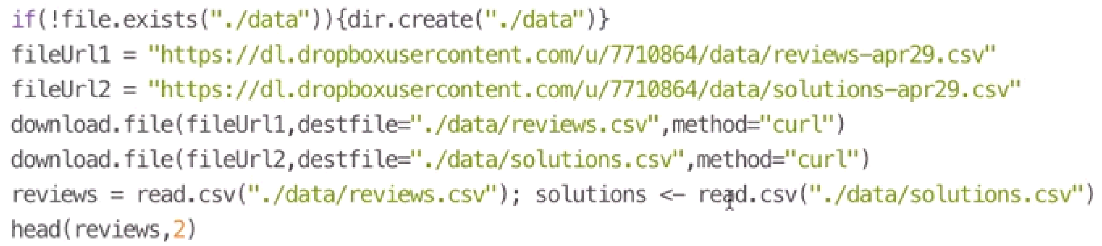
Getting Data

Sunday, December 20, 2015 7:33 PM

- Directory set: setwd(../data)
- Directory up: setwd(../)
- Create directory : file.exist()
- Direct.create("data")
- Download.file()
    - Url,destfile,csv
    - While downloading, store date of download also
- Flat files : read.table
    - Set quote argument so that r won't get confused with it



```
if(!file.exists("./data")){dir.create("./data")}
fileUrl1 = "https://dl.dropboxusercontent.com/u/7710864/data/reviews-apr29.csv"
fileUrl2 = "https://dl.dropboxusercontent.com/u/7710864/data/solutions-apr29.csv"
download.file(fileUrl1,destfile="./data/reviews.csv",method="curl")
download.file(fileUrl2,destfile="./data/solutions.csv",method="curl")
reviews = read.csv("./data/reviews.csv"); solutions <- read.csv("./data/solutions.csv")
head(reviews,2)
```

- Read.xlsx() can read excel
    - XLConnect also can be used
    - It's possible to read specific row and column
- Write.xlsx()
- Reading xml
    - Library xml
    - XmlTreeParse()
    - XmlRoot()
    - XmlName()
    - Rootnode[[1][12]]
    - XmlSApply()
- Xml
    - XpathSApply()
- Html
    - HtmlTreeParse
    - Read the course notes for more
- Json
    - Library jsonlite
    - A<-Fromjason () for getting from a picture
    - Names(A)
    - $ can be used to have inside nested data
    - Tojson () convert to Json
- Data.table
    - Library (data.table)
    - Tables()
    - Copy function should be used to create copy of a variable. No assignment
    - Multi step operation is possible
    - .N is for count
    - Setkey, merge can be used to join tables
    - This has fast read from file and is memory efficient

Analyzing Data

- Use which command to subset when the data contains NA
- Order and Sort is useful to subset

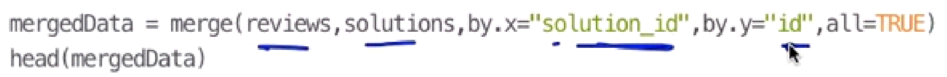
Summarize Data

- Head() to find the head
- Tail() to see bottom
- Summary() to find overall summary
- Str() to find the extra information
- Quantile() to find different percentage
- Table() for list values, not for data frame, useNa='ifAny') is important
  - Tables can also create 2D values of 2 different lists which is helpful for the analysis
- Sum(is.na(list)) can be used to find NA
- Any(is.na(data)) can be used to find NA too
- Colsums(is.na(dataframe)) can be used for data frame
- All(Colsums(is.na(dataframe))) will be TRUE if no NA
- Table(DF$list %in% c("21212","32121")) to find number of existence
- DF[DF$list %in% c("21212","32121"),] to filter the data
- Xtabs can be used to find the cross tab relation with different varibles
  - Xtab(Freq~Gender+Admit,data=DF) will show the freq of gender admitted and rejected)
  - ?Check breaks and ftable() for flat table
- Size_byte<-Object.size(DF) will give size. Then print(size,units="Mb")
- To check a condition that satisfies the condition
  - Ifelse(DF$list <0,TRUE,FALSE)
  - Table(that data)
- Create category variable using cut
  - a<-Cut(DF$list,breaks=quantile(DF$list1)
  - Table(a)
  - Library hmisc has cut2 which has quantiles defined by default
    - Mutate will add thses variables into DF
- Abs(),sqrt(),ceiling(),floor(),signif()

Reshaping The Data

- Library reshape
- Melt function creates a DF which has ID and variables .Separate rows are created for each variables
- Dcast(DF,list1~variable) will show count
- Dcast(DF,list1~variable,mean) will show mean
- Tapply can be used to math based on a factor variable.
  - Tapply(list1,factorList2,mean)

Merging Data



```
mergedData = merge(reviews,solutions,by.x="solution_id",by.y="id",all=TRUE)
head(mergedData)
```

Arrange in plyr can be used also.

Dplr package

plyr() provides a consistent and concise Grammer for manipulating tabular data

tbl_df() -> Create data frame table

Subset column

select(cran,ip_id,package,country) - Selecting the data columns

select(cran, r_arch:country) - Using : to select with values

select(cran, -time) : Print in reverse order

select(DF,list:Value)

Subset row

filter(cran, package == "swirl") - select all rows for which the package variable is equal to "swirl"

filter(cran,size>100500, r_os =="linux-gnu") - Both condition

Arrange column

arrange(cran2, ip_id)

arrange(cran2, desc(ip_id))

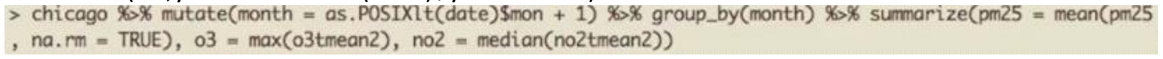
arrange(cran2, package, ip_id) - Multiple arrange columns

desc() can be used for decesing order

Mutate - Create a new column based on another column

mutate(cran3,size_mb = size/2^20)

mutate(DF,year=as.POSIXlt(List1)$year + 1990)



```
> chicago %>% mutate(month = as.POSIXlt(date)$mon + 1) %>% group_by(month) %>% summarize(pm25 = mean(pm25
, na.rm = TRUE), o3 = max(o3tmean2), no2 = median(no2tmean2))
```

Summarize

summarize(cran, avg_bytes = mean(size))

summarize(DF,list1=mean(list1,na.rm=TRUE))

Split

Ddply(DF,.(list),summarize,sum=ave(count,fun=sum))

Rename

rename(DF,newname=oldname)

Group

newDF<- group_by(DF,groupingList)

Cleaning Data

- Editing Texts
  - Tolower and toupper can be used to convert
  - Strsplit: string split, strsplit(names(DF),"\\.")
  - Sub to subsitiute the first occurance but gsub to all
  - Grep("value",list) to find the index of the find. Argument value=true will gives the values
  - Grepl gives logical output
  - Useful package stringr
  - Nchar for size, substr for subset, paste and paste0, str_trim for removing data
- Regular Expressions
  - ^ for start of line
  - $ for end of the line
  - [Bb] [Uu][Ss][Hh] for all occurance of Bush
  - [0-9][a-zA-Z] for 9th or 3am
  - [^?.]$ any line that ends without .
  - . Means any character
  - | or
  - +([a-zA-Z]+) +\1 + -> gives the repeating words
- Cleaning Date
  - Library lubridate

Project

Monday, December 28, 2015 10:55 PM