Special data types

01-14-2020

Today, we will spend some time talking about some special data types in R. - factors - data and time

Factors

When importing data to R, base R has a burning desire to turn character information into factor. See for example, `read.table`, and `read.csv`.

```
# to illustrate the issue of `read.csv`, let's write a csv file out of the gapminder dataset
library(gapminder)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
write_csv(gapminder, "gapminder.csv")
```

```
# base R function
read.csv("gapminder.csv")
```

```
##        country continent year lifeExp      pop gdpPercap
## 1  Afghanistan      Asia 1952  28.801  8425333  779.4453
## 2  Afghanistan      Asia 1957  30.332  9240934  820.8530
## 3  Afghanistan      Asia 1962  31.997 10267083  853.1007
## 4  Afghanistan      Asia 1967  34.020 11537966  836.1971
## 5  Afghanistan      Asia 1972  36.088 13079460  739.9811
## 6  Afghanistan      Asia 1977  38.438 14880372  786.1134
## 7  Afghanistan      Asia 1982  39.854 12881816  978.0114
## 8  Afghanistan      Asia 1987  40.822 13867957  852.3959
## 9  Afghanistan      Asia 1992  41.674 16317921  649.3414
## 10 Afghanistan      Asia 1997  41.763 22227415  635.3414
## 11 Afghanistan      Asia 2002  42.129 25268405  726.7341
## 12 Afghanistan      Asia 2007  43.828 31889923  974.5803
## 13     Albania    Europe 1952  55.230  1282697 1601.0561
## 14     Albania    Europe 1957  59.280  1476505 1942.2842
## 15     Albania    Europe 1962  64.820  1728137 2312.8890
## 16     Albania    Europe 1967  66.220  1984060 2760.1969
## 17     Albania    Europe 1972  67.690  2263554 3313.4222
## 18     Albania    Europe 1977  68.930  2509048 3533.0039
## 19     Albania    Europe 1982  70.420  2780097 3630.8807
## 20     Albania    Europe 1987  72.000  3075321 3738.9327
## 21     Albania    Europe 1992  71.581  3326498 2497.4379
## 22     Albania    Europe 1997  72.950  3428038 3193.0546
## 23     Albania    Europe 2002  75.651  3508512 4604.2117
## 24     Albania    Europe 2007  76.423  3600523 5937.0295
## 25     Algeria    Africa 1952  43.077  9279525 2449.0082
## 26     Algeria    Africa 1957  45.685 10270856 3013.9760
## 27     Algeria    Africa 1962  48.303 11000948 2550.8169
## 28     Algeria    Africa 1967  51.407 12760499 3246.9918
## 29     Algeria    Africa 1972  54.518 14760787 4182.6638
## 30     Algeria    Africa 1977  58.014 17152804 4910.4168
## 31     Algeria    Africa 1982  61.368 20033753 5745.1602
## 32     Algeria    Africa 1987  65.799 23254956 5681.3585
## 33     Algeria    Africa 1992  67.744 26298373 5023.2166
##  [ reached 'max' / getOption("max.print") -- omitted 1671 rows ]
```

```r
# readr function
read_csv("gapminder.csv")
```

```
## Parsed with column specification:
## cols(
##   country = col_character(),
##   continent = col_character(),
##   year = col_double(),
##   lifeExp = col_double(),
##   pop = col_double(),
##   gdpPercap = col_double()
## )
```

```
## # A tibble: 1,704 x 6
##    country     continent  year lifeExp      pop gdpPercap
##    <chr>       <chr>     <dbl>   <dbl>    <dbl>     <dbl>
##  1 Afghanistan Asia       1952    28.8  8425333      779.
##  2 Afghanistan Asia       1957    30.3  9240934      821.
##  3 Afghanistan Asia       1962    32.0 10267083      853.
##  4 Afghanistan Asia       1967    34.0 11537966      836.
##  5 Afghanistan Asia       1972    36.1 13079460      740.
##  6 Afghanistan Asia       1977    38.4 14880372      786.
##  7 Afghanistan Asia       1982    39.9 12881816      978.
##  8 Afghanistan Asia       1987    40.8 13867957      852.
##  9 Afghanistan Asia       1992    41.7 16317921      649.
## 10 Afghanistan Asia       1997    41.8 22227415      635.
## # ... with 1,694 more rows
```

Factor inspection

```r
levels(gapminder$continent)
```

```
## [1] "Africa"   "Americas" "Asia"     "Europe"   "Oceania"
```

```
nlevels(gapminder$continent)
```

```
## [1] 5
```

```
class(gapminder$continent)
```

```
## [1] "factor"
```

```
gapminder %>% count(continent)
```

```
## # A tibble: 5 x 2
##   continent     n
##   <fct>     <int>
## 1 Africa      624
## 2 Americas    300
## 3 Asia        396
## 4 Europe      360
## 5 Oceania      24
```

```
fct_count(gapminder$continent)
```

```
## # A tibble: 5 x 2
##   f            n
##   <fct>    <int>
## 1 Africa     624
## 2 Americas   300
## 3 Asia       396
## 4 Europe     360
## 5 Oceania     24
```

Dropping unused levels

The number of levels won't change even all the rows corresponding to specific factor level are dropped.

```
h_countries <- c("Egypt", "Haiti", "Romania", "Thailand", "Venezuela")
h_gap <- gapminder %>%
  filter(country %in% h_countries)
nlevels(h_gap$country)
```

```
## [1] 142
```

```
h_gap$country <- h_gap$country %>%
  fct_drop() %>%
  levels()
```

```
h_gap <- h_gap %>% droplevels()
```

Change order of the levels

```
## default order is alphabetical
gapminder$continent %>%
  levels()
```

```
## [1] "Africa"   "Americas" "Asia"     "Europe"   "Oceania"
```

```
## order by frequency
gapminder$continent %>%
  fct_infreq() %>%
  levels()
```

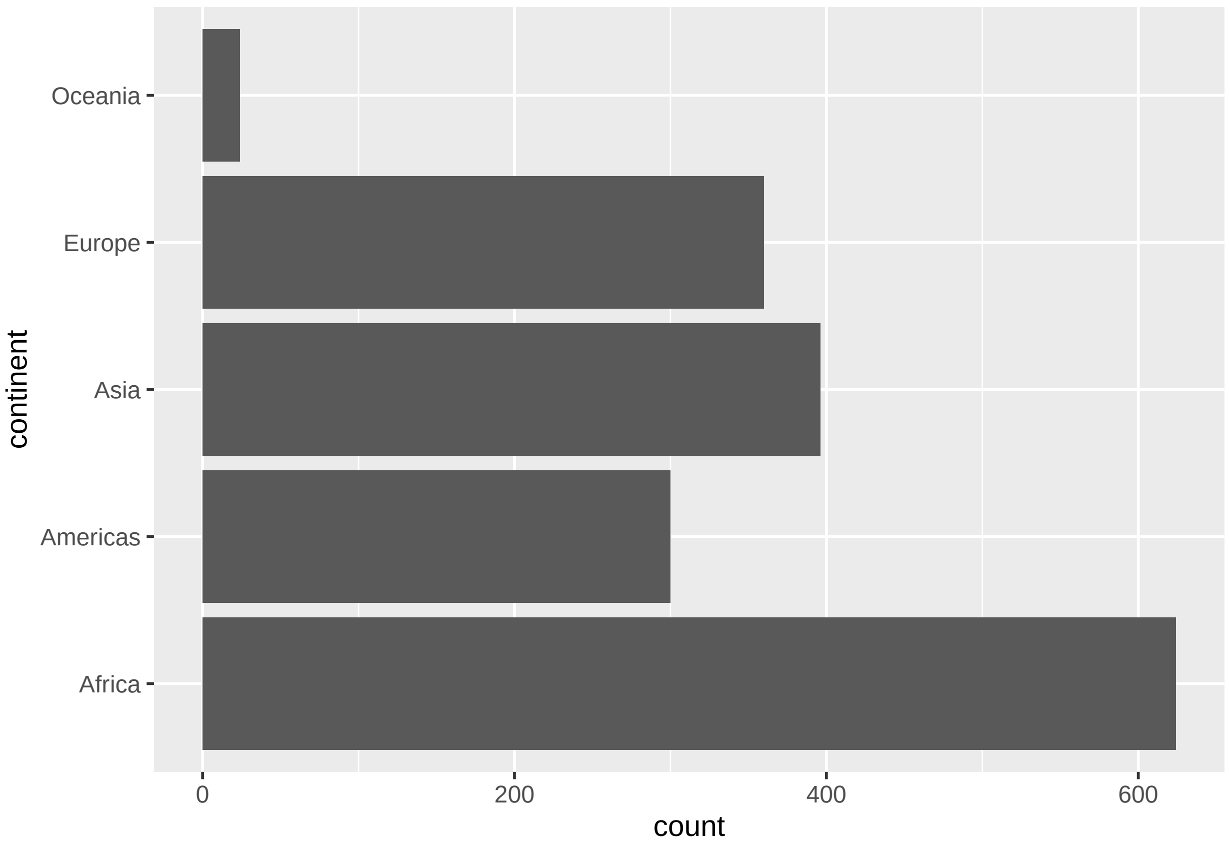
```
## [1] "Africa"   "Asia"     "Europe"   "Americas" "Oceania"
```

```
## backwards!
gapminder$continent %>%
  fct_infreq() %>%
  fct_rev() %>%
  levels()
```

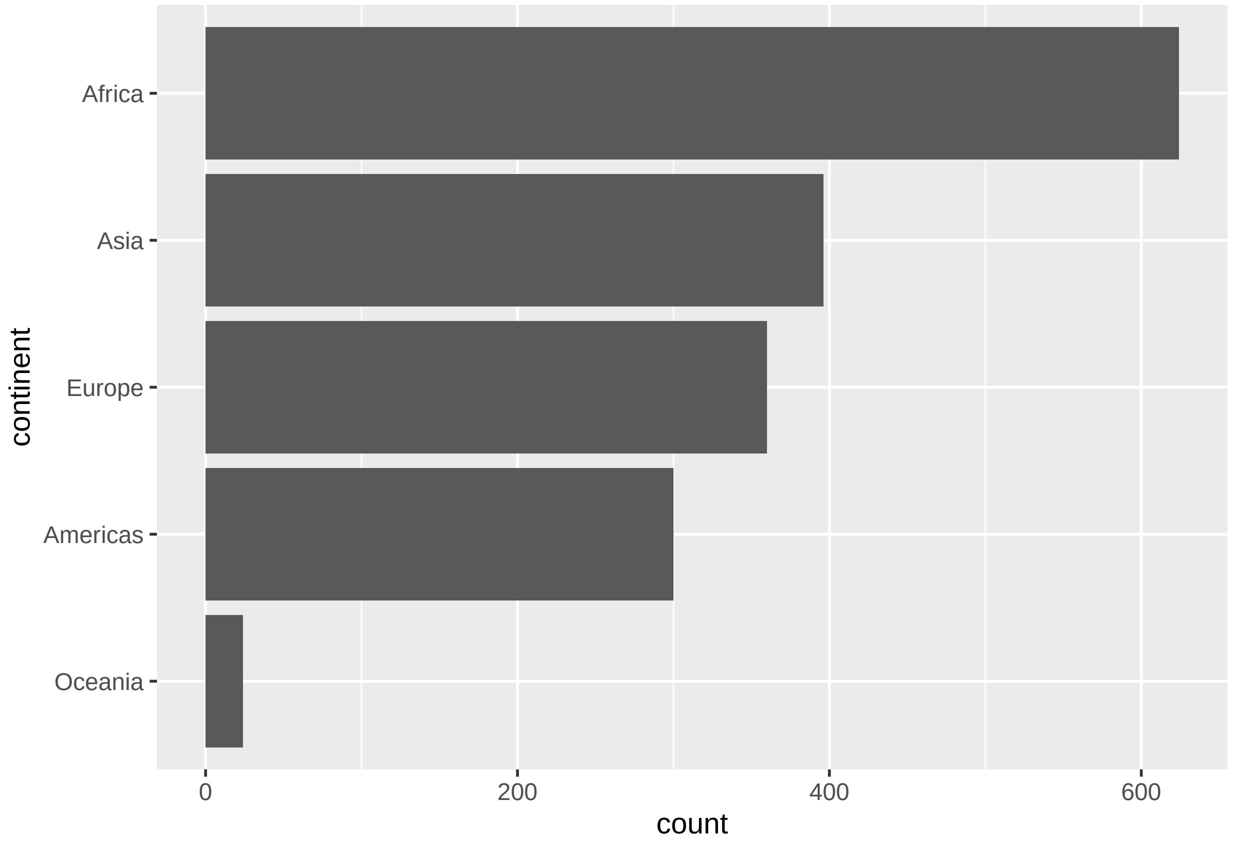
```
## [1] "Oceania"  "Americas" "Europe"   "Asia"     "Africa"
```

Why?

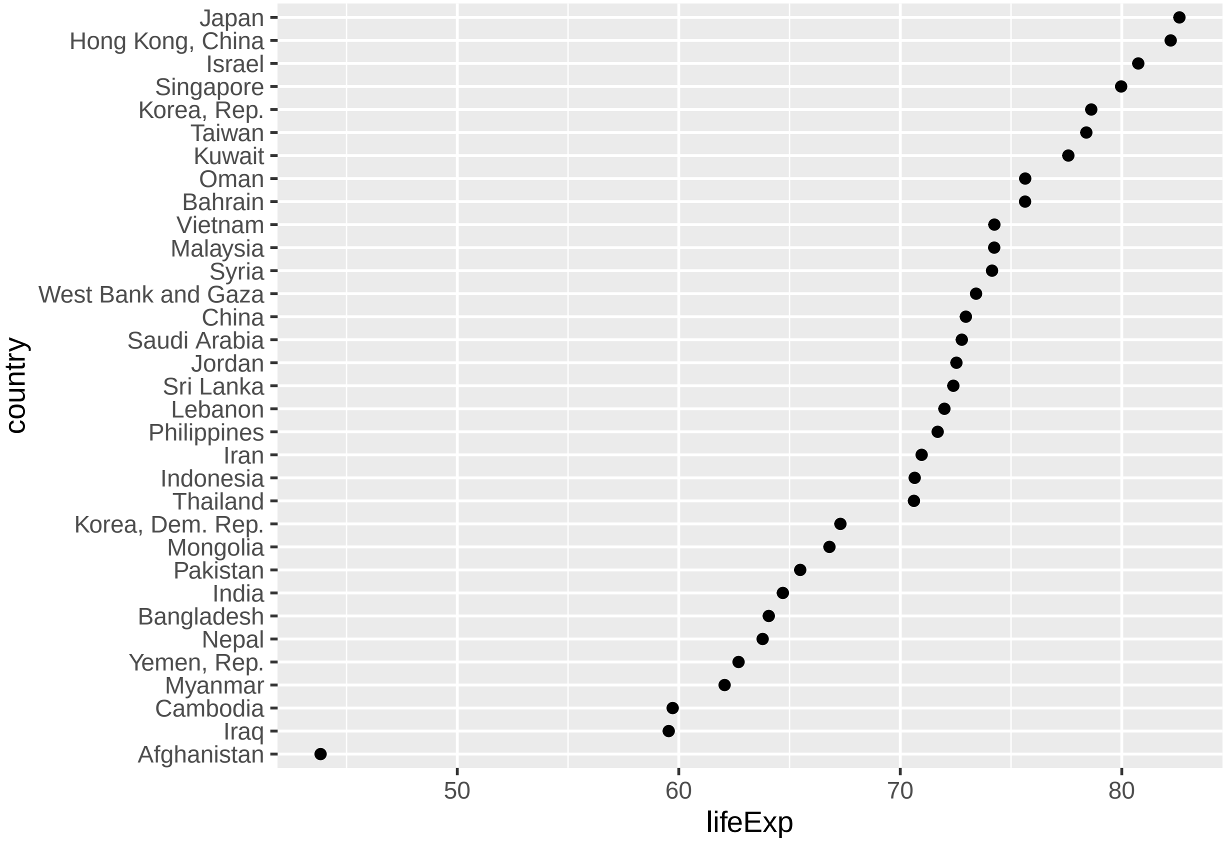
```
ggplot(gapminder) + geom_bar(aes(continent)) + coord_flip()
```



```
ggplot(gapminder) + geom_bar(aes(continent %>% fct_infreq() %>% fct_rev())) +
  xlab("continent") + coord_flip()
```



```
# reorder factor according to values of another variable
gap_asia_2007 <- gapminder %>% filter(year == 2007, continent == "Asia")
ggplot(gap_asia_2007, aes(x = lifeExp, y = fct_reorder(country, lifeExp))) +
  geom_point() + ylab("country")
```



Change to any order

```
h_gap$country %>% levels()
```

```
## NULL
```

```
h_gap$country %>%
  fct_relevel("Romania", "Haiti") %>%
  levels()
```

```
## [1] "Romania"   "Haiti"     "Egypt"     "Thailand"  "Venezuela"
```

Record levels

```
i_gap <- gapminder %>%
  filter(country %in% c("United States", "Sweden", "Australia")) %>%
  droplevels()
i_gap$country %>% levels()
```

```
## [1] "Australia"     "Sweden"        "United States"
```

```r
i_gap$country %>%
  fct_recode("USA" = "United States", "Oz" = "Australia") %>%
  levels()
```

```
## [1] "Oz"     "Sweden" "USA"
```

Date and time

```r
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```r
today()
```

```
## [1] "2020-01-13"
```

```r
now()  # in UTC (Coordinated Universal Time)
```

```
## [1] "2020-01-13 22:57:37 PST"
```

```r
ymd("2017-01-31")
```

```
## [1] "2017-01-31"
```

```r
mdy("January 31st, 2017")
```

```
## [1] "2017-01-31"
```

```r
dmy("31-Jan-2017")
```

```
## [1] "2017-01-31"
```

```r
ymd_hms("2017-01-31 20:11:59")
```

```
## [1] "2017-01-31 20:11:59 UTC"
```

```r
mdy_hm("01/31/2017 08:01")
```

```
## [1] "2017-01-31 08:01:00 UTC"
```

```
mdy_hm("01/31/2017 08:01", tz = "America/New_York")
```

```
## [1] "2017-01-31 08:01:00 EST"
```

```
# all the time zone names
OlsonNames
```

```
## function (tzdir = NULL)
## {
##     if (is.null(tzdir)) {
##         if (.Platform$OS.type == "windows")
##             tzdir <- Sys.getenv("TZDIR", file.path(R.home("share"),
##                 "zoneinfo"))
##         else {
##             tzdirs <- c(Sys.getenv("TZDIR"), file.path(R.home("share"),
##                 "zoneinfo"), "/usr/share/zoneinfo", "/share/zoneinfo",
##                 "/usr/share/lib/zoneinfo", "/usr/lib/zoneinfo",
##                 "/usr/local/etc/zoneinfo", "/etc/zoneinfo", "/usr/etc/zoneinfo")
##             tzdirs <- tzdirs[file.exists(tzdirs)]
##             if (!length(tzdirs)) {
##                 warning("no Olson database found")
##                 return(character())
##             }
##             else tzdir <- tzdirs[1L]
##         }
##     }
##     else if (!dir.exists(tzdir))
##         stop(sprintf("%s is not a directory", sQuote(tzdir)),
##             domain = NA)
##     x <- list.files(tzdir, recursive = TRUE)
##     ver <- if (file.exists(vf <- file.path(tzdir, "VERSION")))
##         readLines(vf, warn = FALSE)
##     else if (file.exists(vf <- file.path(tzdir, "+VERSION")))
##         readLines(vf, warn = FALSE)
##     x <- setdiff(x, "VERSION")
##     ans <- grep("^[ABCDEFGHIJKLMNOPQRSTUVWXYZ]", x, value = TRUE)
##     if (!is.null(ver))
##         attr(ans, "Version") <- ver
##     ans
## }
## <bytecode: 0x7f99ba9b89b8>
## <environment: namespace:base>
```

```
(t1 <- mdy_hm("01/31/2017 08:01", tz = "America/New_York"))
```

```
## [1] "2017-01-31 08:01:00 EST"
```

```
# convert timezone
with_tz(t1, tzone = "America/Los_Angeles")
```

```
## [1] "2017-01-31 05:01:00 PST"
```

```
# fix a timezone
force_tz(t1, tzone = "America/Los_Angeles")
```

```
## [1] "2017-01-31 08:01:00 PST"
```

From individual components

```
library(nycflights13)
flights %>%
  select(year, month, day, hour, minute)
```

```
## # A tibble: 336,776 x 5
##     year month   day  hour minute
##    <int> <int> <int> <dbl>  <dbl>
##  1  2013     1     1     5     15
##  2  2013     1     1     5     29
##  3  2013     1     1     5     40
##  4  2013     1     1     5     45
##  5  2013     1     1     6      0
##  6  2013     1     1     5     58
##  7  2013     1     1     6      0
##  8  2013     1     1     6      0
##  9  2013     1     1     6      0
## 10  2013     1     1     6      0
## # ... with 336,766 more rows
```

```
(flights_dt <- flights %>%
  select(year, month, day, hour, minute) %>%
  mutate(
    date = make_date(year, month, day),
    time = make_datetime(year, month, day, hour, minute,)))
```

```
## # A tibble: 336,776 x 7
##     year month   day  hour minute date       time
##    <int> <int> <int> <dbl>  <dbl> <date>     <dttm>
##  1  2013     1     1     5     15 2013-01-01 2013-01-01 05:15:00
##  2  2013     1     1     5     29 2013-01-01 2013-01-01 05:29:00
##  3  2013     1     1     5     40 2013-01-01 2013-01-01 05:40:00
##  4  2013     1     1     5     45 2013-01-01 2013-01-01 05:45:00
##  5  2013     1     1     6      0 2013-01-01 2013-01-01 06:00:00
##  6  2013     1     1     5     58 2013-01-01 2013-01-01 05:58:00
##  7  2013     1     1     6      0 2013-01-01 2013-01-01 06:00:00
##  8  2013     1     1     6      0 2013-01-01 2013-01-01 06:00:00
##  9  2013     1     1     6      0 2013-01-01 2013-01-01 06:00:00
## 10  2013     1     1     6      0 2013-01-01 2013-01-01 06:00:00
## # ... with 336,766 more rows
```

Get components

```r
datetime <- ymd_hms("2016-07-08 12:34:56")

year(datetime)
```

```
## [1] 2016
```

```r
month(datetime)
```

```
## [1] 7
```

```r
month(datetime, label = TRUE)
```

```
## [1] Jul
## 12 Levels: Jan < Feb < Mar < Apr < May < Jun < Jul < Aug < Sep < ... < Dec
```

```r
mday(datetime)
```

```
## [1] 8
```

```r
yday(datetime)
```

```
## [1] 190
```

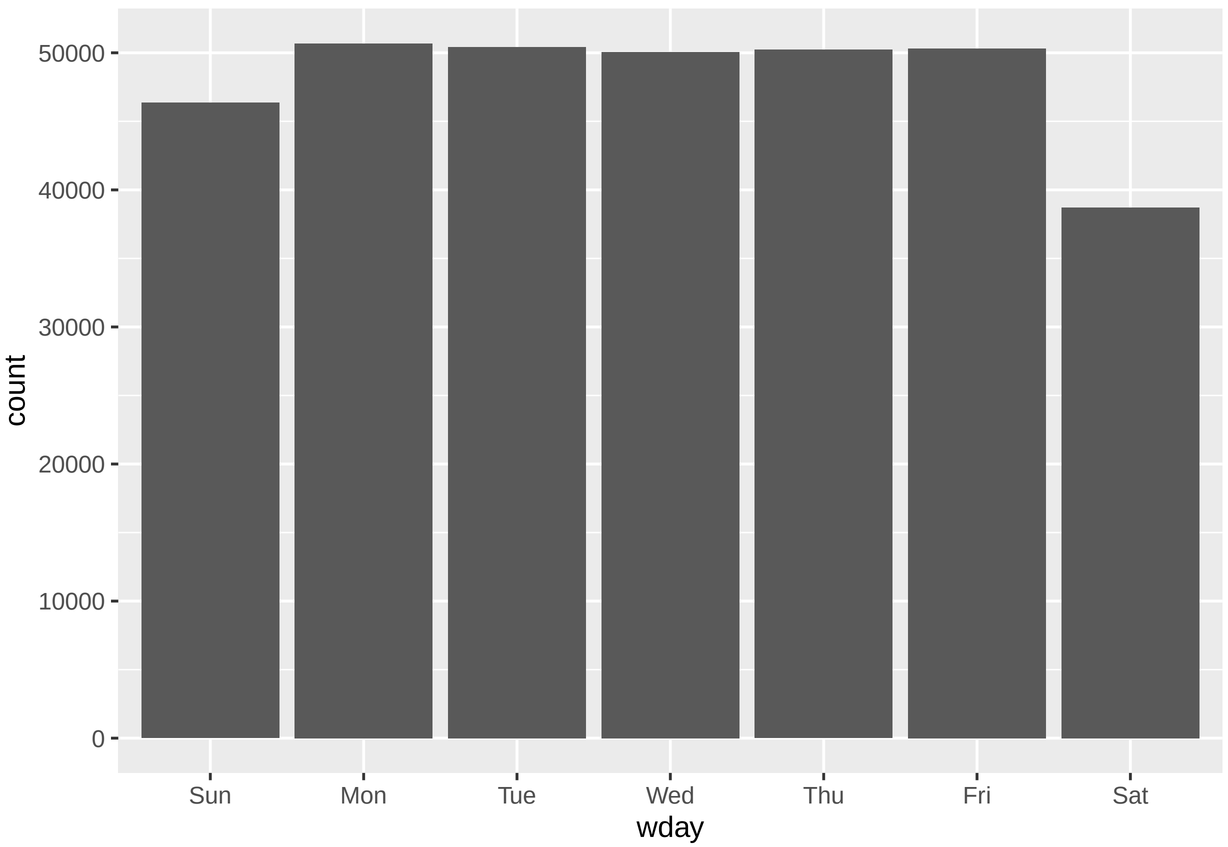
```r
wday(datetime)
```

```
## [1] 6
```

```r
wday(datetime, label = TRUE, abbr = FALSE)
```

```
## [1] Friday
## 7 Levels: Sunday < Monday < Tuesday < Wednesday < Thursday < ... < Saturday
```

```r
flights_dt %>%
  mutate(wday = wday(time, label = TRUE)) %>%
  ggplot(aes(x = wday)) +
    geom_bar()
```



References

https://r4ds.had.co.nz https://lubridate.tidyverse.org/ https://forcats.tidyverse.org/