Statistics Application against Capstone Data

Joe Marco

September 16th 2017

setwd("~/DataScienceFoundationsCourse/Capstone/Statistics")

### This section of the course challenges the students to ask some questions about our capstone data, and specifically practice the skills we have learned thus far (data visualization, cleansing, application of statistics)

#### Before getting into the details, lets load some of the libraries we will need and load our data and hide messages and warnings so the RMD file is not extensively long

suppressMessages(library(tidyverse))  
suppressMessages(library(ggplot2))  
suppressWarnings(suppressMessages((capstonedata <- read\_csv("capstonedata\_clean.csv"))))

## # A tibble: 895,204 x 24  
## X1 sku national\_inv lead\_time in\_transit\_qty forecast\_3\_month  
## <int> <int> <int> <dbl> <int> <int>  
## 1 1 1026827 0 7.872267 0 0  
## 2 2 1043384 2 9.000000 0 0  
## 3 3 1043696 2 7.872267 0 0  
## 4 4 1043852 7 8.000000 0 0  
## 5 5 1044048 8 7.872267 0 0  
## 6 6 1044198 13 8.000000 0 0  
## 7 7 1044643 1095 7.872267 0 0  
## 8 8 1045098 6 2.000000 0 0  
## 9 9 1045815 140 7.872267 0 15  
## 10 10 1045867 4 8.000000 0 0  
## # ... with 895,194 more rows, and 18 more variables:  
## # forecast\_6\_month <int>, forecast\_9\_month <int>, sales\_1\_month <int>,  
## # sales\_3\_month <int>, sales\_6\_month <int>, sales\_9\_month <int>,  
## # min\_stock <int>, potential\_issue <chr>, pieces\_past\_due <int>,  
## # perf\_6\_month\_avg <dbl>, perf\_12\_month\_avg <dbl>, local\_bo\_qty <int>,  
## # deck\_risk <chr>, oe\_constraint <chr>, ppap\_risk <chr>,  
## # stop\_auto\_buy <chr>, rev\_stop <chr>, went\_on\_backorder <chr>

#### Let's start by double checking we have no "NA" values and count unique skus (distinct?)

#the below code can be applies to all columns, for now we will just check all of our skuss (products)  
any(is.na(capstonedata$sku))

## [1] FALSE

nrow(distinct(capstonedata))

## [1] 895204

#### Next we want to explore the total products to found out how many specifically went on backorder vs did not go on backorder. The field "went\_on\_backorder" is what we will explore further

table(capstonedata$went\_on\_backorder)

##   
## No Yes   
## 887601 7602

#### Interesting... now we can see a small percentage of our total products went on backorder, we will want to focus specifically on the products that went on backorder, so lets create a dataframe for those 7602 products

productbackorders <- capstonedata[capstonedata$went\_on\_backorder == "Yes",]

#### I mentioned a small percentage of our total list of products went on backorder, but lets show the EXACT probability of backorder based off the data we have

# We can show the probability of backorder based off our data, to do this we will want to divide products that went on backorer over the total amount of products  
Probability\_backorder <- nrow(productbackorders) / nrow(capstonedata)  
Probability\_backorder

## [1] 0.008493036

# Great - we can see our output, but in decimal format, let's throw a function in there to convet the decimal to percent  
percent <- function(x, digits = 2, format = "f", ...) {  
 paste0(formatC(100 \* x, format = format, digits = digits, ...), "%")  
}  
# now we simply call our function "percent" on the output of our probability  
percent(Probability\_backorder)

## [1] "0.85%"

#### 

#### Calculate some averages around ALL Products (inventory and lead time). Let's not forget to add in standard deviation calculations as well so we can understand how spread out our averages are. Recall the output of standard deviation: A low standard deviation means that most of the numbers are very close to the average. A high standard deviation means that the numbers are spread out

mean\_inventory <- mean(capstonedata$national\_inv[!is.na(capstonedata$national\_inv)])  
mean\_leadtime <- mean(capstonedata$lead\_time[!is.na(capstonedata$lead\_time)])  
  
sd\_inventory <- sd(capstonedata$national\_inv[!is.na(capstonedata$national\_inv)])  
sd\_leadtime <- sd(capstonedata$lead\_time[!is.na(capstonedata$lead\_time)])  
  
mean\_inventory

## [1] 504.6036

sd\_inventory

## [1] 30766.77

mean\_leadtime

## [1] 7.857083

sd\_leadtime

## [1] 6.848415

# We can see our variance for inventory is HUGE - which makes sense (some products have very low inventory, some have very high), while the variance for leadtime is roughly close to the average

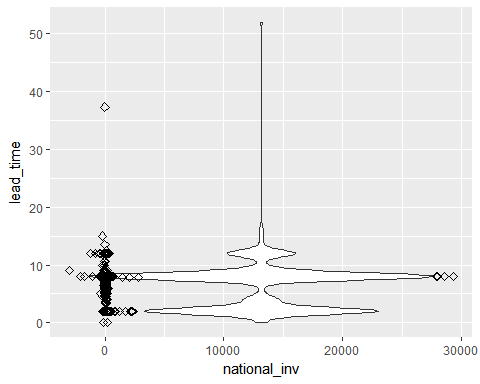
#### Can you find some trends (high, low, increase, decrease, anomalies)?

* Let's try a violin plot. We will compare national inventory with lead time. the below plot tells us that we see some groupings for products that went on backorder with lead time in the following groups (0-3, 7-10, 12-13). We then plot the poins on our averages and what we can see is that we have a average at or close to 0 for many of our products which basically means, I have 0 or negative inventory for products that went on backorder, which makes sense and means I may need to increase my inventory for those items.

ggplot(productbackorders, aes(x = national\_inv, y = lead\_time))+  
 geom\_violin() +  
 stat\_summary(fun.y=mean, geom="point", shape=23, size=2)

## Warning: Removed 1 rows containing non-finite values (stat\_ydensity).

## Warning: Removed 1 rows containing non-finite values (stat\_summary).

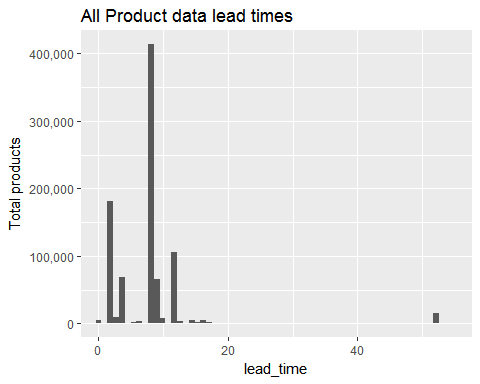


#### Now let's make a histogram looking at ALL products vs Backordered products and their associated lead times. We want to measure if their is some association with how long it takes products to be delivered and if that plays into backorders. With the below graphs that longer lead time does not necessarily == backorder. We see a routine spike in products that take

* Utilize ggplot to create some charts and graphs (geom\_histogram and or geom\_bar)

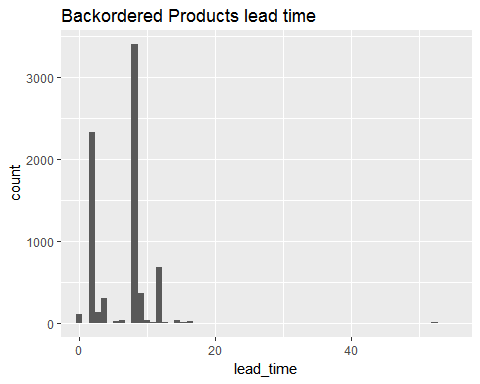
suppressMessages(library(scales))  
  
ggplot (capstonedata, aes(lead\_time)) +   
 geom\_histogram(binwidth = 0.9)+  
 coord\_cartesian(xlim = c(0, 55))+  
 ggtitle("All Product data lead times")+  
 scale\_y\_continuous(name="Total products", labels = comma)

## Warning: Removed 1 rows containing non-finite values (stat\_bin).



ggplot (productbackorders, aes(lead\_time)) +   
 geom\_histogram(binwidth = 0.9)+  
 coord\_cartesian(xlim = c(0, 55)) +  
 ggtitle("Backordered Products lead time")

## Warning: Removed 1 rows containing non-finite values (stat\_bin).



scale\_y\_continuous(name="Total backordered products", labels = comma)

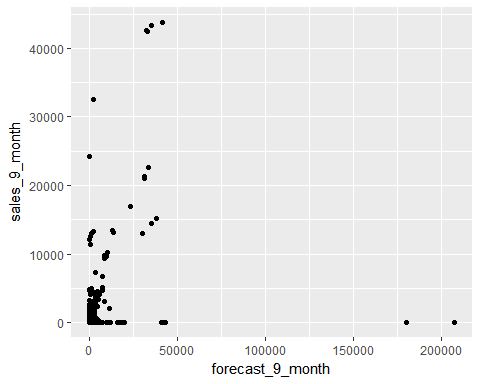
## <ScaleContinuousPosition>  
## Range:   
## Limits: 0 -- 1

#### Lets make a scatter plot looking at SALES VS FORECAST specifically for products that went on backorder. What we can gather from this is that there is some density close to 0, meaning our actual sales is close to $0 and our forecast of sales is close to $0 --- raises the question, WHY are we even making and producing those products?? Additionally but we can see a few data points where we are forcasting $175,000 or more in sales, we NEED to make sure we stock up these products to avoid backorder, otherwise we have a major impact to business revenue. Additionally, there are a few points in which our sales were 40,000 or more over the last 9 months, these appear to be the major revenue making products over the past 9 months that went on backorder, so we should focus on upping our inventory for these products as well

* ggplot (geom\_point)

ggplot(productbackorders, aes(x=forecast\_9\_month, y = sales\_9\_month)) +  
 geom\_point()

## Warning: Removed 1 rows containing missing values (geom\_point).



### Additional items to be investigated in final report

* Conditional probability that an item will go on backorder (refer to formula)
* Distribution of backorders per product type/category??
* Can you compare two related quantities?
* <http://www.sthda.com/english/wiki/visualize-correlation-matrix-using-correlogram>