Propose a project that uses a large, publically accessible dataset. Motivate the problem you are tackling, discuss the data source(s) you are using, and explain the the analysis you are performing. Do enough exploratory data analysis to convince the project is viable and to generate two interesting non-trivial plots. Explain them and give url links to those plots.

The problem:

This project is targeted to people who would like to understand the current job market conditions for specific job types in various cities in the USA. The target audience includes both job seekers who want to know whether a specific city has a high demand for the job type they are pursuing and also employers who want to understand how easy it will be to fill a particular job requisition in particular cities.

Data Source: The data source for this project is the Indeed.com job board, arguably the world’s largest. Indeed.com has a publicly accessible API from which a multitude of data may be harvested. This API has been used for this initial exploration.

Initial exploration: For this exercise, a data set of the 24 largest US cities was generated. For the full project, more cities may be included. The question put to the API was “How many job postings are currently listed for a provided keyword set for each of the 24 US cities?” The results, as expected, showed the differences in industry mix and therefore demand for various job types among the various cities. Job keywords explored in this initial query include “data scientist”, “mechanical engineer”, “truck driver”, and “journalist”.

My initial concept was to provide a web page which would allow the user to input their own keywords and to receive an analysis of which were the cities with the highest demand for jobs featuring those keywords. This should work fine for this top-level analysis.

Questions addressed by a detailed analysis of the Indeed job posting data may include:

* For overall demand; how many jobs are posted for the given keywords in each city? (original question)
* For overall demand density; how many jobs are posted for the given keywords per million residents in each city?
* What is the average age of jobs posted for the given keywords in the given city?
* For the given keywords (in the given city), what skills are most often in the job listings themselves?

I believe this analysis would be useful to both job seekers and employers. The high-level analysis of simply how many jobs are listed by city could be done on the fly, via a web interface. A more in-depth analysis, involving characteristics of the job postings themselves may need to be done on a batch basis.

This project will provide information on the local job market for user-defined job types in metropolitan US cities using the Indeed.com API as a data source.

Questions of interest include:

* For overall demand; how many jobs are posted for the given keywords in each city? (original question)
* What is the average age of jobs posted for the given keywords in the given city?
* For the given keywords (in the given city), what skills are most often in the job listings themselves?

An initial analysis for the keyword “data scientist” provided the information that the 4 US hotspots for data scientists looking for work are Seattle, Boston, Chicago and New York City. indicating that any aspiring data scientist would be most likely to find a job quickly in those cities.

I believe the results of this analysis would be useful to both job seekers and employers looking to hire.

<https://www.youtube.com/edit?video_id=4MGUHgG8Mvs>

Job market characteristics in US cities for user-specified job keywords.

This project is targeted to people who would like to understand the current job market conditions for specific job types in various cities in the USA. The target audience includes both job seekers who want to know whether a specific city has a high demand for the job type they are pursuing and also employers who want to understand how easy it will be to fill a particular job requisition in particular cities.

Data Source: The data source for this project is the Indeed.com job board, arguably the world’s largest. Indeed.com has a publicly accessible API from which a multitude of data may be harvested. This API has been used for this initial exploration.

Initial exploration: For this exercise, a data set of the 24 largest US cities was generated. For the full project, more cities may be included. The question put to the API was “How many job postings are currently listed for a provided keyword set for each of the 24 US cities?” The results, as expected, showed the differences in industry mix and therefore demand for various job types among the various cities. Job keywords explored in this initial query include “data scientist”, “mechanical engineer”, “truck driver”, and “journalist”.

My initial concept was to provide a web page which would allow the user to input their own keywords and to receive an analysis of which were the cities with the highest demand for jobs featuring those keywords. This should work fine for this top-level analysis.

Questions addressed by a detailed analysis of the Indeed job posting data may include:

• For overall demand; how many jobs are posted for the given keywords in each city? (original question)

• For overall demand density; how many jobs are posted for the given keywords per million residents in each city?

• What is the average age of jobs posted for the given keywords in the given city?

• For the given keywords (in the given city), what skills are most often in the job listings themselves?

The 2 charts provided are the result of running the script provided for the 2 keywords "data scientist" and "television". They show the number of jobs currently posted on Indeed for the 24 largest US cities for these keywords. It is possible to conclude from these simple charts, that the best job market for data scientists is located in the cities of Seattle, Boston, Chicago, and New York City. For a job seeker using the keyword "television," the obvious place to look for work is New York City.

I believe this analysis would be useful to both job seekers and employers. The high-level analysis of simply how many jobs are listed by city could be done on the fly, via a web interface. More in-depth analyses, involving characteristics of the job postings themselves may need to be done on a batch basis.

http://www.indeed.com/publisher

http://mitchki.com/datascientist.pdf

<http://mitchki.com/television.pdf>

<https://www.youtube.com/watch?v=4MGUHgG8Mvs>

Question 1 script

## Question 1

numberseats <- 25

seats <- rep(0,numberseats)

fiftythou <- rep(0,50000)

tenthou <- rep(0,10000)

thou <- rep(0,1000)

##generate uniform random number between and length(seats)

set.seed(123)

## test whether seat is empty

##test whether seats on both sides are empty

## if all conditions are true, sit on seat

## Check whether there are any places left to sit

## (3 empty in a row, or 2 empty on an end)

set.seed(123)

## create vectors of empty seats below

## zero denotes an empty seat.. set to 1 when somebody sits in it

numberseats <- 25

seats <- rep(0,numberseats)

fiftythou <- rep(0,50000)

## function below checks whether any seats are available to sit on

## assumes that no rider will occupy a seat next to an occupied seat

## this requires either 3 empty seats in a row,

## or 2 empty seats at the end of the row

seatsAvail <- function(seats) {

## exit and return true as soon as an available seat is identified

if (sum(seats[1:2])==0) {return(TRUE)}

if (sum(seats[c(length(seats)-1,length(seats))])==0) {return(TRUE)}

inner <- c(2:(length(seats)-1))

for (i in inner) {

if ((seats[i-1]==0) & (seats[i]==0) & (seats[i+1]==0)) {return(TRUE)}

}

return(FALSE)

}

seatsAvail(full)

## function willsit calculates whether a person will sit in the selected seat

## assuming the person refuses to sit next to an already occupied seat

willsit <- function(seats,selectSeat) {

if (selectSeat==1) {

if ((!seats[1])&(!seats[2])) {return(TRUE)}

else {return(FALSE)}

}

if (selectSeat==length(seats)) {

if ((!seats[length(seats)])&(!seats[length(seats)-1])) {return(TRUE)}

else {return(FALSE)}

}

if (!seats[selectSeat]&!seats[selectSeat-1]&!seats[selectSeat+1]) {return(TRUE)}

else {return(FALSE)}

}

full <- c(1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0)

## function q1 performs a simulation of the subway seat problem

## as described in question 1

## it takes a vector of 0's corresponding to empty subway seats

## and the number of replications of the exercise

## defaulting to 100 reps if no value is provided

q1 <- function(seats,reps=100) {

totals <- 0; innersum <- 0; outersum <- 0

for (j in 1:reps) {

seats <- rep(0,length(seats))

## check whether any seats are available,

## if so, select a seat and see whether you will sit

while (seatsAvail(seats)) {

selectSeat <- ceiling(runif(1,min=0,max=length(seats)))

if (willsit(seats,selectSeat)) {seats[selectSeat] <- 1}

}

totals[j] <- sum(seats)/length(seats)

innersum[j] <- sum(seats[2:(length(seats)-1)])

outersum[j] <- sum(seats[c(1,length(seats))])

}

## print summary of the results of the simulations

print(paste("# seats =",length(seats)))

print (paste("N=",length(totals)," mean=",mean(totals)," sd=",sd(totals)))

print(paste("inner sum mean = ",mean(innersum)/(length(seats)-2)))

print(paste("outer sum mean = ",mean(outersum)/2))

}