Working with Dates and Times

Noah Foilb

9/16/2020

|  |
| --- |
| ## MAKE SURE YOUR CODE IS PROPERLY COMMENTED THROUGOUT! |

#### Start by loading the needed libraries: readr, lubridate, dplyr, and nycflights13

library(readr)  
library(lubridate)

##   
## Attaching package: 'lubridate'

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

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

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

library(nycflights13)

## Part 1 Working with dates

#### 1. Store your birthday in a variable called bday. Display bday so that you can see that it is in character format. (See step 1 of code 6A)

# Assign Variable to bday  
bday <- "August 23, 1999"  
  
# Use Str to show structure of bday  
str(bday)

## chr "August 23, 1999"

#### 2. Convert your birthday to date format and store it in a variable called birthday. Display birthday so that you can be sure it is in date format. (See step 1 of code 6A)

# Use mdy on bday and assign it to birthday  
birthday <- mdy(bday)  
  
# Use str to show structure of birthday  
str(birthday)

## Date[1:1], format: "1999-08-23"

#### 3. Display both the abbreviated and unabbreviated day of the week on which you were born!

# Weekday without abbreviation  
wday(birthday,label = TRUE, abbr = FALSE)

## [1] Monday  
## 7 Levels: Sunday < Monday < Tuesday < Wednesday < Thursday < ... < Saturday

# Weekday with abbreviation  
wday(birthday,label = TRUE)

## [1] Mon  
## Levels: Sun < Mon < Tue < Wed < Thu < Fri < Sat

#### For this next part, you will be working with the flights data set from the nycflights13 package. Run the head(flights) and help(flights) commands to remind yourself what flights looks like and what the variables mean.

head(flights)

## # A tibble: 6 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  
## # ... with 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>

help(flights)

## starting httpd help server ... done

### Run this code that you saw in the week\_6A code in class to create a data frame called flights3 that contains the columns arr\_delay and dep\_delay, has the original columns of month, day, year, renamed to M, D, and Y, respectively, and uses the make\_date command to create a new column called date containing the calendar date in date format. Show the first 6 rows.

flights3 <- flights %>%   
 select(month, day, year, arr\_delay, dep\_delay) %>%   
 rename(M=month, D=day, Y=year) %>%   
 mutate(date=make\_date(year=Y, month=M, day=D))  
head(flights3)

## # A tibble: 6 x 6  
## M D Y arr\_delay dep\_delay date   
## <int> <int> <int> <dbl> <dbl> <date>   
## 1 1 1 2013 11 2 2013-01-01  
## 2 1 1 2013 20 4 2013-01-01  
## 3 1 1 2013 33 2 2013-01-01  
## 4 1 1 2013 -18 -1 2013-01-01  
## 5 1 1 2013 -25 -6 2013-01-01  
## 6 1 1 2013 12 -4 2013-01-01

#### 4. Stating with flights3, create a new data frame called flights4 which has columns called dia, mes, and ano (Spanish for day, month, and year), where:

* dia is the unabbreviated day of the week (e.g., Monday)
* mes is the unabbreviated month (e.g., September)
* ano is the year in numeric format (e.g., 2013) Display the first 6 rows.
* Hint: you will need to use the wday(), month(), and year() commands within a mutate command.

# Assign Variable  
flights4 <- flights3 %>%  
   
 # Use wday, month, and year functions  
 mutate(dia = wday(date,label = TRUE, abbr = FALSE),mes = month(date, label=TRUE, abbr = FALSE), ano = year(date))

#### 5. Group the data in flights4 by dia (day), summarize the data to show the average arrival and departure delays for each dia (day), then display the results in increasing order by average arrival delay. ANSWER THIS QUESTION: What is the best day of the week to fly in NYC and why?

* You will either need to use drop\_na() or na.rm from inside of the mean() commands to ignore rows for which delay information is not available.
* Notice that if you ask for help(flights), the documentation for the flights data frame shown in the help window includes the following explanation:
  + dep\_delay, arr\_delay: “Departure and arrival delays, in minutes. Negative times represent early departures/arrivals.”

# Assign Variable  
flights4 %>%  
   
 # Group by day  
 group\_by(dia) %>%  
   
 # Summarize  
 summarize(  
   
 # Avg arrival delay  
 Avg\_Arr\_Delay = mean(arr\_delay,na.rm = TRUE),  
   
 # Avg dep delay  
 Avg\_Dep\_Delay = mean(dep\_delay,na.rm = TRUE),  
 .groups = 'drop') %>%  
   
 # Rearrange order  
 arrange(Avg\_Arr\_Delay)

## # A tibble: 7 x 3  
## dia Avg\_Arr\_Delay Avg\_Dep\_Delay  
## <ord> <dbl> <dbl>  
## 1 Saturday -1.45 7.65  
## 2 Sunday 4.82 11.6   
## 3 Tuesday 5.39 10.6   
## 4 Wednesday 7.05 11.8   
## 5 Friday 9.07 14.7   
## 6 Monday 9.65 14.8   
## 7 Thursday 11.7 16.1

# To be able to answer the question, we would need to sum the average delay times and see which one has the lowest. This would be saturday because it has the lowest values in both columns.

#### 6. Group the data in flights4 by mes (month) and summarize the data to show the average arrival and departure delays for each mes (month), then display the results in decreasing order by average departure delay. ANSWER THIS QUESTION: What are the two worst months to fly in NYC and why?

* You will either need to use drop\_na() or na.rm from inside of the mean() commands to ignore rows for which delay information is not available.

# Assign variable  
flights4 %>%  
   
 # Group by month  
 group\_by(mes) %>%  
   
 # SUmmarize  
 summarize(  
   
 # Avg Arrival Delay  
 Avg\_Arr\_Delay = mean(arr\_delay,na.rm = TRUE),  
   
 # Avg dep delay  
 Avg\_Dep\_Delay = mean(dep\_delay,na.rm = TRUE),  
 .groups = 'drop') %>%  
   
 # Rearrange order  
 arrange(desc(Avg\_Dep\_Delay))

## # A tibble: 12 x 3  
## mes Avg\_Arr\_Delay Avg\_Dep\_Delay  
## <ord> <dbl> <dbl>  
## 1 July 16.7 21.7   
## 2 June 16.5 20.8   
## 3 December 14.9 16.6   
## 4 April 11.2 13.9   
## 5 March 5.81 13.2   
## 6 May 3.52 13.0   
## 7 August 6.04 12.6   
## 8 February 5.61 10.8   
## 9 January 6.13 10.0   
## 10 September -4.02 6.72  
## 11 October -0.167 6.24  
## 12 November 0.461 5.44

# To be able to answer the question, we would need to sum the average delay times and see which two are the highest. These would be July and June!

## Part 2 Join

#### 7. Start out by looking at the first six rows of the flights, airlines, and airports data frames. ANSWER THIS QUESTION: What is the common column in flights and airlines?

# Look at first six rows  
head(flights)

## # A tibble: 6 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  
## # ... with 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>

head(airlines)

## # A tibble: 6 x 2  
## carrier name   
## <chr> <chr>   
## 1 9E Endeavor Air Inc.   
## 2 AA American Airlines Inc.   
## 3 AS Alaska Airlines Inc.   
## 4 B6 JetBlue Airways   
## 5 DL Delta Air Lines Inc.   
## 6 EV ExpressJet Airlines Inc.

head(airports)

## # A tibble: 6 x 8  
## faa name lat lon alt tz dst tzone   
## <chr> <chr> <dbl> <dbl> <dbl> <dbl> <chr> <chr>   
## 1 04G Lansdowne Airport 41.1 -80.6 1044 -5 A America/New\_Y~  
## 2 06A Moton Field Municipal Airp~ 32.5 -85.7 264 -6 A America/Chica~  
## 3 06C Schaumburg Regional 42.0 -88.1 801 -6 A America/Chica~  
## 4 06N Randall Airport 41.4 -74.4 523 -5 A America/New\_Y~  
## 5 09J Jekyll Island Airport 31.1 -81.4 11 -5 A America/New\_Y~  
## 6 0A9 Elizabethton Municipal Air~ 36.4 -82.2 1593 -5 A America/New\_Y~

#The common column is the carrier

#### 8. Create data frame FLIGHTS (use all capitals so that it is different than flights) listing number of flights by each carrier as num\_flights, and sort the data from the most flights to least (group\_by, summarize, arrange). Display the entire data frame.

* If done correctly, you’ll have 16 rows with UA as the top carrier with 58,665 flights.

#Assign Variable  
FLIGHTS <- flights %>%  
   
 # Group by carrier  
 group\_by(carrier) %>%  
   
 # Summarize and find number of flights  
 summarize(num\_flights = n(),.groups = 'drop') %>%  
   
 # Rearrange order  
 arrange(desc(num\_flights))  
  
# Display  
print(FLIGHTS)

## # A tibble: 16 x 2  
## carrier num\_flights  
## <chr> <int>  
## 1 UA 58665  
## 2 B6 54635  
## 3 EV 54173  
## 4 DL 48110  
## 5 AA 32729  
## 6 MQ 26397  
## 7 US 20536  
## 8 9E 18460  
## 9 WN 12275  
## 10 VX 5162  
## 11 FL 3260  
## 12 AS 714  
## 13 F9 685  
## 14 YV 601  
## 15 HA 342  
## 16 OO 32

#### 9. Use inner\_join to join FLIGHTS and airlines together by their common column so that one can now see the carrier code, the num\_flights, and the name of the airline. Name this new data frame flights\_by\_carrier. Display the entire data frame.

# Join the two data sets using inner join  
flights\_by\_carrier <- inner\_join(FLIGHTS,airlines,by = c("carrier"))  
  
# Display  
print(flights\_by\_carrier)

## # A tibble: 16 x 3  
## carrier num\_flights name   
## <chr> <int> <chr>   
## 1 UA 58665 United Air Lines Inc.   
## 2 B6 54635 JetBlue Airways   
## 3 EV 54173 ExpressJet Airlines Inc.   
## 4 DL 48110 Delta Air Lines Inc.   
## 5 AA 32729 American Airlines Inc.   
## 6 MQ 26397 Envoy Air   
## 7 US 20536 US Airways Inc.   
## 8 9E 18460 Endeavor Air Inc.   
## 9 WN 12275 Southwest Airlines Co.   
## 10 VX 5162 Virgin America   
## 11 FL 3260 AirTran Airways Corporation  
## 12 AS 714 Alaska Airlines Inc.   
## 13 F9 685 Frontier Airlines Inc.   
## 14 YV 601 Mesa Airlines Inc.   
## 15 HA 342 Hawaiian Airlines Inc.   
## 16 OO 32 SkyWest Airlines Inc.

#### 10. Create a data frame called Destinations for all flights with an origin of JFK which groups the flights by the destination (dest), summarizes the number of flights to each destination as flights\_to\_destination, and arranges the destinations by decreasing number of flights. Display the entire data frame.

* If done correctly, LAX will be the top destination with 11,262 flights.

# Assign Variable  
Destinations <- flights %>%  
   
 # Filter for JFK  
 filter(origin == "JFK") %>%  
   
 # Group by dest  
 group\_by(dest) %>%  
   
 # Summarize and find flights to dest  
 summarize(flights\_to\_destination = n(),.groups = 'drop') %>%  
   
 # Rearrange order  
 arrange(desc(flights\_to\_destination))  
  
# Display Results  
print(Destinations)

## # A tibble: 70 x 2  
## dest flights\_to\_destination  
## <chr> <int>  
## 1 LAX 11262  
## 2 SFO 8204  
## 3 BOS 5898  
## 4 MCO 5464  
## 5 SJU 4752  
## 6 FLL 4254  
## 7 LAS 3987  
## 8 BUF 3582  
## 9 MIA 3314  
## 10 DCA 3270  
## # ... with 60 more rows

#### 10. Join together Destinations and airports to show the names of each destination along with other location data. In Destinations, the common column is called dest, while in airports this column is called faa. Call this data frame Destination\_airports. Display the entire data frame.

* Not all destinations in the Destinations data frame are included in airports. Use the proper join command so that all destinations are displayed even if the name of the airport is unknown.
* Your new data frame should still have 70 rows.

# Join two data sets using left\_ join  
Destination\_airport <- left\_join(Destinations,airports,by = c("dest" = "faa"))  
  
# Display results  
print(Destination\_airport)

## # A tibble: 70 x 9  
## dest flights\_to\_destin~ name lat lon alt tz dst tzone   
## <chr> <int> <chr> <dbl> <dbl> <dbl> <dbl> <chr> <chr>   
## 1 LAX 11262 Los Angeles~ 33.9 -118. 126 -8 A America~  
## 2 SFO 8204 San Francis~ 37.6 -122. 13 -8 A America~  
## 3 BOS 5898 General Edw~ 42.4 -71.0 19 -5 A America~  
## 4 MCO 5464 Orlando Intl 28.4 -81.3 96 -5 A America~  
## 5 SJU 4752 <NA> NA NA NA NA <NA> <NA>   
## 6 FLL 4254 Fort Lauder~ 26.1 -80.2 9 -5 A America~  
## 7 LAS 3987 Mc Carran I~ 36.1 -115. 2141 -8 A America~  
## 8 BUF 3582 Buffalo Nia~ 42.9 -78.7 724 -5 A America~  
## 9 MIA 3314 Miami Intl 25.8 -80.3 8 -5 A America~  
## 10 DCA 3270 Ronald Reag~ 38.9 -77.0 15 -5 A America~  
## # ... with 60 more rows

#### 11. Use the proper R command to display the date and time that you completed this homework assignment!

# Hour now  
hour(now())

## [1] 18

# Minute now  
minute(now())

## [1] 6

# Seconds now  
second(now())

## [1] 14.33116

# Year now  
year(now())

## [1] 2020

#Month Now  
month(now(), label=TRUE, abbr=FALSE)

## [1] October  
## 12 Levels: January < February < March < April < May < June < ... < December

# Day of week now  
wday(now(), label=TRUE, abbr=FALSE)

## [1] Wednesday  
## 7 Levels: Sunday < Monday < Tuesday < Wednesday < Thursday < ... < Saturday