DA5020 ~ Practicum 1 ~ Prasad Raut

# Task 1

(0 pts) Download the PubMed excerpt data set (XML). Load the XML file into a browser or text editing tool and inspect it. Explore the data set as you see fit and that allows you to get a sense of the data and get comfortable with it.

# Installing the required packages using install.packages() function from the console and loading those packages into the R session using library() function.  
library(XML)  
library(RCurl)  
library(tidyverse)  
library(plyr)

# Using getURL() function to download the XML file from a URL.  
xData <- getURL("https://da5020.weebly.com/uploads/8/6/5/9/8659576/pubmedsample.jun18.xml")  
# Using xmlParse() function to extract data from an XML file and store it as an XML document named pubmed in R. Setting the root node for pubmed using xmlRoot() function.  
pubmed<- xmlRoot(xmlParse(xData), useInternalNodes = TRUE)   
# Examining the first record from the pubmed XML document.  
pubmed[[1]]

## <PubmedArticle>  
## <MedlineCitation Status="MEDLINE" Owner="NLM">  
## <PMID Version="1">973217</PMID>  
## <DateCompleted>  
## <Year>1976</Year>  
## <Month>12</Month>  
## <Day>03</Day>  
## </DateCompleted>  
## <DateRevised>  
## <Year>2002</Year>  
## <Month>11</Month>  
## <Day>13</Day>  
## </DateRevised>  
## <Article PubModel="Print">  
## <Journal>  
## <ISSN IssnType="Print">0095-3814</ISSN>  
## <JournalIssue CitedMedium="Print">  
## <Volume>3</Volume>  
## <Issue>1</Issue>  
## <PubDate>  
## <Year>1976</Year>  
## <Season>Fall</Season>  
## </PubDate>  
## </JournalIssue>  
## <Title>Topics in health care financing</Title>  
## <ISOAbbreviation>Top Health Care Financ</ISOAbbreviation>  
## </Journal>  
## <ArticleTitle>Hospital debt management and cost reimbursement.</ArticleTitle>  
## <Pagination>  
## <MedlinePgn>69-81</MedlinePgn>  
## </Pagination>  
## <AuthorList CompleteYN="Y">  
## <Author ValidYN="Y">  
## <LastName>Blume</LastName>  
## <ForeName>F R</ForeName>  
## <Initials>FR</Initials>  
## </Author>  
## </AuthorList>  
## <Language>eng</Language>  
## <PublicationTypeList>  
## <PublicationType UI="D016428">Journal Article</PublicationType>  
## </PublicationTypeList>  
## </Article>  
## <MedlineJournalInfo>  
## <Country>United States</Country>  
## <MedlineTA>Top Health Care Financ</MedlineTA>  
## <NlmUniqueID>7509107</NlmUniqueID>  
## <ISSNLinking>0095-3814</ISSNLinking>  
## </MedlineJournalInfo>  
## <CitationSubset>IM</CitationSubset>  
## <MeshHeadingList>  
## <MeshHeading>  
## <DescriptorName UI="D000066" MajorTopicYN="N">Accounting</DescriptorName>  
## </MeshHeading>  
## <MeshHeading>  
## <DescriptorName UI="D004469" MajorTopicYN="Y">Economics, Hospital</DescriptorName>  
## </MeshHeading>  
## <MeshHeading>  
## <DescriptorName UI="D006739" MajorTopicYN="Y">Hospital Administration</DescriptorName>  
## </MeshHeading>  
## <MeshHeading>  
## <DescriptorName UI="D014481" MajorTopicYN="N" Type="Geographic">United States</DescriptorName>  
## </MeshHeading>  
## </MeshHeadingList>  
## </MedlineCitation>  
## <PubmedData>  
## <History>  
## <PubMedPubDate PubStatus="pubmed">  
## <Year>1976</Year>  
## <Month>1</Month>  
## <Day>1</Day>  
## </PubMedPubDate>  
## <PubMedPubDate PubStatus="medline">  
## <Year>1976</Year>  
## <Month>1</Month>  
## <Day>1</Day>  
## <Hour>0</Hour>  
## <Minute>1</Minute>  
## </PubMedPubDate>  
## <PubMedPubDate PubStatus="entrez">  
## <Year>1976</Year>  
## <Month>1</Month>  
## <Day>1</Day>  
## <Hour>0</Hour>  
## <Minute>0</Minute>  
## </PubMedPubDate>  
## </History>  
## <PublicationStatus>ppublish</PublicationStatus>  
## <ArticleIdList>  
## <ArticleId IdType="pubmed">973217</ArticleId>  
## </ArticleIdList>  
## </PubmedData>  
## </PubmedArticle>

# Task 2

(30 pts) Load the data into R and create two linked tibbles: one for publication and one for journal. Use ISSN as the key to link them. Only load the following information into the publication tibble: PMID (primary key for publication), ISSN and publication year (foreign key for journal), date completed (as one date field), date revised (as one date field), number of authors (derived/calculated field from the authors), publication type, title of article. Load this information into the journal tibble: ISSN (primary key), medium (from CitedMedium attribute), publication year (primary key), publication season, language, and journal title. In cases where there are multiple languages for a publication, pick the first language. Same for publication type: pick the first one. The primary key for journal is (ISSN, publication year). Also, exclude any journals that do not have an ISSN as the primary key cannot be empty.

# Using function() function to create the xmlToColumn() function which takes an argument named path containin the path of node in XML document. This function will be used to extract the data from the pubmed xml document into columns for creating tibbles.  
xmlToColumn <- function(path){  
 # Using xpathSApply() function to set the path to the nodes from pubmed XML document.   
 xpathSApply(pubmed, "//PubmedArticle/MedlineCitation", function(x) {  
 # Using length() function with if statement to check if the path exists.  
 if(length(xpathSApply(x, path)) != 0) {  
 # Using xpathSApply() function to get the value from the XML node specified by the path.  
 xpathSApply(x, path, xmlValue)  
 }   
 # Using else statement to put NA values if the node is missing.  
 else {  
 NA  
 }  
 })  
}  
  
# Creating the publication tibble:  
  
# Using xpathSApply() to get the PMID values from the nodes specified by the given path. Converting the PMID values to integer using as.integer() function.  
PMID <- as.integer(xpathSApply(pubmed, '//PubmedArticle/MedlineCitation/PMID', xmlValue, simplify = TRUE))  
  
# Creating the ISSN variable using xmlToColumn() function.  
ISSN <- xmlToColumn("./Article/Journal/ISSN")  
# Creating the Publication\_year variable using xmlToColum() function.  
Publication\_year <- xmlToColumn("./Article/Journal/JournalIssue/PubDate/Year")  
# Combining the ISSN and the Publication\_year to form a foreign key for publication tibble and a primary key for journal tibble using paste() function.  
ISSN <- paste(ISSN, Publication\_year, sep = ".")   
  
# Creating the Date\_completed variable using xmlToColumn() function.   
Date\_completed <- xmlToColumn("./DateCompleted")  
# Using str\_replace() function to add "-" between year, month, and day of DateComplete values. Using as.Date() function to convert the resulting vector into type date and storing the result back in DateComplete.  
Date\_completed <- as.Date(str\_replace(Date\_completed,"(\\d{4})(\\d{2})(\\d{2})$","\\1-\\2-\\3"))  
  
# Creating the Date\_revised variable using xmlToColumn() function.  
Date\_revised <- xmlToColumn("./DateRevised")  
# Using str\_replace() function to add "-" between year, month, and day of DateRevised values. Using as.Date() function to convert the resulting vector into type date and storing the result back in DateRevised.  
Date\_revised <- as.Date(str\_replace(Date\_revised,"(\\d{4})(\\d{2})(\\d{2})$","\\1-\\2-\\3"))  
  
# Using xpathSApply() function to set the path to the nodes from pubmed XML document to get the number of authors.  
Number\_of\_authors <- xpathSApply(pubmed, "//PubmedArticle/MedlineCitation", function(x) {  
 # Using length() function with if statement to check if the path exists for getting the authors.  
 if(length(xpathSApply(x, "./Article/AuthorList")) != 0) {  
 # Using xpathSApply() function to get the authors from the Author XML node specified by the path. Using the count() from plyr package and nrow() functions to count the number of authors in each node.  
 nrow(plyr::count(xpathSApply(x, "./Article/AuthorList/Author",xmlValue)))  
 }  
 # Using else statement to put NA values if the Authors are missing.  
 else {  
 NA  
 }  
})  
  
# Creating the Publication\_type variable using xmlToColumn() function.  
Publication\_type <- xmlToColumn("./Article/PublicationTypeList/PublicationType")  
# Using sapply() function to extract only the first element of all elements in the list Publication\_type and storing it back as a vector.  
Publication\_type <- sapply(1:length(Publication\_type), function(x) Publication\_type[[x]][1])  
  
# Creating the Article\_title variable using xmlToColumn() function.  
Article\_title <- xmlToColumn("./Article/ArticleTitle")  
# Using data\_frame() function to add the specified variables as columns to the publication tibble.  
(publication <- data\_frame(PMID, ISSN, Date\_completed, Date\_revised, Number\_of\_authors, Publication\_type, Article\_title))

## # A tibble: 177 x 7  
## PMID ISSN Date\_completed Date\_revised Number\_of\_authors  
## <int> <chr> <date> <date> <int>  
## 1 973217 0095-3814.1976 1976-12-03 2002-11-13 1  
## 2 1669026 0377-8231.1991 1993-11-15 2008-11-21 NA  
## 3 1875346 0022-2623.1991 1991-09-25 2013-11-21 10  
## 4 5757641 NA.1968 1970-03-22 2003-11-14 1  
## 5 3549656 0021-8820.1987 1987-05-15 2014-11-20 6  
## 6 8119288 0014-2956.1994 1994-04-01 2016-10-17 2  
## 7 8219565 1051-0443.NA 1993-12-08 2016-11-23 4  
## 8 8655018 0017-0011.1996 1996-07-30 2017-03-03 2  
## 9 8863847 0026-895X.1996 1996-11-25 2014-11-20 5  
## 10 9110943 1059-2725.1996 1996-10-23 2011-11-17 4  
## # ... with 167 more rows, and 2 more variables: Publication\_type <chr>,  
## # Article\_title <chr>

# Using View() function to display the created publication tibble in the R Data Viewer.  
View(publication)

# Creating the Journal tibble:  
  
# Using xpathSApply() function to set the path to the nodes from pubmed XML document to get the Medium of the journals.  
Medium <- xpathSApply(pubmed, "//PubmedArticle/MedlineCitation", function(x) {  
 # Using length() function with if statement to check if the path exists for getting the Medium from the Journal Issue nodes.  
 if(length(xpathSApply(x, "./Article/Journal/JournalIssue")) != 0) {  
 # Using xpathSApply() function to get the attribute CitedMedium from the JournalIssue XML node specified by the given path.  
 xpathSApply(x, "./Article/Journal/JournalIssue", xmlGetAttr, "CitedMedium")  
 }   
 # Using else statement to put NA values if the Authors are missing.  
 else {  
 NA  
 }  
})  
  
# Creating the Publication\_season variable using xmlToColumn() function.  
Publication\_season <- xmlToColumn("./Article/Journal/JournalIssue/PubDate/Season")  
  
# Creating the Language variable using xmlToColumn() function.  
Language <- xmlToColumn("./Article/Language")  
# Using sapply() function to extract only the first element of all elements in the variable Language and storing it back as a vector.  
Language <- sapply(1:length(Language), function(x) Language[[x]][1])  
  
# Creating the Journal\_title variable using xmlToColumn() function.  
Journal\_title <- xmlToColumn("./Article/Journal/Title")  
  
# Using data\_frame() function to add the specified variables as columns to the journal tibble.   
journal <- data\_frame(ISSN, Medium, Publication\_year, Publication\_season, Language, Journal\_title)  
# Removing all the ISSN values which are NA using filter() with substring() function and eliminating the duplicate rows from the journal tibble using the unique() function.  
(journal <- base::unique(filter(journal, substring(ISSN, 1, 2) != "NA")))

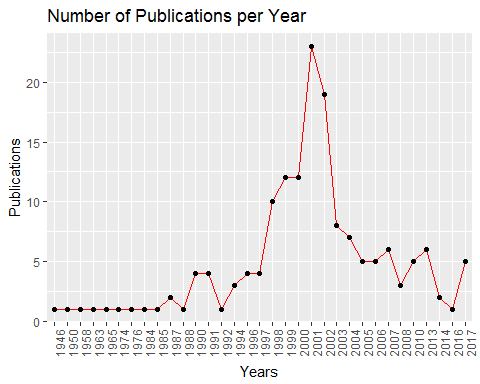
## # A tibble: 150 x 6  
## ISSN Medium Publication\_year Publication\_sea~ Language Journal\_title   
## <chr> <chr> <chr> <chr> <chr> <chr>   
## 1 0095-~ Print 1976 Fall eng Topics in hea~  
## 2 0377-~ Print 1991 <NA> fre Bulletin et m~  
## 3 0022-~ Print 1991 <NA> eng Journal of me~  
## 4 0021-~ Print 1987 <NA> eng The Journal o~  
## 5 0014-~ Print 1994 <NA> eng European jour~  
## 6 1051-~ Print <NA> <NA> eng Journal of va~  
## 7 0017-~ Print 1996 <NA> pol Ginekologia p~  
## 8 0026-~ Print 1996 <NA> eng Molecular pha~  
## 9 1059-~ Print 1996 <NA> eng The Online jo~  
## 10 0009-~ Print 1996 <NA> eng Circulation   
## # ... with 140 more rows

# Using View() function to display the created journal tibble in the R Data Viewer.  
View(journal)

# Task 3

(10 pts) Create a line graph of the number of publications per year.

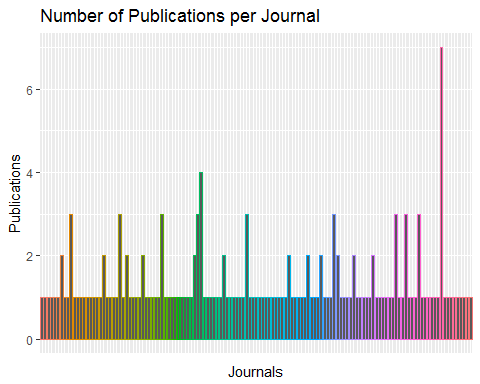
# Using inner\_join() function to join the journal tibble with publication tibble by ISSN variable and associate publication years with their respective Publications.  
lg <- inner\_join(publication, journal, by = "ISSN")   
# Using table() function to get the frequency of publications each year. Converting the table to data frame named linegraph using as.data.frame() function.  
linegraph <- as.data.frame(table(lg$Publication\_year))  
  
# Using ggplot() function to create a line graph of number of publications per year for the linegraph data frame.  
ggplot(data = linegraph, aes(x = Var1, y = Freq, group = 1)) +  
 geom\_line(color = "red") + geom\_point() +   
 ggtitle("Number of Publications per Year") + xlab("Years") + ylab("Publications") +   
 theme(axis.text.x = element\_text(angle = 90, hjust = 1))



# Task 4

(10 pts) Create a histogram of the number of publication per journal.

# Using inner\_join() function to join the journal tibble with publication tibble by ISSN variable and associate publications with their respective journal titles.   
hist<- inner\_join(publication, journal, by = "ISSN")   
  
# Using ggplot() and geom\_bar() functions to create a histogram of the number of publication per journal.  
ggplot(hist, aes(x = Journal\_title, color = Journal\_title)) + geom\_bar() + ggtitle("Number of Publications per Journal") + xlab("Journals") + ylab("Publications") + theme(axis.text.x = element\_blank(), axis.ticks.x = element\_blank(), legend.position = "none")



# Task 5

(10 pts) Find the article that had the most number of authors and list the article, journal, and number of authors.

# Using filter() function to get the article having the most number of authors.  
filter(publication, Number\_of\_authors == max(Number\_of\_authors, na.rm = T)) %>%  
 # Using inner\_join() function to join the journal tibble with publication tibble by ISSN variable and associate article titles with their respective journal titles.  
 inner\_join(journal, by = "ISSN") %>%  
 # Using select() function to list the columns containing article, journal, and number of authors.  
 select("Article" = Article\_title, "Journal" = Journal\_title, "Number of Authors" = Number\_of\_authors)

## # A tibble: 1 x 3  
## Article Journal `Number of Autho~  
## <chr> <chr> <int>  
## 1 Measurement of B --> K\*gamma branchin~ Physical revie~ 621

# Task 6

(10 pts) Find the average number of authors for articles. Display a single number.

# Using summarize() with mean() function to get the average number of authors for articles. Using round() function to display a single number.  
summarize(publication, "Average Number of Authors" = round(mean(Number\_of\_authors, na.rm = T), digits = 0))

## Average Number of Authors  
## 1 10

# Task 7

(10 pts) What is the shortest and the longest time periods between date completed and date revised. Display the time elapsed in days. Only consider cases where the difference is a positive number.

# Calculating the time difference in days between the date revised and date completed.  
timediff <- (publication$Date\_revised - publication$Date\_completed)  
# Using which() function to remove the time difference which is a negative number.   
timediff <- timediff[timediff > 0]  
# Using min() function to get the shortest time periods between date completed and date revised.   
min(timediff, na.rm = T)

## Time difference of 4 days

# Using max() function to get the longest time periods between date completed and date revised.  
max(timediff, na.rm = T)

## Time difference of 14755 days

# Task 8

(10 pts) In how many different languages are the articles published in PubMed? Note that this will actually give an incorrect answer because the tibble only contains the first language of publication.

# Using unique() function to get different languages of articles published in PubMed.  
(lang <- unique(journal$Language))

## [1] "eng" "fre" "pol" "spa" "rus" "swe" "dut" "chi" "ita" "kor"

# Using length() function to count the different languages of articles published in PubMed.  
length(lang)

## [1] 10

# Task 9

(20 pts) Using the XML data (not the tibbles created above), inspect the abstract text fields and find how many articles contained the words “in vitro”, “drug resistance”, “virus”, “transmission”, “clinical study” in any capitalization in any abstract section. Note that drug resistance could be spelled as “drug resistance” or “drug-resistance” or “drug resistant” or “drug resistent” – use regular expressions to deal with the variation.

# Creating the abstract list containing the abstracts for all publications using the xmlToColumn() function.   
abstract <- xmlToColumn("./Article/Abstract/AbstractText")  
# Using head() function to examine the first few elements of the list abstract.  
head(abstract)

## [[1]]  
## [1] NA  
##   
## [[2]]  
## [1] NA  
##   
## [[3]]  
## [1] "Modification of the hexahydronaphthalene ring 5-position in simvastatin 2a via oxygenation and oxa replacement afforded two series of derivatives which were evaluated in vitro for inhibition of 3-hydroxy-3-methylglutaryl-coenzyme A reductase and acutely in vivo for oral effectiveness as inhibitors of cholesterogenesis in the rat. Of the compounds selected for further biological evaluation, the 6 beta-methyl-5-oxa 10 and 5 alpha-hydroxy 16 derivatives of 3,4,4a,5-tetrahydro 2a, as well as, the 6 beta-epimer 14 of 16 proved orally active as hypocholesterolemic agents in cholestyramine-primed dogs. Subsequent acute oral metabolism studies in dogs demonstrated that compounds 14 and 16 evoke lower peak plasma drug activity and area-under-the-curve values than does compound 10 and led to the selection of 14 and 16 for toxicological evaluation."  
##   
## [[4]]  
## [1] NA  
##   
## [[5]]  
## [1] "The resistance of 6-[(R)-2-[3-(3,4-dihydroxybenzoyl)-3-(3-hydroxypropyl)-1-ureido]-2- phenylacetamido]penicillanic acid (1a) to metabolism by catechol-O-methyl-transferase (COMT) was increased by introduction of the chlorine atom into the catechol moiety. Penicillins (1b-1d) having one or two chlorine atoms at the positions adjacent to the hydroxyl group were found to have greater stability to COMT. This resulted in greater efficiency in vivo in experimental Pseudomonas aeruginosa and Escherichia coli infections. In vitro activities were essentially unchanged."  
##   
## [[6]]  
## [1] "Bacillus subtilis ferrochelatase is encoded by the hemH gene of the hemEHY gene cluster and catalyses the incorporation of Fe2+ into protoporphyrin IX. B. subtilis ferrochelatase produced in Escherichia coli was purified. It was found to be a monomeric, water-soluble enzyme of molecular mass 35 kDa which in addition to Fe2+ can incorporate Zn2+ and Cu2+ into protoporphyrin IX. Chemical modification experiments indicated that the single cysteine residue in the ferrochelatase is required for enzyme activity although it is not a conserved residue compared to other ferrochelatases. In growing B. subtilis, the ferrochelatase constitutes approximately 0.05% (by mass) of the total cell protein, which corresponds to some 600 ferrochelatase molecules/cell. The turnover number of isolated ferrochelatase, 18-29 min-1, was found to be consistent with the rate of haem synthesis in exponentially growing cells (0.2 mol haem formed/min/mol enzyme). It is concluded that the B. subtilis ferrochelatase has enzymic properties which are similar to those of other characterised ferrochelatases of known primary structure, i.e. ferrochelatases of the mitochondrial inner membrane of yeast and mammalian cells. However, in contrast to these enzymes the B. subtilis enzyme is a water-soluble protein and should be more amenable to structural analysis."

# Using str\_to\_lower() to convert all characters of strings to lower case.   
abstract <- str\_to\_lower(abstract)  
  
# Using str\_detect() function to find how many articles contained the word "in vitro".  
sum(str\_detect(abstract, "in vitro"))

## [1] 7

# Using str\_detect() function to find how many articles contained the words "drug resistance" or "drug-resistance" or "drug resistant" or "drug resistent".  
sum(str\_detect(abstract, "drug[ |-]resist[a|e]n[ce|t]"))

## [1] 2

# Using str\_detect() function to find how many articles contained the word "virus".  
sum(str\_detect(abstract, "virus"))

## [1] 2

# Using str\_detect() function to find how many articles contained the word "transmission".  
sum(str\_detect(abstract, "transmission"))

## [1] 1

# Using str\_detect() function to find how many articles contained the word "clinical study".  
sum(str\_detect(abstract, "clinical study"))

## [1] 0