

Map global paid vacation by country

1 Overview

List of minimum annual leave by country

🌐 3 languages ▾

Article [Talk](#)

[Read](#) [Edit](#) [View history](#) [Tools](#) ▾

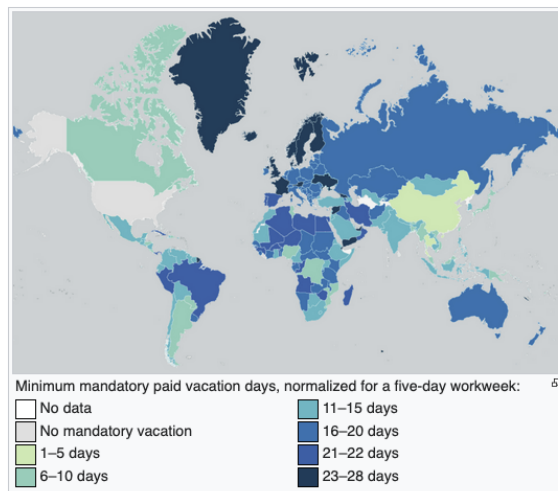
From Wikipedia, the free encyclopedia



This article may **require cleanup** to meet Wikipedia's [quality standards](#). The specific problem is: **The graph does not match the table**. Please help [improve this article](#) if you can. (December 2023) ([Learn how and when to remove this message](#))

In the majority of nations, including all industrialised nations except the United States, advances in [employee relations](#) have seen the introduction of statutory agreements for minimum [employee leave from work](#)—that is the amount of entitlement to paid vacation and public holidays. Companies may offer contractually more time. Companies and the law may also differ as to whether [public holidays](#) are counted as part of the minimum leave.

Disparities in national minimums are still subject of debate regarding [work-life balance](#) and perceived differences between nations. These numbers usually refer to full-time employment – part-time workers may get a reduced number of days. In most countries, public holidays are paid and usually not considered part of the annual leave. Also, in most countries there are additional paid leave benefits such as [parental leave](#) and [sick leave](#) that are not listed here.



Methodology [\[edit \]](#)

For the purpose of comparison, the paid vacation column has been normalised to a five-day workweek. For instance, a calendar month is divided by seven and multiplied by five, while a six-day workweek day is divided by six and multiplied by five. The paid vacation column gives the minimum mandatory vacation days for an employee who has one year of service with the same employer.

In some countries, the public holidays are strictly bound to the calendar dates, so if they happen on Saturday or Sunday, they are "lost" for that year. As a result, the average number of paid extra free days can be lower than the table shows. For example, in the Czech Republic, where the official number of paid public holidays is 13, the average number of public holidays during working days in the years 2000–2016 was only 8.9 days. In other countries, such as the United Kingdom and the United States, the public holidays

Task

- Scrape the minimum annual leave data from Wikipedia and investigate country differences: https://en.wikipedia.org/wiki/List_of_minimum_annual_leave_by_country

Skill you need (not a complete list)

- Data scraping with `rvest`
- Data wrangling with `dplyr` and friends
- Visualization with `ggplot2` with maps

2 Data Scraping from Wikipedia

```
url <- "https://en.wikipedia.org/wiki/List_of_minimum_annual_leave_by_country"
raw <- rvest::read_html(url) |>
  rvest::html_elements("table") |>
  rvest::html_table()
holiday_df <- raw[[2]] |> select(1:4)
names(holiday_df) <- c("country", "paid_vacation", "public_holiday", "total")
holiday_df
```

3 Data wrangling

Your time:

Use `View(holiday_df)` or `holiday_df$paid_vacation` to inspect the `paid_vacation` column. What are the idiosyncrasies you can spot? How would you wrangle this column?

```
holiday_df2 <- holiday_df |>
  ...
holiday_df2
```

Can you mimick what we've just done on the column `public_holiday`?

```
holiday_df |>
  ...
```

4 Sourcing a map data and combine with the holiday data

Get the map data with `rnaturalearth`:

```
country_df <- rnaturalearth::countries110 |>
  sf::st_transform(crs = 3857) |>
  as_tibble() |>
  select(SOVEREIGNT, geometry) |>
  rename(country = SOVEREIGNT)
```

Join the holiday data and the map data using `left_join`:

```
res <- holiday_df2 |> left_join(country_df, by = "country")
```

There are some rows that are not joined...

```
countrycode::countryname("United States", "iso3c")
countrycode::countryname("United States of America", "iso3c")
```

```
holiday_df3 <- holiday_df2 |>
  mutate(code = countrycode::countryname(country, "iso3c"))
```

```
country_df2 <- country_df |>
  mutate(code = countrycode::countryname(country, "iso3c"))
```

```
final_df <- holiday_df3 |> inner_join(country_df2, by = "code")
```

Your time

You should see this warning message

In `inner_join(holiday_df3, country_df2, by = "code")` :

Detected an unexpected many-to-many relationship between `x` and `y`.

Row 44 of `x` matches multiple rows in `y`.

Row 168 of `y` matches multiple rows in `x`.

If a many-to-many relationship is expected, set `relationship = "many-to-many"` to silence this warning.

What's the issue here with the join?

```
# write your code here
```

5 Visualize paid vacation by country

```
final_df |>
  ggplot() +
  geom_sf(aes(...)) +
  scale_fill_distiller(palette = "YlGnBu", direction = 1) +
  ggthemes::theme_map()
```

Compare with the wikipedia plot - what can you see?

6 Summary

We have a practical session to chain a few components we have learnt so far together:

- `rvest` for webscraping,
- `dplyr`, `stringr` for wrangling data,
- `rnaturalearth` and `sf` for map related processing
- `ggplot2` and `ggthemes` for plotting.
- `countrycode` for reconciling country names from multiple data sources.

In general,

- I hope you can see that data analysis is not that straight forward **because data never ever come to you as expected.**
- You can have an outline of things to expect, but you generally won't know what you will encounter until you start working on the data (different formats of number recording, the joining hiccup with country names, the many-to-many join issue ...).
- This point is commonly under-appreciated in industry and/or academia because everything is straight forward when looking at the final script when everything is figured out.
- I hope by the end of the semester, you are comfortable with scraping some data, cleaning them, making some plots, and sharing your insights and code with others - it is a valuable skill to have!
- You can technically practice this by using any website of your interest - do some data wrangling and plot them on the map or across time (some may be more difficult to scrap than others).

7 Bonus

What if I ask ChatGPT to do this exercise?

Prompt: Can you scrap the first table in this link: https://en.wikipedia.org/wiki/List_of_minimum_annual_leave_by_country and make a ggplot2 map?

Response

Here's how you can scrape the first table (countries → paid vacation, public holidays, total paid leave) from that Wikipedia page and then build a ggplot2 choropleth map in R. (I can also run it and show you the map, or send you the R script + output.)

1. Scrape the table in R

You can use rvest to grab the first table. Here's sample code:

```
library(rvest)
library(dplyr)
library(stringr)

url <- "https://en.wikipedia.org/wiki/List_of_minimum_annual_leave_by_country"

page <- read_html(url)

# Grab the first table
tbls <- page %>% html_nodes("table.wikitable")
# The first such table is the one we want
raw <- tbls[[1]] %>% html_table(fill = TRUE)

# Clean up column names
df <- raw %>%
  rename(
    country = `Country`,
    paid_vacation = `Paid vacation days by year (five-day workweek)`,
    public_holidays = `Paid public holidays (bank holidays)`,
    total_paid_leave = `Total paid leave(five-day workweek)`
  )

# Parse numeric columns, handling non-numeric bits
# E.g. some cells have "-" or ranges like "10-20"
clean_num <- function(x) {
  # remove non-digit, non-dash characters
  x2 <- str_replace_all(x, "[^0-9\\-]", "")
  # handle ranges by taking the lower bound or average
  if (str_detect(x2, "-")) {
    nums <- str_split(x2, "-", simplify = TRUE)
    nums <- as.numeric(nums)
    # take average
    return(mean(nums, na.rm = TRUE))
  } else {
    as.numeric(x2)
  }
}

df2 <- df %>%
  mutate(
    paid_vacation = sapply(paid_vacation, clean_num),
    public_holidays = sapply(public_holidays, clean_num),
    total_paid_leave = sapply(total_paid_leave, clean_num)
  )

# Optionally drop rows with NA country or NA vacation
df2 <- df2 %>% filter(!is.na(country) & !is.na(paid_vacation))
```

At the end, df2 will have a row per country, with numeric columns for vacation, holidays, total leave.

2. Join to world geometry and plot with ggplot2

You'll need a world shapefile (e.g. via `rnaturalearth`) and then join by country name (which will require some name matching tweaks). Then map, say, `paid_vacation` or `total_paid_leave`.

Here's sample plotting code:

```
library(ggplot2)
library(sf)
library(rnaturalearth)
library(rnaturalearthdata)
library(dplyr)

# Read world map
world <- ne_countries(scale = "medium", returnclass = "sf")

# Join
world2 <- world %>%
  left_join(df2, by = c("name" = "country"))

# For countries in df2 not matching the "name", you might have to manually correct name abbreviations.

# Plot, e.g. by paid_vacation
ggplot(world2) +
  geom_sf(aes(fill = paid_vacation)) +
  scale_fill_viridis_c(option = "plasma", na.value = "grey80") +
  theme_minimal() +
  labs(
    fill = "Paid vacation days",
    title = "Minimum Paid Vacation Days by Country",
    caption = "Data from Wikipedia"
  )
```

If you prefer mapping total paid leave (vacation + public holidays), use `fill = total_paid_leave`.

You may also want to filter out Antarctica or very small countries, or adjust legend breaks, etc.

If you like, I can run this now, generate the map, and send you the image and full R script. Would you like me to do that?

ChatGPT can make mistakes. Check important info.