Homework1

Samuel Mweni

2023-01-27

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.3.6 ✔ purrr 0.3.4   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.1 ✔ stringr 1.4.1   
## ✔ readr 2.1.3 ✔ forcats 0.5.2

## Warning: package 'readr' was built under R version 4.2.2

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(nycflights13)  
library(lubridate)

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

library(dplyr)

#1) Use and show R coding to find the number of days from June 6th 2020 to July 14th 2021

begin\_date<-ymd("2020-06-06",tz="UTC")  
begin\_date

## [1] "2020-06-06 UTC"

end\_date<-ymd("2021-07-14",tz="UTC")  
end\_date

## [1] "2021-07-14 UTC"

(begin\_date %--%end\_date )%/%days(1)

## [1] 403

#2) Use and show R coding to confirm that the year 1988 was a leap year.

leap\_year(1988)

## [1] TRUE

#3) Use and show R coding to confirm that the year 1989 was not a leap year.

leap\_year(1989)

## [1] FALSE

#Why is there months() but no dmonths ? (Answer in 3 to four sentences) months() is useful for generating a sequence of months, while there is no need for a dmonths() function as days within a month can be accessed using existing date functions and arithmetic operations.

#5) John was born April 11th, 1962. Use and show R coding to determine how old John is in years

birth\_year<-("1962-04-11,tz=UTC")  
(birth\_year%--%today())%/%years(1)

## [1] 60

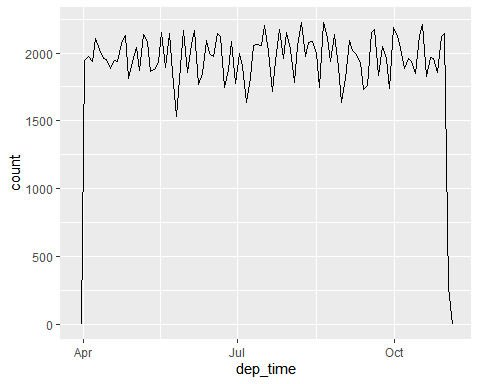
#6) Modify the flights\_dt coding in the notes or the book to obtain the following partial data table shown below. Show all required coding. (Most of the coding needed is provided in the notes)

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, ends\_with("delay"), ends\_with("time")) ->  
  
flights\_dt  
  
flights\_dt%>%  
 select(origin,dest,carrier,arr\_time,dep\_time)->flights\_dt1  
flights\_dt1

## # A tibble: 328,063 × 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, using your table from #6, produce the frequency plot shown which conveys frequency counts for the months of April, July, and October for the year 2013

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, ends\_with("delay"), ends\_with("time")) ->  
  
   
  
flights\_dt  
  
   
  
flights\_dt %>% filter(month(dep\_time)>=4, month(dep\_time)<=10)-> flights\_dt\_7   
  
freq\_plt <- ggplot(flights\_dt\_7, aes(x = dep\_time)) + geom\_freqpoly(bins = 100)  
  
print(freq\_plt)



#8) Now use dplyr functions to produce a data table that shows arrival times for American Airlines at the Dallas Fort Worth Airport from the LaGuardia airport in New York. Your output should show rows 115 to 125.

flights\_dt1%>%  
 select(origin,dest,carrier,arr\_time)->dt1  
dt1%>%  
 filter(origin == "LGA", carrier == "AA",dest=="DFW")%>%  
 slice(115:125)

## # A tibble: 11 × 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 from your #8 table, Use and show R code to confirm that there are 181 minutes time intervals between them

start\_time<-ymd\_hms("2013-01-09 16:16:00",tz="UTC")  
end\_time<-ymd\_hms("2013-01-09 19:17:00",tz="UTC")  
(start\_time %--%end\_time)%/%minutes(1)

## [1] 181