Rent dataset - Sliding windows with slider

Tural Sadigov

2022-08-04

DISCLAIMER: This Quarto document is prepared using Julia Silge's YouTube video. There are edits here and there.

Libraries and data

```
library(tidyverse)
```

```
-- Attaching packages --------------------------------------- tidyverse 1.3.2 --
v ggplot2 3.3.6     v purrr   0.3.4
v tibble  3.1.8     v dplyr   1.0.9
v tidyr   1.2.0     v stringr 1.4.0
v readr   2.1.2     v forcats 0.5.1
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```
library(tidytuesdayR)
library(skimr)
tuesdata <- tidytuesdayR::tt_load('2022-07-05')
```

```
--- Compiling #TidyTuesday Information for 2022-07-05 ----
--- There are 3 files available ---
--- Starting Download ---


    Downloading file 1 of 3: `rent.csv`
    Downloading file 2 of 3: `sf_permits.csv`
    Downloading file 3 of 3: `new_construction.csv`
```

```
--- Download complete ---
```

```
tuesdata
```

```
Available datasets:
    rent
    sf_permits
    new_construction
```

Take the Rent data.

```
rent <- tuesdata$rent
rent
```

```
# A tibble: 200,796 x 17
   post_id         date  year nhood city  county price  beds baths  sqft room_~1
   <chr>          <dbl> <dbl> <chr> <chr> <chr>  <dbl> <dbl> <dbl> <dbl>   <dbl>
 1 pre2013_1341~ 2.01e7  2005 alam~ alam~ alame~  1250     2     2    NA       0
 2 pre2013_1356~ 2.01e7  2005 alam~ alam~ alame~  1295     2    NA    NA       0
 3 pre2013_1271~ 2.00e7  2004 alam~ alam~ alame~  1100     2    NA    NA       0
 4 pre2013_68671 2.01e7  2012 alam~ alam~ alame~  1425     1    NA   735       0
 5 pre2013_1275~ 2.00e7  2004 alam~ alam~ alame~   890     1    NA    NA       0
 6 pre2013_1523~ 2.01e7  2006 alam~ alam~ alame~   825     1    NA    NA       0
 7 pre2013_27543 2.01e7  2007 alam~ alam~ alame~  1500     1     1    NA       0
 8 6379096957    2.02e7  2017 alam~ alam~ alame~  2925     3    NA    NA       0
 9 pre2013_6254  2.01e7  2009 alam~ alam~ alame~   450    NA     1    NA       0
10 pre2013_1522~ 2.01e7  2006 alam~ alam~ alame~  1395     2    NA    NA       0
# ... with 200,786 more rows, 6 more variables: address <chr>, lat <dbl>,
#   lon <dbl>, title <chr>, descr <chr>, details <chr>, and abbreviated
#   variable name 1: room_in_apt
# i Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

Skim it.

```
skimr::skim(rent)
```

Table 1: Data summary

Name	rent
Number of rows	200796
Number of columns	17
_____	
Column type frequency:	
character	8
numeric	9
_____	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
post_id	0	1.00	9	14	0	200796	0
nhood	0	1.00	4	43	0	167	0
city	0	1.00	5	19	0	104	0
county	1394	0.99	4	13	0	10	0
address	196888	0.02	1	38	0	2869	0
title	2517	0.99	2	298	0	184961	0
descr	197542	0.02	13	16975	0	3025	0
details	192780	0.04	4	595	0	7667	0

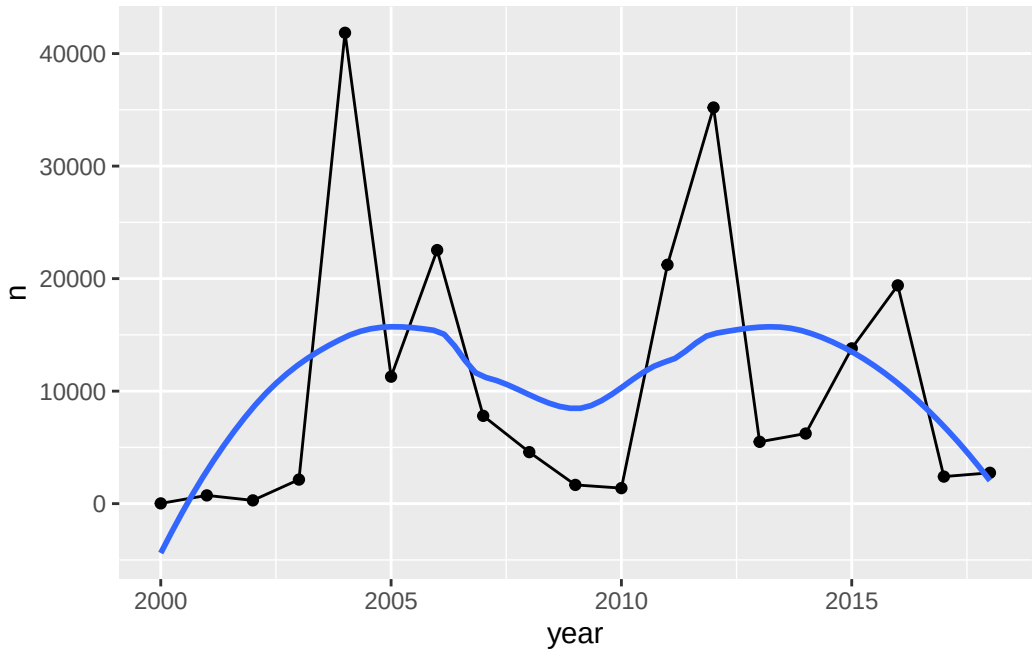
Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
date	0	1.00	20095718436694.07	70000902200502272011092420120805.00					20180717.00	
year	0	1.00	2009.51	4.48	2000.00	2005.00	2011.00	2012.0	2018.00	
price	0	1.00	2135.36	1427.75	220.00	1295.00	1800.00	2505.0	40000.00	
beds	6608	0.97	1.89	1.08	0.00	1.00	2.00	3.0	12.00	
baths	158121	0.21	1.68	0.69	1.00	1.00	2.00	2.0	8.00	
sqft	136117	0.32	1201.83	5000.22	80.00	750.00	1000.00	1360.0	900000.00	
room_in_apt	0	1.00	0.00	0.04	0.00	0.00	0.00	0.0	1.00	
lat	193145	0.04	37.67	0.35	33.57	37.40	37.76	37.8	40.43	
lon	196484	0.02	-122.21	0.78	-123.20	-122.42	-122.26	-122.0	-74.20	

Count and plot the number of houses for rent throughout the years.

```
rent %>%
  count(year) %>%
  ggplot(aes(year, n)) +
  geom_point() +
  geom_line() +
  geom_smooth(se = F)
```

```
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Filter out the apartments that only rented out rooms but not the whole house, and look at only years above 2005. Choose few columns only and change the format of the date.

```
rent_df <-
  rent %>%
  filter(room_in_apt <1, year>2005) %>%
  select(beds, date, price) %>%
  mutate(date = lubridate::ymd(date)) %>%
  arrange(date)
skimr::skim(rent_df)
```

Table 4: Data summary

Name	rent_df
Number of rows	144261
Number of columns	3
_____	
Column type frequency:	
Date	1
numeric	2
_____	
Group variables	None

Variable type: Date

skim_variable	n_missing	complete_rate	min	max	median	n_unique
date	0	1	2006-01-01	2018-07-17	2012-02-13	604

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
beds	4756	0.97	1.94	1.10	0	1	2	3	12	
price	0	1.00	2335.98	1539.65	220	1435	1995	2785	40000	

Not-sliding mean

Look at each month of data, and obtain mean price. Note that this is still not really a sliding window, maybe since windows do not overlap ('jumping' window, maybe?!).

```
library(slider)
slide_period_dbl(.x = rent_df,
                 .i = rent_df$date,
                 .period = 'month',
                 .f = ~mean(.x$price))
```

```
 [1] 1702.348 1764.308 1680.277 1780.290 1741.493 1792.470 1937.523 1916.761
 [9] 2158.055 1981.265 2000.490 1837.366 2863.852 1944.671 1886.720 1929.790
[17] 1949.142 2122.238 2145.564 2095.632 2213.338 2162.576 1836.880 1935.268
[25] 2093.470 2101.785 2158.061 2069.820 2321.663 1402.000 2053.995 2075.493
```

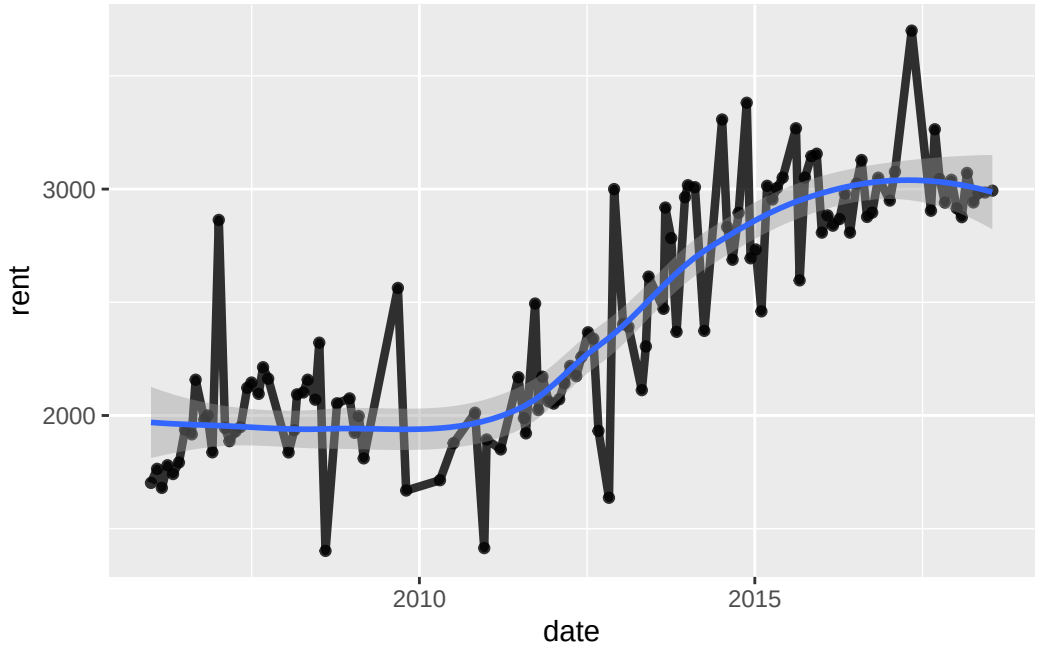
```
 [33] 1923.436 1997.835 1810.720 2563.304 1668.604 1714.088 1877.848 2012.075
 [41] 1414.392 1894.130 1850.200 2169.115 1988.202 1921.168 2494.451 2024.189
 [49] 2171.402 2061.801 2052.503 2071.554 2142.095 2218.942 2173.790 2258.142
 [57] 2367.558 2339.565 1932.022 1636.667 3000.000 2403.243 2392.396 2112.032
 [65] 2304.910 2613.418 2471.167 2918.508 2783.562 2369.468 2964.150 3017.188
 [73] 3008.525 2373.783 3307.472 2832.411 2687.975 2895.371 3381.402 2694.425
 [81] 2733.273 2459.929 3014.436 2954.046 3004.527 3050.929 3268.769 2596.823
 [89] 3051.025 3145.838 3156.274 2807.752 2883.929 2838.391 2867.761 2980.510
 [97] 2807.714 3024.339 3128.416 2877.165 2896.006 3050.295 2949.307 3076.659
[105] 3700.000 2904.348 3264.213 3044.426 2940.444 3041.136 2915.409 2876.117
[113] 3071.498 2941.088 2982.688 2984.801 2993.061
```

We could obtain more statistics via 'jumping' windows.

```
mean_rent <- function(df) {
  summarise(df, date = min(date), rent = mean(price), n = n())
}

slide_period_dfr(.x = rent_df,
                 .i = rent_df$date,
                 .period = 'month',
                 .f = mean_rent) %>%
  ggplot(aes(date, rent)) +
  geom_point(size = 1.5, alpha = 0.8) +
  geom_line(size = 1.5, alpha = 0.8) +
  geom_smooth(se=T)
```

```
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

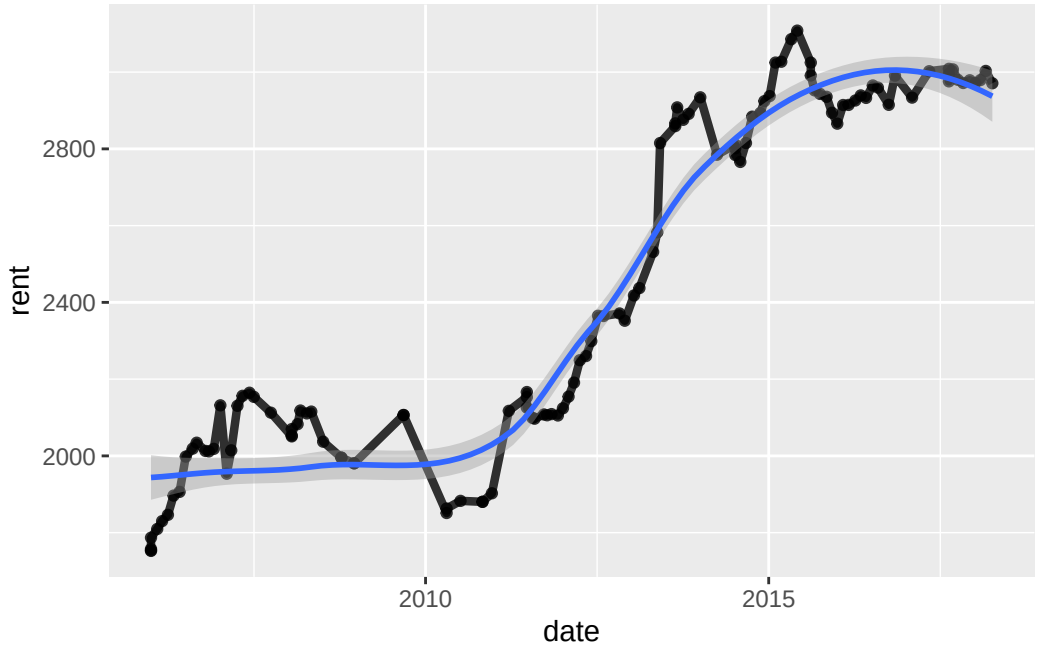


Sliding means.

We could look at moving averages for each month in local neighborhood.

```
slide_period_dfr(.x = rent_df,
                 .i = rent_df$date,
                 .period = 'month',
                 .f = mean_rent,
                 .before = 3,
                 .after = 3) %>%
  ggplot(aes(date, rent)) +
  geom_point(size = 1.5, alpha = 0.8) +
  geom_line(size = 1.5, alpha = 0.8) +
  geom_smooth(se=T)
```

```
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



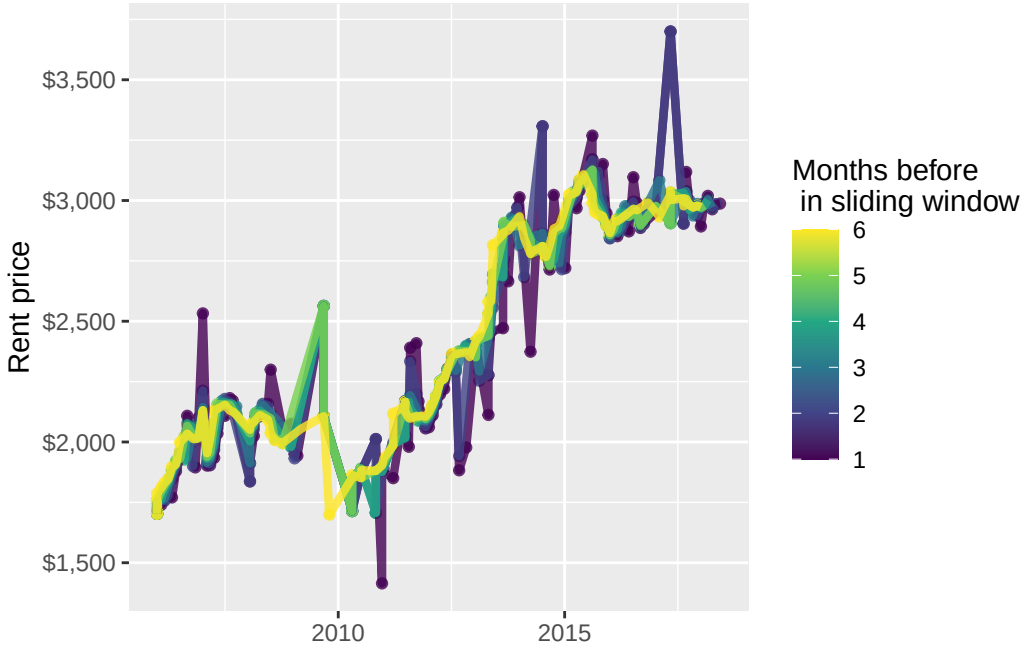
Different sliding windows

Same idea as above, but this time we will be looking at averages of previous one to six months.

```
res <-
  tibble(.before = 1:6) %>%
  mutate(
    mean_rent = map(
      .before,
      ~ slide_period_dfr(.x = rent_df,
                 .i = rent_df$date,
                 .period = 'month',
                 .f = mean_rent,
                 .before = .x)
    )
  )
```

```
res %>%
  unnest(mean_rent) %>%
  ggplot(aes(date, rent, color = .before, group = .before)) +
```

```
  geom_point(size = 1.5, alpha = 0.8) +
  geom_line(size = 1.5, alpha = 0.8) +
  scale_color_viridis_c() +
  scale_y_continuous(labels = scales::dollar) +
  labs(x=NULL, y = 'Rent price', color = "Months before \n in sliding window")
```



Plot above shows these smoothing effect of averaging.