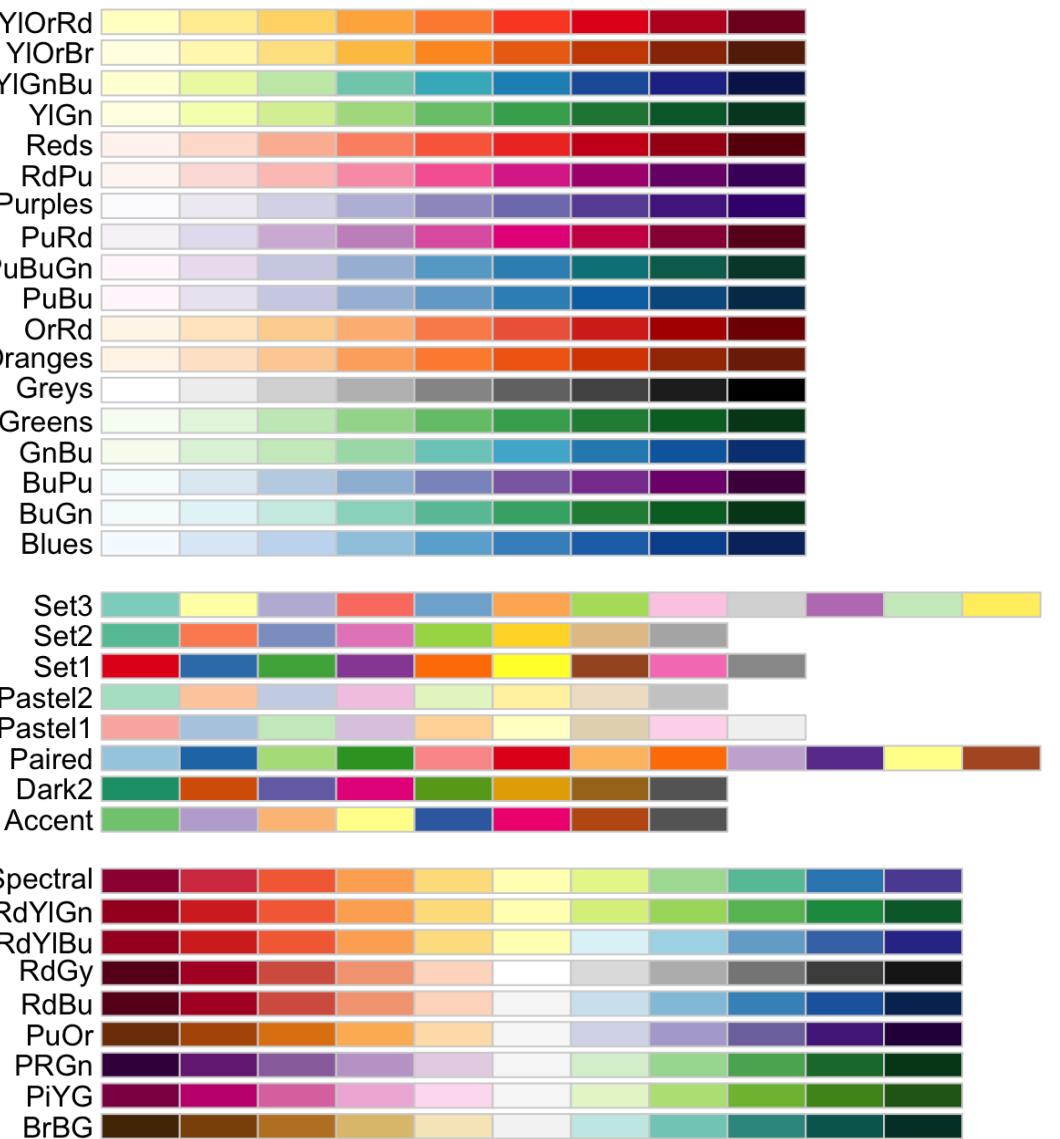
# More Readability: Increase Font Size

```
1 accident_bike |>
2   ggplot(aes(x = fct_rev(type_person),
3               fill = fct_rev(gender))) +
4   geom_bar(position = "dodge") +
5   coord_flip() +
6   labs(x = NULL, y = NULL, fill = NULL) +
7   theme_minimal() +
8   theme(panel.grid.minor = element_blank(),
9         panel.grid.major.y = element_blank(),
10        legend.position = c(0.9, 0.1),
11        axis.text.x = element_text(size = 20),
12        axis.text.y = element_text(size = 25),
13        legend.text = element_text(size = 20)) +
14   guides(fill = guide_legend(reverse = TRUE))
```

Number of Persons Hospitalized



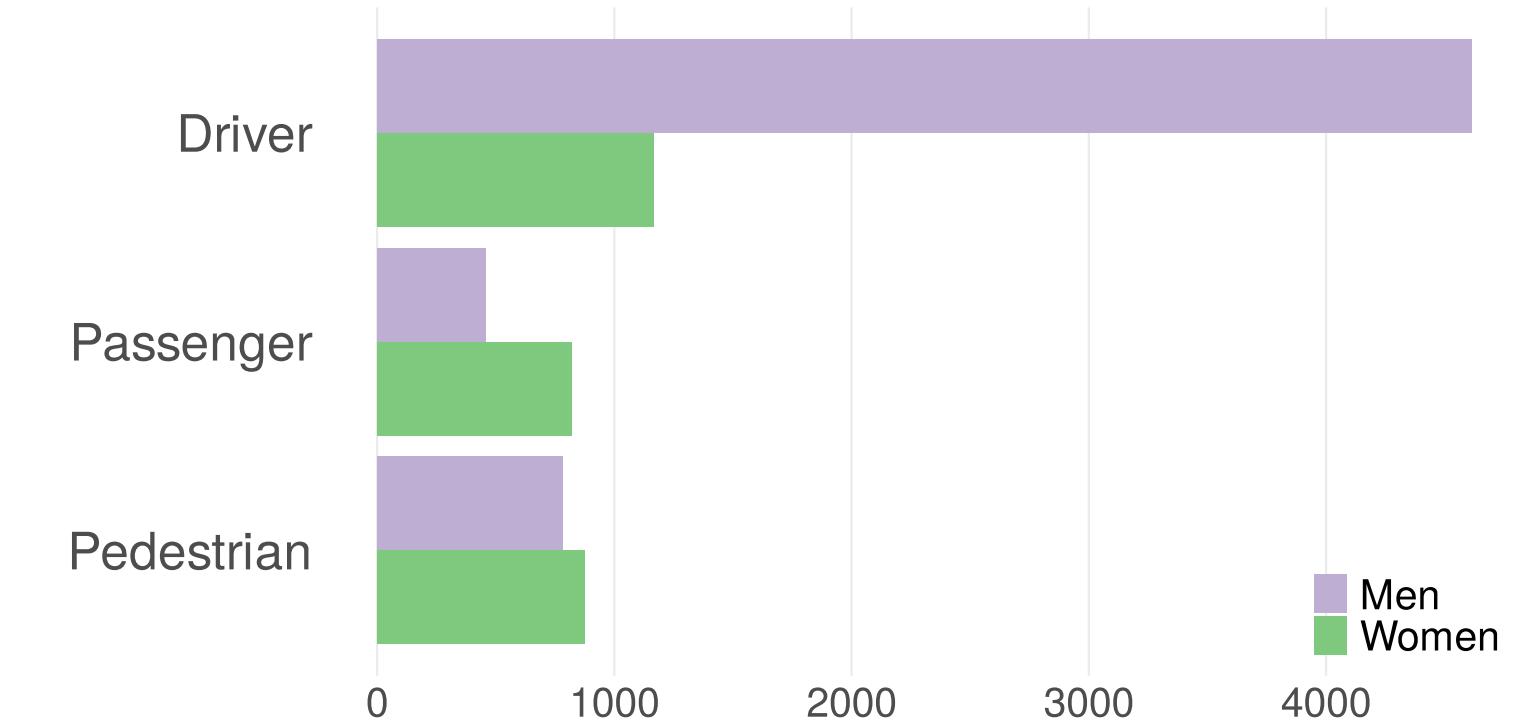
# R Color Brewer's Palettes



# R Color Brewer's Palettes

```
1 accident_bike |>
2   ggplot(aes(x = fct_rev(type_person),
3               fill = fct_rev(gender))) +
4   geom_bar(position = "dodge") +
5   coord_flip() +
6   labs(x = NULL, y = NULL, fill = NULL) +
7   scale_fill_brewer(palette = "Accent") +
8   theme_minimal() +
9   theme(panel.grid.minor = element_blank(),
10         panel.grid.major.y = element_blank(),
11         legend.position = c(0.9, 0.1),
12         axis.text.x = element_text(size = 20),
13         axis.text.y = element_text(size = 25),
14         legend.text = element_text(size = 20)) +
15   guides(fill = guide_legend(reverse = TRUE))
```

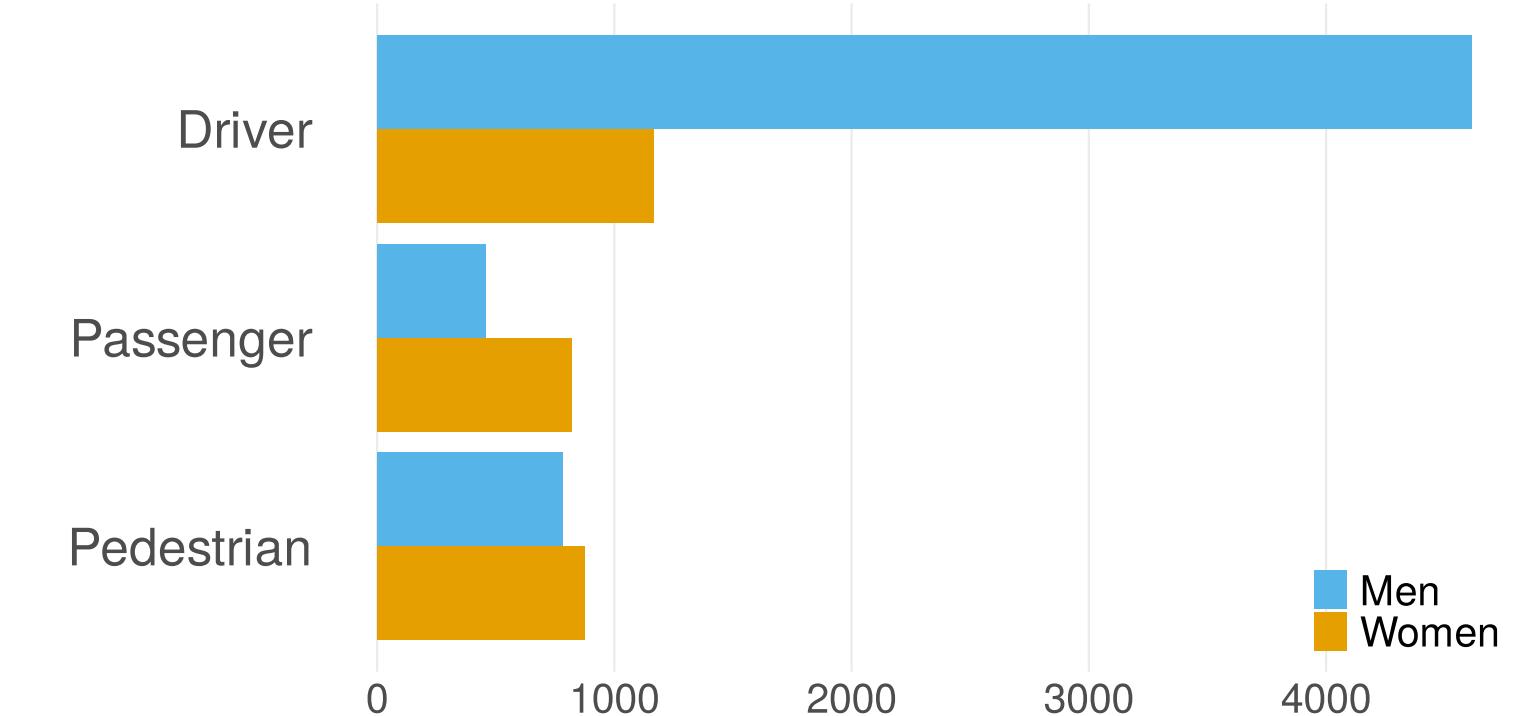
Number of Persons Hospitalized



# Color-Safe Palette: Okabe-Ito Palette

```
1 accident_bike |>
2   ggplot(aes(x = fct_rev(type_person),
3               fill = fct_rev(gender))) +
4   geom_bar(position = "dodge") +
5   coord_flip() +
6   labs(x = NULL, y = NULL, fill = NULL) +
7   see::scale_fill_okabeito() +
8   theme_minimal() +
9   theme(panel.grid.minor = element_blank(),
10         panel.grid.major.y = element_blank(),
11         legend.position = c(0.9, 0.1),
12         axis.text.x = element_text(size = 20),
13         axis.text.y = element_text(size = 25),
14         legend.text = element_text(size = 20)) +
15   guides(fill = guide_legend(reverse = TRUE))
```

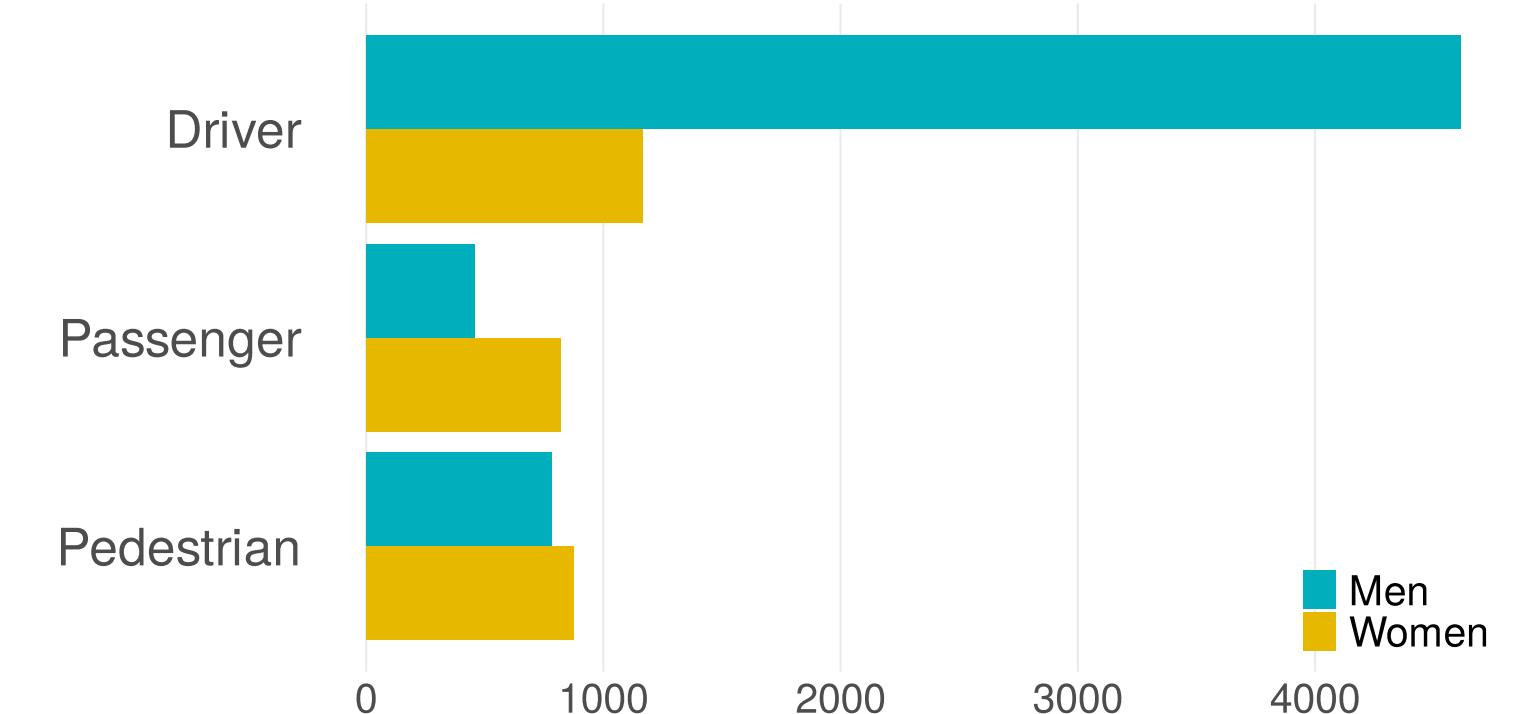
Number of Persons Hospitalized



# Custom Palette

```
1 accident_bike |>
2   ggplot(aes(x = fct_rev(type_person),
3               fill = fct_rev(gender))) +
4   geom_bar(position = "dodge") +
5   coord_flip() +
6   labs(x = NULL, y = NULL, fill = NULL) +
7   scale_fill_manual(values = c("#E7B800", "#00AFBB")) +
8   theme_minimal() +
9   theme(panel.grid.minor = element_blank(),
10         panel.grid.major.y = element_blank(),
11         legend.position = c(0.9, 0.1),
12         axis.text.x = element_text(size = 20),
13         axis.text.y = element_text(size = 25),
14         legend.text = element_text(size = 20)) +
15   guides(fill = guide_legend(reverse = TRUE))
```

Number of Persons Hospitalized



# Fonts

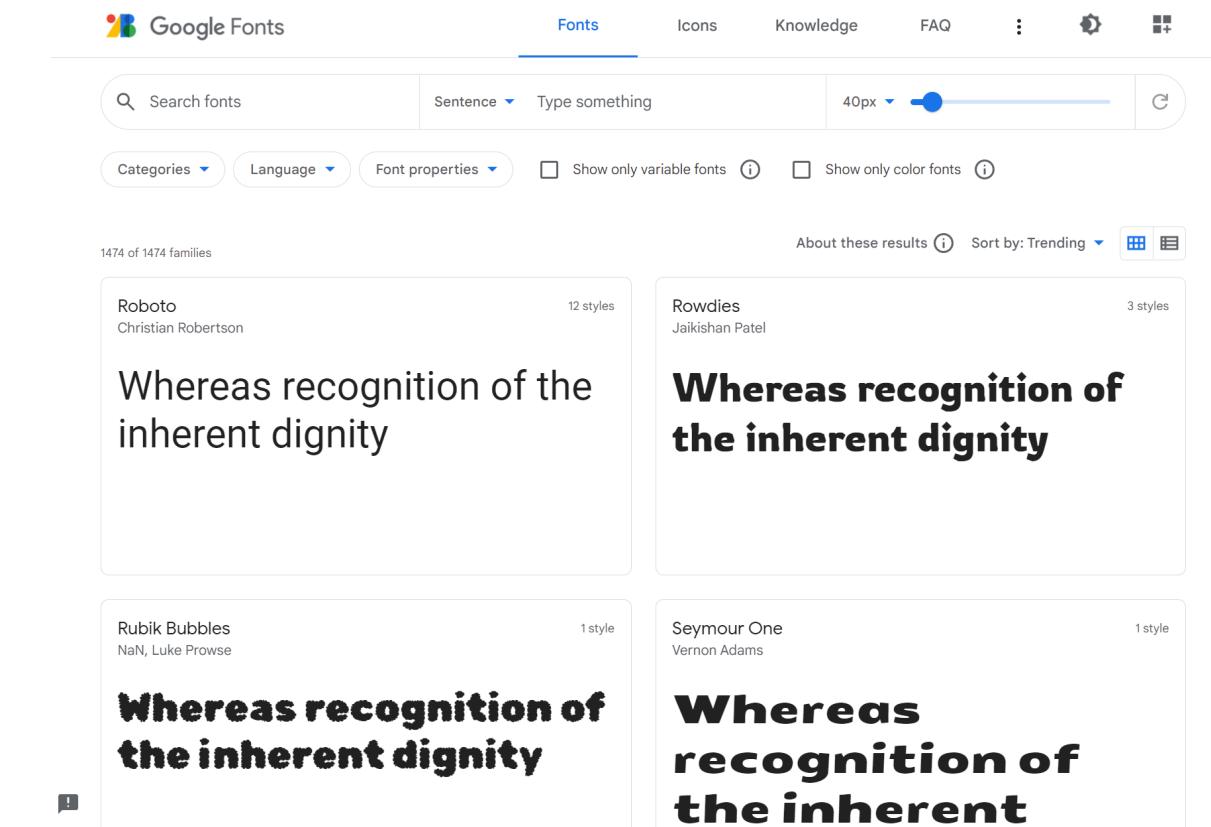
## Goolge Fonts

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

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

## showtext

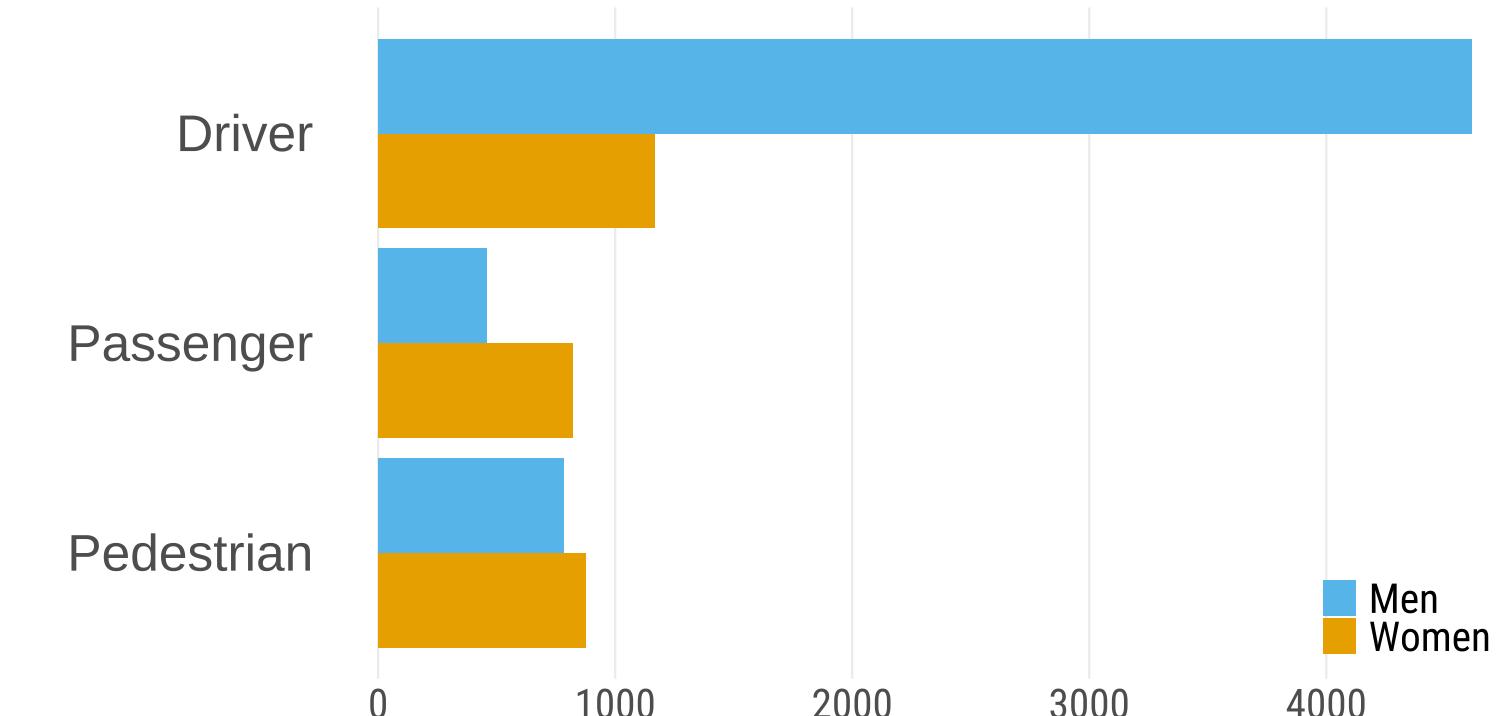
- Your collaborators need to download the fonts
- `font_add_google()` and `showtext_auto()` automatically solve the problem



# Roboto Condensed

```
1 library(showtext)
2 font_base <- "Roboto Condensed"
3 font_light <- "Roboto Condensed Light 300"
4 font_add_google(font_base, font_light)
5 showtext_auto()
6
7 accident_bike |>
8   ggplot(aes(x = fct_rev(type_person), fill = fct_rev(gender)))
9   geom_bar(position = "dodge") +
10  coord_flip() +
11  labs(x = NULL, y = NULL, fill = NULL) +
12  see::scale_fill_okabeito() +
13  theme_minimal() +
14  theme(panel.grid.minor = element_blank(),
15        panel.grid.major.y = element_blank(),
16        legend.position = c(0.9, 0.1),
17        axis.text.x = element_text(size = 20, family =
18          "Londonderry-Lament+Font+Condensed-200 Family"),
19        axis.text.y = element_text(size = 25, family =
20          "Londonderry-Lament+Font+Condensed-200 Family"))
```

Number of Persons Hospitalized



# Global Options

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

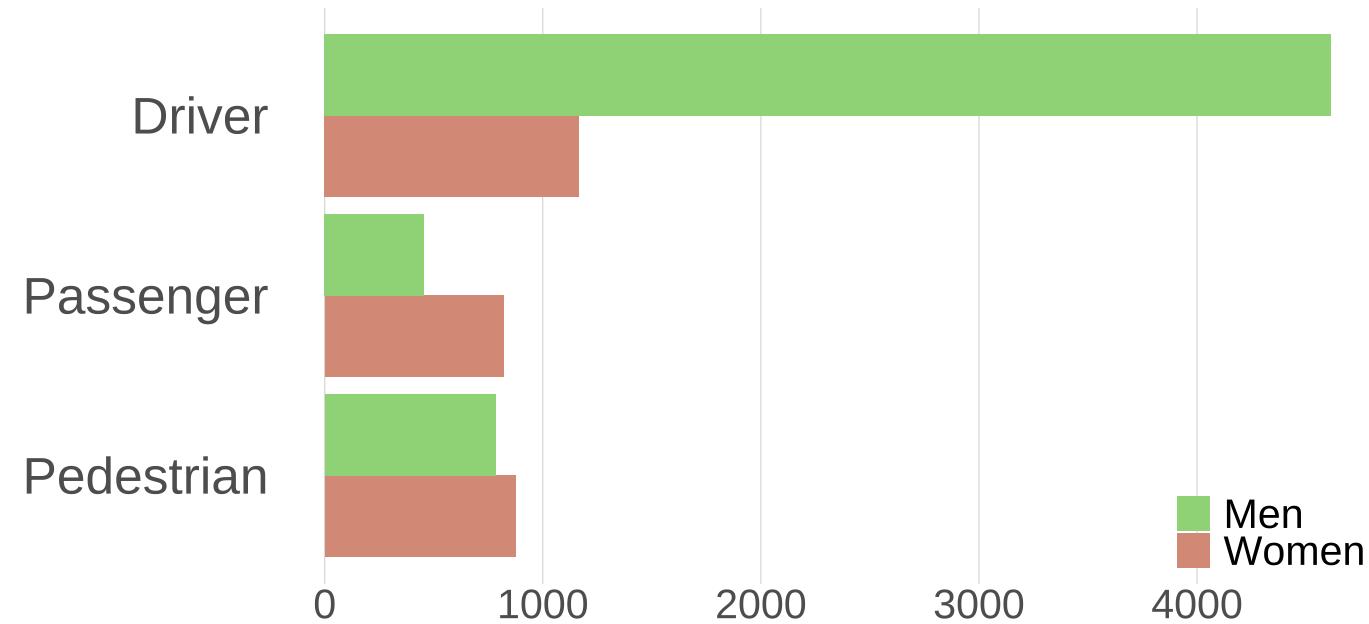
```
1 theme_set(theme_minimal(base_size = 12, base_family = "Roboto Condensed"))
2 theme_update(
3   axis.ticks = element_line(color = "grey92"),
4   axis.ticks.length = unit(.5, "lines"),
5   panel.grid.minor = element_blank(),
6   legend.title = element_text(size = 12),
7   legend.text = element_text(color = "grey30"),
8   plot.title = element_text(size = 18, face = "bold"),
9   plot.subtitle = element_text(size = 12, color = "grey30"),
10  plot.caption = element_text(size = 9, margin = margin(t = 15))
11 )
```

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

# Third-party Themes: hrbrthemes

```
1 accident_bike |>
2   ggplot(aes(x = fct_rev(type_person),
3               fill = fct_rev(gender))) +
4   geom_bar(position = "dodge") +
5   coord_flip() +
6   labs(x = NULL, y = NULL, fill = NULL) +
7   hrbrthemes::scale_fill_ipsum() +
8   hrbrthemes::theme_ipsum_rc() +
9   theme(panel.grid.minor = element_blank(),
10         panel.grid.major.y = element_blank(),
11         legend.position = c(0.9, 0.1),
12         axis.text.x = element_text(size = 20),
13         axis.text.y = element_text(size = 25),
14         legend.text = element_text(size = 20)) +
15   guides(fill = guide_legend(reverse = TRUE))
```

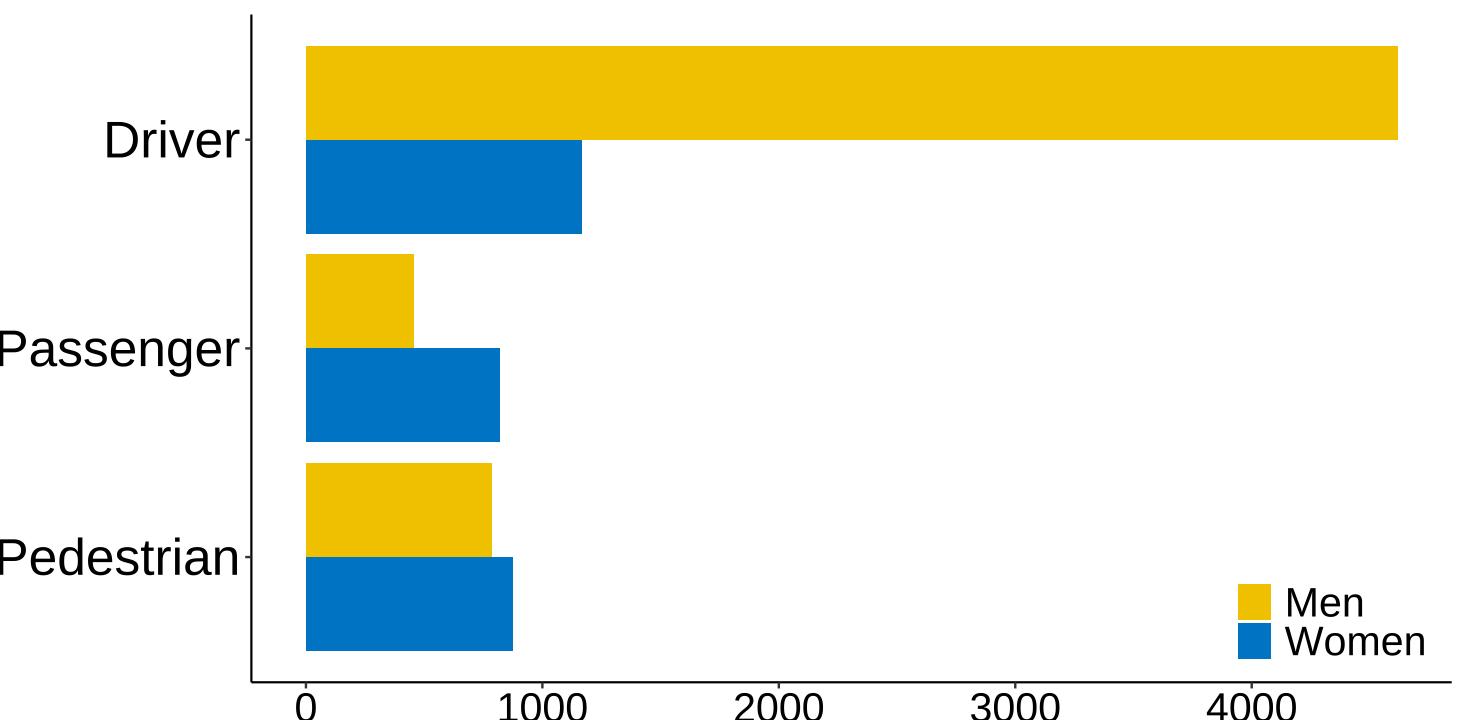
Number of Persons Hospitalized



# Third-party Themes: ggpublisher & ggsci Plaette

```
1 p <- accident_bike |>
2   ggplot(aes(x = fct_rev(type_person),
3               fill = fct_rev(gender))) +
4   geom_bar(position = "dodge") +
5   coord_flip() +
6   labs(x = NULL, y = NULL, fill = NULL) +
7   ggpublisher::theme_pubr() +
8   theme(panel.grid.minor = element_blank(),
9         panel.grid.major.y = element_blank(),
10        legend.position = c(0.9, 0.1),
11        axis.text.x = element_text(size = 20),
12        axis.text.y = element_text(size = 25),
13        legend.text = element_text(size = 20)) +
14   guides(fill = guide_legend(reverse = TRUE))
15
16 ggpublisher::set_palette(p, "jco") # choose one of ggsci pa
```

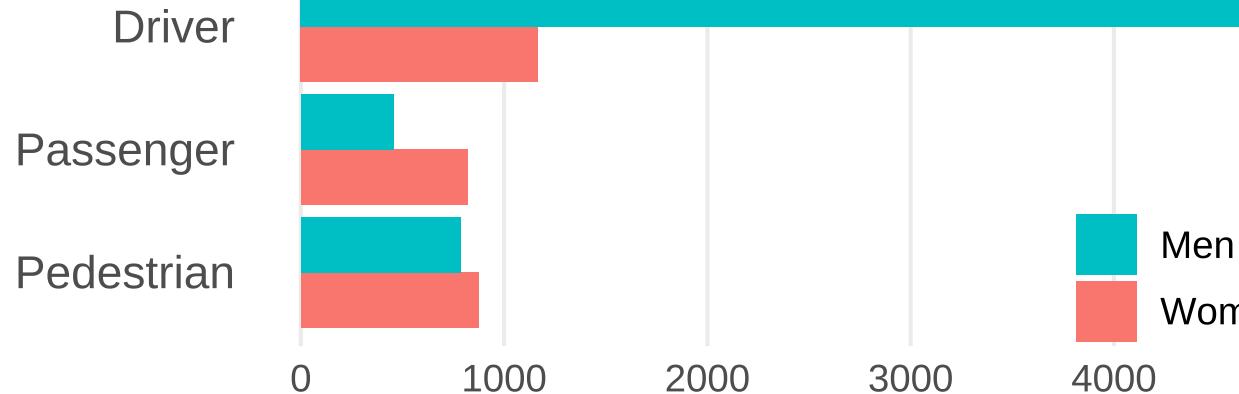
Number of Persons Hospitalized



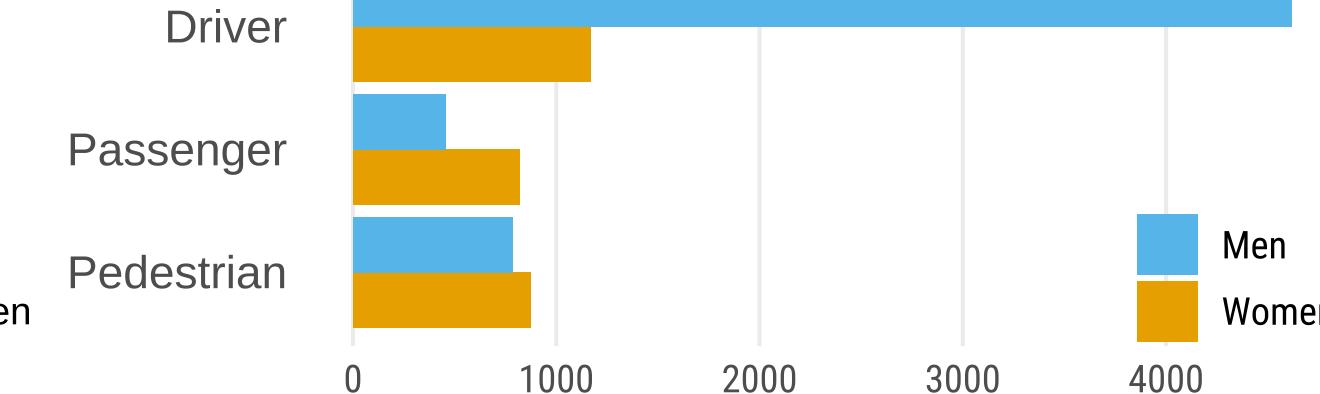
# Patchwork

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

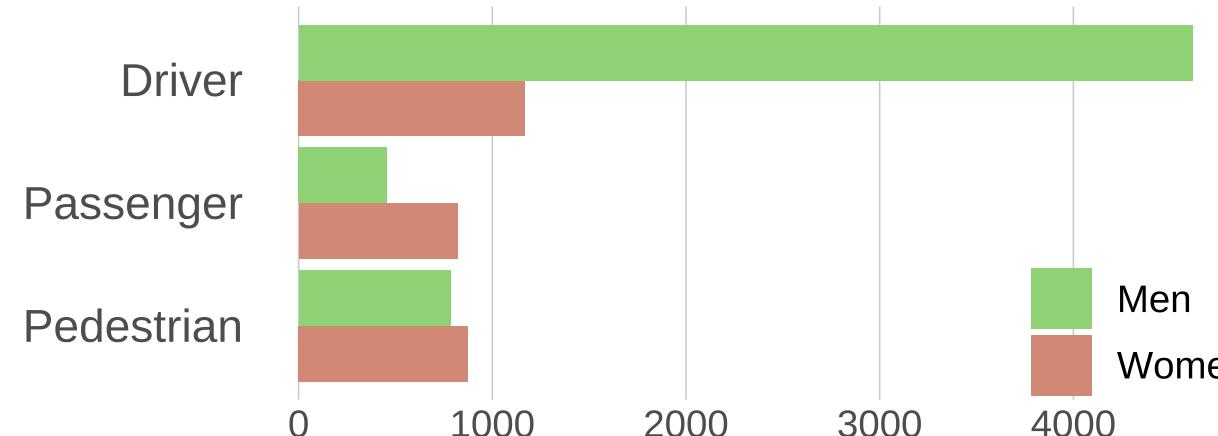
Default



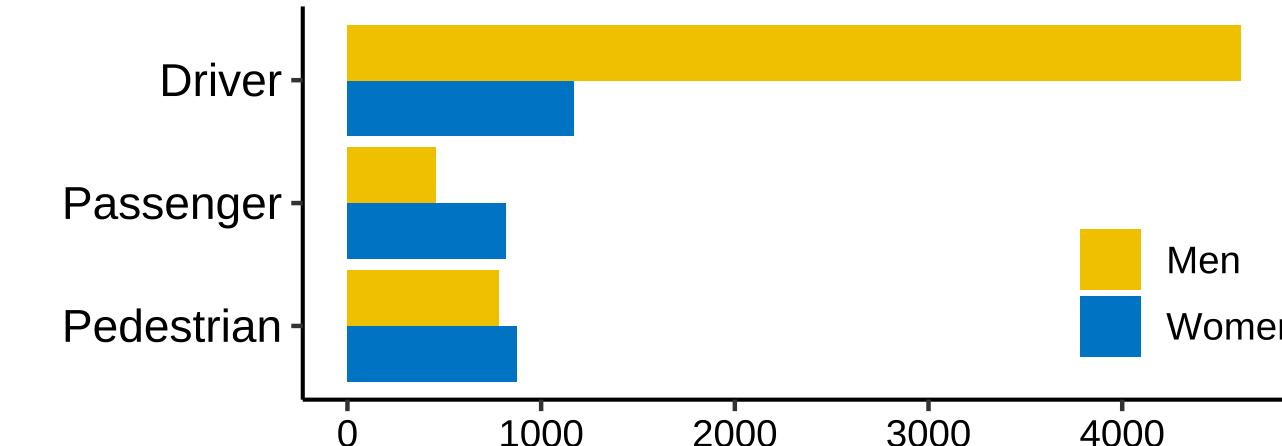
Custom Theme



hrbrthemes



ggpubr & ggsci



# 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
  <fct>    <int>     <int>     <int>     <int>     <int>     <int>
  <int>    <int>
1 sunny      24399    14969    19208    19420    11971    6958
2 cloud       1159     1190     1325     1633      555     554
3 soft rain    2126     1198     1281     1408     1068     542
4 hard rain     386     202      386      352      222      96
5 snow         2         2      124        5       NA       NA
  n_Wom...⁶ n_Wom...⁷
  <fct>    <int>
```

```
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
<b>Good</b>								
sunny	24399	14969	19208	19420	11971	6958	9417	9298
cloud	1159	1190	1325	1633	555	554	630	774
<b>Bad</b>								
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	NA	NA	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,

```
1 library(fixest) # for faster regression with fixed effect
2
3 models <- list(
4     "(1)" = feglm(is_hospitalized ~ type_person + positive_alcohol + positive_drug | age_c + gender,
5                  family = binomial(logit), data = data),
6     "(2)" = feglm(is_hospitalized ~ type_person + positive_alcohol + positive_drug | age_c + gender + type_vehicle,
7                  family = binomial(logit), data = data),
8     "(3)" = feglm(is_hospitalized ~ type_person + positive_alcohol + positive_drug | age_c + gender + type_vehicle +
9                  family = binomial(logit), data = data),
10    "(4)" = feglm(is_died ~ type_person + positive_alcohol + positive_drug | age_c + gender,
11                  family = binomial(logit), data = data),
12    "(5)" = feglm(is_died ~ type_person + positive_alcohol + positive_drug | age_c + gender + type_vehicle,
13                  family = binomial(logit), data = data),
14    "(6)" = feglm(is_died ~ type_person + positive_alcohol + positive_drug | age_c + gender + type_vehicle + weather,
15                  family = binomial(logit), data = data)
16 )
```

# modelsummary: Init

```
1 modelsummary(models)
```

	(1)	(2)	(3)	(4)	(5)	(6)
type_personPassenger	0.049 (0.104)	0.530 (0.071)	0.507 (0.070)	-1.781 (0.759)	-1.575 (0.783)	-1.565 (0.784)
type_personPedestrian	2.124 (0.115)	2.402 (0.066)	2.323 (0.064)	2.280 (0.301)	2.418 (0.287)	2.422 (0.285)
positive_alcoholTRUE	-0.077 (0.088)	0.310 (0.095)	0.353 (0.093)	-13.710 (0.053)	-13.455 (0.064)	-13.492 (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

```
1 cm <- c(  
2   "type_personpassenger" = "Passenger",  
3   "type_personpedestrian" = "Pedestrian",  
4   "positive_alcoholTRUE" = "Positive Alcohol"  
5 )  
6  
7 modelsummary(models,  
8   coef_map = cm  
9 )
```

	(1)	(2)	(3)	(4)	(5)	(6)
Positive Alcohol	-0.077 (0.088)	0.310 (0.095)	0.353 (0.093)	-13.710 (0.053)	-13.455 (0.064)	-13.492 (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 Statistics

```
1 cm <- c(
2   "type_personpassenger" = "Passenger",
3   "type_personpedestrian" = "Pedestrian",
4   "positive_alcoholTRUE" = "Positive Alcohol"
5 )
6
7 gm <- tibble(
8   raw = c("nobs", "FE: age_c", "FE: gender", "FE: type_vehicle",
9   clean = c("Observations", "FE: Age Group", "FE: Gender", "FE: T
10  fmt = c(0, 0, 0, 0, 0)
11 )
12
13 modelsummary(models,
14   coef_map = cm,
15   gof_map = gm
16 )
```

	(1)	(2)	(3)	(4)	(5)	(6)
Positive Alcohol	-0.077 (0.088)	0.310 (0.095)	0.353 (0.093)	-13.710 (0.053)	-13.455 (0.064)	-13.492 (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	X
FE: Weather			X			X

# modelsummary: Stars & Headers

```
1 code_line_numbers="7,16"
2 cm <- c(
3   "type_personpassenger" = "Passenger",
4   "type_personpedestrian" = "Pedestrian",
5   "positive_alcoholTRUE" = "Positive Alcohol"
6 )
7
8 gm <- tibble(
9   raw = c("nobs", "FE: age_c", "FE: gender", "FE: type_vehicle",
10  clean = c("Observations", "FE: Age Group", "FE: Gender", "FE: T
11  fmt = c(0, 0, 0, 0, 0)
12 )
13
14 modelsummary(models,
15   stars = c("+" = .1, "*" = .05, "**" = .01),
16   coef_map = cm,
17   gof_map = gm) |>
18   add_header_above(c(" ", "Hospitalization" = 3, "Died within 24 hours" = 3))
```

	Hospitalization			Died within 24 hours		
	(1)	(2)	(3)	(4)	(5)	(6)
Positive Alcohol	-0.077 (0.088)	0.310** (0.095)	0.353** (0.093)	-13.710** (0.053)	-13.455** (0.064)	-13.492** (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	X
FE: Weather				X		X

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

# modelsummary: Export to *LATEX*

```
1 cm <- c(
2   "type_personpassenger" = "Passenger",
3   "type_personpedestrian" = "Pedestrian",
4   "positive_alcoholTRUE" = "Positive Alcohol"
5 )
6
7 gm <- tibble(
8   raw = c("nobs", "FE: age_c", "FE: gender", "FE: type_vehicle",
9   clean = c("Observations", "FE: Age Group", "FE: Gender", "FE: T
10  fmt = c(0, 0, 0, 0, 0)
11 )
12
13 modelsummary(models,
14   output = "latex_tabular",
15   stars = c("+ = .1, "*" = .05, "** = .01,
16   coef_map = cm,
17   gof_map = gm) |>
18   add_header_above(c(" ", "Hospitalization" = 3, "Died within 24 ho
19   nts" = 7, hline_after = TRUE)) |>
```

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 (0.104)	0.530** (0.071)	0.507** (0.070)	-1.781* (0.759)	-1.575+ (0.783)	-1.565+ (0.784)
Pedestrian	2.124** (0.115)	2.402** (0.066)	2.323** (0.064)	2.280** (0.301)	2.418** (0.287)	2.422** (0.285)
Positive Alcohol	-0.077 (0.088)	0.310** (0.095)	0.353** (0.093)	-13.710** (0.053)	-13.455** (0.064)	-13.492** (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

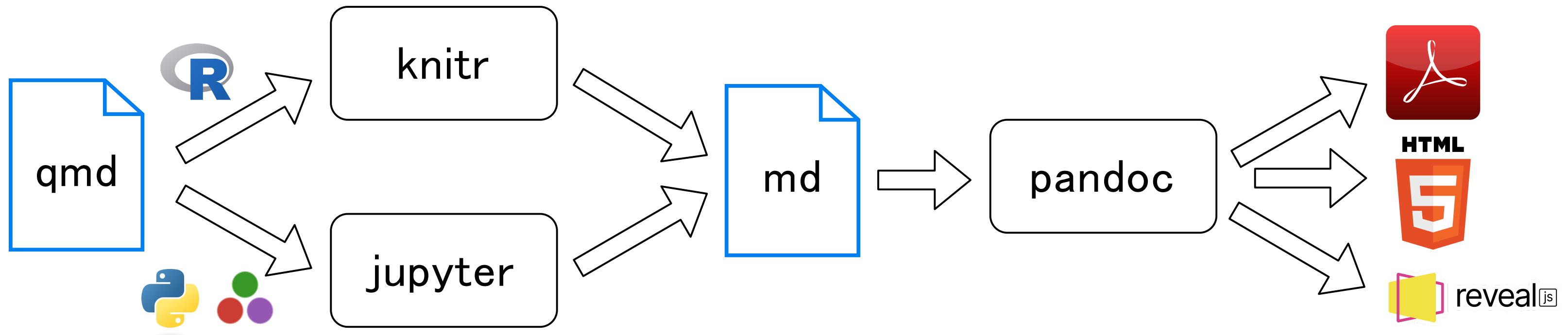
- You can quickly export tibble (dataframe) as latex table by **kableExtra**
- **modelsummary** produces kableExtra object from regression results
- You can see the latex table in **output/tex/** and the compiled results in **code/thesis/**

## 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

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

*LATEX*

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

### Bullet points

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

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

### Enumerate

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

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

# Quarto (Markdown) Is Easy-version of *LATEX*!

Quarto (Markdown)

Text Formatting

```
1 **bold letters**  
2 _italic letters_  
3 $f_n(x)$
```

Display Math

```
1 $$  
2 \begin{aligned}  
3 u(x) &= \frac{c^{1 - \gamma}}{1 - \gamma} \\  
4 u'(x) &= c^{1 - \gamma}  
5 \end{aligned}  
6 $$
```

Cross References

```
1 @bib_tex_key  
2 @fig-label_fig  
3 @tbl-label_tbl
```

*LATEX*

```
1 \textbf{bold letters}  
2 \textit{italic letters}  
3 \underline{$f_n(x)$}
```

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

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

# Quarto Presentation

## Quarto (Reveal.js)

```
1 ## First Slide  
2  
3 Blah, Blah, Blah  
4  
5 ## Second Slide  
6  
7 Yeah, Yeah, Yeah
```

## *LATEX* (Beamer)

```
1 \begin{frame}{First Slide}  
2  
3 Blah, Blah, Blah  
4  
5 \end{frame}  
6  
7 \begin{frame}{Second Slide}  
8  
9 Yeah, Yeah, Yeah  
10  
11 \end{frame}
```

# Quarto Presentation: Fragments

Quarto (Reveal.js)

Pause

```
1 First fragment  
2  
3 . . .  
4  
5 Second fragment
```

Incremental List

```
1 :::: {.incremental}  
2  
3 - 1st element  
4 - 2nd element  
5 - 3rd element  
6  
7 :::
```

*LATEX* (Beamer)

```
1 First fragment  
2  
3 \pause  
4  
5 Second fragment
```

```
1 \begin{itemize} [<+>]  
2   \item 1st element  
3   \item 2nd element  
4   \item 3rd element  
5 \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 JAIO. <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. <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](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 In 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](https://cran.r-project.org/web/packages/kableExtra/vignettes/awesome_table_in_pdf.pdf).