Assignment 3 - Exploratory Data Analysis

Team 1

1. **Data Cleaning**

Data cleaning was necessary for our data because there was a high enough number of outliers that would skew the data when we applied different techniques to give information about the data.

In Table B, there are three columns that indicate the number of items that a customer received early, late, or on time. These three columns then equaled another column, "Total Parts." In an additional column that we created, the three columns were summed and the "Total Parts" column was subtracted to indicate if there were any orders that had not been fulfilled. We found that 5,618 orders had negative values indicating objects not yet being delivered to customers. Since these orders make up only a one hundredth of the total data set, these orders were eliminated from the data set. This was necessary to keep the data from being skewed or having too high of a prediction error.In Table D, the values that are intended to be in column "type" are 0 and 1. There were values from 2-2000 found in the column which were incorrect data types. These orders with incorrect data types only made up 2% of the data, so they were eliminated from the data sheet. This makes it so only working data is left that contains 0s and 1s.

In Table E, there are two date columns, one which indicates when an order was placed and one when an order was marked as complete. An additional column was added to the table which subtracted the date of order placement from the date of order completion. This allowed for the total number of days surpassed for an order to be seen. After completing the calculation for each order, we found that 157 orders had negative values which indicates that the orders completion date was input as a date that was older than the date in the order placed column. Since this made up a small percentage of our data for this table, these rows were eliminated from the table. Also, there are orders that have null values in the order completion column. To handle these rows, an if-else statement will be used to ignore those orders since we do not have complete data for those orders.

The Customer Info table has a type column which indicates what type of customer they are. Values that are seen in this column are 0,1, and 2. When filtering the data, there are two outliers. A value of -1 and 27. After looking up the corresponding “Customer ID” we found that those were data entry errors because the customer type was not consistent for the same customer throughout the data set. The values were changed to reflect the correct customer type which came out to be 1 and 2 respectively.

Looking at the data set, there is a column in each table that indicates customer ID numbers. Each column name is different but after closer inspection, the same values can be seen in each table. To make it easier to access different values for a single customer ID, these tables will be joined using a JOIN operator in SQL.

1. **Data Reshaping**

The only reshaping done so far was the conversion of the date columns to day, month and year columns. This was done to make machine learning and other analytics by day, month, and year easier.

**Reference: convert\_dates.Rmd**

**convert\_dates**

**library**(reticulate)

**library**(readxl)

**Replace use\_python with your package**

**use\_python**("/Users/Bryton/Miniconda3/python")  
**#py\_config()**

**run if first time**

*#py\_install("pandas")*  
 *#py\_install("xlrd")*  
 *#py\_install("datetime")*

import pandas  
 import datetime  
 import numpy

# Customer Info

customerInfo **=** pandas.read\_excel("Data/WSU Capstone Project - Working Data.xlsx",sheet\_name**=**"Customer Info", usecols**=**"A:C")  
 customerInfo.to\_csv("Data/customerInfo.csv")

# table A

tableA **=** pandas.read\_excel("Data/WSU Capstone Project - Working Data.xlsx",sheet\_name**=**"Table A", usecols**=**"A:D")  
 dates **=** pandas.DatetimeIndex(tableA.Date)  
 tableA["Year"] **=** dates.year  
 tableA["Month"] **=** dates.month  
 tableA["Day"] **=** dates.day

print(tableA.Year)

## 0 2014  
 ## 1 2014  
 ## 2 2014  
 ## 3 2014  
 ## 4 2014  
 ## ...   
## 96225 2017  
 ## 96226 2017  
 ## 96227 2017  
 ## 96228 2017  
 ## 96229 2017  
 ## Name: Year, Length: 96230, dtype: int64

tableA.to\_csv("Data/tableA.csv")

# Table B

tableB **=** pandas.read\_excel("Data/WSU Capstone Project - Working Data.xlsx",sheet\_name**=**"Table B", usecols**=**"A:G")  
 dates **=** pandas.DatetimeIndex(tableB.Date)  
 tableB["Year"] **=** dates.year  
 tableB["Month"] **=** dates.month  
 tableB["Day"] **=** dates.day

tableB.to\_csv("Data/tableB.csv")

# Table D

tableD **=** pandas.read\_excel("Data/WSU Capstone Project - Working Data.xlsx",sheet\_name**=**"Table D", usecols**=**"A:D")  
 dates **=** pandas.DatetimeIndex(tableD.Date)  
 tableD["Year"] **=** dates.year  
 tableD["Month"] **=** dates.month  
 tableD["Day"] **=** dates.day

tableD.to\_csv("Data/tableD.csv")

# Table E

tableE **=** pandas.read\_excel("Data/WSU Capstone Project - Working Data.xlsx",sheet\_name**=**"Table E", usecols**=**"A:D")  
 dates **=** pandas.DatetimeIndex(tableE.Open)  
 tableE["Open\_Year"] **=** dates.year  
 tableE["Open\_Month"] **=** dates.month  
 tableE["Open\_Day"] **=** dates.day  
 dates **=** pandas.DatetimeIndex(tableE.Closed)  
 tableE["Closed\_Year"] **=** dates.year  
 tableE["Closed\_Month"] **=** dates.month  
 tableE["Closed\_Day"] **=** dates.day

tableE.to\_csv("Data/tableE.csv")

1. **Data Management**
   1. **Apply previously defined data management strategies**

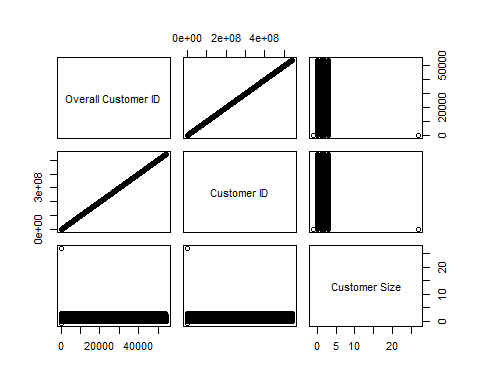
To manage our data better we plan to implement a SQLITE database

1. **Plot distributions of Key variables**

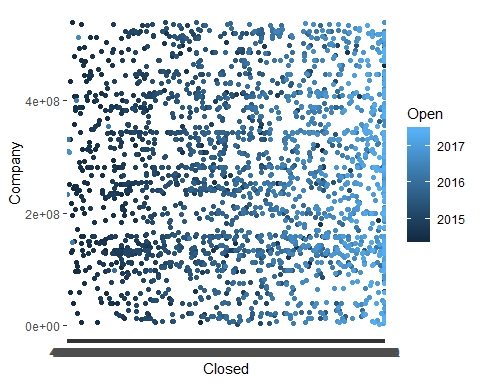
Basic plots were created to help us visualize issues in the data

**Reference Data\_Discussion.Rmd:**

**plot**(CustomerInfo)



**ggplot**(tableE, **aes**(x = Closed, y = Company, color = Open)) **+** **geom\_point**()



1. **Create Key metrics/Measurements/Statistics that summarize the data**

Customer Info: 116,158 observations with 3 values for each observation Table A: 96,230 observations with 6 values for each observation Table B: 546,963 observations, with 7 values for each observation Table D: 270,296 observations with 6 values for each observation Table E: 7,205 observations with 5 values for each observation

1. **Find out if there are any outliers/anomalies in your dataset, and if there are outliers, develop strategies for dealing with them**

Outliers were found and dealt with in the data cleaning process.

1. **Deploy graphical tools (scatterplots, histograms, etc) and look for relationships (correlations) within the data.**

**Reference EDA\_G.Rmd:**

**library**(readxl)  
library(tidyverse)

**library**(ggplot2)  
**library**(devtools)

## Loading required package: usethis

CustomerInfo <- **read\_excel**("Data.xlsx", sheet = "Customer Info")  
tableA <- **read\_excel**("Data.xlsx", sheet = "Table A")

## New names:  
## \* `` -> ...5

tableB <- **read\_excel**("Data.xlsx", sheet = "Table B")  
tableD <- **read\_excel**("Data.xlsx", sheet = "Table D")

## New names:  
## \* `` -> ...5

tableE <- **read\_excel**("Data.xlsx", sheet = "Table E")  
  
  
CustInfo = **read.csv**("Data\_customerInfo.csv")  
TableA = **read.csv**("Data\_tableA.csv")  
TableB = **read.csv**("Data\_tableB.csv")  
TableD = **read.csv**("Data\_tableD.csv")  
TableE = **read.csv**("Data\_tableE.csv")

## Part G

tableA = tableA[,1**:**4]  
tableB = tableB[,1**:**7]  
tableD = tableD[,1**:**4]  
tableE = tableE[,1**:**5]  
head(tableA)

## # A tibble: 6 x 4  
## RowID CustomerLocation Date Type  
## <dbl> <dbl> <dttm> <dbl>  
## 1 1 519700015 2014-06-06 00:00:00 0  
## 2 2 43930003 2014-06-06 00:00:00 0  
## 3 3 43930003 2014-06-06 00:00:00 0  
## 4 4 420760005 2014-06-06 00:00:00 0  
## 5 5 510840011 2014-06-06 00:00:00 0  
## 6 6 110120145 2014-06-06 00:00:00 0

**head**(CustomerInfo)

## # A tibble: 6 x 3  
## `Overall Customer ID` `Customer ID` `Customer Size`  
## <dbl> <dbl> <dbl>  
## 1 1 10001 1  
## 2 2 20001 1  
## 3 3 30001 0  
## 4 3 30002 0  
## 5 4 40001 1  
## 6 4 40002 1

**head**(tableB)

## # A tibble: 6 x 7  
## RowID `Customer Key` Date Requested Early Late `On Time`  
## <dbl> <dbl> <dttm> <dbl> <dbl> <dbl> <dbl>  
## 1 1 316170027 2014-06-06 00:00:00 1 1 0 0  
## 2 2 316170027 2014-06-06 00:00:00 1 1 0 0  
## 3 3 316170027 2014-06-06 00:00:00 1 0 1 0  
## 4 4 316170027 2014-06-06 00:00:00 1 1 0 0  
## 5 5 316170027 2014-06-06 00:00:00 1 0 1 0  
## 6 6 316170027 2014-06-06 00:00:00 1 0 1 0

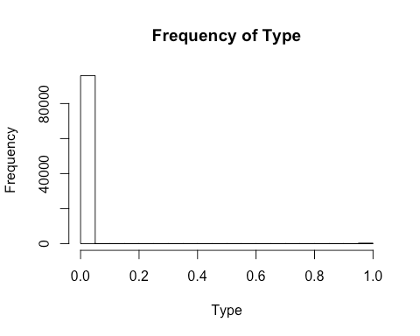
**head**(tableD)

## # A tibble: 6 x 4  
## RowID Customer Date Type  
## <dbl> <dbl> <chr> <dbl>  
## 1 1 511530001 41801 1  
## 2 2 48880011 41801 1  
## 3 3 294210001 41801 1  
## 4 4 387830038 41801 1  
## 5 5 387830038 41801 1  
## 6 6 435710010 41801 1

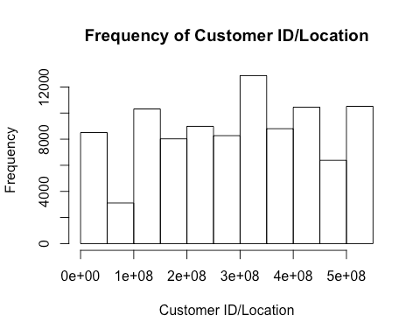
**head**(tableE)

## # A tibble: 6 x 5  
## RowID Company Open Closed Type  
## <dbl> <dbl> <dttm> <chr> <dbl>  
## 1 1 331360071 2014-06-06 00:00:00 42035 0  
## 2 2 331360071 2014-06-06 00:00:00 42035 0  
## 3 3 331360071 2014-06-06 00:00:00 42035 0  
## 4 4 331360071 2014-06-06 00:00:00 42035 0  
## 5 5 331360071 2014-06-06 00:00:00 42035 0  
## 6 6 331360071 2014-06-06 00:00:00 42035 0

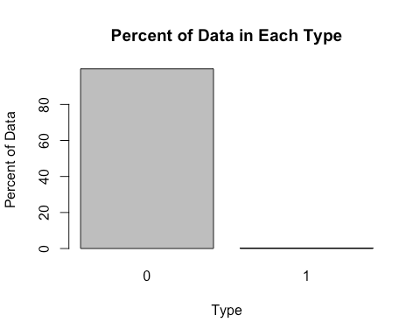
**hist**(tableA**$**Type, xlab="Type", main="Frequency of Type")



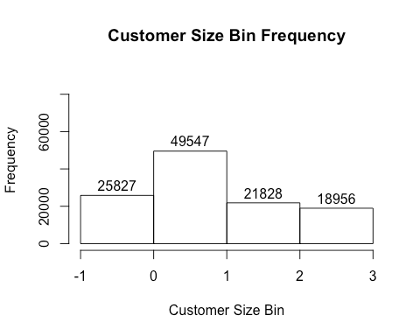
**hist**(tableA**$**CustomerLocation, xlab="Customer ID/Location", main= "Frequency of Customer ID/Location")



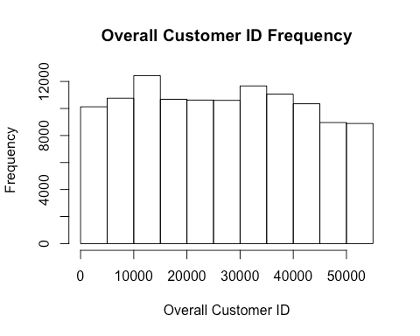
s= tableA **%>%**  
 **group\_by**(Type) **%>%**  
 **summarise**(percent= 100**\*** **n**() **/** **nrow**(tableA))  
barplot(s**$**percent,names.arg = **c**(0,1), xlab = "Type", ylab = "Percent of Data", main="Percent of Data in Each Type")



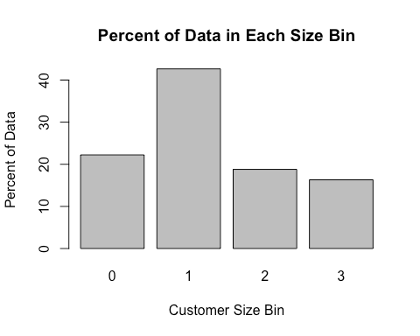
h =**hist**(CustomerInfo**$**`Customer Size`, main="Customer Size Bin Frequency", xlab = "Customer Size Bin", breaks=**c**(**-**1,0,1,2,3), ylim = **c**(0,90000))  
text(h**$**mids,h**$**counts,labels=h**$**counts, adj=**c**(.5,**-**.5))



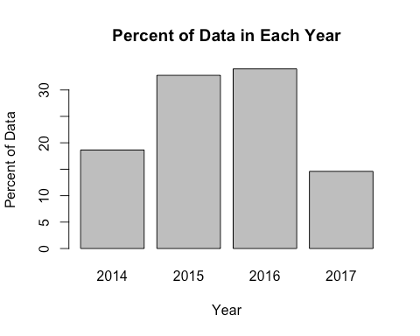
**hist**(CustomerInfo**$**`Overall Customer ID`, main="Overall Customer ID Frequency", xlab="Overall Customer ID")



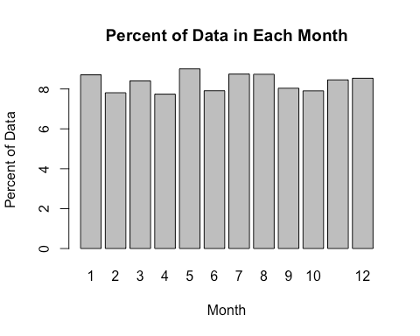
s= CustomerInfo **%>%**  
 **group\_by**(`Customer Size`) **%>%**  
 **summarise**(percent= 100**\*** **n**() **/** **nrow**(CustomerInfo))  
barplot(s**$**percent,names.arg = **c**(0,1,2,3), xlab="Customer Size Bin", ylab = "Percent of Data", main="Percent of Data in Each Size Bin")



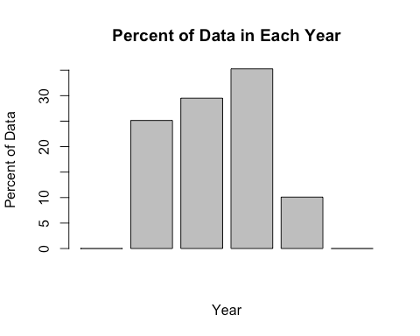
s= TableB **%>%**  
 **group\_by**(Year) **%>%**  
 **summarise**(percent= 100**\*** **n**() **/** **nrow**(TableB))  
barplot(s**$**percent,names.arg = **c**(2014,2015,2016,2017), xlab = "Year", ylab = "Percent of Data", main="Percent of Data in Each Year")



s= TableB **%>%**  
 **group\_by**(Month) **%>%**  
 **summarise**(percent= 100**\*** **n**() **/** **nrow**(TableB))  
barplot(s**$**percent,names.arg = **c**(1,2,3,4,5,6,7,8,9,10,11,12),xlab = "Month", ylab = "Percent of Data", main="Percent of Data in Each Month")



s= TableD **%>%**  
 **group\_by**(Year) **%>%**  
 **summarise**(percent= 100**\*** **n**() **/** **nrow**(TableD))  
barplot(s**$**percent,xlab = "Year", ylab = "Percent of Data", main="Percent of Data in Each Year")



org = "1899-12-30"  
tmp = **as.integer**(tableD**$**Date)

## Warning: NAs introduced by coercion

tmp = **as.Date**(tmp,origin=org, tz = "PST") *#in pst because thats where SEL is i dont think this will matter*  
  
*tableD***$**Date = tmp  
  
tmp = tmp = **as.integer**(tableE**$**Closed)

## Warning: NAs introduced by coercion

tmp = **as.Date**(tmp,origin=org, tz = "PST")  
  
tableE**$**Closed = tmp  
  
head(tableD)

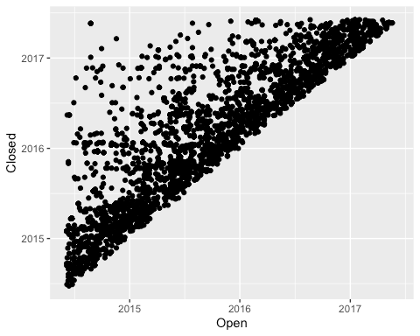
## # A tibble: 6 x 4  
## RowID Customer Date Type  
## <dbl> <dbl> <date> <dbl>  
## 1 1 511530001 2014-06-11 1  
## 2 2 48880011 2014-06-11 1  
## 3 3 294210001 2014-06-11 1  
## 4 4 387830038 2014-06-11 1  
## 5 5 387830038 2014-06-11 1  
## 6 6 435710010 2014-06-11 1

**head**(tableE)

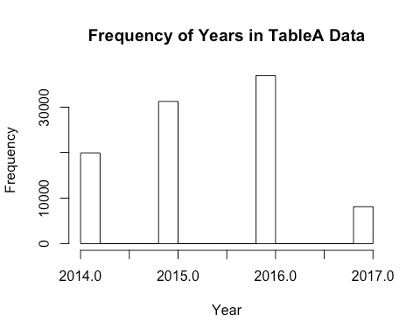
## # A tibble: 6 x 5  
## RowID Company Open Closed Type  
## <dbl> <dbl> <dttm> <date> <dbl>  
## 1 1 331360071 2014-06-06 00:00:00 2015-01-31 0  
## 2 2 331360071 2014-06-06 00:00:00 2015-01-31 0  
## 3 3 331360071 2014-06-06 00:00:00 2015-01-31 0  
## 4 4 331360071 2014-06-06 00:00:00 2015-01-31 0  
## 5 5 331360071 2014-06-06 00:00:00 2015-01-31 0  
## 6 6 331360071 2014-06-06 00:00:00 2015-01-31 0

**ggplot**(data = tableE, **aes**(x =Open, y=Closed)) **+** **geom\_point**() **+** **geom\_hline**(yintercept = 2015)

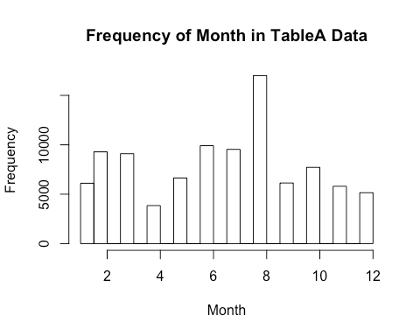
## Warning: Removed 392 rows containing missing values (geom\_point).



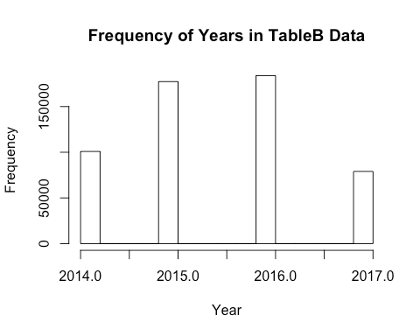
**hist**(TableA**$**Year, xlab="Year", main = "Frequency of Years in TableA Data")



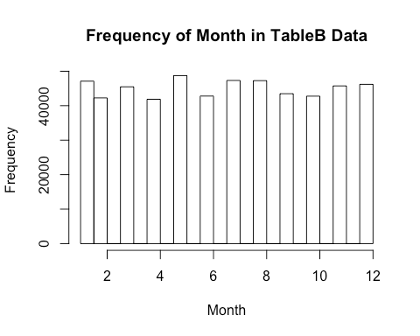
**hist**(TableA**$**Month, xlab = "Month",main= "Frequency of Month in TableA Data")



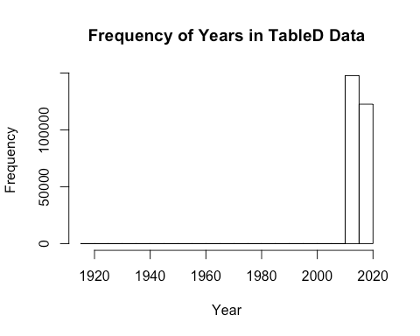
**hist**(TableB**$**Year, xlab="Year", main = "Frequency of Years in TableB Data")



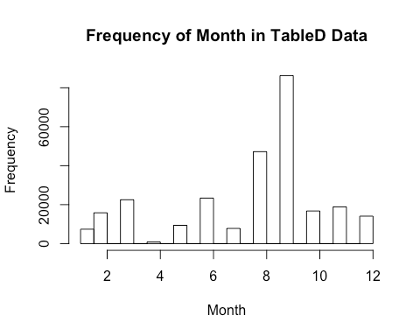
**hist**(TableB**$**Month, xlab = "Month", main="Frequency of Month in TableB Data")



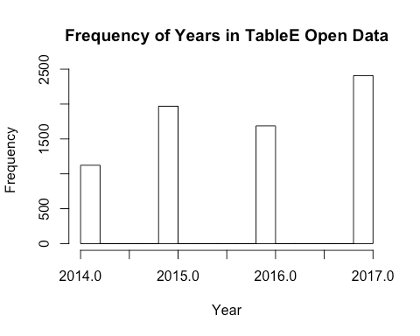
**hist**(TableD**$**Year, xlab="Year", main = "Frequency of Years in TableD Data")



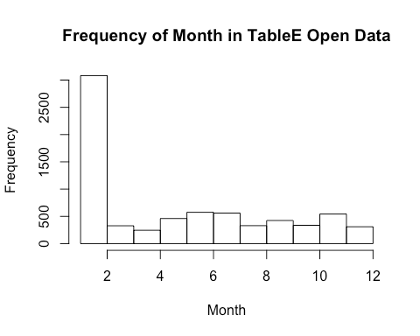
**hist**(TableD**$**Month, xlab = "Month", main="Frequency of Month in TableD Data")



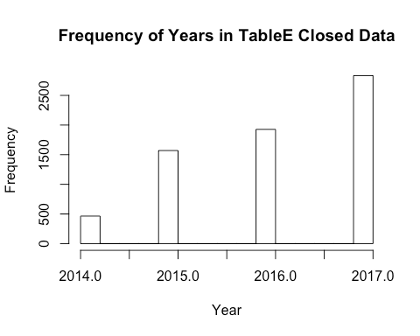
**hist**(TableE**$**Open\_Year, xlab="Year", main = "Frequency of Years in TableE Open Data")



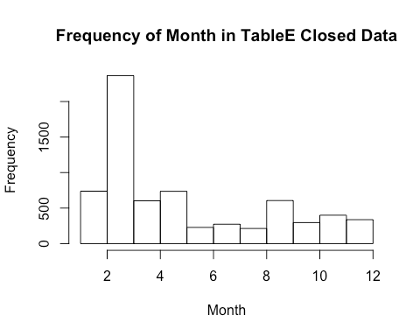
**hist**(TableE**$**Open\_Month, xlab = "Month", main="Frequency of Month in TableE Open Data")



**hist**(TableE**$**Closed\_Year, xlab="Year", main = "Frequency of Years in TableE Closed Data")



**hist**(TableE**$**Closed\_Month, xlab = "Month", main="Frequency of Month in TableE Closed Data")



1. **Analyze your data over time (if appropriate)**

**Reference EDA\_D.Rmd:**

**library**(dplyr)

**library**(tidyr)

**library**(ggplot2)

# load in files

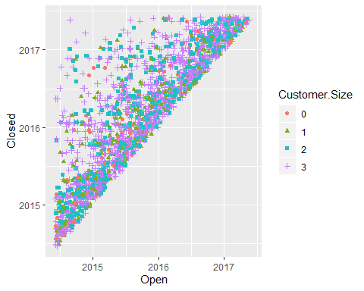
customerInfo <- **read.csv**("~/team\_1/Data/customerInfo.csv")  
tableA <- **read.csv**("~/team\_1/Data/tableA.csv")  
tableB <- **read.csv**("~/team\_1/Data/tableB.csv")  
tableD <- **read.csv**("~/team\_1/Data/tableD.csv")  
tableE <- **read.csv**("~/team\_1/Data/tableE.csv")

**colnames**(tableE)[**colnames**(tableE) **==** "Company"] = "Customer.ID"  
colnames(tableB)[**colnames**(tableB) **==** "Customer.Key"] = "Customer.ID"  
colnames(tableD)[**colnames**(tableD) **==** "Customer"] = "Customer.ID"

tblE\_CustInfo = **merge**(customerInfo,tableE,by="Customer.ID")  
tblE\_CustInfo**$**Customer.Size = **as.factor**(tblE\_CustInfo**$**Customer.Size)  
tblE\_CustInfo**$**Open = **as.Date.factor**(tblE\_CustInfo**$**Open)  
tblE\_CustInfo**$**Closed = **as.Date.factor**(tblE\_CustInfo**$**Closed)

### plot open vs. closed colored by company size

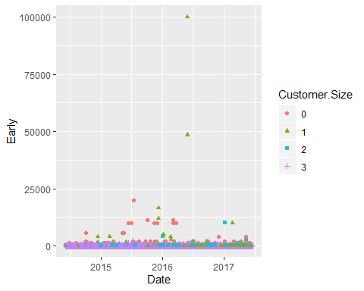
**ggplot**(tblE\_CustInfo, **aes**(x = Open, y = Closed, shape = Customer.Size, color = Customer.Size)) **+** **geom\_point**()



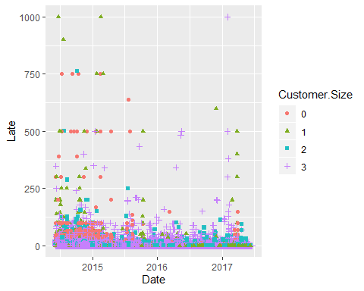
tblB\_CustInfo = **merge**(customerInfo,tableB,by="Customer.ID")  
tblB\_CustInfo**$**Customer.Size = **as.factor**(tblB\_CustInfo**$**Customer.Size)  
tblB\_CustInfo**$**Date = **as.Date.factor**(tblB\_CustInfo**$**Date)

### Plot date by early, late, requested, and on-time

**ggplot**(tblB\_CustInfo, **aes**(x = Date, y = Early, shape = Customer.Size, color = Customer.Size)) **+** **geom\_point**()

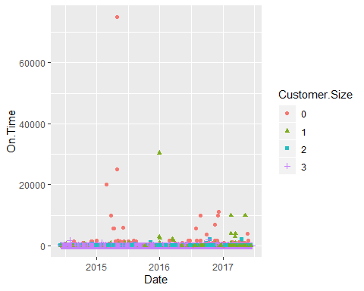


**ggplot**(tblB\_CustInfo, **aes**(x = Date, y = Late, shape = Customer.Size, color = Customer.Size)) **+** **geom\_point**() **+** **lims**(y =**c**(0,1000))



Note: It looks like the company reduced the amount of Late orders over time

**ggplot**(tblB\_CustInfo, **aes**(x = Date, y = On.Time, shape = Customer.Size, color = Customer.Size)) **+** **geom\_point**()



Note: smaller companies paid on time more often? That seems wrong given what I know of small companies. maybe 0 is large and 3 is small contrary to my previous understanding.

webshot**::install\_phantomjs**()

## It seems that the version of `phantomjs` installed is greater than or equal to the requested version.To install the requested version or downgrade to another version, use `force = TRUE`.

**library**(readxl)  
 **library**(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──

## ✓ ggplot2 3.3.0 ✓ purrr 0.3.3  
 ## ✓ tibble 2.1.3 ✓ dplyr 0.8.4  
 ## ✓ tidyr 1.0.0 ✓ stringr 1.4.0  
 ## ✓ readr 1.3.1 ✓ forcats 0.4.0

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
 ## x dplyr::filter() masks stats::filter()  
 ## x dplyr::lag() masks stats::lag()

**library**(ggplot2)  
 **library**(devtools)

## Loading required package: usethis

**library**(dplyr)  
 **library**(plotly)

##   
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':  
 ##   
## last\_plot

## The following object is masked from 'package:stats':  
 ##   
## filter

## The following object is masked from 'package:graphics':  
 ##   
## layout

**library**(hrbrthemes)

## NOTE: Either Arial Narrow or Roboto Condensed fonts are required to use these themes.

## Please use hrbrthemes::import\_roboto\_condensed() to install Roboto Condensed and

## if Arial Narrow is not on your system, please see <https://bit.ly/arialnarrow>

CustInfo = **read.csv**("Data\_customerInfo.csv")  
 TableA = **read.csv**("Data\_tableA.csv")  
 TableB = **read.csv**("Data\_tableB.csv")  
 TableD = **read.csv**("Data\_tableD.csv")  
 TableE = **read.csv**("Data\_tableE.csv")

**head**(TableA)

## X RowID CustomerLocation Date Type Year Month Day  
 ## 1 0 1 519700015 2014-06-06 0 2014 6 6  
 ## 2 1 2 43930003 2014-06-06 0 2014 6 6  
 ## 3 2 3 43930003 2014-06-06 0 2014 6 6  
 ## 4 3 4 420760005 2014-06-06 0 2014 6 6  
 ## 5 4 5 510840011 2014-06-06 0 2014 6 6  
 ## 6 5 6 110120145 2014-06-06 0 2014 6 6

**head**(CustInfo)

## X Overall.Customer.ID Customer.ID Customer.Size  
 ## 1 0 1 10001 1  
 ## 2 1 2 20001 1  
 ## 3 2 3 30001 0  
 ## 4 3 3 30002 0  
 ## 5 4 4 40001 1  
 ## 6 5 4 40002 1

**head**(TableB)

## X RowID Customer.Key Date Requested Early Late On.Time Year Month Day  
 ## 1 0 1 316170027 2014-06-06 1 1 0 0 2014 6 6  
 ## 2 1 2 316170027 2014-06-06 1 1 0 0 2014 6 6  
 ## 3 2 3 316170027 2014-06-06 1 0 1 0 2014 6 6  
 ## 4 3 4 316170027 2014-06-06 1 1 0 0 2014 6 6  
 ## 5 4 5 316170027 2014-06-06 1 0 1 0 2014 6 6  
 ## 6 5 6 316170027 2014-06-06 1 0 1 0 2014 6 6

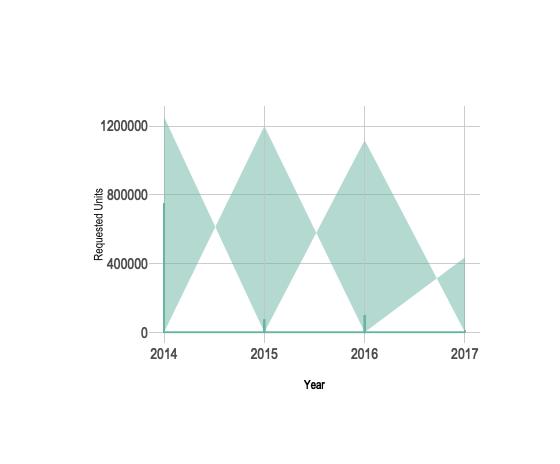
TableD=TableD[**!**(TableD**$**Type**>**1),]  
 **head**(TableD)

## X RowID Customer Date Type Year Month Day  
 ## 1 0 1 511530001 2014-06-11 00:00:00 1 2014 6 11  
 ## 2 1 2 48880011 2014-06-11 00:00:00 1 2014 6 11  
 ## 3 2 3 294210001 2014-06-11 00:00:00 1 2014 6 11  
 ## 4 3 4 387830038 2014-06-11 00:00:00 1 2014 6 11  
 ## 5 4 5 387830038 2014-06-11 00:00:00 1 2014 6 11  
 ## 6 5 6 435710010 2014-06-11 00:00:00 1 2014 6 11

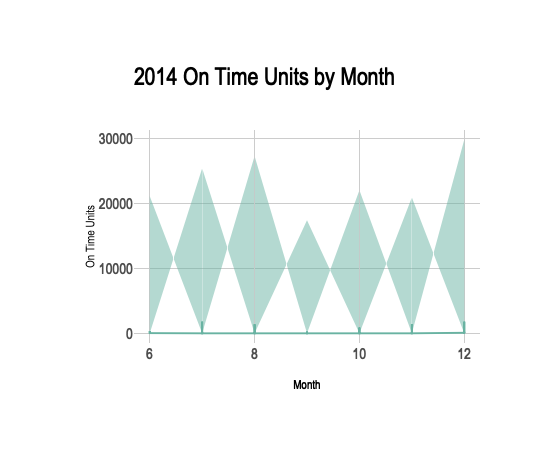
**head**(TableE)

## X RowID Company Open Closed Open\_Year Open\_Month Open\_Day  
 ## 1 0 1 331360071 2014-06-06 2015-01-31 2014 6 6  
 ## 2 1 2 331360071 2014-06-06 2015-01-31 2014 6 6  
 ## 3 2 3 331360071 2014-06-06 2015-01-31 2014 6 6  
 ## 4 3 4 331360071 2014-06-06 2015-01-31 2014 6 6  
 ## 5 4 5 331360071 2014-06-06 2015-01-31 2014 6 6  
 ## 6 5 6 331360071 2014-06-06 2015-01-31 2014 6 6  
 ## Closed\_Year Closed\_Month Closed\_Day  
 ## 1 2015 1 31  
 ## 2 2015 1 31  
 ## 3 2015 1 31  
 ## 4 2015 1 31  
 ## 5 2015 1 31  
 ## 6 2015 1 31

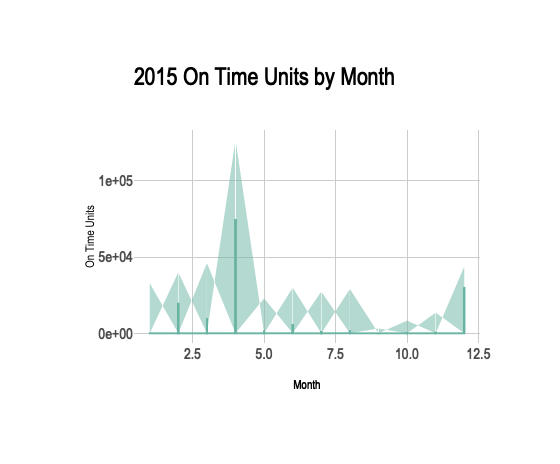
p = TableB **%>%**  
 **ggplot**(**aes**(x=Year, y=Requested))**+**  
 **geom\_area**(fill="#69b3a2", alpha=.5) **+**  
 **geom\_line**(color="#69b3a2")**+**  
 **ylab**("Requested Units") **+**  
 **theme\_ipsum**()  
 p = **ggplotly**(p)  
 p



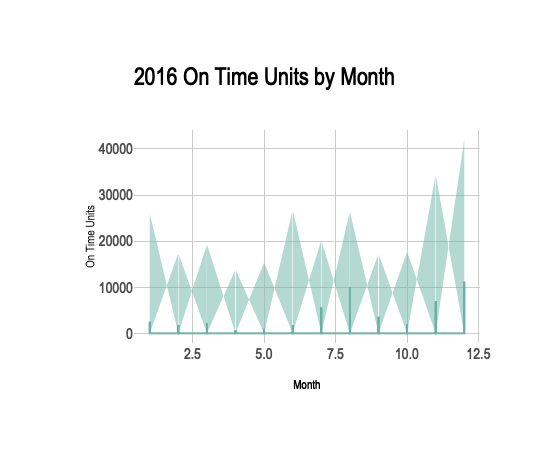
OnTimeYear = **filter**(TableB, TableB**$**Year**==**2014)  
 p = OnTimeYear **%>%**  
 **ggplot**(**aes**(x=Month, y=On.Time))**+**  
 **geom\_area**(fill="#69b3a2", alpha=.5) **+**  
 **geom\_line**(color="#69b3a2")**+**  
 **ylab**("On Time Units") **+**  
 **ggtitle**("2014 On Time Units by Month")**+**  
 **theme\_ipsum**()  
 p = **ggplotly**(p)  
 p



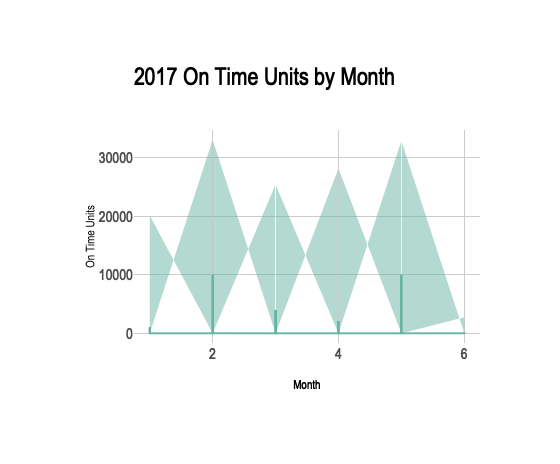
OnTimeYear = **filter**(TableB, TableB**$**Year**==**2015)  
 p = OnTimeYear **%>%**  
 **ggplot**(**aes**(x=Month, y=On.Time))**+**  
 **geom\_area**(fill="#69b3a2", alpha=.5) **+**  
 **geom\_line**(color="#69b3a2")**+**  
 **ylab**("On Time Units") **+**  
 **ggtitle**("2015 On Time Units by Month")**+**  
 **theme\_ipsum**()  
 p = **ggplotly**(p)  
 p



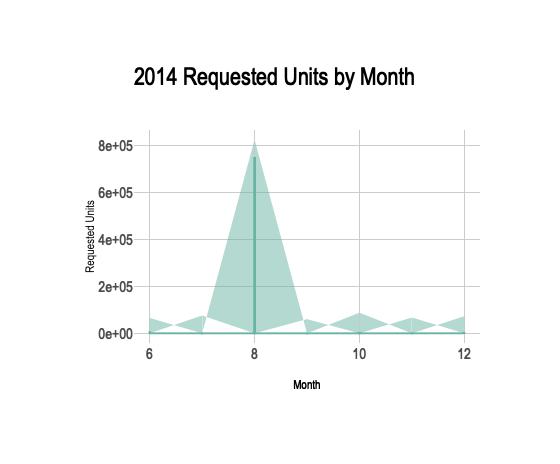
OnTimeYear = **filter**(TableB, TableB**$**Year**==**2016)  
 p = OnTimeYear **%>%**  
 **ggplot**(**aes**(x=Month, y=On.Time))**+**  
 **geom\_area**(fill="#69b3a2", alpha=.5) **+**  
 **geom\_line**(color="#69b3a2")**+**  
 **ylab**("On Time Units") **+**  
 **ggtitle**("2016 On Time Units by Month")**+**  
 **theme\_ipsum**()  
 p = **ggplotly**(p)  
 p



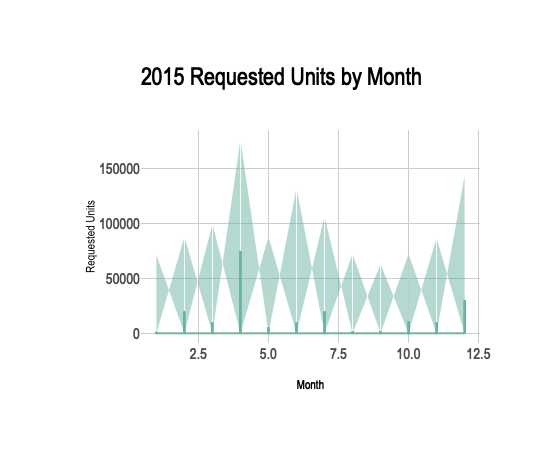
OnTimeYear = **filter**(TableB, TableB**$**Year**==**2017)  
 p = OnTimeYear **%>%**  
 **ggplot**(**aes**(x=Month, y=On.Time))**+**  
 **geom\_area**(fill="#69b3a2", alpha=.5) **+**  
 **geom\_line**(color="#69b3a2")**+**  
 **ylab**("On Time Units") **+**  
 **ggtitle**("2017 On Time Units by Month")**+**  
 **theme\_ipsum**()  
 p = **ggplotly**(p)  
 p



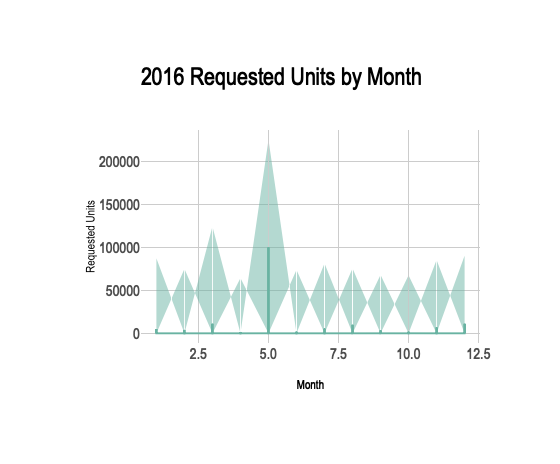
RequestedYear = **filter**(TableB, TableB**$**Year**==**2014)  
 q = RequestedYear **%>%**  
 **ggplot**(**aes**(x=Month, y=Requested))**+**  
 **geom\_area**(fill="#69b3a2", alpha=.5) **+**  
 **geom\_line**(color="#69b3a2")**+**  
 **ylab**("Requested Units") **+**  
 **ggtitle**("2014 Requested Units by Month")**+**  
 **theme\_ipsum**()  
 q = **ggplotly**(q)  
 q



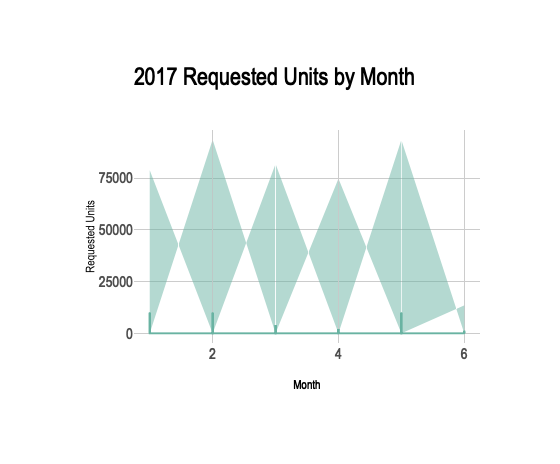
RequestedYear = **filter**(TableB, TableB**$**Year**==**2015)  
 q = RequestedYear **%>%**  
 **ggplot**(**aes**(x=Month, y=Requested))**+**  
 **geom\_area**(fill="#69b3a2", alpha=.5) **+**  
 **geom\_line**(color="#69b3a2")**+**  
 **ylab**("Requested Units") **+**  
 **ggtitle**("2015 Requested Units by Month")**+**  
 **theme\_ipsum**()  
 q = **ggplotly**(q)  
 q



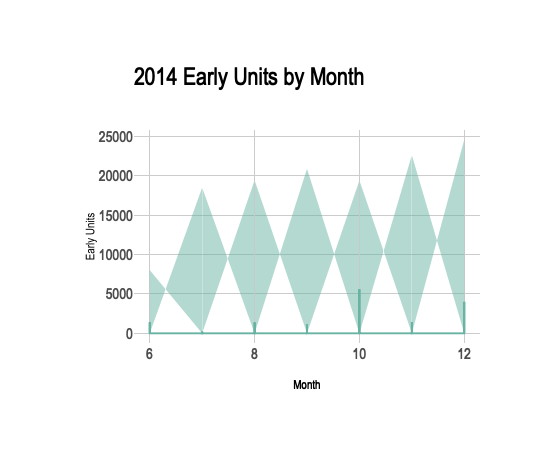
RequestedYear = **filter**(TableB, TableB**$**Year**==**2016)  
 q = RequestedYear **%>%**  
 **ggplot**(**aes**(x=Month, y=Requested))**+**  
 **geom\_area**(fill="#69b3a2", alpha=.5) **+**  
 **geom\_line**(color="#69b3a2")**+**  
 **ylab**("Requested Units") **+**  
 **ggtitle**("2016 Requested Units by Month")**+**  
 **theme\_ipsum**()  
 q = **ggplotly**(q)  
 q



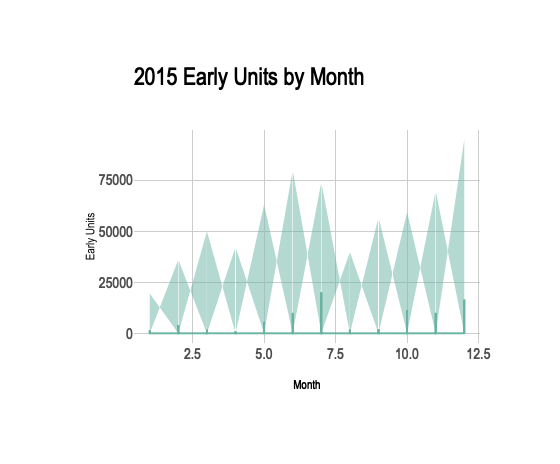
RequestedYear = **filter**(TableB, TableB**$**Year**==**2017)  
 q = RequestedYear **%>%**  
 **ggplot**(**aes**(x=Month, y=Requested))**+**  
 **geom\_area**(fill="#69b3a2", alpha=.5) **+**  
 **geom\_line**(color="#69b3a2")**+**  
 **ylab**("Requested Units") **+**  
 **ggtitle**("2017 Requested Units by Month")**+**  
 **theme\_ipsum**()  
 q = **ggplotly**(q)  
 q



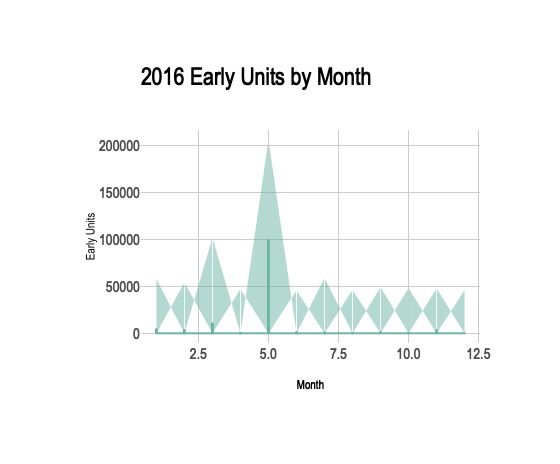
EarlyYear = **filter**(TableB, TableB**$**Year**==**2014)  
 q = EarlyYear **%>%**  
 **ggplot**(**aes**(x=Month, y=Early))**+**  
 **geom\_area**(fill="#69b3a2", alpha=.5) **+**  
 **geom\_line**(color="#69b3a2")**+**  
 **ylab**("Early Units") **+**  
 **ggtitle**("2014 Early Units by Month")**+**  
 **theme\_ipsum**()  
 q = **ggplotly**(q)  
 q



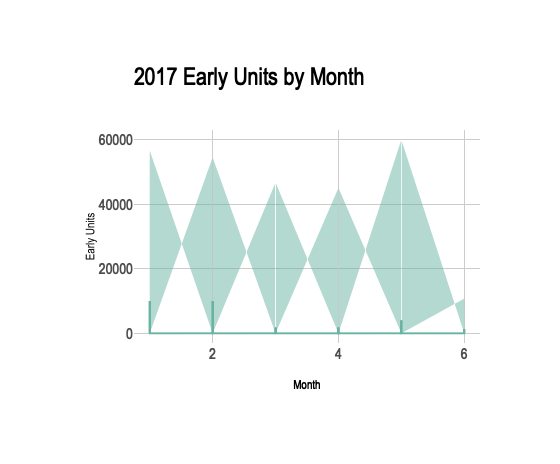
EarlyYear = **filter**(TableB, TableB**$**Year**==**2015)  
 q = EarlyYear **%>%**  
 **ggplot**(**aes**(x=Month, y=Early))**+**  
 **geom\_area**(fill="#69b3a2", alpha=.5) **+**  
 **geom\_line**(color="#69b3a2")**+**  
 **ylab**("Early Units") **+**  
 **ggtitle**("2015 Early Units by Month")**+**  
 **theme\_ipsum**()  
 q = **ggplotly**(q)  
 q



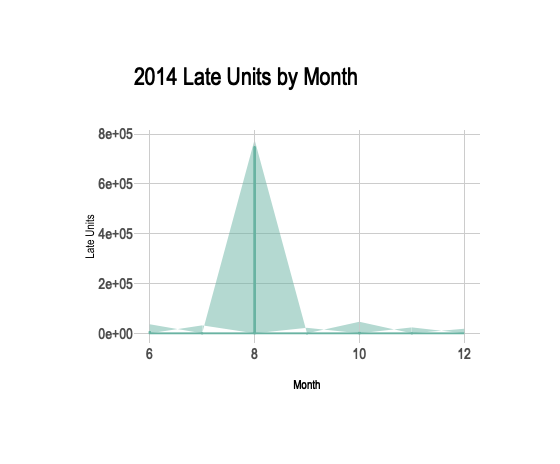
EarlyYear = **filter**(TableB, TableB**$**Year**==**2016)  
 q = EarlyYear **%>%**  
 **ggplot**(**aes**(x=Month, y=Early))**+**  
 **geom\_area**(fill="#69b3a2", alpha=.5) **+**  
 **geom\_line**(color="#69b3a2")**+**  
 **ylab**("Early Units") **+**  
 **ggtitle**("2016 Early Units by Month")**+**  
 **theme\_ipsum**()  
 q = **ggplotly**(q)  
 q



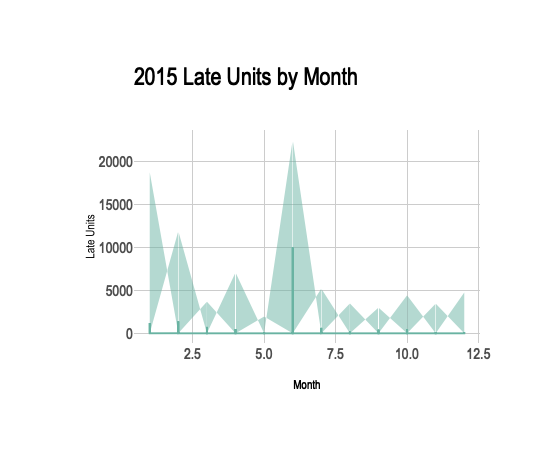
EarlyYear = **filter**(TableB, TableB**$**Year**==**2017)  
 q = EarlyYear **%>%**  
 **ggplot**(**aes**(x=Month, y=Early))**+**  
 **geom\_area**(fill="#69b3a2", alpha=.5) **+**  
 **geom\_line**(color="#69b3a2")**+**  
 **ylab**("Early Units") **+**  
 **ggtitle**("2017 Early Units by Month")**+**  
 **theme\_ipsum**()  
 q = **ggplotly**(q)  
 q



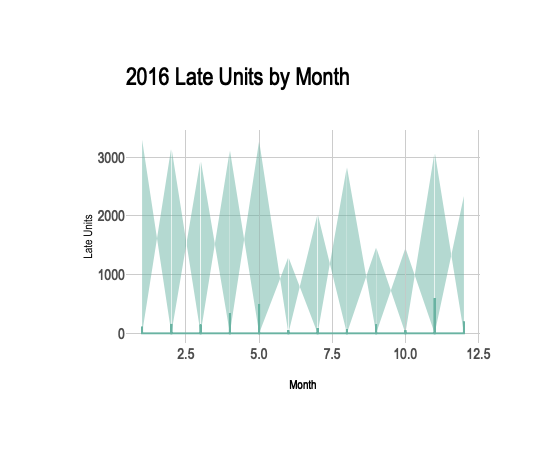
LateYear = **filter**(TableB, TableB**$**Year**==**2014)  
 q = LateYear **%>%**  
 **ggplot**(**aes**(x=Month, y=Late))**+**  
 **geom\_area**(fill="#69b3a2", alpha=.5) **+**  
 **geom\_line**(color="#69b3a2")**+**  
 **ylab**("Late Units") **+**  
 **ggtitle**("2014 Late Units by Month")**+**  
 **theme\_ipsum**()  
 q = **ggplotly**(q)  
 q



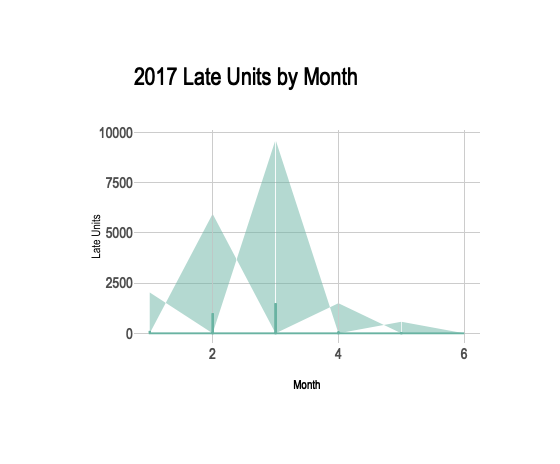
LateYear = **filter**(TableB, TableB**$**Year**==**2015)  
 q = LateYear **%>%**  
 **ggplot**(**aes**(x=Month, y=Late))**+**  
 **geom\_area**(fill="#69b3a2", alpha=.5) **+**  
 **geom\_line**(color="#69b3a2")**+**  
 **ylab**("Late Units") **+**  
 **ggtitle**("2015 Late Units by Month")**+**  
 **theme\_ipsum**()  
 q = **ggplotly**(q)  
 q



LateYear = **filter**(TableB, TableB**$**Year**==**2016)  
 q = LateYear **%>%**  
 **ggplot**(**aes**(x=Month, y=Late))**+**  
 **geom\_area**(fill="#69b3a2", alpha=.5) **+**  
 **geom\_line**(color="#69b3a2")**+**  
 **ylab**("Late Units") **+**  
 **ggtitle**("2016 Late Units by Month")**+**  
 **theme\_ipsum**()  
 q = **ggplotly**(q)  
 q



LateYear = **filter**(TableB, TableB**$**Year**==**2017)  
 q = LateYear **%>%**  
 **ggplot**(**aes**(x=Month, y=Late))**+**  
 **geom\_area**(fill="#69b3a2", alpha=.5) **+**  
 **geom\_line**(color="#69b3a2")**+**  
 **ylab**("Late Units") **+**  
 **ggtitle**("2017 Late Units by Month")**+**  
 **theme\_ipsum**()  
 q = **ggplotly**(q)  
 q



1. **Make visual and quantitative comparison across categories/segments in your data**
2. **Look for, describe and interpret the patterns you find as you go in the Rmarkdown. For each section, provide a description of the activity that is being completed, any issues, success, or key facts about the process being completed. Create a story-style EDA report that includes technical details that are generally interpretable by any audience. Note that not all information will be relevant to LOB’s direct interest, still include all key facts in the report. Your final presentation will trim your analysis down to key facts. Please include reference points to data and ensure all code is executable. This will require that you setup your project in a manner that share a data repository and code repository. The instructor should be able to go into the master branch and execute the Rmarkdown.**