CUNY 607 HW5

Todd Weigel

October 2, 2016

Assignment Overview

This assignment was to get an understanding of "Tidy" data, both by manipulating data in a cluttered or "untidy" format as well as by using some of the newer R packages, tidy.r and dplyr.r.

Below is the code module containing functions to initialize the environment, as well as encapsulate some of the more cumbersome processing, including transforming the data into a "tidy" one, as well as some analysis functions.

```
initialize <-function()
{
    library(stringr)
    library(tidyr)
    library(dplyr)
}


getFlightDF <- function(filename)
{
    flights <- read.csv(filename) %>%      #use piping
        gather(destination, counts, -X, -X.1, na.rm = TRUE) %>%  #rotate the Destination Columns
        rename(delayed = X.1) %>%    # name the column with delayed or ontime indicators
                #second row col 1 should always have 1st row col 1 values, so use lag function to move
                #1row and assign to  rows 2,4,6.... (note is.na needed because 1st row from lag is na a
        mutate(airline= str_c(X, unlist(lapply(lag(X), function(x) {x[is.na(x)] <-"" ; x}))))

    flights$X <- NULL  #(easier than listing column names in mutate)
    flights <- flights[c("airline", "delayed", "destination", "counts")]  #just rearrange columns
    flights$delayed <- (!flights$delayed == "on time")                    # set delayed column to bool
    return(flights)
}


getMeanFlights <- function(df, bDelayed = TRUE)
{
    return(df%>%
        filter(delayed == bDelayed) %>%
        summarise(mean(counts)))
}


getWorstDestinationByCount <- function(df)
{
    byDest <- df %>% group_by(destination)  %>%
```

```
                    filter(delayed == TRUE) %>%
                    summarise(cityCounts = sum(counts))
    return(byDest[which.max(byDest$cityCounts),])
}

getBestDestinationByCount <- function(df)
{
    byDest <- df %>% group_by(destination)  %>%
        filter(delayed == TRUE) %>%
        summarise(cityCounts = sum(counts))
    return(byDest[which.min(byDest$cityCounts),])
}



getBestWorstDestinationByPctDelayed <- function(df, worst = TRUE)
{
    grouped <- df %>% group_by(destination)
    delayed <- grouped %>% filter(delayed == TRUE) %>%
                           summarise(cityCounts = mean(counts))
    notDelayed <- grouped %>% filter(delayed == FALSE) %>%
                              summarise(cityCounts = mean(counts))
    pctDelayed <- (delayed[,2]/notDelayed[,2])

    if (worst)
    {
        delayed$destination[which(pctDelayed == max(pctDelayed))]
    }
    else
    {
        delayed$destination[which(pctDelayed == min(pctDelayed))]
    }
}
```

Lets first look at dataset before tidying:

```
read.csv("flights.csv")
```

```
##         X     X.1 Los.Angeles Phoenix San.Diego San.Francisco Seattle
## 1  Alaska on time         497     221       212           503    1841
## 2         delayed          62      12        20           102     305
## 3                          NA      NA        NA            NA      NA
## 4 AM WEST on time         694    4840       383           320     201
## 5         delayed         117     415        65           129      61
```

And as you can observe, it is a mess. Now lets initialize our environment, run the function to tidy the data, and then look at the tidy dataset:

```r
initialize()
```

```
##
## 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
```

```r
fDf <- getFlightDF("flights.csv")
fDf
```

```
##    airline delayed   destination counts
## 1   Alaska   FALSE   Los.Angeles    497
## 2   Alaska    TRUE   Los.Angeles     62
## 3  AM WEST   FALSE   Los.Angeles    694
## 4  AM WEST    TRUE   Los.Angeles    117
## 5   Alaska   FALSE       Phoenix    221
## 6   Alaska    TRUE       Phoenix     12
## 7  AM WEST   FALSE       Phoenix   4840
## 8  AM WEST    TRUE       Phoenix    415
## 9   Alaska   FALSE     San.Diego    212
## 10  Alaska    TRUE     San.Diego     20
## 11 AM WEST   FALSE     San.Diego    383
## 12 AM WEST    TRUE     San.Diego     65
## 13  Alaska   FALSE San.Francisco    503
## 14  Alaska    TRUE San.Francisco    102
## 15 AM WEST   FALSE San.Francisco    320
## 16 AM WEST    TRUE San.Francisco    129
## 17  Alaska   FALSE       Seattle   1841
## 18  Alaska    TRUE       Seattle    305
## 19 AM WEST   FALSE       Seattle    201
## 20 AM WEST    TRUE       Seattle     61
```

Much better looking. . . as one can see, there is one observation per row of data, basically that observation consists of the counts of an airline's flights for a given destination city, and whether the count is on-time flights or delayed flights.

The code in that get function was as follows. First a the reading of the file with read.csv. Then the "gather" function was use to basically rotate the 5 city columns and their corresponding counts to be rows of data rather than columns. Next some renames occurred, to add readability. Next we used the lag function to populate the now "missing" airline info in every other row. Lastly, some rearranging of the columns was done, as well as changing the "on-time" and "delayed" text to instead be boolean TRUE or FALSE values. Let's now call some of the functions for analysis:

```r
getBestDestinationByCount(fDf)
```

```
## # A tibble: 1 x 2
##   destination cityCounts
##         <chr>      <int>
## 1   San.Diego         85
```

```r
getWorstDestinationByCount(fDf)
```

```
## # A tibble: 1 x 2
##   destination cityCounts
##         <chr>      <int>
## 1     Phoenix        427
```

Those returned the best and worst destinations by total delayed flights. That was done by using the group by function, a filter on rows with delays, getting the sum of counts, and returning the max or min as relevant.

```r
getBestWorstDestinationByPctDelayed(fDf, TRUE)
```

```
## [1] "San.Francisco"
```

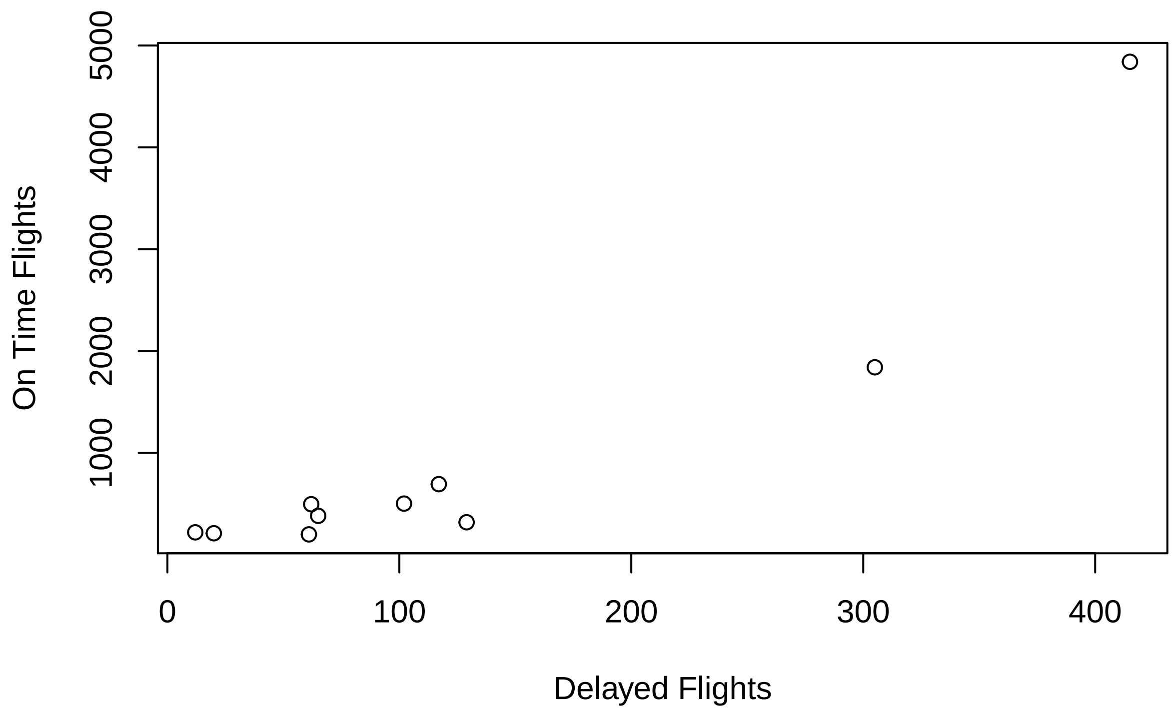
```r
getBestWorstDestinationByPctDelayed(fDf, FALSE)
```

```
## [1] "Phoenix"
```

This returned either the best or worst destination by the percentage of delayed flights, which is probably a more useful metric. This required similar techniques to the others, with the addition of having to divide by grouped sums and then finding the appropriate subsetted row.

Here is a plot of the ontime versus delayed flights, showing that flights are much more likely to be ontime versus delayed.

```r
plot(subset(fDf, delayed==TRUE)[,"counts"], subset(fDf, delayed==FALSE)
     [,"counts"], xlab = "Delayed Flights", ylab = "On Time Flights")
```



##Some toughts regarding "tidy"" data. Clearly, it does take some work to make one's data neat, but as many a mother or teacher has likley said, preparation is worth it's weight in gold. It would have been far more difficult to try and accomplish any kind of analysis on the original data (even the simple analysis done here), and so this is a skill that will need to be well learned. One thing "sort of" learned, but still needing much research is the new "tibble" object that some of the dplyr functions return...that was a surprise and actually made some of the work slightly harder, as there was just a lack of knowledge as to what one should do with it.

Lastly, so as not to take up space, the original data set was expanded to seven cities and three airlines to test that the "tidying" of the data was at least a little scalable. Below is the load and view of that bigger dataset, as well as the data.frame that it was "tidied"" into.

```
read.csv("flightsbig.csv")
```

```
##                X     X.1 Los.Angeles Phoenix San.Diego San.Francisco Seattle
## 1       Alaska on time         497     221       212           503    1841
## 2              delayed          62      12        20           102     305
## 3                               NA      NA        NA            NA      NA
## 4      AM WEST on time         694    4840       383           320     201
## 5              delayed         117     415        65           129      61
## 6                               NA      NA        NA            NA      NA
## 7 Flights R Us on time         442    3083       244           204     128
## 8              delayed          75     264        41            82      39
```

```
##   Tokyo New.York
## 1   500      305
## 2    20       55
## 3    NA       NA
## 4   438       40
## 5    21       10
## 6    NA       NA
## 7   279       25
## 8    13        6
```

```r
getFlightDF("flightsbig.csv")
```

```
##         airline delayed   destination counts
## 1        Alaska   FALSE   Los.Angeles    497
## 2        Alaska    TRUE   Los.Angeles     62
## 3       AM WEST   FALSE   Los.Angeles    694
## 4       AM WEST    TRUE   Los.Angeles    117
## 5  Flights R Us   FALSE   Los.Angeles    442
## 6  Flights R Us    TRUE   Los.Angeles     75
## 7        Alaska   FALSE       Phoenix    221
## 8        Alaska    TRUE       Phoenix     12
## 9       AM WEST   FALSE       Phoenix   4840
## 10      AM WEST    TRUE       Phoenix    415
## 11 Flights R Us   FALSE       Phoenix   3083
## 12 Flights R Us    TRUE       Phoenix    264
## 13       Alaska   FALSE     San.Diego    212
## 14       Alaska    TRUE     San.Diego     20
## 15      AM WEST   FALSE     San.Diego    383
## 16      AM WEST    TRUE     San.Diego     65
## 17 Flights R Us   FALSE     San.Diego    244
## 18 Flights R Us    TRUE     San.Diego     41
## 19       Alaska   FALSE San.Francisco    503
## 20       Alaska    TRUE San.Francisco    102
## 21      AM WEST   FALSE San.Francisco    320
## 22      AM WEST    TRUE San.Francisco    129
## 23 Flights R Us   FALSE San.Francisco    204
## 24 Flights R Us    TRUE San.Francisco     82
## 25       Alaska   FALSE       Seattle   1841
## 26       Alaska    TRUE       Seattle    305
## 27      AM WEST   FALSE       Seattle    201
## 28      AM WEST    TRUE       Seattle     61
## 29 Flights R Us   FALSE       Seattle    128
## 30 Flights R Us    TRUE       Seattle     39
## 31       Alaska   FALSE         Tokyo    500
## 32       Alaska    TRUE         Tokyo     20
## 33      AM WEST   FALSE         Tokyo    438
## 34      AM WEST    TRUE         Tokyo     21
## 35 Flights R Us   FALSE         Tokyo    279
## 36 Flights R Us    TRUE         Tokyo     13
## 37       Alaska   FALSE      New.York    305
## 38       Alaska    TRUE      New.York     55
## 39      AM WEST   FALSE      New.York     40
## 40      AM WEST    TRUE      New.York     10
## 41 Flights R Us   FALSE      New.York     25
```

```
## 42 Flights R Us    TRUE       New.York      6
```