DS413613 HOMEWORK1

James Dickens

1/20/2022

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.0.5

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

## v ggplot2 3.3.3 v purrr 0.3.4  
## v tibble 3.1.2 v dplyr 1.0.5  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.1

## Warning: package 'ggplot2' was built under R version 4.0.5

## Warning: package 'tidyr' was built under R version 4.0.5

## Warning: package 'readr' was built under R version 4.0.5

## Warning: package 'forcats' was built under R version 4.0.5

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(nycflights13)

## Warning: package 'nycflights13' was built under R version 4.0.5

library(lubridate) # new package

## Warning: package 'lubridate' was built under R version 4.0.5

##   
## Attaching package: 'lubridate'

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

library(dplyr)  
  
next\_year <- today() + years(1)  
next\_year

## [1] "2023-01-20"

(today() %--% next\_year)%/%months(1)

## [1] 12

# 1 Use and show R coding to find the number of  
# days from June 6th 2020 to July 14th 2021  
  
(("2020-06-06") %--% ("2021-07-14"))%/%days(1)

## [1] 403

# 2 Use and show R coding to confirm that the year  
# 1988 was a leap year  
  
(("1988-01-01") %--% ("1989-01-01"))%/%days(1)

## [1] 366

# 3 Use and show R coding to confirm that the year  
# 1989 was not a leap year  
  
(("1989-01-01") %--% ("1990-01-01"))%/%days(1)

## [1] 365

# 4 Why is there months() but no dmonths ? (Answer  
# in 3 to four sentences)  
  
  
# 5 John was born April 11th, 1962. Use and show  
# R coding to determine how old John is in years  
  
(("1962-04-11") %--% today())%/%years(1)

## [1] 59

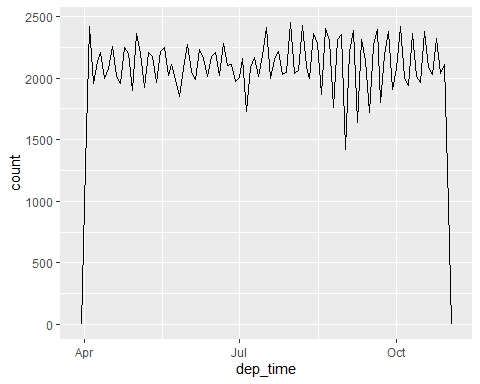
flights

## # A tibble: 336,776 x 19  
## year month day dep\_time sched\_dep\_time dep\_delay arr\_time sched\_arr\_time  
## <int> <int> <int> <int> <int> <dbl> <int> <int>  
## 1 2013 1 1 517 515 2 830 819  
## 2 2013 1 1 533 529 4 850 830  
## 3 2013 1 1 542 540 2 923 850  
## 4 2013 1 1 544 545 -1 1004 1022  
## 5 2013 1 1 554 600 -6 812 837  
## 6 2013 1 1 554 558 -4 740 728  
## 7 2013 1 1 555 600 -5 913 854  
## 8 2013 1 1 557 600 -3 709 723  
## 9 2013 1 1 557 600 -3 838 846  
## 10 2013 1 1 558 600 -2 753 745  
## # ... with 336,766 more rows, and 11 more variables: arr\_delay <dbl>,  
## # carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,  
## # air\_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time\_hour <dttm>

# 6 Modify the flights\_dt coding in the notes or the book  
# to obtain the following partial data table shown below.  
# Show all required coding.  
  
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   
# Now produce the frequency plot shown which conveys frequency  
# counts for the months of April, July, and October for the year   
# 2013  
  
flights\_dt %>%   
 filter(dep\_time < ymd(20131031),dep\_time > ymd(20130401)) %>%   
 ggplot(aes(dep\_time)) +   
 geom\_freqpoly(binwidth = 200000) # 600 s = 10 minutes



#8  
# Now use dplyr functions to produce a data table that  
# shows arrival times for American Airlines at the Dallas  
# Fort Worth Airpot from the Laguardia airport in New York.  
# Your output should show rows 115 to 125.  
# A partial table is provided below.  
  
  
flights\_dt %>%  
 select(origin,dest,carrier, arr\_time)%>%  
 filter( carrier == "AA", dest == "DFW", origin == "LGA")%>%  
 slice(115:125) -> flights\_dt4  
flights\_dt4

## # 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  
# Using the first two observational date time designations, Use   
# and show R code to confirm that there are 181 minutes timme  
# intervals between them.  
  
(("2013-01=09\_16:16:00") %--% ("2013-01-09\_19:17:00"))%/%minutes(1)

## [1] 181