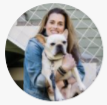
Dates

The correct data format: YYYY-MM-DD

Tweet



Mara Averick @dataandme · Oct 5

In lieu of a eulogy, just read @kwbroman and @kara_woo's "Data Organization in Spreadsheets" aloud at mine: doi.org/10.1080/000313... (I'm sure everyone will choke up at the mere thought of using font-colour or highlighting as encoding for data 😭)

> **HaoOooooooooOOooo** @Hao_and_Y · Oct 5
>
> When I die, please inscribe "observations in rows, variables in columns" onto the rocket that will carry my ashes into the sun. Thx twitter.com/standupmaths/s...
>
> Show this thread

4 11 76

Joyce Robbins @jtrnyc · Oct 5

Please write dates on my tombstone in YYYY-MM-DD format.

2 6 96

Carina M. McCormick, Ph.D.
@CMMc_PhD

Replying to @jtrnyc @dataandme and 2 others

I shared this idea with my husband, who clearly expressed that this is actually what he wants me to do in the event of his untimely death. Who knew you would make such a lasting impact with this quick comment?

3:35 PM · Oct 8, 2020 · Twitter Web App

Converting to `Date` class

You can convert character data to `Date` class with `as.Date()`:

```
dchar <- "2018-10-12"
ddate <- as.Date(dchar)
```

Class

```
dchar
```

```
## [1] "2018-10-12"
```

```
ddate
```

```
## [1] "2018-10-12"
```

```
class(dchar)
```

```
## [1] "character"
```

```
class(ddate)
```

```
## [1] "Date"
```

Specifying the format

```
as.Date("Thursday, January 6, 2005", format = "%A, %B %d, %Y")
```

```
## [1] "2005-01-06"
```

For a list of the conversion specifications available in R, see `?strptime`.

parse_date

similar to `as.Date()`, more strict but some options aren't implemented

```
as.Date("1/12/2019", format="%m/%d/%y")
```

```
## [1] "2020-01-12"
```

```
readr::parse_date("1/12/2019", format="%m/%d/%y")
```

```
## [1] NA
```

```
as.Date("Thursday, January 6, 2005", format = "%A, %B %d, %Y")
```

```
## [1] "2005-01-06"
```

```
readr::parse_date("Thursday, January 6, 2005", format = "%A, %B %d, %Y")
```

```
## Error: Invalid %A auto parser
```

See `?parse_date`

Challenge

Convert **Date: Mon, Mar-15-2021** to Date class using `as.Date()`

First person to complete in the chat window gets a virtual dessert treat

lubridate package

The tidyverse **lubridate** makes it easy to convert dates that are not in standard format with `ymd()`, `ydm()`, `mdy()`, `myd()`, `dmy()`, and `dym()` (among many other useful date-time functions):

```
lubridate::mdy("April 13, 1907")
```

```
## [1] "1907-04-13"
```

Working with `Date` Class

It is well worth the effort to convert to `Date` class, because there's a lot you can do with dates in a `Date` class that you can't do if you store the dates as character data.

Number of days between dates:

```
as.Date("2017-11-02") - as.Date("2017-01-01")
```

```
## Time difference of 305 days
```

Compare dates

```
as.Date("2017-11-12") > as.Date("2017-3-3")
```

```
## [1] TRUE
```

Note that `Sys.Date()` returns today's date as a `Date` class:

```
Sys.Date()
```

```
## [1] "2021-10-25"
```

```
class(Sys.Date())
```

```
## [1] "Date"
```

base R functions

```
today <- Sys.Date()
weekdays(today, abbreviate = TRUE)
```

```
## [1] "Mon"
```

```
months(today, abbreviate = TRUE)
```

```
## [1] "Oct"
```

```
quarters(today)
```

```
## [1] "Q4"
```

lubridate functions

The **lubridate** package provides additional functions to extract information from a date:

```
today <- Sys.Date()
lubridate::year(today)
```

```
## [1] 2021
```

```
lubridate::yday(today)
```

```
## [1] 298
```

```
lubridate::month(today, label = TRUE)
```

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

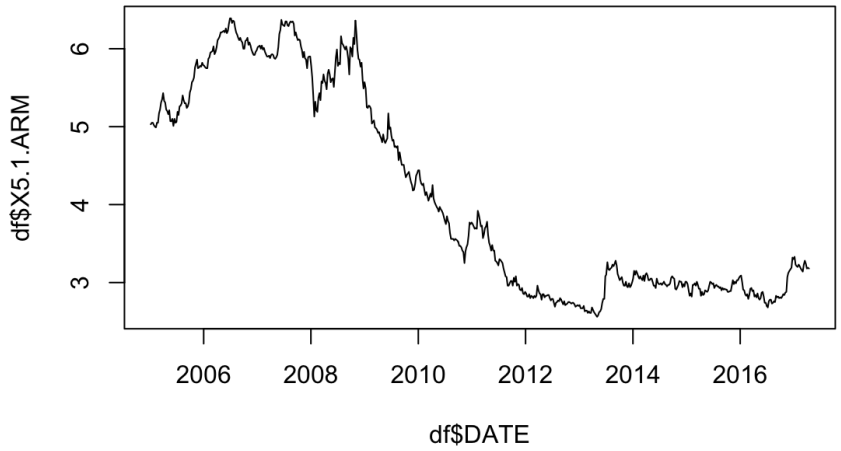
```
lubridate::week(today)
```

```
## [1] 43
```

Plotting with a `Date` class variable

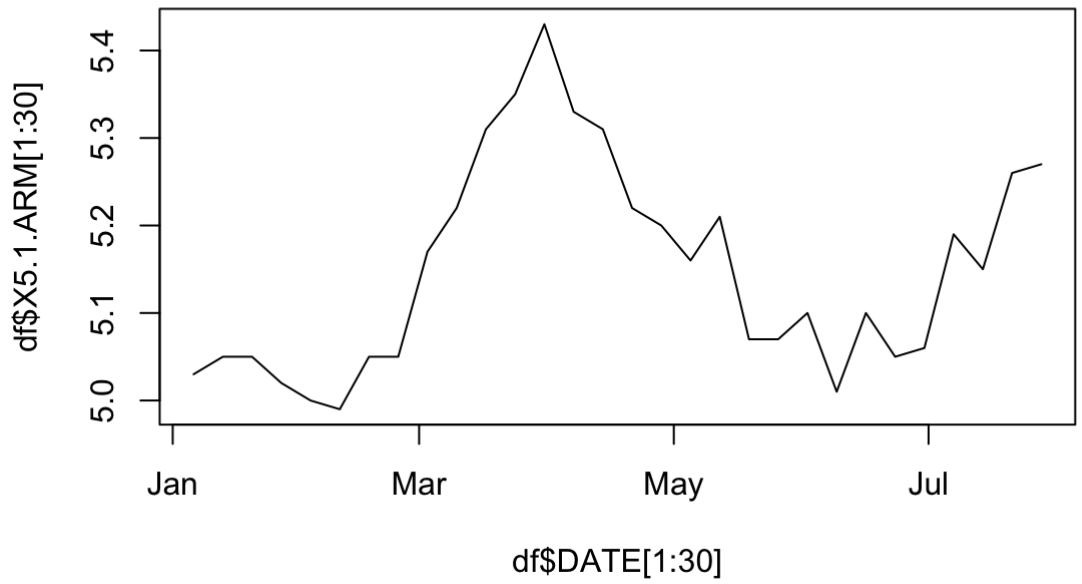
Both base R graphics and **ggplot2** "know" how to work with a `Date` class variable, and label the axes properly:

```
df <- read.csv("mortgage.csv")
df$DATE <- as.Date(df$DATE)
plot(df$DATE, df$X5.1.ARM, type = "l") # on the order of years
```



base R

```
plot(df$DATE[1:30], df$X5.1.ARM[1:30], type = "l") # switch to months
```



ggplot2

```
# readr
library(tidyverse)
```

Note that unlike base R `read.csv()`, `readr::read_csv()` automatically reads DATE in as a `Date` class since it's in YYYY-MM-DD format:

```
df <- readr::read_csv("mortgage.csv")
glimpse(df)
```

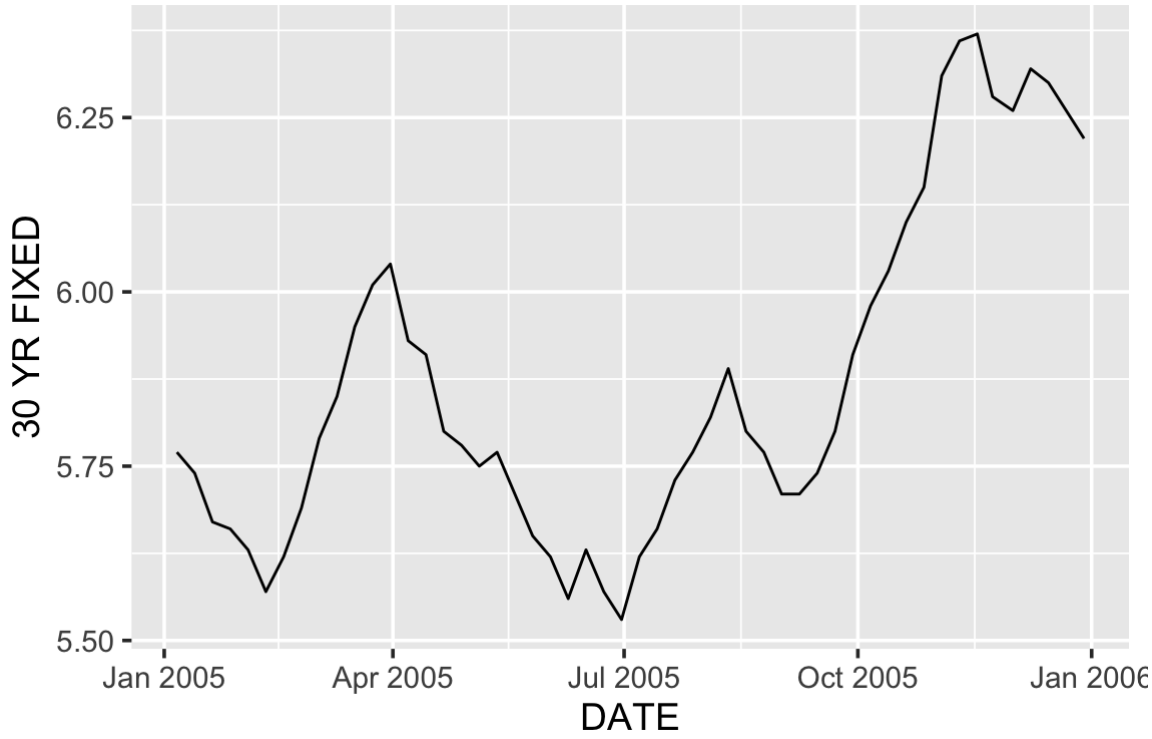
```
## Rows: 641
## Columns: 4
## $ DATE          <date> 2005-01-06, 2005-01-13, 2005-01-20, 2005-01-27, 2005-02…
## $ `5/1 ARM`     <dbl> 5.03, 5.05, 5.05, 5.02, 5.00, 4.99, 5.05, 5.05, 5.17, 5.…
## $ `15 YR FIXED` <dbl> 5.21, 5.19, 5.15, 5.14, 5.14, 5.10, 5.14, 5.22, 5.33, 5.…
## $ `30 YR FIXED` <dbl> 5.77, 5.74, 5.67, 5.66, 5.63, 5.57, 5.62, 5.69, 5.79, 5.…
```

ggplot2

```
g <- ggplot(df, aes(DATE, `30 YR FIXED`)) +
  geom_line() +
  theme_grey(14)
```

ggplot2

```
ggplot(df %>% filter(DATE < as.Date("2006-01-01")),
       aes(DATE, `30 YR FIXED`)) +
  geom_line() +
  theme_grey(14)
```

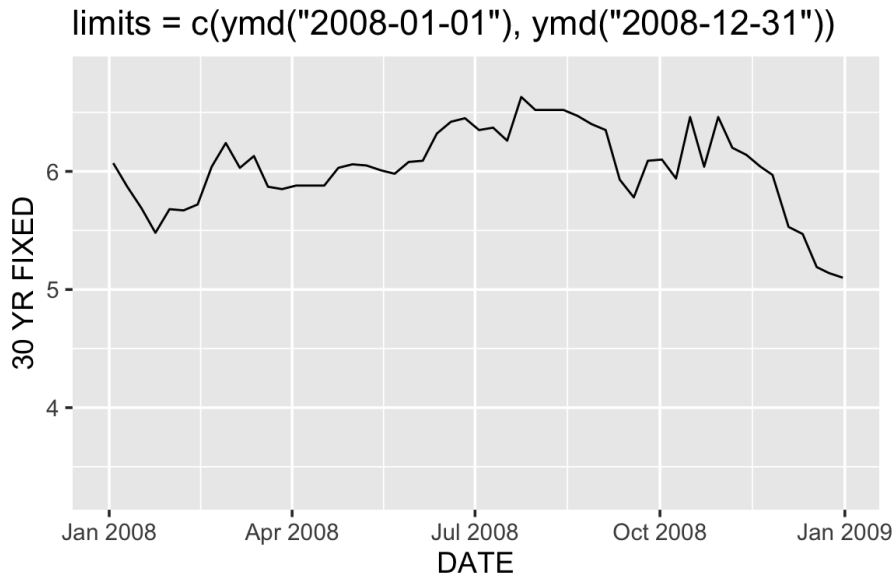


Again, when the data is filtered, the x-axis labels switch from years to months.

Limits

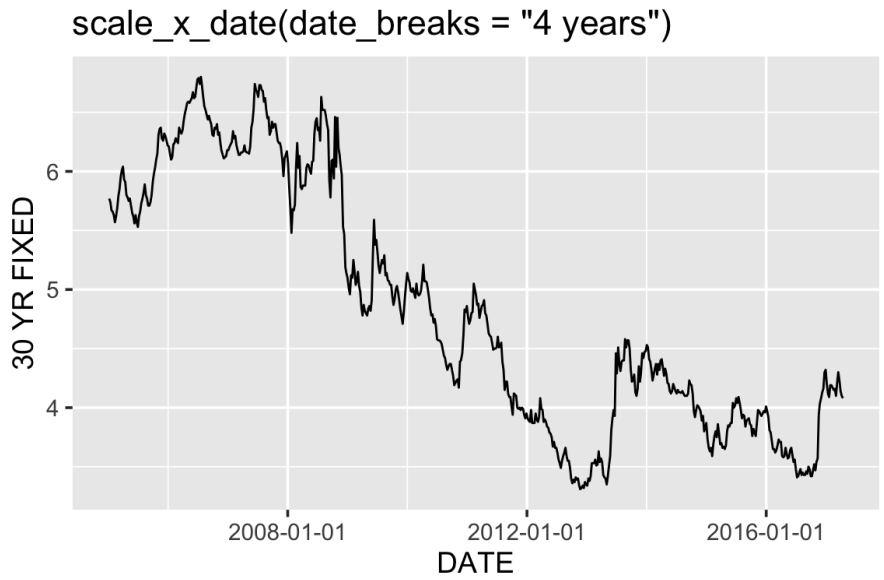
We can control the x-axis breaks, limits, and labels with `scale_x_date()`:

```
library(lubridate)
g + scale_x_date(limits = c(ymd("2008-01-01"), ymd("2008-12-31"))) +
  ggtitle("limits = c(ymd(\"2008-01-01\"), ymd(\"2008-12-31\"))")
```



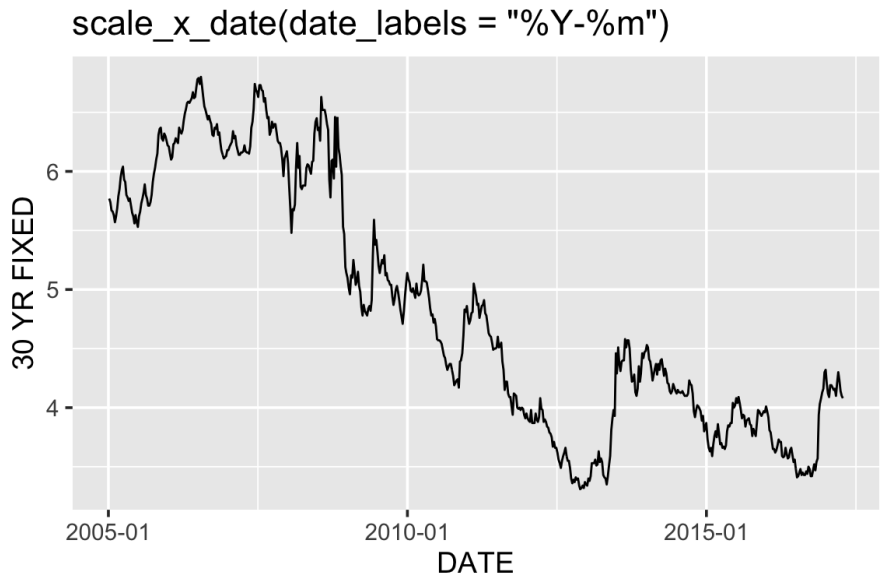
Breaks

```
g + scale_x_date(date_breaks = "4 years") +
  ggtitle("scale_x_date(date_breaks = \"4 years\")")
```



Labels

```
g + scale_x_date(date_labels = "%Y-%m") +
  ggtitle("scale_x_date(date_labels = \"%Y-%m\")")
```



(Remember `?parse_date` and `?strptime` for help.)

Annotations

We can use `geom_vline()` with `annotate()` to mark specific events in a time series:

```
ggplot(df, aes(DATE, `30 YR FIXED`)) +
  geom_line() +
  geom_vline(xintercept = ymd("2008-09-29"), color = "blue") +
  annotate("text", x = ymd("2008-09-29"), y = 3.75,
           label = " Market crash\n 9/29/08", color = "blue",
           hjust = 0) +
  scale_x_date(limits = c(ymd("2008-01-01"), ymd("2009-12-31")),
               date_breaks = "1 year",
               date_labels = "%Y") +
  theme_grey(16) +
  ggtitle("`geom_vline()` with `annotate()`")
```