

Journal (reproducible report)

Johannes Julius Halbe

2020-11-27

Contents

Challenge 1	1
Challenge 2	4
Challenge 3	6
Challenge 4	9

This is an .Rmd file. It is plain text with special features. Any time you write just like this, it will be compiled to normal text in the website. If you put a # in front of your text, it will create a top level-header.

Challenge 1

Last compiled: 2020-12-05

```
# Challenge 01 ----

# 1.0 Load libraries ----
library(tidyverse)

#Excel Files
library(readxl)

# 2.0 Importing Files ----
bikes_tbl      <- read_xlsx("docs/00_data/01_bike_sales/01_raw_data/bikes.xlsx")
orderlines_tbl <- read_xlsx("docs/00_data/01_bike_sales/01_raw_data/orderlines.xlsx")
bikeshops_tbl  <- read_xlsx("docs/00_data/01_bike_sales/01_raw_data/bikeshops.xlsx")

# 3.0 Examining Data ----

#orderlines_tbl

#glimpse(orderlines_tbl)

#view(orderlines_tbl)
```

```

# 4.0 Joining Data ----

bike_orderlines_joined_tbl <- orderlines_tbl %>%
  left_join(bikes_tbl, by = c("product.id" = "bike.id")) %>%
  left_join(bikeshops_tbl, by = c("customer.id" = "bikeshop.id"))

# 5.0 Wrangling Data ----
bike_orderlines_wrangled_tbl <- bike_orderlines_joined_tbl %>%
  select(-...1) %>%
  rename(bikeshop = name) %>%
  set_names(names(.) %>% str_replace_all("\\\\.", "_")) %>%
  separate(col = location,
            into = c("city", "state"),
            sep = ", ") %>%
  mutate(total_price = price * quantity)

# 6.0 Business Insights ----
# 6.1 Sales by location ----

# Step 1 - Manipulate

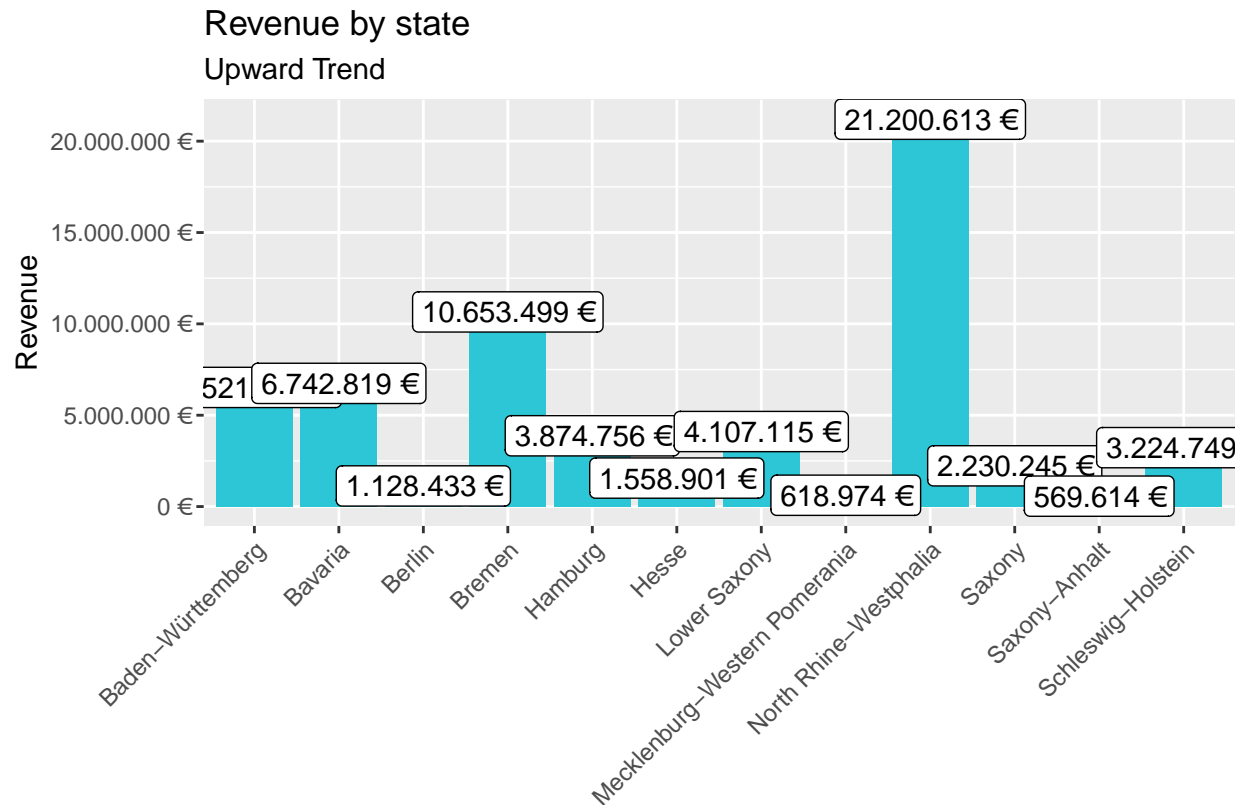
sales_by_location_tbl <- bike_orderlines_wrangled_tbl %>%
  select(state, total_price) %>%
  group_by(state) %>%
  summarize(sales = sum(total_price)) %>%
  mutate(sales_text = scales::dollar(sales, big.mark = ".",
                                     decimal.mark = ",",
                                     prefix = "",
                                     suffix = " €"))

# Step 2 - Visualize

sales_by_location_tbl %>%
  ggplot(aes(x = state, y = sales)) +
  geom_col(fill = "#2DC6D6") +
  geom_label(aes(label = sales_text)) +
  geom_smooth(method = "lm", se = FALSE) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_y_continuous(labels = scales::dollar_format(big.mark = ".",
                                                    decimal.mark = ",",
                                                    prefix = "",
                                                    suffix = " €")) +

  labs(
    title = "Revenue by state",
    subtitle = "Upward Trend",
    x = "",
    y = "Revenue"
  )

```



```
# 6.2 Sales by location & year ----

# Step 1 - Manipulate
library(lubridate)

sales_by_location_year_tbl <- bike_orderlines_wrangled_tbl %>%

  select(state, total_price, order_date) %>%
  mutate(year = year(order_date)) %>%
  group_by(state, year) %>%
  summarise(sales = sum(total_price)) %>%
  ungroup() %>%

  mutate(sales_text = scales::dollar(sales, big.mark = ".",
    decimal.mark = ",",
    prefix = "",
    suffix = " €"))

# Step 2 - Visualize
sales_by_location_year_tbl %>%

  # Set up x, y, fill
  ggplot(aes(x = year, y = sales)) +

  # Geometries
  geom_col() + # Run up to here to get a stacked bar plot
```

```

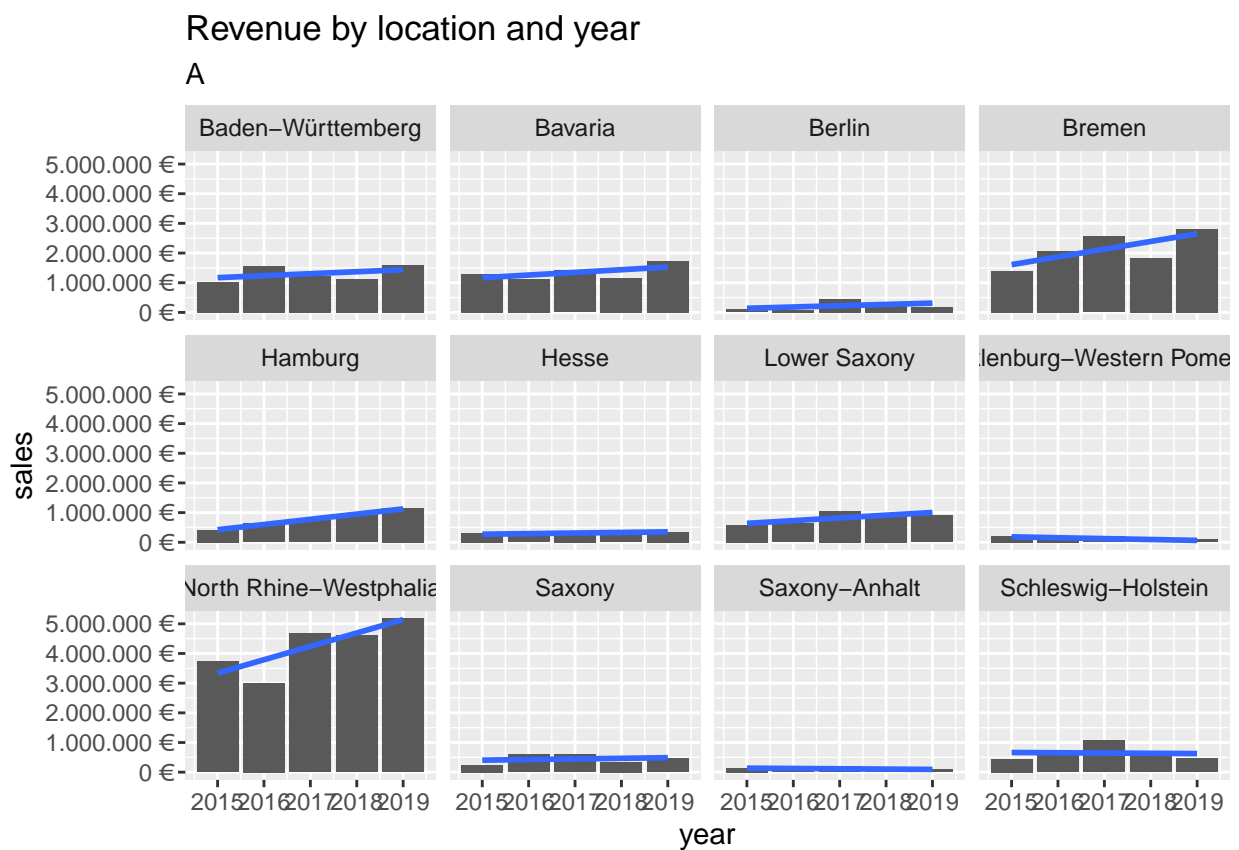
geom_smooth(method = "lm", se = FALSE) +

# Facet
facet_wrap(~ state) +

# Formatting
scale_y_continuous(labels = scales::dollar_format(big.mark = ".",
                                                    decimal.mark = ",",
                                                    prefix = "",
                                                    suffix = " €")) +

labs(
  title = "Revenue by location and year",
  subtitle = "A"
)

```



Challenge 2

Last compiled: 2020-12-05

#1. API

```

library(tidyverse)
library(httr)
library(jsonlite)

```

```

library(tibble)
library(keyring)

keyring::key_set("token")

resp <- GET("https://www.ncdc.noaa.gov/cdo-web/api/v2/stations?limit=1000", add_headers(token = key_get("token")))

stations_tbl <- resp %>%
  .$content %>%
  rawToChar() %>%
  fromJSON() %>% .$results

head(stations_tbl,10)

```

```

##      elevation   mindate   maxdate latitude      name datacoverage
## 1      139.0 1948-01-01 2014-01-01 31.57020      ABBEVILLE, AL US      0.8813 COOP:010
## 2      249.3 1938-01-01 2015-11-01 34.25530      ADDISON, AL US      0.5059 COOP:010
## 3      302.1 1940-05-01 1962-03-01 34.41667      ADDISON CENTRAL TOWER, AL US      0.9658 COOP:010
## 4      172.2 1995-04-01 2015-11-01 33.17833      ALABASTER SHELBY CO AIRPORT, AL US      0.8064 COOP:010
## 5      183.8 1949-01-01 1949-12-01 34.68910      BELLE MINA 2 N, AL US      1.0000 COOP:010
## 6       34.1 1935-05-01 1936-11-01 31.13333      ALAGA, AL US      0.2624 COOP:010
## 7       53.3 1940-11-01 2014-12-01 32.23220      ALBERTA, AL US      0.9888 COOP:010
## 8      348.1 1931-01-01 1977-06-01 34.23333      ALBERTVILLE, AL US      0.9535 COOP:010
## 9      195.1 1969-10-01 2015-11-01 32.94520      ALEXANDER CITY, AL US      0.9946 COOP:010
## 10     200.9 1942-11-01 1969-10-01 32.98333      ALEXANDER CITY 6 NE, AL US      0.9629 COOP:010
##      elevationUnit longitude
## 1      METERS -85.24820
## 2      METERS -87.18140
## 3      METERS -87.31667
## 4      METERS -86.78167
## 5      METERS -86.88190
## 6      METERS -85.06667
## 7      METERS -87.41040
## 8      METERS -86.16667
## 9      METERS -85.94800
## 10     METERS -85.86667

```

#2. Web scraping

LIBRARIES ----

```

library(tidyverse) # Main Package - Loads dplyr, purrr, etc.
library(rvest)     # HTML Hacking & Web Scraping
library(xopen)     # Quickly opening URLs
library(jsonlite)  # converts JSON files to R objects
library(glue)      # concatenate strings
library(stringi)   # character string/text processing

url <- "https://www.rosebikes.de/fahrr%C3%A4der/rennrad"
html <- url %>%
  read_html()

```

```

model_name <- html %>%
  html_nodes(".catalog-category-bikes__title > span") %>%
  html_text() %>%
  stringr::str_extract("(?<=\n).*(?=\n)")

model_price_cent <- html %>%
  html_nodes(".catalog-category-bikes__price-title") %>%
  html_text() %>%
  stringr::str_extract("(?<=ab\\s).*(?=\s€)") %>%
  str_replace_all(c("\\." = "", "," = "")) %>%
  as.numeric()

model_price_EUR = model_price_cent / 100

bikes_tbl <- tibble(model_name, model_price_EUR)
head(bikes_tbl, 10)

```

```

## # A tibble: 9 x 2
##   model_name      model_price_EUR
##   <chr>          <dbl>
## 1 PRO SL DISC      1599
## 2 PRO SL           1199
## 3 REVEAL FOUR DISC 2499
## 4 REVEAL FOUR      2099
## 5 REVEAL SIX DISC  3499
## 6 X-LITE FOUR DISC 2699
## 7 X-LITE FOUR      2199
## 8 X-LITE SIX DISC  3899
## 9 X-LITE SIX       3499

```

Challange 3

```

# 1.0 Libraries-----

library(vroom)
library(tidyverse)
library(data.table)
library(tictoc)
library(lubridate)

# 2.0 10 US Companies with most patents-----

col_types <- list(
  id = col_character(),
  type = col_integer(),
  name_first = col_skip(),
  name_last = col_skip(),
  organization = col_character()
)

assignee_tbl <- vroom(

```

```

file      = "docs/02_data_wrangling/assignee.tsv",
delim     = "\t",
col_types = col_types,
na        = c("", "NA", "NULL")
)

setDT(assignee_tbl)

col_types <- list(
  patent_id = col_character(),
  assignee_id = col_character(),
  location_id = col_skip()
)

patent_assignee_tbl <- vroom(
  file      = "docs/02_data_wrangling/patent_assignee.tsv",
  delim     = "\t",
  col_types = col_types,
  na        = c("", "NA", "NULL")
)

setDT(patent_assignee_tbl)

combined_data_t1 <- merge(x = patent_assignee_tbl, y = assignee_tbl,
  by.x      = "assignee_id",
  by.y      = "id",
  all.x     = FALSE,
  all.y     = FALSE)

top_ten_US <- combined_data_t1[type == 2, .N, by = organization][order(-N)]
head(top_ten_US, 10)

# 3.0 US company with the most patents granted in 2019

col_types <- list(
  id = col_character(),
  type = col_skip(),
  number = col_skip(),
  country = col_skip(),
  date = col_date("%Y-%m-%d"),
  abstract = col_skip(),
  title = col_skip(),
  kind = col_skip(),
  num_claims = col_skip(),
  filename = col_skip(),
  withdrawn = col_skip()
)

patent_tbl <- vroom(

```

```

file      = "docs/02_data_wrangling/patent.tsv",
delim     = "\t",
col_types = col_types,
na        = c("", "NA", "NULL")
)
setDT(patent_tbl)

combined_data_t2 <- merge(x = combined_data_t1, y = patent_tbl,
  by.x = "patent_id",
  by.y = "id",
  all.x = FALSE,
  all.y = FALSE)

patents_granted <- combined_data_t2[lubridate::year(date) == "2019" & type == 2,.N, by=organization][order]

head(patents_granted,10)

# 4.0 -----

col_types <- list(
  uuid = col_skip(),
  patent_id = col_character(),
  mainclass_id = col_character(),
  subclass_id = col_skip(),
  sequence = col_skip()
)

uspc_tbl <- vroom(
  file      = "docs/02_data_wrangling/uspc.tsv",
  delim     = "\t",
  col_types = col_types,
  na        = c("", "NA", "NULL")
)
setDT(uspc_tbl)

combined_data_t3 <- merge(x = combined_data_t1, y = uspc_tbl,
  by      = "patent_id",
  all.x = FALSE,
  all.y = FALSE)

combined_data_t3[, ":="(assignee_id = NULL)]

# 4.1 Most innovative tech sector?
#

tic()
most_inno_tech <- combined_data_t3[, unique(patent_id), by=mainclass_id][, .N , by =mainclass_id][order]

head(most_inno_tech$mainclass_id,1)
toc()

# 4.2 Top 5 USPTO tech main classes

```



```
tic()
top10_ww <- combined_data_t1[type == 2 | type == 3, .N , by = organization][order(-N)][1:10]
toc()
```

```
tic()
most_inno_tech_top10 <- combined_data_t3[organization %in% top10_ww$organization , unique(patent_id), by = organization]
head(most_inno_tech$mainclass_id,5)
toc()
```

```
read_rds("docs/top_ten_US")
```

```
##              organization      N
## 1: International Business Machines Corporation 139091
## 2:              General Electric Company 47121
## 3:              Intel Corporation 42156
## 4:  Hewlett-Packard Development Company, L.P. 35572
## 5:              Microsoft Corporation 30085
## 6:              Micron Technology, Inc. 28000
## 7:              QUALCOMM Incorporated 24702
## 8:              Texas Instruments Incorporated 24181
## 9:              Xerox Corporation 23173
## 10:              Apple Inc. 21820
```

```
read_rds("docs/patents_granted")
```

```
##              organization      N
## 1: International Business Machines Corporation 9265
## 2:              Intel Corporation 3526
## 3:  Microsoft Technology Licensing, LLC 3106
## 4:              Apple Inc. 2817
## 5:  Ford Global Technologies, LLC 2624
## 6:  Amazon Technologies, Inc. 2533
## 7:  QUALCOMM Incorporated 2359
## 8:              Google Inc. 2290
## 9:  General Electric Company 1860
## 10: Hewlett-Packard Development Company, L.P. 1589
```

```
read_rds("docs/most_inno_tech")
```

```
## [1] "257"
```

```
read_rds("docs/most_inno_main_class")
```

```
## [1] "257" "438" "370" "709" "365"
```

Challange 4

```

# 1.0 Libraries

library(tidyverse)
library(lubridate)
library(ggrepel)
library(maps)
library(ggthemes)
library(viridis)

# 2.0 Map the time course of the cumulative Covid-19 cases!

# 2.1 Import Data

covid_data_tbl <- read_csv("https://opendata.ecdc.europa.eu/covid19/casedistribution/csv")

# 2.2 Data wrangling

cum_c19_cases_tbl <- covid_data_tbl %>%

  mutate(date := lubridate::dmy(dateRep)) %>%
  select(date, countriesAndTerritories, cases) %>%
  filter(countriesAndTerritories %in% c("Germany",
                                         "France",
                                         "Spain",
                                         "United_Kingdom",
                                         "United_States_of_America"),
         year(date) == "2020") %>%

  group_by(countriesAndTerritories) %>%
  arrange(date, .by_group = TRUE) %>%
  mutate(cum_cases = cumsum(cases)) %>%
  ungroup()

# 2.3. Data visualization
cum_c19_cases_tbl %>%
  ggplot(aes(date, cum_cases, color = countriesAndTerritories)) +
  geom_line(aes(color = countriesAndTerritories), size = 1) +
  geom_label_repel(
    data = cum_c19_cases_tbl %>%
      filter(date %in% max(date),
             countriesAndTerritories == "United_States_of_America"),
    label = scales::dollar(max(cum_c19_cases_tbl$cum_cases),
                           big.mark = ".",
                           decimal.mark = ",",
                           prefix = ""),
    segment.size = 0.2,
    min.segment.length = 1,
    box.padding = 1.5
  ) +

  scale_y_continuous(labels = scales::dollar_format(scale = 1e-6,
                                                    prefix = ""),

```

```

suffix = " M")) +
scale_x_date(date_labels = "%B", date_breaks = "1 month") +

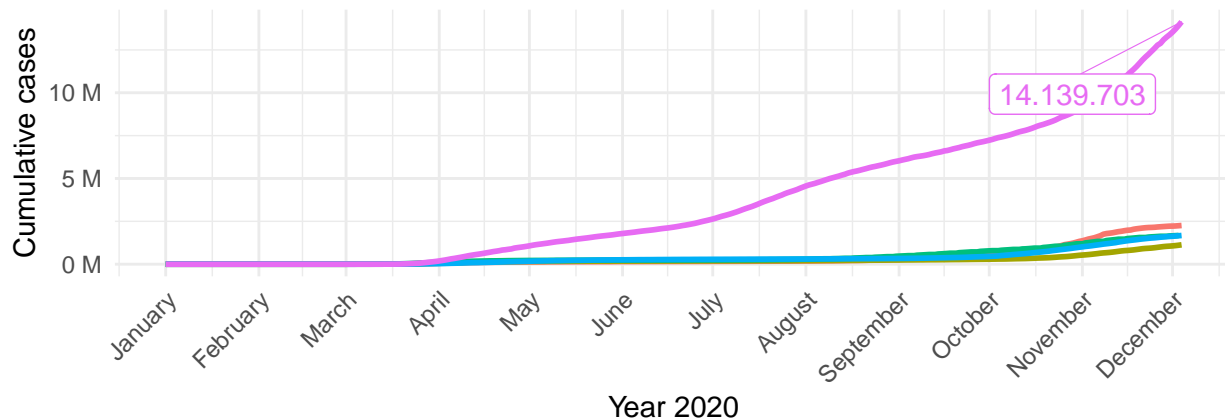
labs(
  title = "COVID-19 confirmed cases worldwide",
  subtitle = str_glue("As of {Sys.Date()}, the USA had a lot more cases than all european countries"),
  x = "Year 2020",
  y = "Cumulative cases",
  color = "Countries"
) +

theme_minimal() +
theme(
  axis.text.x = element_text(angle = 45, hjust = 1),
  legend.position = "bottom",
  plot.title = element_text(face = "bold"),
  legend.margin = margin(0.2, 0.2, 0.2, 0.2, "cm"),
  legend.direction = "vertical"
)

```

COVID-19 confirmed cases worldwide

As of 2020-12-05, the USA had a lot more cases than all european countries



Countries

14.139.703	France
14.139.703	Germany
14.139.703	Spain
14.139.703	United_Kingdom
14.139.703	United_States_of_America

3.0 Mortality rate -----

3.1 Import Data

```

covid_data_tbl <- read_csv("https://opendata.ecdc.europa.eu/covid19/casedistribution/csv")

world <- map_data("world")

# 3.2 Data wrangling

covid_deaths_tbl <- covid_data_tbl %>%
  mutate(date := lubridate::dmy(dateRep)) %>%
  select(date, countriesAndTerritories, deaths, popData2019) %>%
  filter( year(date) == "2020") %>%

  group_by(countriesAndTerritories) %>%
  arrange(date, .by_group = TRUE) %>%
  mutate(total_deaths = cumsum(deaths)) %>%
  ungroup() %>%

  filter(date == as.Date(date("2020-12-01"))) %>%
  mutate(mortality_pct := 100 * total_deaths / popData2019) %>%
  select(date, countriesAndTerritories, mortality_pct) %>%
  mutate(across(countriesAndTerritories, str_replace_all, "_", " ")) %>%
  mutate(countriesAndTerritories = case_when(
    countriesAndTerritories == "United Kingdom" ~ "UK",
    countriesAndTerritories == "United States of America" ~ "USA",
    countriesAndTerritories == "Czechia" ~ "Czech Republic",
    TRUE ~ countriesAndTerritories
  ))

covid_mortality_tbl <- covid_deaths_tbl %>%
  full_join(world %>% select(region,long,lat), by = c("countriesAndTerritories" = "region"))

# 3.3. Data visualization

covid_mortality_tbl %>% ggplot()+
  geom_map(map = world,
    aes(long, lat, map_id = countriesAndTerritories),
    color="#2b2b2b", fill=NA, size=0.15) +
  geom_map(map = world,
    aes(fill=mortality_pct,
      map_id = countriesAndTerritories),
    color="white", size=0.15) +

  scale_fill_continuous(
    name = "Mortality Rate",
    low  = "#FFD700",
    high = "#800000")+

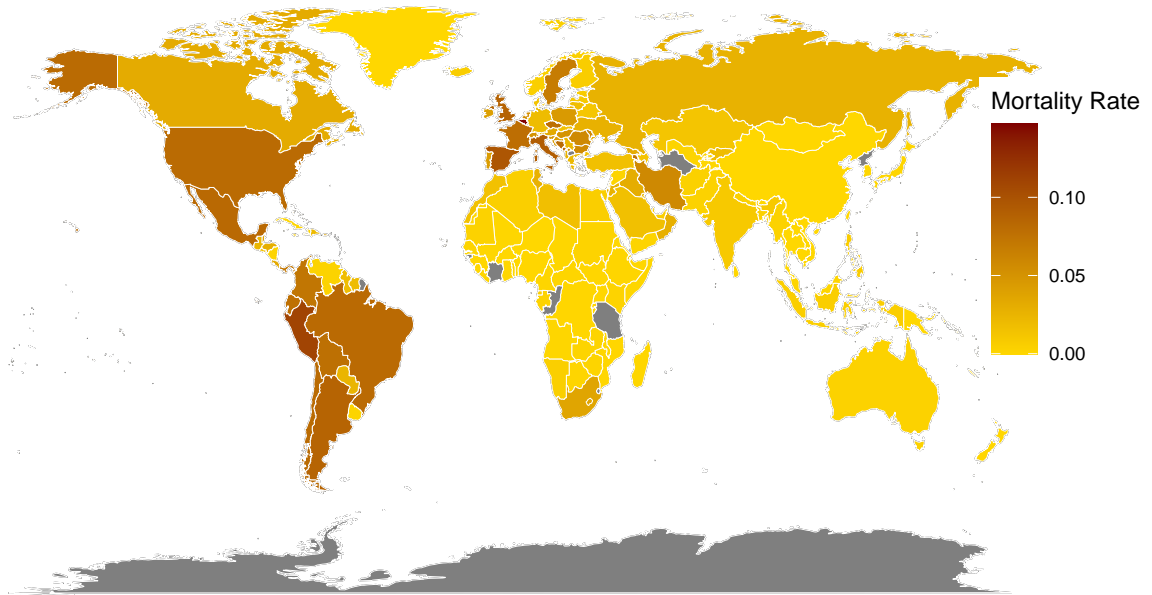
  labs(title = "Confirmed COVID-19 deaths relative to the size of the population",
    subtitle = "More than X-Million confirmed COVID-19 deaths worldwide",
    caption = "Date: 2020-12-01") +

  theme_map() +

```

```
theme(  
  plot.margin=margin(20,20,20,20),  
  legend.position = c(0.9, 0.4))
```

Confirmed COVID-19 deaths relative to the size of the population
More than X-Million confirmed COVID-19 deaths worldwide



Date: 2020-12-01

The End!