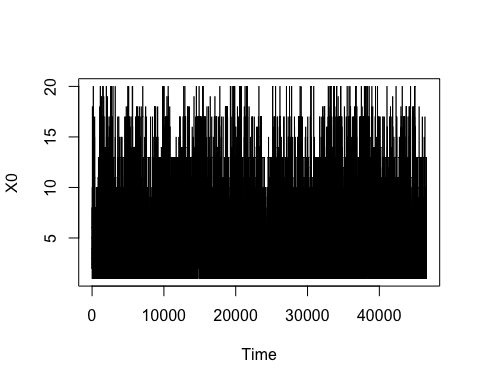
Probability-Markov-Chain-Project

## Set-up

Our data set was found on the Center for Machine Learning and Intelligent Systems website in its archived data set section. While looking through the Time Series data sets provided, we decided to work with the Online Retail Data Set that contained 541,909 rows of data. This data set provides “all the transactions that occurred between 01/12/2010 and 09/12/2011 for a UK-based and registered non-store online retails” (UCI). The data set contain 8 columns for attribute information. These columns included: the number of the invoice, the product code for each item, the product name for each item, the quantities of each product per transaction, the unit price of each item, the customer number and the name of the country where each customer resides. We found the unit prices column to be the best suited for our Markov Chain analysis as it was the only one that provided a numerical value that could be rounded into unique states. In order to keep from having too many data points at first, we decided to only look at the first 50,000 entries for our first Markov Chain. After deciding this, we determined we wanted to limit our states to 20 – each state being determined by the rounded value of the price provided. In order to calculate the state for each data value, we used the Round() function in Microsoft Excel for the values, rounding each value to the nearest integer. Most of the prices provided were in the single digits, while some were between $10 and $20. There were only a few values over $20, in comparison to the other prices, so we decided to filter them as they could not stand alone as their own state. We could have combined multiple prices into a single state to include these prices, but felt as though our approach was better suited for this data set. This left us with 46,540 total data points over our 20 states to create our Markov Chain. With our 20 states set, and each value mapped into a unique state, we transferred our data into R for the analysis part of the project.

## Plot Time Series Data

library(readxl)  
Online\_Retail <- read\_excel("~/Documents/Graduate-Math-Projects/Online Retail.xlsx", col\_names = FALSE)  
retailtimeseries <- ts(Online\_Retail)  
plot.ts(retailtimeseries)

 We imported our data set, which only has the rounded values of the united prices. Since the data set only contains the states of our Markov Chain, it is easy to put the data into a time series in order to plot it.

## Empirical Distribution of Markov Chain

fre1 <- table(Online\_Retail)  
fre1 <- fre1 / 46540  
fre1.data <- data.frame(fre1)  
colnames(fre1.data) <- c("States","frequency")  
library(ggplot2)  
plot1 <- ggplot(data = fre1.data, aes(x= States, y = frequency, group = 1)) + geom\_line()

To calculate the empirical distribution of the Markov Chain, we were tasked with computing the occupation frequencies for each state. This would end up showing the fraction of time spent in each state, also known as the empirical distribution.  
 We calculated the fraction of time spent in each state by dividing each frequency by the overall observations in the data set. We then plotted this against the states of the Markov Chain to show the empirical distribution.

## Computing the Transition Matrix

library(markovchain)

## Package: markovchain  
## Version: 0.6.9.8-1  
## Date: 2017-08-15  
## BugReport: http://github.com/spedygiorgio/markovchain/issues

Online\_Retail\_tpm = createSequenceMatrix(Online\_Retail, toRowProbs = TRUE)

To calculate the transition matrix, we were tasked with computing the frequencies of jumps between each pair of states. To do this, we used the MarkovChain package to create a sequence maxtrix of our time series. This ultimately divided by occupation frequency at each state so the total jump probability out of each state would be 1.

## Computing the Stationary Distribution

library(expm)

## Loading required package: Matrix

##   
## Attaching package: 'expm'

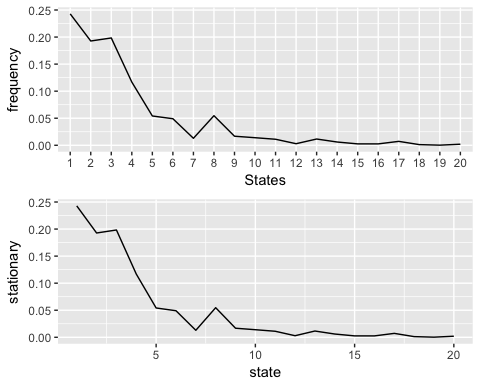
## The following object is masked from 'package:Matrix':  
##   
## expm

x2 <- Online\_Retail\_tpm%^%50  
stationary <- x2[1,]  
stationary.data <- data.frame(stationary)  
stationary.data <- stationary.data[order(as.numeric(rownames(stationary.data))),,drop = FALSE]  
stationary.data$state <- c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20)

To calculate the stationary distribution, we set our transition matrix to the power of 50 in order to reach the moment where all rows are equal for the transition matrix. This ultimately means that regardless of the initial state, the probability of ending up with a certain state is the same.  
 We then converted this stationary distribution into a data frame and included a column to indicate each state for the Markov Chain.

## Comparing the Empirical Distribution and Stationary Distribution

plot2 <- ggplot(data = stationary.data, aes(x= state, y = stationary, group = 1)) + geom\_line()  
library(gridExtra)  
grid.arrange(plot1,plot2)

 By plotting these two graphs together, we can notice that the empirical distribution and stationary distribution are very similar with little to no differences. With this information, we are able to determine that the Markov Chain method does produce a good model for this time series.