# Data analysis work

Phillip Hughes

2023-05-13

**Assignment Instructions** Complete all questions below. After completing the assignment, knit your document, and download both your .Rmd and knitted output. Upload your files for peer review.

For each response, include comments detailing your response and what each line does.

---

**Question 1.** Using the nycflights13 dataset, find all flights that departed in July, August, or September using the helper function between().

```
flights %>%
    filter(between(month, 7, 9)) %>%
    drop_na(dep_time)
```

```
## # A tibble: 84,448 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     7     1        1           2029       212      236           2359
##  2  2013     7     1        2           2359         3      344            344
##  3  2013     7     1       29           2245       104      151              1
##  4  2013     7     1       43           2130       193      322             14
##  5  2013     7     1       44           2150       174      300            100
##  6  2013     7     1       46           2051       235      304           2358
##  7  2013     7     1       48           2001       287      308           2305
##  8  2013     7     1       58           2155       183      335             43
##  9  2013     7     1      100           2146       194      327             30
## 10  2013     7     1      100           2245       135      337            135
## # i 84,438 more rows
## # i 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>
```

**Question 2.** Using the nycflights13 dataset sort flights to find the 10 flights that flew the furthest. Put them in order of fastest to slowest.

```
distance<-flights%>%
  arrange(desc(distance), air_time)%>%
  select(distance,air_time,everything())
head(distance, n=10)
```

```
## # A tibble: 10 x 19
##    distance air_time  year month   day dep_time sched_dep_time dep_delay
##       <dbl>    <dbl> <int> <int> <int>    <int>          <int>     <dbl>
##  1     4983      580  2013     5     7      959           1000        -1
##  2     4983      580  2013     6     6     1044           1000        44
##  3     4983      580  2013     9    29      957           1000        -3
##  4     4983      581  2013     6     7      952           1000        -8
##  5     4983      582  2013     6     8      951           1000        -9
##  6     4983      582  2013     9     6      955           1000        -5
##  7     4983      584  2013     2    26     1000            900        60
##  8     4983      584  2013     5     6      956           1000        -4
##  9     4983      584  2013     9    28      955           1000        -5
## 10     4983      585  2013     7     3      957           1000        -3
## # i 11 more variables: arr_time <int>, sched_arr_time <int>, arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>
```

**Question 3.** Using the nycflights13 dataset, calculate a new variable called "hr_delay" and arrange the flights dataset in order of the arrival delays in hours (longest delays at the top). Put the new variable you created just before the departure time.Hint: use the experimental argument .before.

```
flights%>%
  mutate(hr_delay=dep_delay/60)%>%
  relocate(hr_delay, .before = dep_time)%>%
  arrange(desc(hr_delay))
```

```
## # A tibble: 336,776 x 20
##     year month   day hr_delay dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <dbl>    <int>          <int>     <dbl>    <int>
##  1  2013     1     9     21.7      641            900      1301     1242
##  2  2013     6    15     19.0     1432           1935      1137     1607
##  3  2013     1    10     18.8     1121           1635      1126     1239
##  4  2013     9    20     16.9     1139           1845      1014     1457
##  5  2013     7    22     16.8      845           1600      1005     1044
##  6  2013     4    10     16       1100           1900       960     1342
##  7  2013     3    17     15.2     2321            810       911      135
##  8  2013     6    27     15.0      959           1900       899     1236
##  9  2013     7    22     15.0     2257            759       898      121
## 10  2013    12     5     14.9      756           1700       896     1058
## # i 336,766 more rows
## # i 12 more variables: sched_arr_time <int>, 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>
```

**Question 4.** Using the nycflights13 dataset, find the most popular destinations (those with more than 2000 flights) and show the destination, the date info, the carrier. Then show just the number of flights for each popular destination.

```
PopDests <- flights %>%
  group_by(dest) %>%
  filter(n() > 2000)
```

```r
PopDestsData <- PopDests %>%
  select(dest, year, month, day, carrier)

NumFlightsToEach <- PopDests %>%
  group_by(dest) %>%
  summarize(NumFlights = n()) %>%
  arrange(desc(NumFlights))

PopDestsData %>%
  inner_join(NumFlightsToEach, by = "dest")%>%
  group_by(dest)
```

```
## # A tibble: 302,969 x 6
## # Groups:   dest [46]
##    dest   year month   day carrier NumFlights
##    <chr> <int> <int> <int> <chr>        <int>
##  1 IAH    2013     1     1 UA            7198
##  2 IAH    2013     1     1 UA            7198
##  3 MIA    2013     1     1 AA           11728
##  4 ATL    2013     1     1 DL           17215
##  5 ORD    2013     1     1 UA           17283
##  6 FLL    2013     1     1 B6           12055
##  7 IAD    2013     1     1 EV            5700
##  8 MCO    2013     1     1 B6           14082
##  9 ORD    2013     1     1 AA           17283
## 10 PBI    2013     1     1 B6            6554
## # i 302,959 more rows
```

```r
PopDestsData%>% count(dest,name = 'flights', sort = TRUE)
```

```
## # A tibble: 46 x 2
## # Groups:   dest [46]
##    dest  flights
##    <chr>   <int>
##  1 ORD     17283
##  2 ATL     17215
##  3 LAX     16174
##  4 BOS     15508
##  5 MCO     14082
##  6 CLT     14064
##  7 SFO     13331
##  8 FLL     12055
##  9 MIA     11728
## 10 DCA      9705
## # i 36 more rows
```

**Question 5.** Using the nycflights13 dataset, find the flight information (flight number, origin, destination, carrier, number of flights in the year, and percent late) for the flight numbers with the highest percentage of arrival delays. Only include the flight numbers that have over 100 flights in the year.

```
flights %>%
group_by(flight, origin, dest, carrier) %>%
summarize(AmmountOfFlights = n(),Delayed = sum(arr_delay > 0, na.rm = TRUE)) %>%
filter(AmmountOfFlights > 100) %>%
mutate(DelayPerc = (Delayed / AmmountOfFlights)) %>%
arrange(desc(DelayPerc))
```

```
## `summarise()` has grouped output by 'flight', 'origin', 'dest'. You can
## override using the `.groups` argument.
```

```
## # A tibble: 1,114 x 7
## # Groups:   flight, origin, dest [1,113]
##    flight origin dest  carrier AmmountOfFlights Delayed DelayPerc
##     <int> <chr>  <chr> <chr>              <int>   <int>     <dbl>
##  1    425 JFK    TPA   B6                   101      81     0.802
##  2    985 LGA    TPA   B6                   170     132     0.776
##  3   3075 JFK    CVG   MQ                   162     115     0.710
##  4    527 EWR    MCO   B6                   311     214     0.688
##  5   1103 JFK    SJU   B6                   137      94     0.686
##  6   1201 JFK    FLL   B6                   139      95     0.683
##  7   3616 LGA    MSP   MQ                   127      86     0.677
##  8   4224 EWR    MKE   EV                   257     174     0.677
##  9    381 LGA    FLL   B6                   170     115     0.676
## 10   3433 JFK    DCA   MQ                   111      75     0.676
## # i 1,104 more rows
```