Intro to R for Epidemiologists

Lab 6 (2/19/15)

Many of these questions go beyond the information provided in the lecture. Therefore, you may need to use R help files and the internet to search for answers. Feel free to ask questions of the instructor, the TAs, or your classmates, but try to work through as much as you can independently.

For the lab, you are expected to create an R script (.R file in the R editor) with your code corresponding to each question. Begin each question with a commented line of code indicating the question. As an example:

```
# Jenna Krall

# Question 1.
head(iris)
```

Part 1. Manipulating Datasets with `dplyr`

We will use the `hflights` dataset in R. This dataset contains information on flights departing from Houston's airports in 2011 (Source: Bureau of Transporation, Research and Innovation Technology Administration). In this section, you should use functions from the `dplyr` package.

1. Install the `hflights` package and load the `hflights` dataset (Hint: After installing the package, use the command `data(hflights)` to load the dataset)
2. Create a data frame tbl from the `hflights` dataset (Hint: `?tbl_df`) and call this data frame tbl `flights`
3. Create a new dataset that only includes observations from August where the taxi in time was less than 15 minutes.
4. Using the dataset in (3.), create a new dataset with only the following variables: DayofMonth, DepTime, ArrTime, UniqueCarrier, ActualElapsedTime, DepDelay, ArrDelay, Origin, Dest, and Distance.
5. Sort the new dataset created in (4.) by descending delay in departure time.
6. Group your dataset in (5.) dataset by airline carrier.
   (a) How many unique carriers are included in this dataset?
   (b) How many observations are in each airline carrier group?
7. Calculate the mean arrival delay to each destination using your `flights` dataset from (2.).
8. For each day of the year, count the total number of flights in `flights` and sort in descending order.
9. Create a new variable named "late" that takes the value of 1 if departure delay $\geq$ 15 minutes and 0 if departure delay $<$ 15 minutes. What percentage of flights depart over 15 minutes late?

```
# Part 1 Load libraries
library(dplyr)
library(hflights)

# 1 Load the data
data(hflights)
head(hflights)
```

```
##      Year Month DayofMonth DayOfWeek DepTime ArrTime UniqueCarrier
## 5424 2011     1          1         6    1400    1500            AA
## 5425 2011     1          2         7    1401    1501            AA
## 5426 2011     1          3         1    1352    1502            AA
## 5427 2011     1          4         2    1403    1513            AA
## 5428 2011     1          5         3    1405    1507            AA
## 5429 2011     1          6         4    1359    1503            AA
##      FlightNum TailNum ActualElapsedTime AirTime ArrDelay DepDelay Origin
## 5424       428  N576AA                60      40      -10        0    IAH
## 5425       428  N557AA                60      45       -9        1    IAH
## 5426       428  N541AA                70      48       -8       -8    IAH
## 5427       428  N403AA                70      39        3        3    IAH
## 5428       428  N492AA                62      44       -3        5    IAH
## 5429       428  N262AA                64      45       -7       -1    IAH
##      Dest Distance TaxiIn TaxiOut Cancelled CancellationCode Diverted
## 5424  DFW      224      7      13         0                          0
## 5425  DFW      224      6       9         0                          0
## 5426  DFW      224      5      17         0                          0
## 5427  DFW      224      9      22         0                          0
## 5428  DFW      224      9       9         0                          0
## 5429  DFW      224      6      13         0                          0
```

```
# 2 Make a tbl_df
flights <- tbl_df(hflights)

# 3. Restrict to flights in august
aug <- filter(flights, Month == 8 & TaxiIn < 15)

# 4 Select some variables
aug <- select(aug, DayofMonth, DepTime, ArrTime, UniqueCarrier, ActualElapsedTime,
    DepDelay, ArrDelay, Origin, Dest, Distance)

# 5. Sort by descending Departure Delay
sort_aug <- arrange(aug, desc(DepDelay))


# 6 Group your dataset
fl_carrier <- group_by(sort_aug, UniqueCarrier)

# 6a How many groups
n_groups(fl_carrier)
```

```
## [1] 15
```

```
# 6b Size of each group
group_size(fl_carrier)
```

```
##  [1]  251   30   53 5681  137  153   88  206  366 1521  182  298 3824 6536
## [15]    1
```

```
# 7.  Group by destination
dest <- group_by(flights, Dest)
```

```r
# Find mean arrival delay for each destination
summarise(dest, avg_delay = mean(ArrDelay, na.rm = TRUE))
```

```
## Source: local data frame [116 x 2]
##
##    Dest  avg_delay
## 1   ABQ   7.226259
## 2   AEX   5.839437
## 3   AGS   4.000000
## 4   AMA   6.840095
## 5   ANC  26.080645
## 6   ASE   6.794643
## 7   ATL   8.233251
## 8   AUS   7.448718
## 9   AVL   9.973988
## 10  BFL -13.198807
## ..  ...         ...
```

```r
# 8 Group by month and day
days <- group_by(flights, Month, DayofMonth)
# Tally results
tally(days, sort = T)
```

```
## Source: local data frame [365 x 3]
## Groups: Month
##
##    Month DayofMonth   n
## 1      1          3 702
## 2      1          2 678
## 3      1         20 663
## 4      1         27 663
## 5      1         13 662
## 6      1          7 661
## 7      1         14 661
## 8      1         21 661
## 9      1         28 661
## 10     1          6 660
## ..   ...        ... ...
```

```r
# 9. Compute flights more than 15 minutes late departing
flight2 <- mutate(flights, late = 1 * (flights$DepDelay >= 15))
# Compute proportion of flights
table(flight2$late)/sum(table(flight2$late))
```

```
##
##         0         1
## 0.8100414 0.1899586
```

Part 2. Manipulating Datasets with `tidyr`

1. Download the file "artificialLongData.csv" from the course website.

2. Using the `tidyr` package, convert the dataset from wide-form to long-form data

```
library(tidyr)
long <- read.csv("artificialLongData.csv")
head(long)
```

```
##   id    t0    t1    t2    t3    t4    t5    t6    t7    t8    t9   t10
## 1 s1  3.82  2.42 -1.85 -2.05  1.01  1.56  0.34  0.52 -0.06 -1.09  0.44
## 2 s2 -4.88 -2.95 -2.38  3.73 -2.77  1.72 -0.99 -0.70  0.15  2.38 -0.72
## 3 s3  0.57 -0.86  1.46 -2.04 -1.18  4.89 -3.94  0.50  4.90 -0.52  1.52
## 4 s4 -3.83 -1.22  0.17 -0.22 -3.05 -0.74  1.93  1.85 -3.93 -2.91 -1.13
## 5 s5 -3.99 -2.82 -2.13 -4.29 -2.33 -0.79 -0.86 -6.23 -8.13 -8.53 -0.87
## 6 s6  6.67  5.87  8.45  4.80  3.78  4.18  4.16  2.10  3.14  3.46  5.49
```

```
long_data <- gather(long, key = time, value = val, t0:t10)
long_data <- arrange(long_data, id)
head(long_data)
```

```
##   id time   val
## 1 s1   t0  3.82
## 2 s1   t1  2.42
## 3 s1   t2 -1.85
## 4 s1   t3 -2.05
## 5 s1   t4  1.01
## 6 s1   t5  1.56
```