DataViz Homework 02 solutions

Your Name Here

Homework 02: Data manipulation, grammar of graphics

General instructions for homeworks:

- Make a new R Markdown file (.Rmd), referring to the assignment on the course Github page
- Change the heading to include your author name
- Save the R Markdown file (named as: [MikeID]-[Homework01].Rmd – e.g. "mlopez-Lab01.Rmd") to somewhere where you'll be able to access it later (zip drive, My Documents, Dropbox, etc)
- Your file should contain the code/commands to answer each question in its own code block, which will also produce plots that will be automatically embedded in the output file
- **Each answer must be supported by written statements (unless otherwise specified) as well as any code used**: In other words, if the answer is `24`, you should write "The answer is 24" (as opposed to just showing the code and output).
- Include the names of anyone you collaborated with at the top of the assignment

Problem 1

What can we learn from Harambe?

In the spring of 2016, Harambe, a gorilla at the Cincinnati Zoo, was killed after dragging a three-year-old boy who had climbed into the gorilla enclosure. In the months following his death, Harambe was remembered on social media in several strange fashions.

One specific article (link) looks at the ways in which Harambe was remembered, including a chart (link) showing common words associated with the gorilla (read the article for specifics)

- Describe the variables in the plot, and identify their types

The variables are word type (categorical) and number of mentions (continuous)

- Describe the seven characteristics in the grammar of graphics with respect to this chart

The data is tweets, the variables mapped to the graph are the word type and number of mentions. The layers are the bars (its a barplot), there is no scale and the coordinates reflect counts from 0 to 35. There are no facets and the theme reflects white background.

- Identify if there are any data-ink or distortion factors with respect to this chart.

We are in pretty good shape here, IMO. Arguably, the title is too bolded

Problem 2

The journal PLOS-ONE publishes some of it's better data visualization from academic journals, available at the link here:

http://journals.plos.org/plosone/browse/data_visualization

One specific article looks at the performance deterioration in online user sessions on Reddit. In other words, comments get worse over time. The chart shown here shows the score (higher is more pleasant, lower is more critical) based on comment index, where comment index is a 1 for the first comment, a 2 for the second comment, etc.

- Comment on the aesthetics of this graph

The x-axis is comment index and the y-axis is the change in score (which is averaged across different Reddit users)

- Comment on the data-ink ratio of this graph

There's a decent amount of wasted ink, including, but not limited to, parts of the background, the shapes, and the scale

- From a statistical perspective, one may remain skeptical that these differences are meaningful. What additional information might be useful as far as identifying if the two groups are different? And how could the authors have shown this information?

To start, the graph is relativeley unclear regarding its labels. This makes it more difficult to read and understand exactly what is being compared. One may remain skeptical of whether or not the blue and red lines are different given that there is no mention of sample size (how many of each comment are being compared?) nor of error. As a result, it is difficult to decipher if differences shown are meaningful

You can read the full journal article here: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0161636

Question 3

Return to Lab 02 and the flights data set.

A researcher is interested in analyzing the performance of all airlines which have at least 20000 departing flights from NYC. Our goal is to obtain the airline with the lowest average delay time (departure delay) as well as the highest proportion of flights that left on time.

```
airlines <- flights %>%
  mutate(on.time = arr_delay <= 0) %>%
  group_by(carrier) %>%
  summarise(count = n(), delay.ave = mean(arr_delay), on.time.ave = mean(on.time)) %>%
  filter(count >= 20000) %>%
  arrange(delay.ave)
airlines
```

You can identify airline codes by visiting `http://www.iata.org/publications/Pages/code-search.aspx`.

Describe each of the steps above, and consider how the order in which the steps are made is important - Why can the filter command not come before the summarise command, for example.

Using the flights data, we'll create a new data frame called ***airlines****. The first step is to make a new variable* ***on.time*** *which is a TRUE/FALSE based on if the flight arrived on time or earlier. Next, we group flights by each carrier. Within each of these groups, we obtain the number of flights (*`count`*), average delay (*`delay.ave`*), and on time percentage (*`on.time.ave`*). Next, we focus on carriers with at least 20,000 flights (these will be the bigger carriers), and arrange in order of average delay time. The final step prints this data frame*

Question 4

a. Find the five individual flights with the longest arrival delay times in the data. Find the five flights with the shortest arrival delay times in the data.

```
library(nycflights13); library(dplyr)
flights <- na.omit(flights)
flights %>%
  arrange(arr_delay) %>%
  head(5)
```

```
## # A tibble: 5 × 19
##    year month   day dep_time sched_dep_time dep_delay arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>
## 1  2013     5     7     1715           1729       -14     1944
## 2  2013     5    20      719            735       -16      951
## 3  2013     5     2     1947           1949        -2     2209
## 4  2013     5     6     1826           1830        -4     2045
## 5  2013     5     4     1816           1820        -4     2017
## # ... with 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>
```

```
flights %>%
  arrange(-arr_delay) %>%
  head(5)
```

```
## # A tibble: 5 × 19
##    year month   day dep_time sched_dep_time dep_delay arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>
## 1  2013     1     9      641            900      1301     1242
## 2  2013     6    15     1432           1935      1137     1607
## 3  2013     1    10     1121           1635      1126     1239
## 4  2013     9    20     1139           1845      1014     1457
## 5  2013     7    22      845           1600      1005     1044
## # ... with 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>
```

```
### Note: you can also use the `tail()` command
```

The five flights with the longest delay time (flights 51, 3535, etc) and shortest delay times (flights 193, 11, etc) are shown above.

b. Find the five flight numbers (`tailnum`) with the longest average arrival delay times.

```r
flights %>%
  group_by(tailnum) %>%
  summarise(ave.delay = mean(arr_delay)) %>%
  arrange(-ave.delay) %>%
  head(5)
```

```
## # A tibble: 5 × 2
##   tailnum ave.delay
##     <chr>     <dbl>
## 1  N844MH       320
## 2  N911DA       294
## 3  N922EV       276
## 4  N587NW       264
## 5  N851NW       219
```

The flight tail numbers are N844MH, . . . , N851NW.

c. Identify the total departure delay time and average departure delay time within each month.

```r
flights %>%
  group_by(month) %>%
  summarise(ave.delay = mean(dep_delay), total.delay = sum(dep_delay))
```

```
## # A tibble: 12 × 3
##    month ave.delay total.delay
##    <int>     <dbl>       <dbl>
## 1      1  9.985491      263597
## 2      2 10.760239      254060
## 3      3 13.164289      367310
## 4      4 13.849187      381739
## 5      5 12.891709      362618
## 6      6 20.725614      561146
## 7      7 21.522179      608927
## 8      8 12.570524      361478
## 9      9  6.630285      179084
## 10    10  6.233175      178381
## 11    11  5.420340      146192
## 12    12 16.482161      445348
```

The average departure delay and total departure delays are shown above.

d. Create a new variable, `nighttime`, which is an indicator (TRUE/FALSE) for whether or not the flight took place at 7:00 PM or later (`hour >= 19`), and estimate the average arrival delay time within both `nighttime` groups.

```r
flights %>%
  mutate(nighttime = (hour >= 19)) %>%   #Note: the parenthesis are not needed, but shown for clarity
  group_by(nighttime) %>%
  summarise(ave.delay = mean(arr_delay))
```

```
## # A tibble: 2 × 2
##   nighttime ave.delay
```

```
##       <lgl>     <dbl>
## 1     FALSE  5.065749
## 2      TRUE 16.885556
```

The average delay of nighttime flights is 16.9 minutes, and for daytime flights its roughly 5 minutes.