

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`.

```
library(tidyverse)
```

```
# to illustrate the issue of `read.csv`, let's write a csv file out of the flights dataset
library(nycflights13)
write_csv(flights %>% sample_n(100), "flights.csv")
```

```
# base R function, character variables are automatically converted to factors
read.csv("flights.csv")
```

```
# readr function, character variables are imported as is
read_csv("flights.csv")
# there are several workarounds,
# 1. we could use `mutate_if` to force the conversion
read_csv("flights.csv") %>%
  mutate_if(is.character, as_factor)
read_csv("flights.csv") %>%
  mutate_if(~ is.character(.) && n_distinct(.) < 20, as_factor)

# 2. we could specify the column types
read_csv("flights.csv", col_types = "iidddddffccffdddT")

# 3. use the rstudio import interface
```

Factor inspection

```
class(gss_cat$partyid)
```

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

```
levels(gss_cat$partyid)
```

```
## [1] "No answer"          "Don't know"         "Other party"
## [4] "Strong republican"  "Not str republican"  "Ind,near rep"
## [7] "Independent"        "Ind,near dem"        "Not str democrat"
## [10] "Strong democrat"
```

```
nlevels(gss_cat$partyid)
```

```
## [1] 10
```

```
gss_cat %>% count(partyid)
```

```
## # A tibble: 10 x 2
##   partyid      n
##   <fct>      <int>
## 1 No answer    154
## 2 Don't know     1
## 3 Other party   393
## 4 Strong republican 2314
## 5 Not str republican 3032
## 6 Ind,near rep   1791
## 7 Independent   4119
## 8 Ind,near dem   2499
## 9 Not str democrat 3690
## 10 Strong democrat 3490
```

```
fct_count(gss_cat$partyid, sort = TRUE)
```

```
## # A tibble: 10 x 2
##   f      n
##   <fct>  <int>
## 1 Independent   4119
## 2 Not str democrat 3690
## 3 Strong democrat 3490
## 4 Not str republican 3032
## 5 Ind,near dem   2499
## 6 Strong republican 2314
## 7 Ind,near rep   1791
## 8 Other party    393
## 9 No answer     154
## 10 Don't know      1
```

Dropping unused levels

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

```
gss_cat2 <- gss_cat %>%
  filter(partyid %in% c("Independent", "Strong democrat", "Strong republican"))
nlevels(gss_cat2$partyid)
```

```
## [1] 10
```

```
# drop a specific factor
gss_cat2$partyid <- gss_cat2$partyid %>% fct_drop()
levels(gss_cat2$partyid)
```

```
## [1] "Strong republican" "Independent" "Strong democrat"
```

```
# drop all the factors in a data frame
gss_cat2 <- gss_cat2 %>% droplevels()
```

Change order of the levels

```
gss_cat$partyid %>%
  levels()
```

```
## [1] "No answer" "Don't know" "Other party"
## [4] "Strong republican" "Not str republican" "Ind,near rep"
## [7] "Independent" "Ind,near dem" "Not str democrat"
## [10] "Strong democrat"
```

```
## order by frequency
gss_cat %>% mutate(partyid = partyid %>% fct_infreq())
```

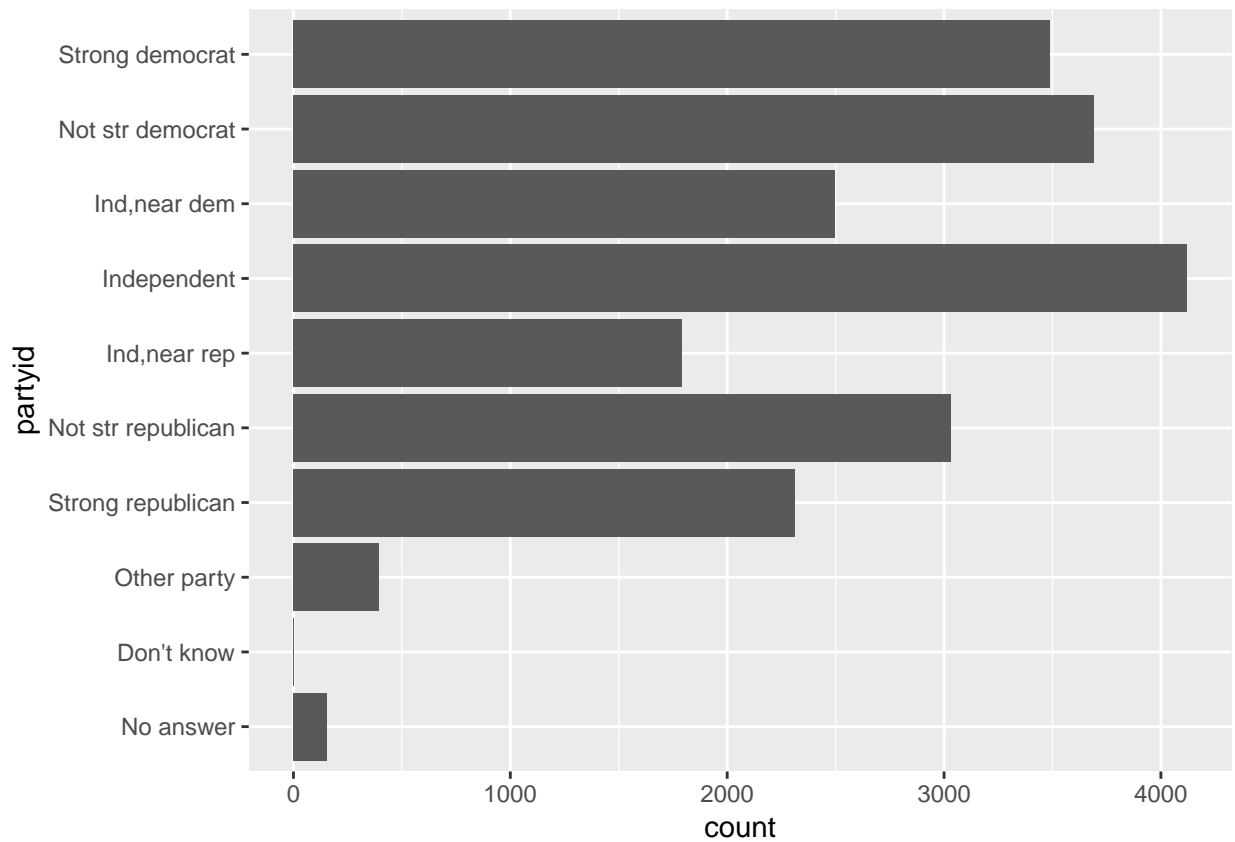
```
## # A tibble: 21,483 x 9
##   year marital age race rincome partyid relig denom tvhours
##   <int> <fct> <int> <fct> <fct> <fct> <fct> <fct> <int>
## 1 2000 Never ma~ 26 White $8000 to ~ Ind,near r~ Protesta~ Souther~ 12
## 2 2000 Divorced 48 White $8000 to ~ Not str re~ Protesta~ Baptist~ NA
## 3 2000 Widowed 67 White Not appli~ Independent Protesta~ No deno~ 2
## 4 2000 Never ma~ 39 White Not appli~ Ind,near r~ Orthodox~ Not app~ 4
## 5 2000 Divorced 25 White Not appli~ Not str de~ None Not app~ 1
## 6 2000 Married 25 White $20000 - ~ Strong dem~ Protesta~ Souther~ NA
## 7 2000 Never ma~ 36 White $25000 or~ Not str re~ Christian Not app~ 3
## 8 2000 Divorced 44 White $7000 to ~ Ind,near d~ Protesta~ Luthera~ NA
## 9 2000 Married 44 White $25000 or~ Not str de~ Protesta~ Other 0
## 10 2000 Married 47 White $25000 or~ Strong rep~ Protesta~ Souther~ 3
## # ... with 21,473 more rows
```

```
## backwards!
gss_cat %>% mutate(partyid = partyid %>% fct_infreq() %>% fct_rev())
```

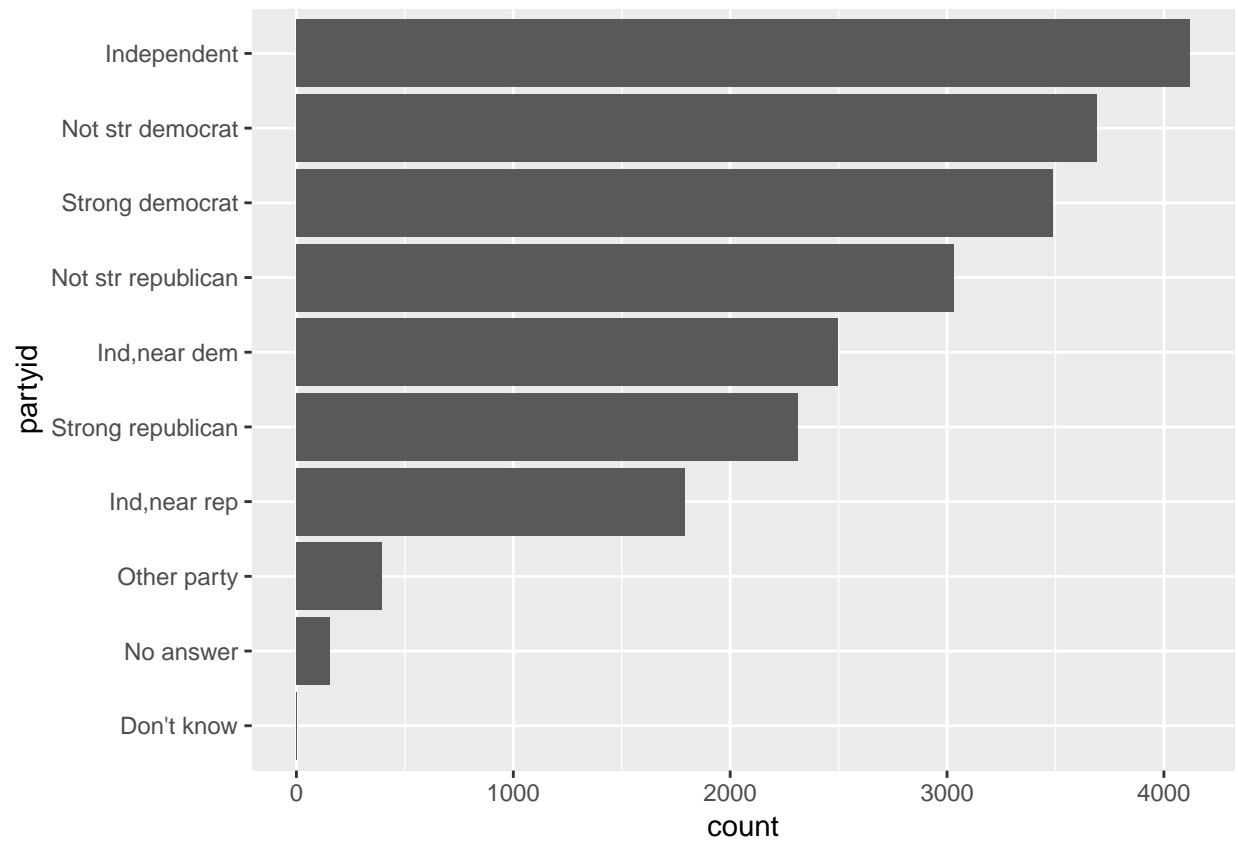
```
## # A tibble: 21,483 x 9
##   year marital age race rincome partyid relig denom tvhours
##   <int> <fct> <int> <fct> <fct> <fct> <fct> <fct> <int>
## 1 2000 Never ma~ 26 White $8000 to ~ Ind,near r~ Protesta~ Souther~ 12
## 2 2000 Divorced 48 White $8000 to ~ Not str re~ Protesta~ Baptist~ NA
## 3 2000 Widowed 67 White Not appli~ Independent Protesta~ No deno~ 2
## 4 2000 Never ma~ 39 White Not appli~ Ind,near r~ Orthodox~ Not app~ 4
## 5 2000 Divorced 25 White Not appli~ Not str de~ None Not app~ 1
## 6 2000 Married 25 White $20000 - ~ Strong dem~ Protesta~ Souther~ NA
## 7 2000 Never ma~ 36 White $25000 or~ Not str re~ Christian Not app~ 3
## 8 2000 Divorced 44 White $7000 to ~ Ind,near d~ Protesta~ Luthera~ NA
## 9 2000 Married 44 White $25000 or~ Not str de~ Protesta~ Other 0
## 10 2000 Married 47 White $25000 or~ Strong rep~ Protesta~ Souther~ 3
## # ... with 21,473 more rows
```

Why?

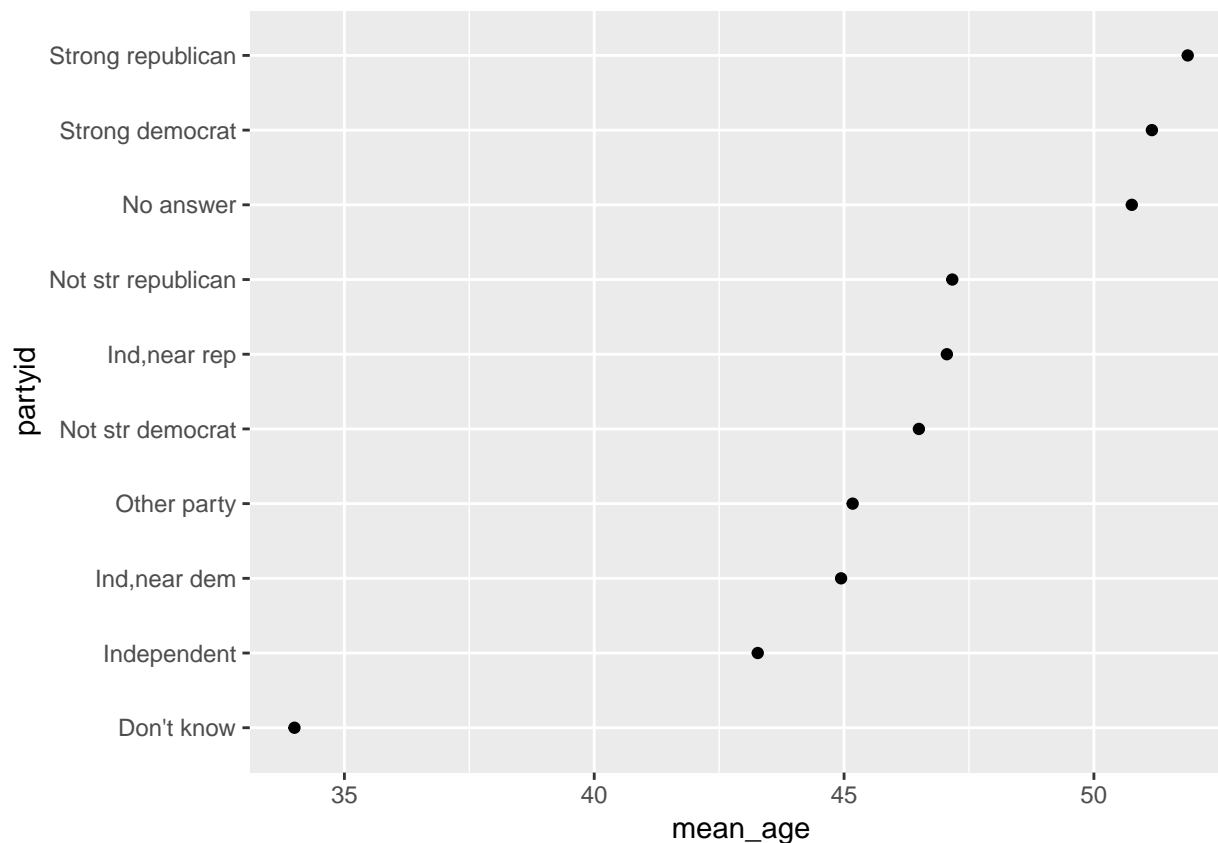
```
ggplot(gss_cat) + geom_bar(aes(partyid)) + coord_flip()
```



```
ggplot(gss_cat) + geom_bar(aes(partyid %>% fct_infreq() %>% fct_rev())) +  
  xlab("partyid") + coord_flip()
```



```
# reorder factor according to values of another variable
gss_cat %>%
  group_by(partyid) %>%
  summarize(mean_age = mean(age, na.rm = TRUE)) %>%
  ggplot(aes(x = mean_age, y = fct_reorder(partyid, mean_age))) +
  geom_point() + ylab("partyid")
```



Change to any order

```
gss_cat$partyid %>% levels()
```

```
## [1] "No answer"      "Don't know"      "Other party"
## [4] "Strong republican" "Not str republican" "Ind,near rep"
## [7] "Independent"     "Ind,near dem"     "Not str democrat"
## [10] "Strong democrat"
```

```
gss_cat$partyid %>%
  fct_relevel("Strong republican", "Strong democrat") %>%
  levels()
```

```
## [1] "Strong republican" "Strong democrat"   "No answer"
## [4] "Don't know"        "Other party"       "Not str republican"
## [7] "Ind,near rep"      "Independent"       "Ind,near dem"
## [10] "Not str democrat"
```

```
# use mutate verb to modify the data frame
gss_cat %>% mutate(partyid = partyid %>% fct_relevel("Strong republican", "Strong democrat"))
```

```
## # A tibble: 21,483 x 9
```

```
##   year marital    age race rincome partyid relig denom tvhours
##   <int> <fct>    <int> <fct> <fct>    <fct>    <fct>    <fct>    <int>
## 1  2000 Never ma~  26 White $8000 to ~ Ind,near r~ Protesta~ Souther~    12
## 2  2000 Divorced  48 White $8000 to ~ Not str re~ Protesta~ Baptist~    NA
## 3  2000 Widowed   67 White Not appli~ Independent Protesta~ No deno~     2
## 4  2000 Never ma~  39 White Not appli~ Ind,near r~ Orthodox~ Not app~     4
## 5  2000 Divorced  25 White Not appli~ Not str de~ None      Not app~     1
## 6  2000 Married   25 White $20000 - ~ Strong dem~ Protesta~ Souther~    NA
## 7  2000 Never ma~  36 White $25000 or~ Not str re~ Christian Not app~     3
## 8  2000 Divorced  44 White $7000 to ~ Ind,near d~ Protesta~ Luthera~    NA
## 9  2000 Married   44 White $25000 or~ Not str de~ Protesta~ Other      0
## 10 2000 Married   47 White $25000 or~ Strong rep~ Protesta~ Souther~     3
## # ... with 21,473 more rows
```

Recode levelsx

```
gss_cat$partyid %>% levels()
```

```
## [1] "No answer"          "Don't know"          "Other party"
## [4] "Strong republican"  "Not str republican"  "Ind,near rep"
## [7] "Independent"        "Ind,near dem"        "Not str democrat"
## [10] "Strong democrat"
```

```
gss_cat$partyid %>%
  fct_recode(
    "Independent,near rep" = "Ind,near rep",
    "Independent,near dem" = "Ind,near dem"
  ) %>%
  levels()
```

```
## [1] "No answer"          "Don't know"          "Other party"
## [4] "Strong republican"  "Not str republican"  "Independent,near rep"
## [7] "Independent"        "Independent,near dem" "Not str democrat"
## [10] "Strong democrat"
```

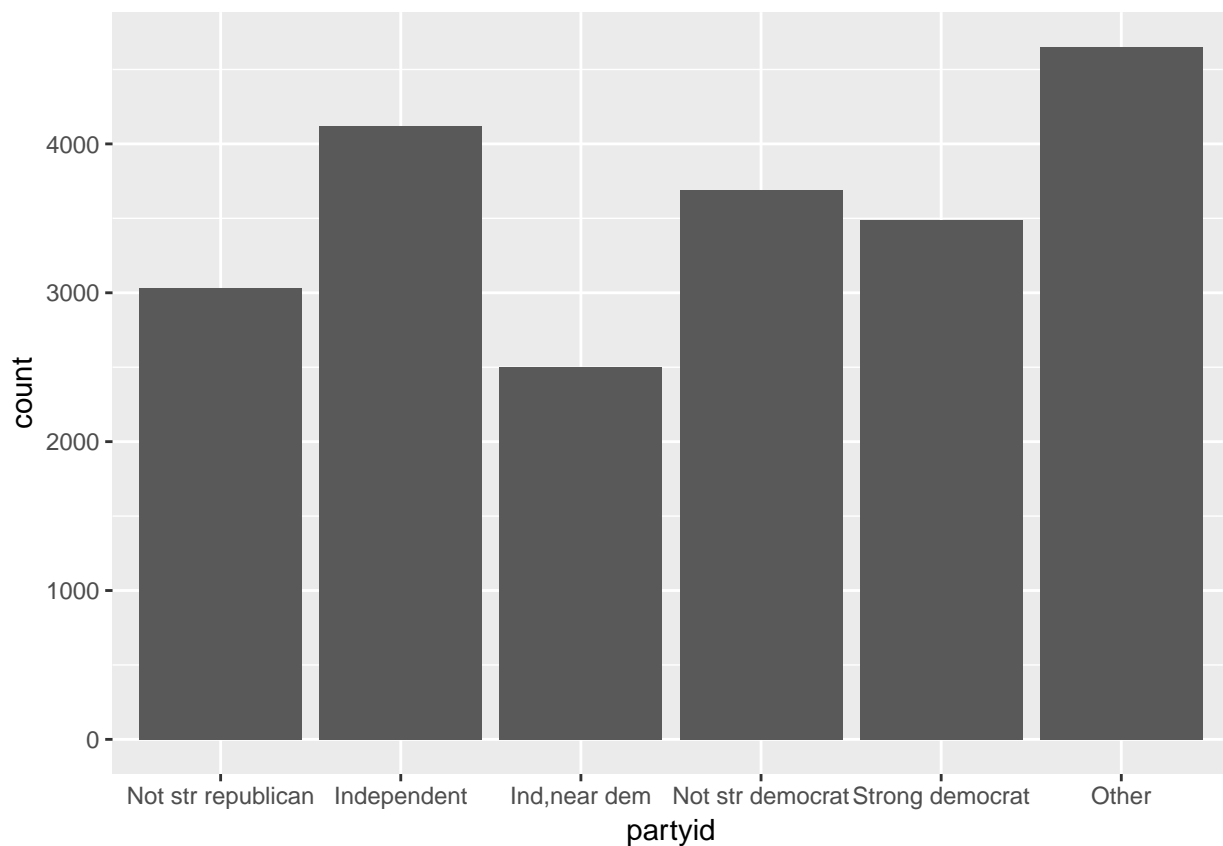
```
# if we need to modify the data frame, then
gss_cat %>% mutate(partyid = partyid %>%
  fct_recode(
    "Independent,near rep" = "Ind,near rep",
    "Independent,near dem" = "Ind,near dem"
  ))
```

```
## # A tibble: 21,483 x 9
##   year marital    age race rincome partyid relig denom tvhours
##   <int> <fct>    <int> <fct> <fct>    <fct>    <fct>    <fct>    <int>
## 1  2000 Never ma~  26 White $8000 to ~ Independen~ Protesta~ Souther~    12
## 2  2000 Divorced  48 White $8000 to ~ Not str re~ Protesta~ Baptist~    NA
## 3  2000 Widowed   67 White Not appli~ Independent Protesta~ No deno~     2
## 4  2000 Never ma~  39 White Not appli~ Independen~ Orthodox~ Not app~     4
## 5  2000 Divorced  25 White Not appli~ Not str de~ None      Not app~     1
```

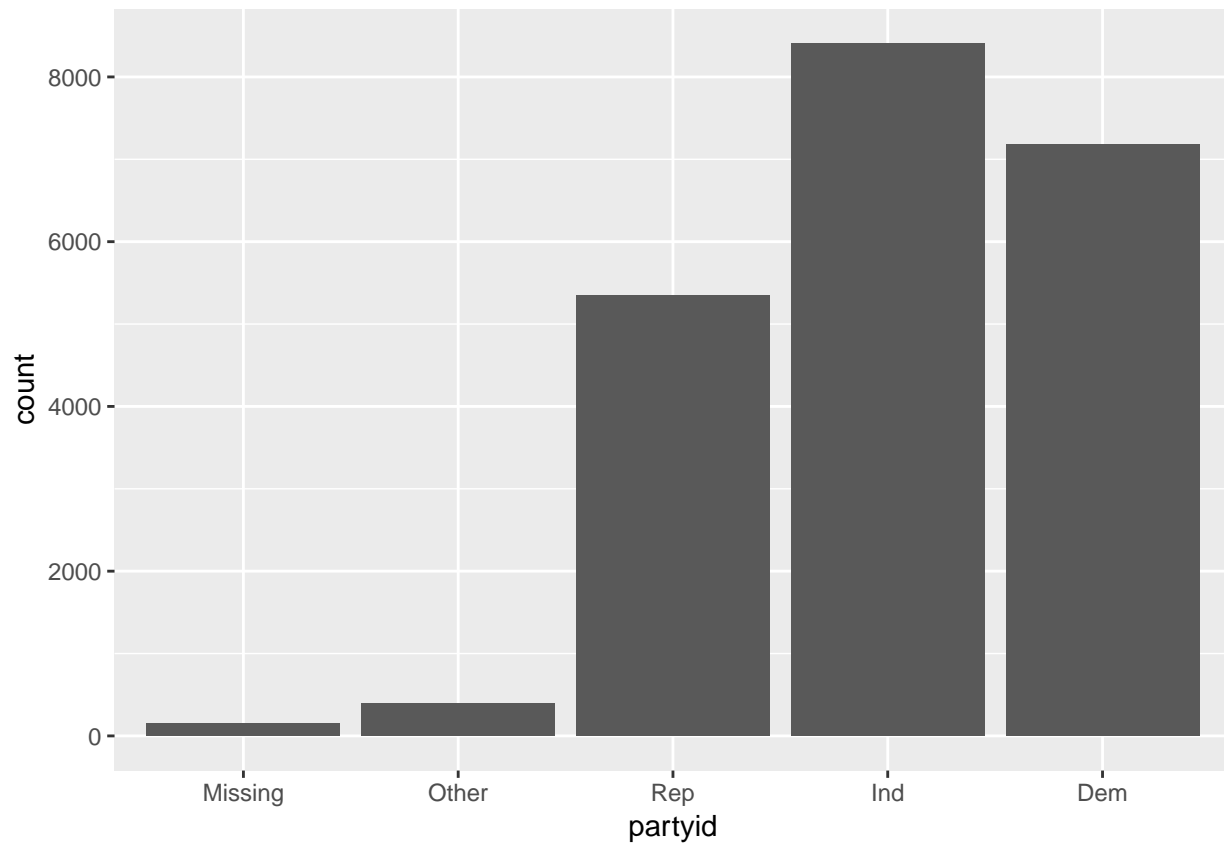
```
## 6 2000 Married      25 White $20000 - ~ Strong dem~ Protest~ Souther~ NA
## 7 2000 Never ma~   36 White $25000 or~ Not str re~ Christian Not app~ 3
## 8 2000 Divorced    44 White $7000 to ~ Independen~ Protest~ Luthera~ NA
## 9 2000 Married     44 White $25000 or~ Not str de~ Protest~ Other 0
## 10 2000 Married    47 White $25000 or~ Strong rep~ Protest~ Souther~ 3
## # ... with 21,473 more rows
```

Collapse levels

```
# collapse automatically
gss_cat %>%
  mutate(partyid = partyid %>% fct_lump(5)) %>%
  ggplot() + geom_bar(aes(partyid))
```



```
# collapse manually
gss_cat %>%
  mutate(partyid = partyid %>% fct_collapse(
    Missing = c("No answer", "Don't know"),
    Rep = c("Strong republican", "Not str republican"),
    Ind = c("Ind,near rep", "Independent", "Ind,near dem"),
    Dem = c("Not str democrat", "Strong democrat"),
    Other = c("Other party")
  )) %>%
  ggplot() + geom_bar(aes(partyid))
```

Remark: there is a bug in forcats v0.4.0 such that the argument `group_other` in `fct_collapse` is malfunction.

Date and time

```
library(lubridate)
```

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

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

```
today()
```

```
## [1] "2020-01-15"
```

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

```
## [1] "2020-01-15 23:20:13 PST"
```

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

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

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

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

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

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

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

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

```
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: 0x7fdfba99f5b0>
## <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, tzzone = "America/Los_Angeles")
```

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

```
# fix a timezone
force_tz(t1, tzzone = "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>    <dtm>
## 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
```

Remark: something was wrong above!

Get components

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

```
year(datetime)
```

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

```
month(datetime)
```

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

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

```
## [1] Jul
```

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

```
mday(datetime)
```

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

```
yday(datetime)
```

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

```
wday(datetime)
```

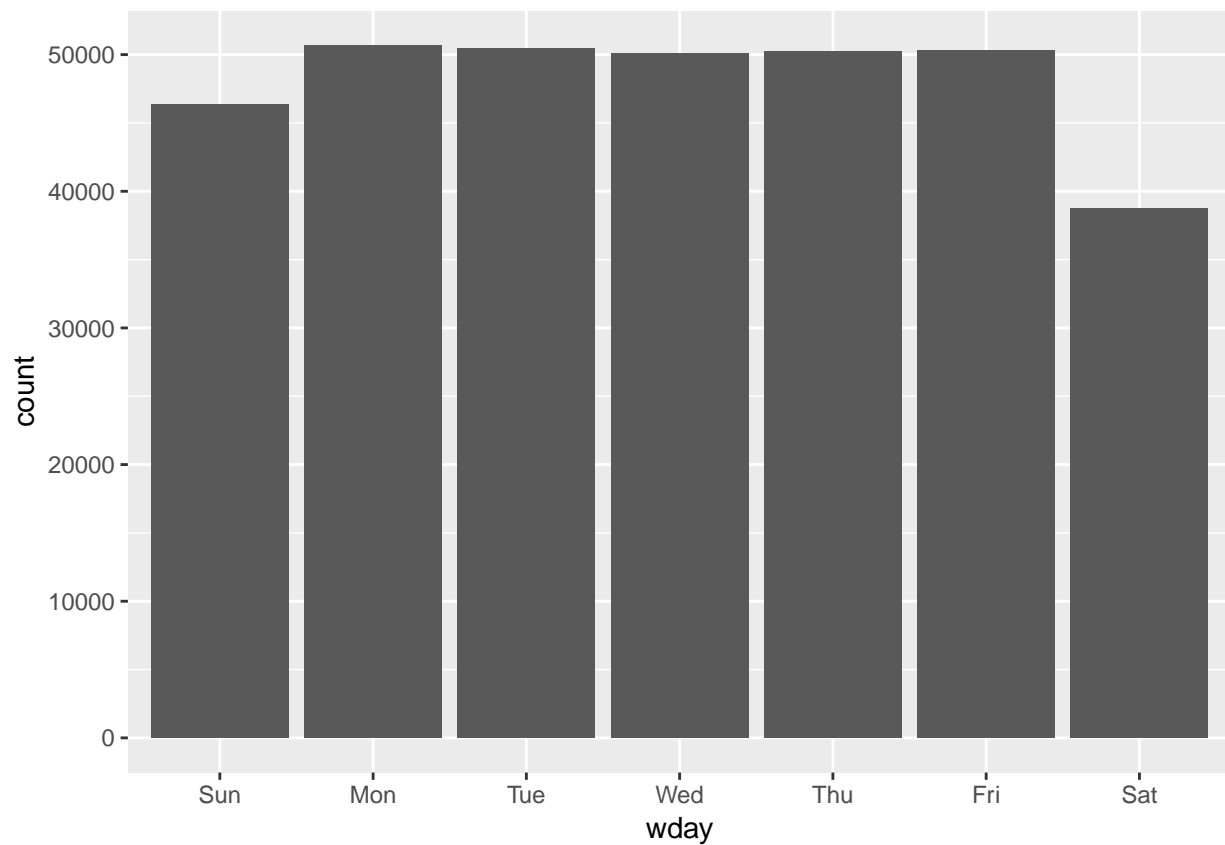
```
## [1] 6
```

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

```
## [1] Friday
```

```
## 7 Levels: Sunday < Monday < Tuesday < Wednesday < Thursday < ... < Saturday
```

```
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/>