Lesson 8: Chapter 16 Dates and Time

0. Load libraries

```
knitr::opts_chunk$set(echo = TRUE)

library(tidyverse)
library(rlang)

# A library for string operations
library(stringr)

# A library for working with Categorical variables, i.e. factors
library(forcats)

# A library for working with date and time
library(lubridate)

# Load NYC flight dataset
library(nycflights13)
```

1. Working with Date and Time in Tidyverse

lubridate is the package in tidyverse to handle all Date and Time related work. Here is the link to package *lubridate*

Useful functions:

- today(): 2022-03-24
- now(): 2022-03-24 14:01:23

Creating date, date-time from a string:

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

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

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

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

```
mdy("Jan 31, 2017")
```

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

```
date_1 <- ymd("2017-01-31")
class(date_1)
```

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

Date objects are stored in R as integer values, allowing for dates to be compared and manipulated as you would a numeric vector. Logical comparisons are a simple.

```
date_num <- as.numeric(date_1)
date_num
```

```
## [1] 17197
```

```
as_date(date_num #, origin = "1970-01-01"
        )
```

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

Getting components from a date-time

- functions: year(), month(), mday(), yday(), wday(), hour(), minute(), second()

Rounding date

- functions: floor_date(), ceiling_date()

```
# create a funcion
make_datetime_100 <- function(year, month, day, time) {
  make_datetime(year, month, day, time %/% 100, time %% 100)
}

flights_dt <- flights %>%
  filter(!is.na(dep_time), !is.na(arr_time)) %>%
  mutate(
    dep_time = make_datetime_100(year, month, day, dep_time),
    arr_time = make_datetime_100(year, month, day, arr_time),
    sched_dep_time = make_datetime_100(year, month, day, sched_dep_time),
    sched_arr_time = make_datetime_100(year, month, day, sched_arr_time)
  ) %>%
  select(origin, dest, ends_with("delay"), ends_with("time"))

head(flights_dt)
```

```
## # A tibble: 6 x 9
##   origin dest  dep_delay arr_delay dep_time            sched_dep_time
##   <chr>  <chr>     <dbl>     <dbl> <dttm>              <dttm>
## 1 EWR    IAH           2        11 2013-01-01 05:17:00 2013-01-01 05:15:00
## 2 LGA    IAH           4        20 2013-01-01 05:33:00 2013-01-01 05:29:00
## 3 JFK    MIA           2        33 2013-01-01 05:42:00 2013-01-01 05:40:00
## 4 JFK    BQN          -1       -18 2013-01-01 05:44:00 2013-01-01 05:45:00
## 5 LGA    ATL          -6       -25 2013-01-01 05:54:00 2013-01-01 06:00:00
## 6 EWR    ORD          -4        12 2013-01-01 05:54:00 2013-01-01 05:58:00
## # ... with 3 more variables: arr_time <dttm>, sched_arr_time <dttm>,
## #   air_time <dbl>
```

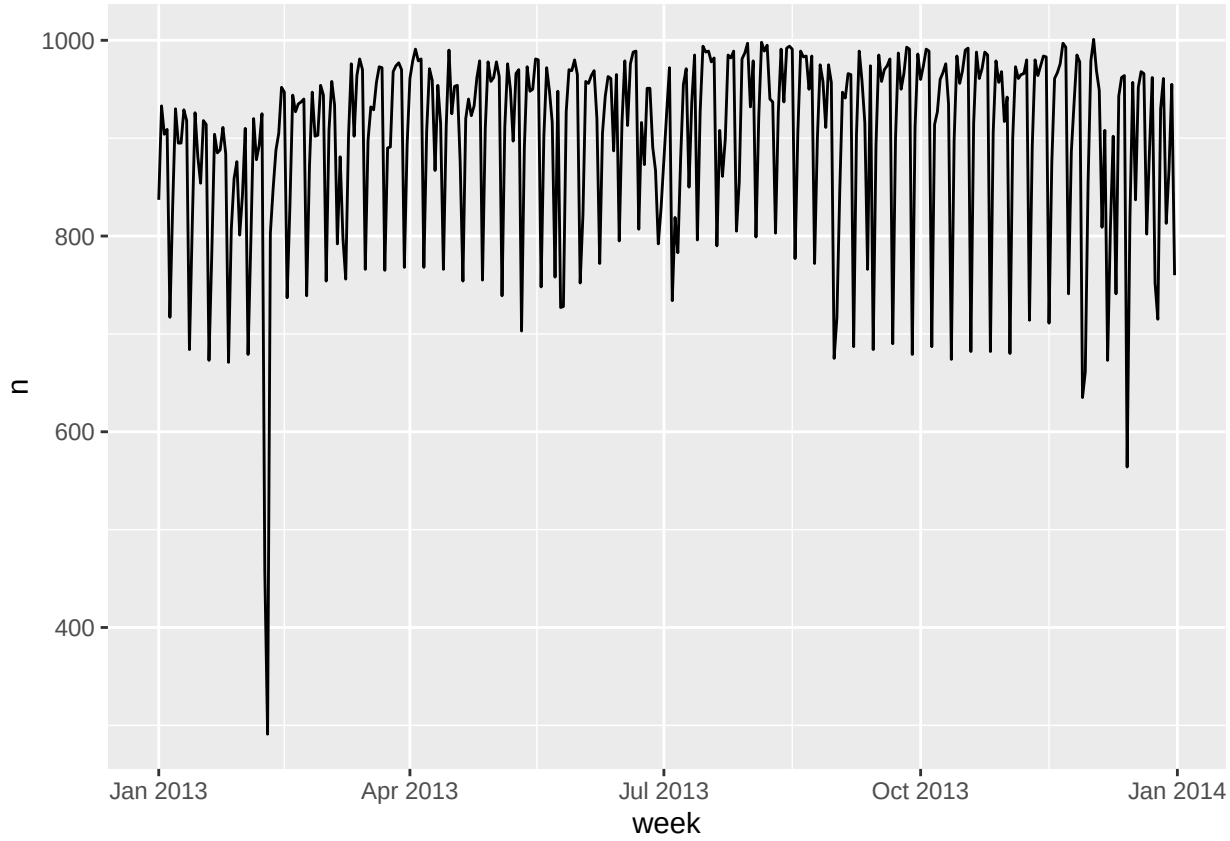
The function floor_date return a date of the starting day of the week, the default starting day is Sunday.

```
flights_dt %>% mutate(week = floor_date(dep_time, "week")) %>% head()
```

```
## # A tibble: 6 x 10
##   origin dest  dep_delay arr_delay dep_time            sched_dep_time
##   <chr>  <chr>     <dbl>     <dbl> <dttm>              <dttm>
## 1 EWR    IAH           2        11 2013-01-01 05:17:00 2013-01-01 05:15:00
## 2 LGA    IAH           4        20 2013-01-01 05:33:00 2013-01-01 05:29:00
## 3 JFK    MIA           2        33 2013-01-01 05:42:00 2013-01-01 05:40:00
## 4 JFK    BQN          -1       -18 2013-01-01 05:44:00 2013-01-01 05:45:00
## 5 LGA    ATL          -6       -25 2013-01-01 05:54:00 2013-01-01 06:00:00
## 6 EWR    ORD          -4        12 2013-01-01 05:54:00 2013-01-01 05:58:00
## # ... with 4 more variables: arr_time <dttm>, sched_arr_time <dttm>,
## #   air_time <dbl>, week <dttm>
```

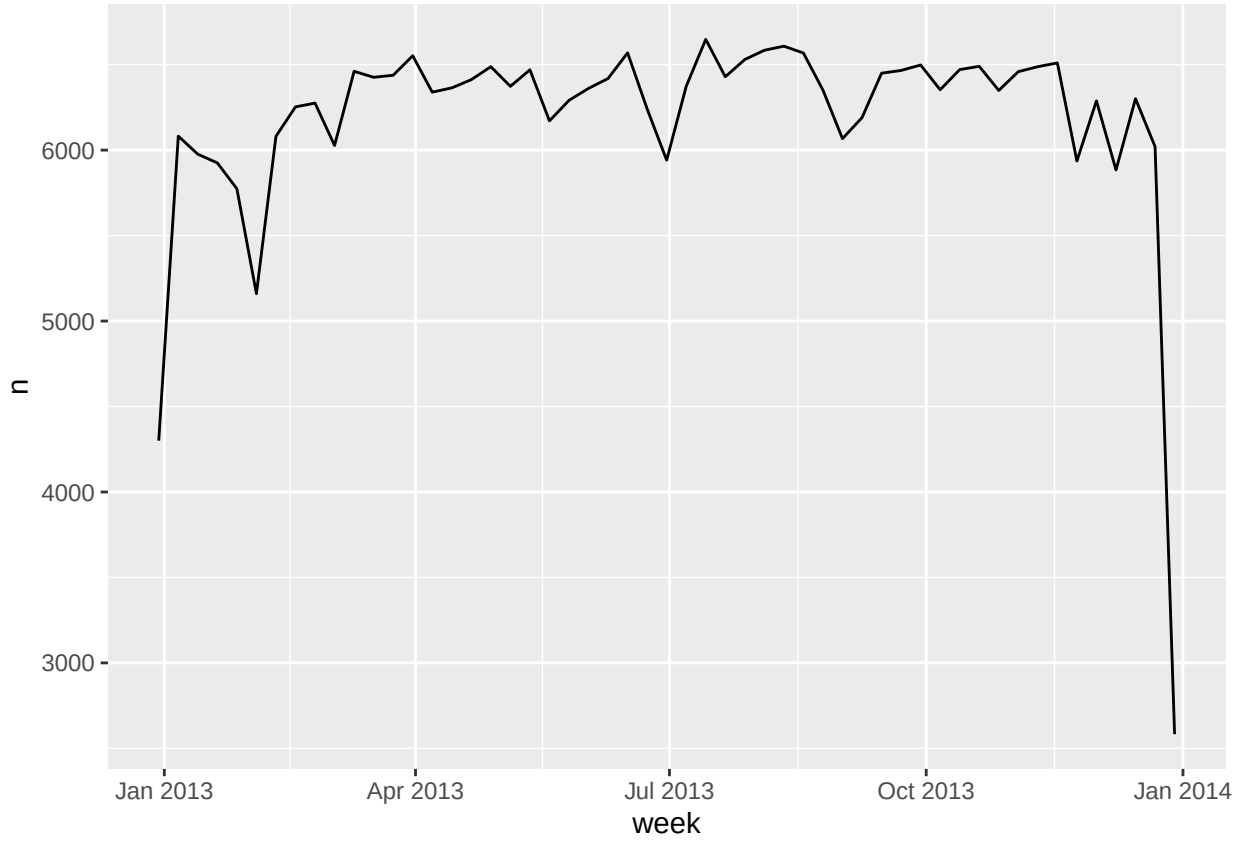
Use the count and floor_date to count the number of flight by **day**.

```
flights_dt %>%
  count(week = floor_date(dep_time, "day")) %>%
  ggplot(aes(week, n)) +
    geom_line()
```



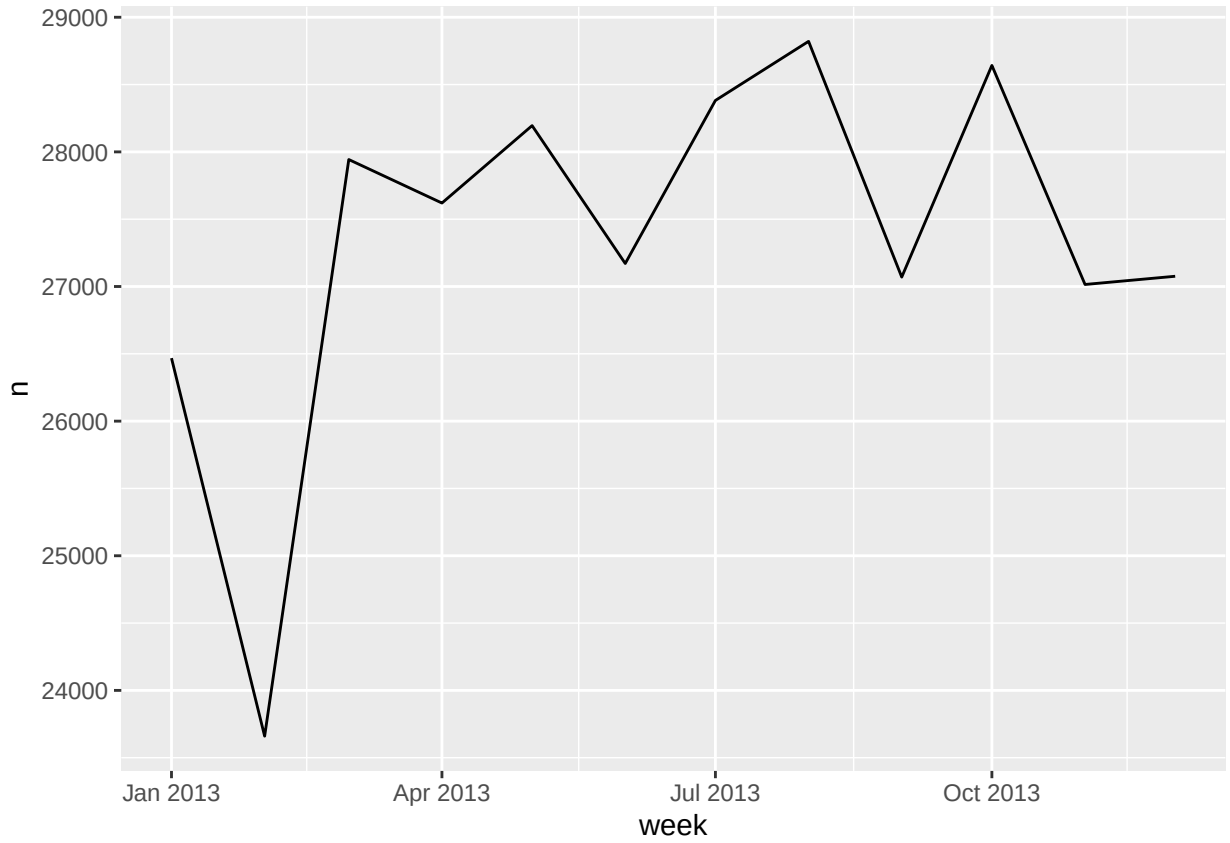
Use the count and floor_date to count the number of flight by **week**.

```
flights_dt %>%
  count(week = floor_date(dep_time, "week")) %>%
  ggplot(aes(week, n)) +
    geom_line()
```



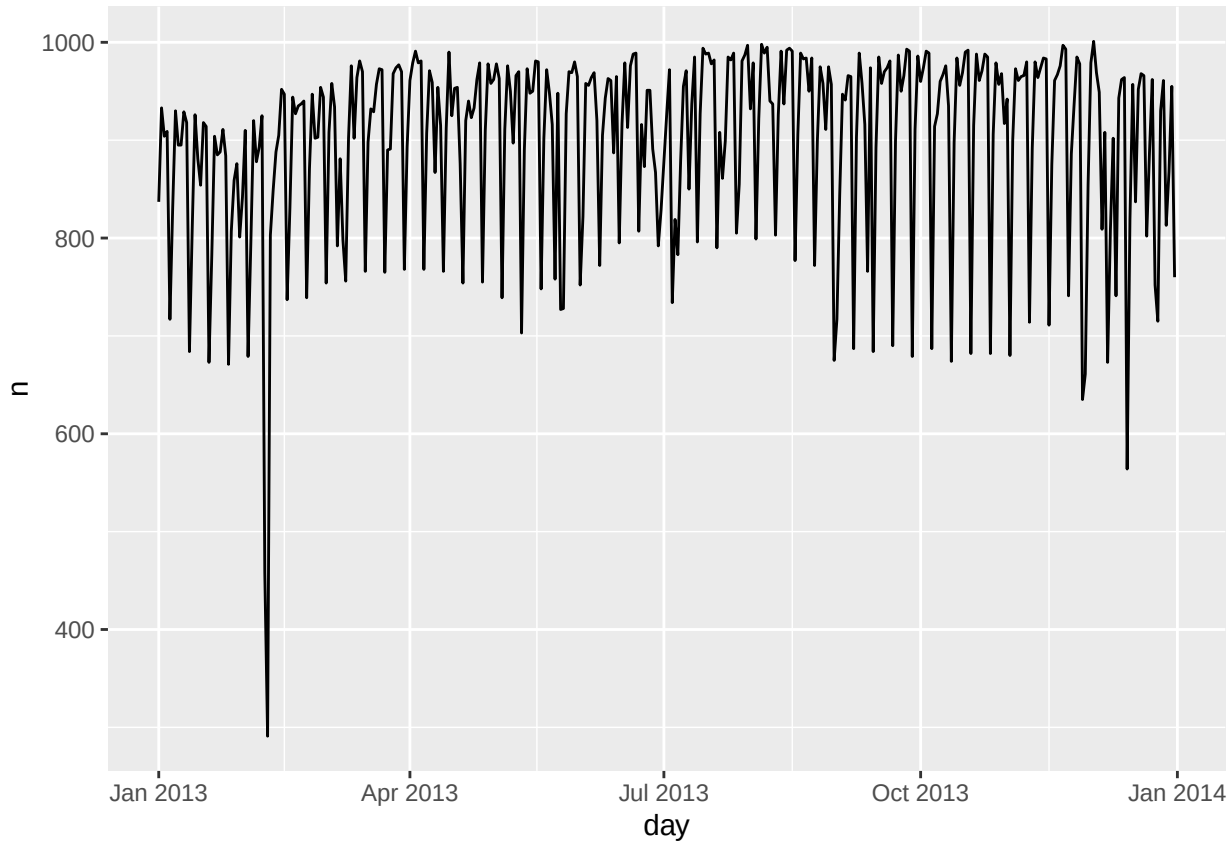
Use the count and floor_date to count the number of flight by **month**.

```
flights_dt %>%
  count(week = floor_date(dep_time, "month")) %>%
  ggplot(aes(week, n)) +
    geom_line()
```



Another way of counting the number of flight by **day**.

```
flights_dt %>%
  count(day = as_date(dep_time)) %>%
  ggplot(aes(day, n)) +
    geom_line()
```



Time Zone

```
Sys.timezone()
```

```
## [1] "America/New_York"
```

2. A Quick Summary about what we have learned

- Chapter 2 data visualization: ggplot2
- Chapter 5 data transformation: dplyr
- Chapter 11 data import: readr, odbc, readxl
- Chapter 12 data transformation: rlang
- Chapter 13 data join:
- Chapter 14 string functions: stringr
- Chapter 15 Categorical variable: forcats
- Chapter 16 Date and Time: lubridate