CS5346 Mini-Project 2
Conference Information Visualizations

Student Name	Dong Yizhou	Li Yuanda
Matric Number	A0078006J	A0078501J

Introduction

According to research, number of science papers published had passed 50 million in 2009 with roughly 2.5 million new scientific papers published each year (Sarah Boon, 2016). Digging out meaningful information such as research topic trends, author ranks and article ranks can be a great challenge with such huge data source.

In this mini project, we use the first 200,000 lines from database of open research corpus with over 20 million published research papers in Computer Science, Neuroscience, and Biomedical as the data source to perform visualization and information digging. For this project we use both R and D3 to achieve the visualization. There are 2 main steps of our tasks:

1. Raw data preprocessing in preparation for the visualization
    a. Data format converting, to do the visualization in R, we managed to use the package rjson to convert the json object into dataframe before we proceed the visualization, while in D3 we proceed the visualization with json directly.
    b. Data cleaning, we noticed that there are quite a number of authors with no ids when we do the visualization for top authors. We dropped the rows with no author ids.
    c. Data aggregation, to calculate the count of authors and articles, we used the aggregation function in R.
2. Visualization with R and Javascript D3
    a. We use the library ggplot and plotly in R to perform the visualization. For Javascript we used D3.
    b. We want to do a cross comparison of the 2 programming language R and Javascript to understand the pros and cons of each language.

Visualizations

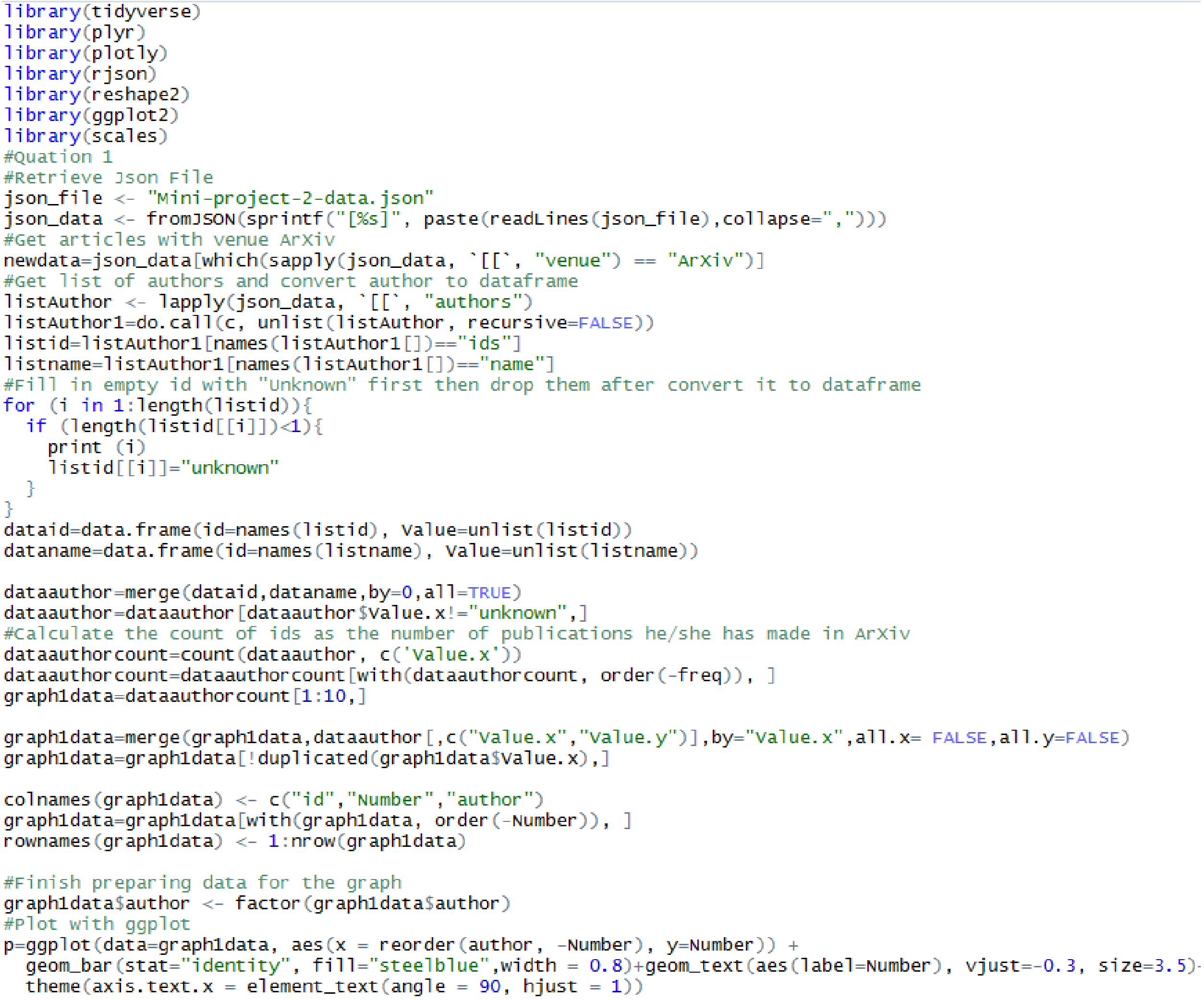
Objective	Visualization
1	Barchart
2	Barchart, Bubble Chart
3	Line Chart, Donut Chart

Objective 1 - Visualize the top 10 authors for venue arXiv

Visualization with R

Data Preparation:

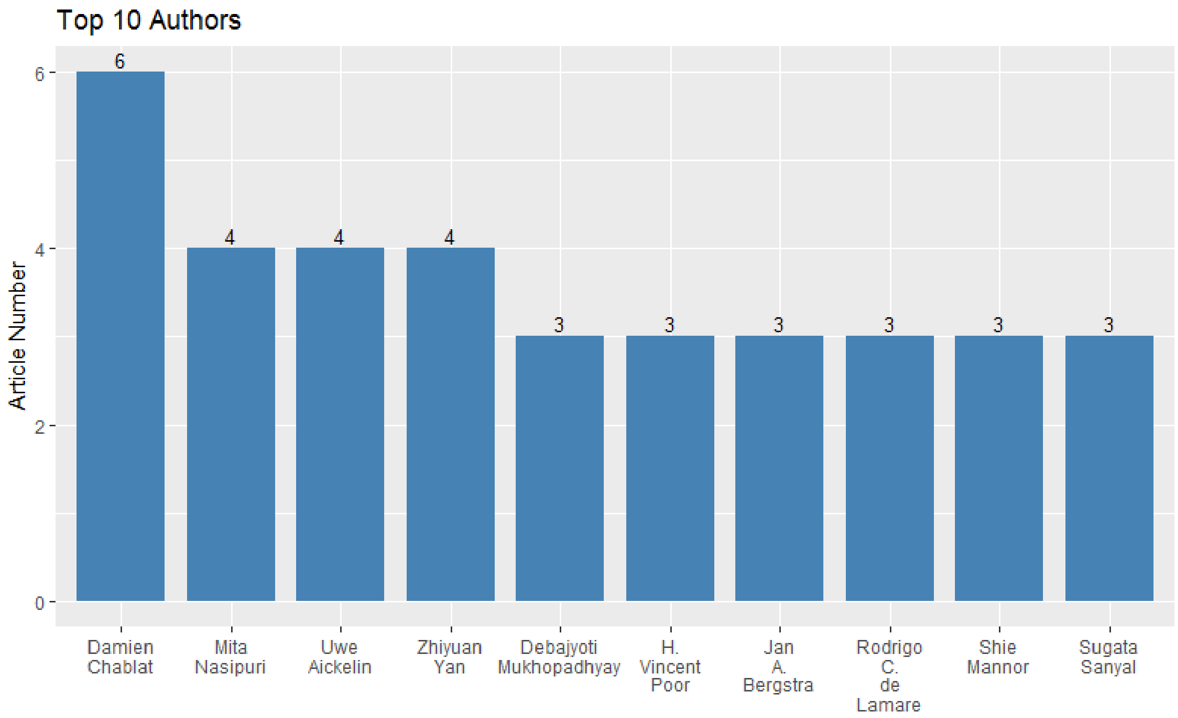
Library “rjson” was used to read in the json file and we retrieved the articles within venue arVix. Then we get the list of authors within the venue and dropped the authors with no ids. Since 1 author id will only be in 1 article once, so the total ids counts represent how many articles a particular author published. We get the id counts and match the id with the authors. After that, we use ggplot to plot the barchat with x axis as author names, y axis as article counts.



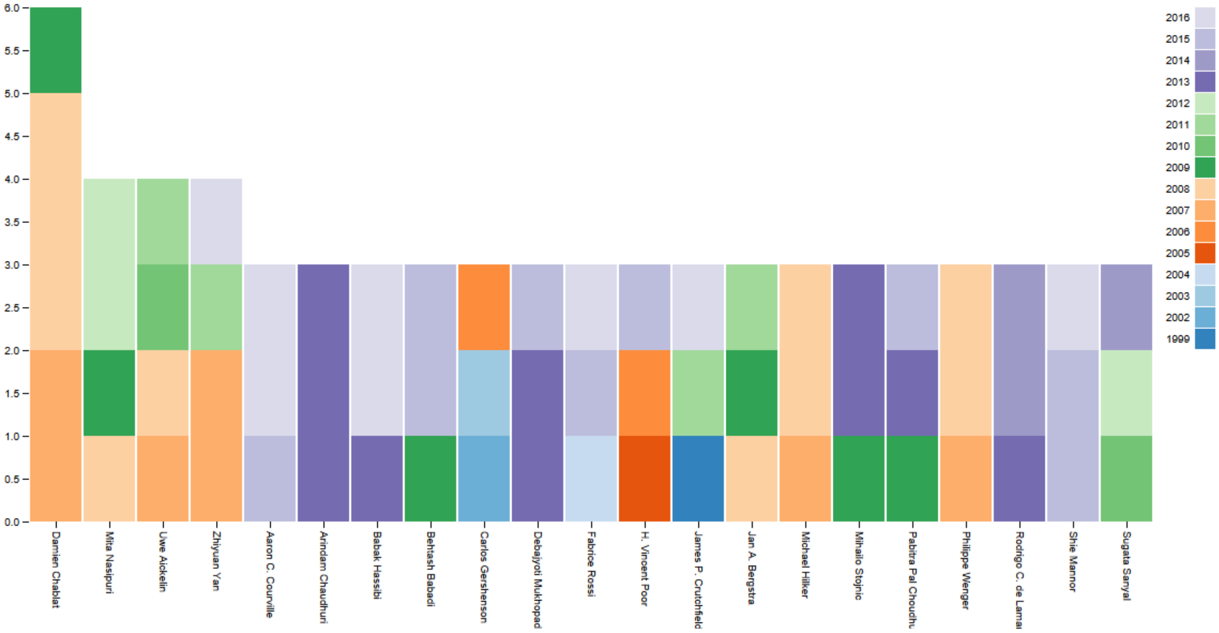
```
library(tidyverse)
library(plyr)
library(plotly)
library(rjson)
library(reshape2)
library(ggplot2)
library(scales)
#Quation 1
#Retrieve Json File
json_file <- "Mini-project-2-data.json"
json_data <- fromJSON(sprintf("[%s]", paste(readLines(json_file),collapse=",")))
#Get articles with venue ArXiv
newdata=json_data[which(sapply(json_data, `[[`, "venue") == "ArXiv")]
#Get list of authors and convert author to dataframe
listAuthor <- lapply(json_data, `[[`, "authors")
listAuthor1=do.call(c, unlist(listAuthor, recursive=FALSE))
listid=listAuthor1[names(listAuthor1[])=="ids"]
listname=listAuthor1[names(listAuthor1[])=="name"]
#Fill in empty id with "Unknown" first then drop them after convert it to dataframe
for (i in 1:length(listid)){
  if (length(listid[[i]])<1){
    print (i)
    listid[[i]]="unknown"
  }
}
dataid=data.frame(id=names(listid), Value=unlist(listid))
dataname=data.frame(id=names(listname), Value=unlist(listname))

dataauthor=merge(dataid,dataname,by=0,all=TRUE)
dataauthor=dataauthor[dataauthor$Value.x!="unknown",]
#Calculate the count of ids as the number of publications he/she has made in ArXiv
dataauthorcount=count(dataauthor, c('Value.x'))
dataauthorcount=dataauthorcount[with(dataauthorcount, order(-freq)), ]
graph1data=dataauthorcount[1:10,]

graph1data=merge(graph1data,dataauthor[,c("Value.x","Value.y")],by="Value.x",all.x= FALSE,all.y=FALSE)
graph1data=graph1data[!duplicated(graph1data$Value.x),]

colnames(graph1data) <- c("id","Number","author")
graph1data=graph1data[with(graph1data, order(-Number)), ]
rownames(graph1data) <- 1:nrow(graph1data)

#Finish preparing data for the graph
graph1data$author <- factor(graph1data$author)
#Plot with ggplot
p=ggplot(data=graph1data, aes(x = reorder(author, -Number), y=Number)) +
  geom_bar(stat="identity", fill="steelblue",width = 0.8)+geom_text(aes(label=Number), vjust=-0.3, size=3.5)
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

R code for object 1

Barchart was chosen as it can represent the number of articles and the rank of authors directly. Note that there are many authors published 3 articles our R code select the 6 authors with 3 articles in the barchart based on their ID rank. We can find author Damlen Chablat published the most number of articles for venue arVix.



R graph for object 1



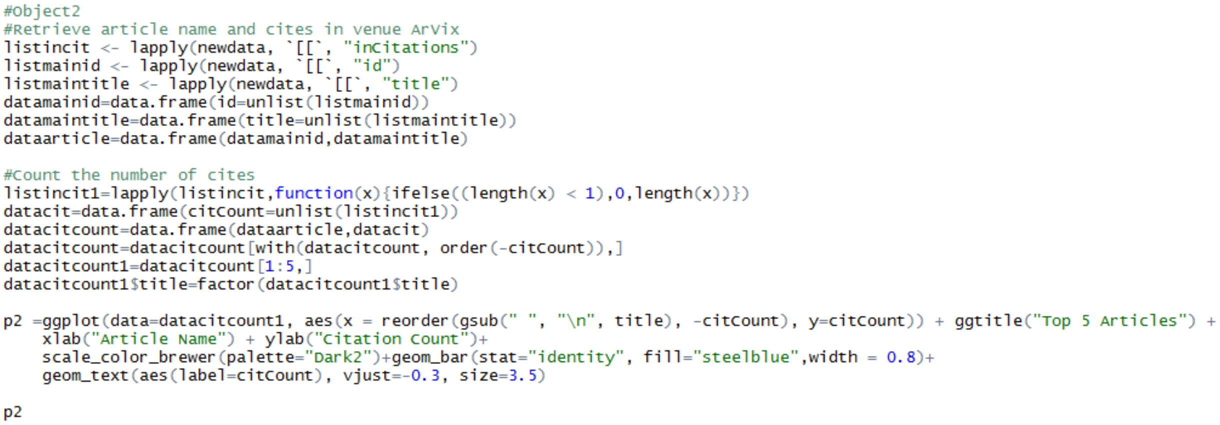
Splitting by years

Objective 2 - Visualize the top 5 papers for venue arXiv

Visualization with R

Data Preparation:

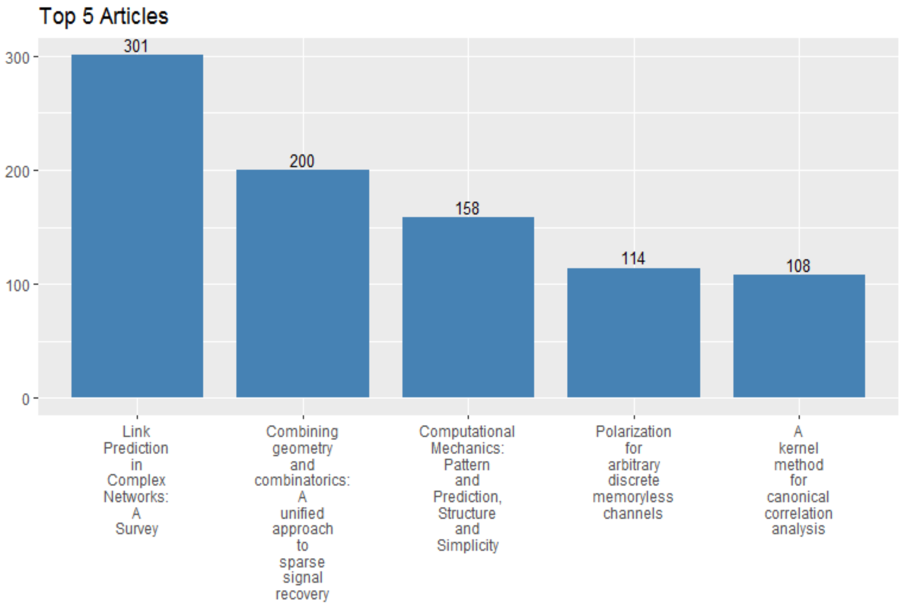
For object 2 R code, we retrieve the list of articled in venue arXiv and get the number of incitations for each article.



```r
#object2
#Retrieve article name and cites in venue ArVix
listincit <- lapply(newdata, `[[`, "inCitations")
listmainid <- lapply(newdata, `[[`, "id")
listmaintitle <- lapply(newdata, `[[`, "title")
datamainid=data.frame(id=unlist(listmainid))
datamaintitle=data.frame(title=unlist(listmaintitle))
dataarticle=data.frame(datamainid,datamaintitle)

#Count the number of cites
listincit1=lapply(listincit,function(x){ifelse((length(x) < 1),0,length(x))})
datacit=data.frame(citCount=unlist(listincit1))
datacitcount=data.frame(dataarticle,datacit)
datacitcount=datacitcount[with(datacitcount, order(-citCount)),]
datacitcount1=datacitcount[1:5,]
datacitcount1$title=factor(datacitcount1$title)

p2 =ggplot(data=datacitcount1, aes(x = reorder(gsub(" ", "\n", title), -citCount), y=citCount)) + ggtitle("Top 5 Articles") +
    xlab("Article Name") + ylab("Citation Count")+
    scale_color_brewer(palette="Dark2")+geom_bar(stat="identity", fill="steelblue",width = 0.8)+
    geom_text(aes(label=citCount), vjust=-0.3, size=3.5)

p2
```

R code for object 2

Same as object 1 barchart was chosen as it can represent the number of citations and the rank of citations directly. We can tell the top article directly from the graph below and the top article in arVix is "Link Prediction in Complex Networks: A Survey".



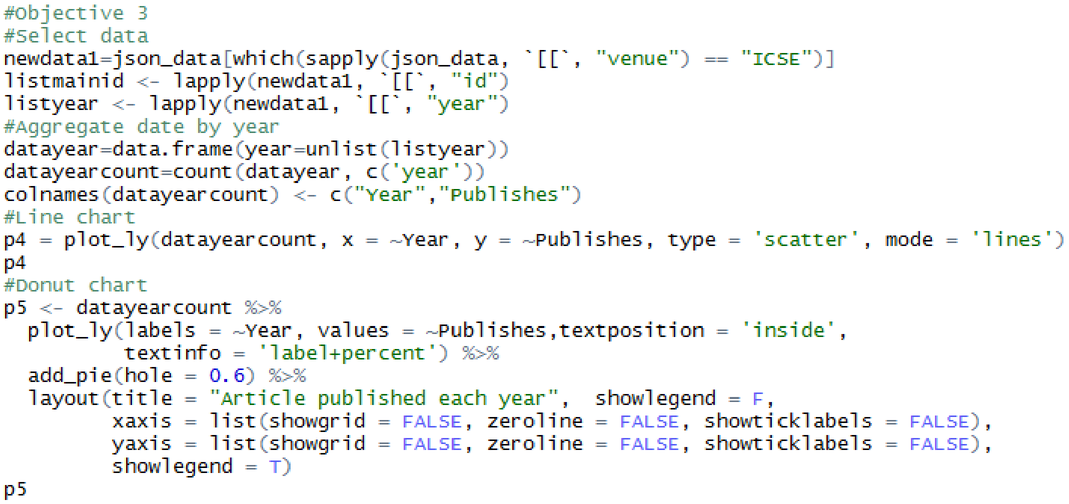
R graph for object 2

Objective 3 - Visualize the trend of the amount of publications

Visualization with R

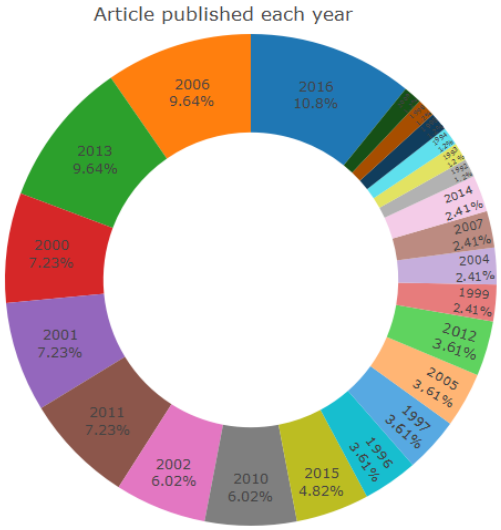
Data Preparation:

We select all the articles in venue ICSE first. Then aggregate the data according to the years.

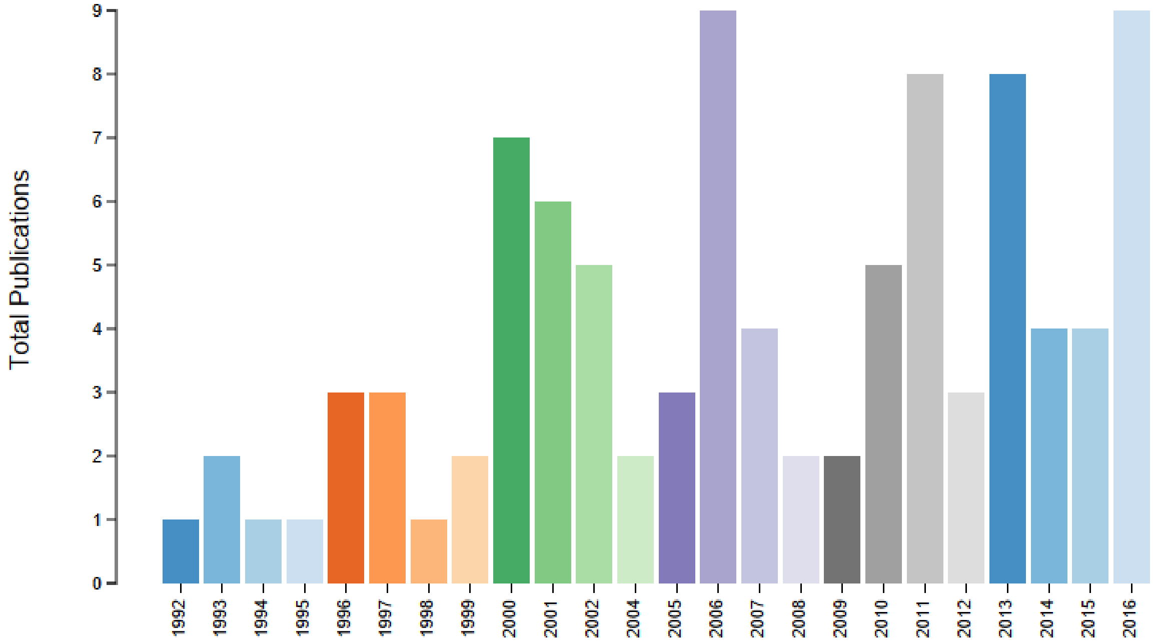


```
#Objective 3
#Select data
newdata1=json_data[which(sapply(json_data, `[[`, "venue") == "ICSE")]
listmainid <- lapply(newdata1, `[[`, "id")
listyear <- lapply(newdata1, `[[`, "year")
#Aggregate date by year
datayear=data.frame(year=unlist(listyear))
datayearcount=count(datayear, c('year'))
colnames(datayearcount) <- c("Year","Publishes")
#Line chart
p4 = plot_ly(datayearcount, x = ~Year, y = ~Publishes, type = 'scatter', mode = 'lines')
p4
#Donut chart
p5 <- datayearcount %>%
  plot_ly(labels = ~Year, values = ~Publishes,textposition = 'inside',
          textinfo = 'label+percent') %>%
  add_pie(hole = 0.6) %>%
  layout(title = "Article published each year",  showlegend = F,
         xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
         yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
         showlegend = T)
p5
```

For objective 3 we created 2 graphs a Line Chart to visualize the general trend of article published each year and a Donut Chart to visualize the percentage contribution for the articles published each year. From the Line Chart we can see the article posted each year varies significantly and seems not following a time order. From the donut chart the percentage contribution is not in year sequence either. So there's no significant trend for the article published.



R graph for object 3


Total Publications
0
1
2
3
4
5
6
7
8
9
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016