R Notebook

```
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.1.1

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.6
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

## Warning: package 'ggplot2' was built under R version 4.1.1

## Warning: package 'tidyr' was built under R version 4.1.2

## Warning: package 'readr' was built under R version 4.1.2

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

library(lubridate)

## Warning: package 'lubridate' was built under R version 4.1.2

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union

library(nycflights13)

## Warning: package 'nycflights13' was built under R version 4.1.1

#1
```

```
('2020-06-06' %--% '2021-07-14')%/%days(1)
```

```
## [1] 403
```

#2

```
leap_year('1988-01-01')
```

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

Yes, 1988 is a leap year #3

```
leap_year('1989-01-01')
```

```
## [1] FALSE
```

1989 is not a leap year. #4

There's dmonth() function as well. Which tells the information about the weeks and seconds in month.

```
dmonths(2)
```

```
## [1] "5259600s (~8.7 weeks)"
```

#5

```
as.period(interval(start = '1962-04-11', end = today()))
```

```
## [1] "59y 9m 5d 0H 0M 0S"
```

#6

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

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),
```

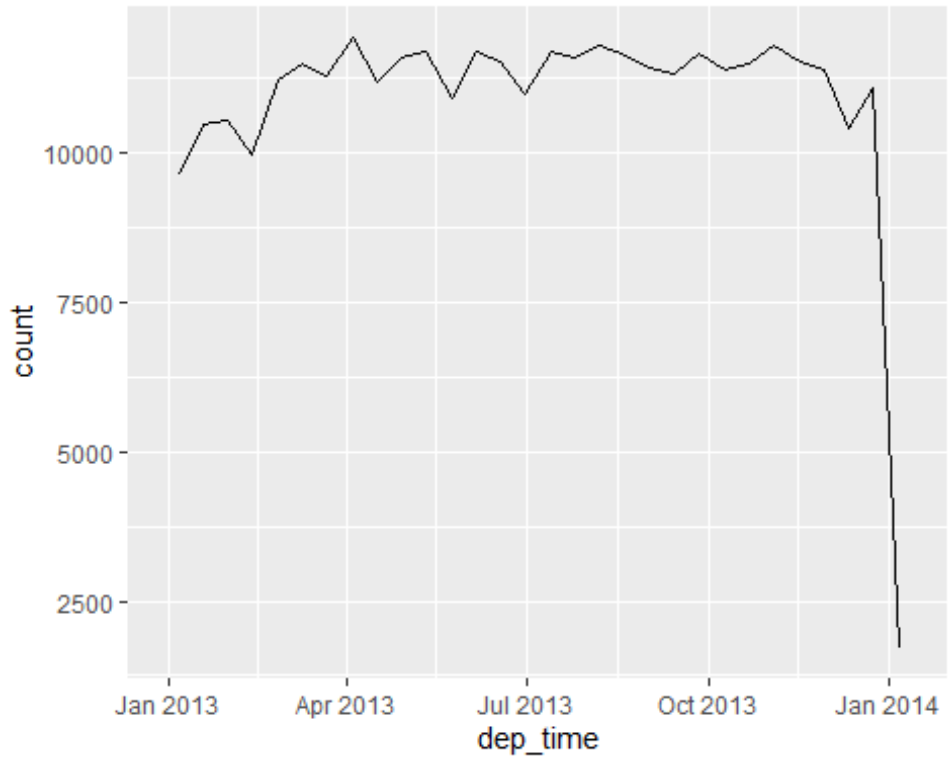
```
  sched_arr_time = make_datetime_100(year, month, day, sched_arr_time)
 ) %>%
 select(origin, dest, carrier, arr_time,dep_time) ->

 flights_dt
flights_dt
```

```
## # A tibble: 328,063 x 5
##    origin dest  carrier arr_time            dep_time
##    <chr>  <chr> <chr>   <dttm>              <dttm>
##  1 EWR    IAH   UA      2013-01-01 08:30:00 2013-01-01 05:17:00
##  2 LGA    IAH   UA      2013-01-01 08:50:00 2013-01-01 05:33:00
##  3 JFK    MIA   AA      2013-01-01 09:23:00 2013-01-01 05:42:00
##  4 JFK    BQN   B6      2013-01-01 10:04:00 2013-01-01 05:44:00
##  5 LGA    ATL   DL      2013-01-01 08:12:00 2013-01-01 05:54:00
##  6 EWR    ORD   UA      2013-01-01 07:40:00 2013-01-01 05:54:00
##  7 EWR    FLL   B6      2013-01-01 09:13:00 2013-01-01 05:55:00
##  8 LGA    IAD   EV      2013-01-01 07:09:00 2013-01-01 05:57:00
##  9 JFK    MCO   B6      2013-01-01 08:38:00 2013-01-01 05:57:00
## 10 LGA    ORD   AA      2013-01-01 07:53:00 2013-01-01 05:58:00
## # ... with 328,053 more rows
```

#7

```
ggplot(flights_dt,aes(x = dep_time)) + geom_line(aes(group=1),stat='bin')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



#8

```
m=flights_dt%>%
 filter(origin == 'LGA'&dest == 'DFW'&carrier =='AA')%>%
 select(origin, dest, carrier, arr_time)%>%
 slice(115:125)
m
```

```
## # A tibble: 11 x 4
##    origin dest  carrier arr_time
##    <chr>  <chr> <chr>   <dttm>
##  1 LGA    DFW   AA      2013-01-09 16:16:00
##  2 LGA    DFW   AA      2013-01-09 19:17:00
##  3 LGA    DFW   AA      2013-01-09 19:36:00
##  4 LGA    DFW   AA      2013-01-09 20:53:00
##  5 LGA    DFW   AA      2013-01-09 22:24:00
##  6 LGA    DFW   AA      2013-01-10 08:37:00
##  7 LGA    DFW   AA      2013-01-10 10:20:00
##  8 LGA    DFW   AA      2013-01-10 11:22:00
```

```
##  9 LGA    DFW   AA      2013-01-10 12:16:00
## 10 LGA    DFW   AA      2013-01-10 13:19:00
## 11 LGA    DFW   AA      2013-01-10 13:23:00
```

#9

```
(m$arr_time[1] %--% m$arr_time[2])/dminutes(1)
```

```
## [1] 181
```