sample \_exploratory data analysis

Kevin A. Ryan (JHUAPL)

Wednesday, October 28, 2015

This markdown document explores data from the [Rossman Store Sales](https://www.kaggle.com/c/rossmann-store-sales) competition hosted on Kaggle. The objective of this competition was to forecast sales using store, promotion and competitor data. Here is the extract from the competition:

Rossmann operates over 3,000 drug stores in 7 European countries. Currently, Rossmann store managers are tasked with predicting their daily sales for up to six weeks in advance. Store sales are influenced by many factors, including promotions, competition, school and state holidays, seasonality, and locality. With thousands of individual managers predicting sales based on their unique circumstances, the accuracy of results can be quite varied.

In their first Kaggle competition, Rossmann is challenging you to predict 6 weeks of daily sales for 1,115 stores located across Germany. Reliable sales forecasts enable store managers to create effective staff schedules that increase productivity and motivation. By helping Rossmann create a robust prediction model, you will help store managers stay focused on what’s most important to them: their customers and their teams!

Prior to building any models, the data we have been provided should be explored; it's probably not good practice to just throw in all variables into a model and hope for the best, tempting as it is to quickly get your name on the Kaggle leaderboard! The following could be done to explore, deal with and clean data.

1. Review raw data
2. Explore relationships
   * Between Sales (output) and input variables
   * Between input variables
3. Dealing with NA/missing values

Exploring relationships between input variables facilitates two things: identifying whether variables need to be dropped due to high correlation, and feature engineering. There are many guides and tips on feature engineering, but SAP have a simple guide that explains what it is and contains some nice tips: [Feature Engineering Tops for Data Scientists](http://go.sap.com/docs/download/2015/09/2a16f496-3f7c-0010-82c7-eda71af511fa.pdf). As they describe, "more data can increase model accuracy, which makes feature engineering an essential part of the modeling process."

Some people deal with these procedures separately, and some do these concurrently. For the purpose of this document, rather than deciding what to do with NA/missing values, I pose some alternatives to deal with them. Then, when exploring relationships between variables, I pose some ideas to create new features, though I am not an expert at feature engineering.

## The Data

Three data sets were provided. As usual, a train and test data set were provided. An additional data set for Store data was also provided. The variable names are quite descriptive, but some are explained further [here](https://www.kaggle.com/c/rossmann-store-sales/data). For the purpose of exploring the data, the test data was not used.

Some data preprocessing was applied to the data prior to analysis.

In the training data:

* Date was converted from "character" to "date"
* Years, Months and Days were extracted from the Date variable

In the store data:

* CompetitionOpenSinceYear, CompetitionOpenSinceMonth and "1" were combined to create a "CompetitionOpenDate" variable

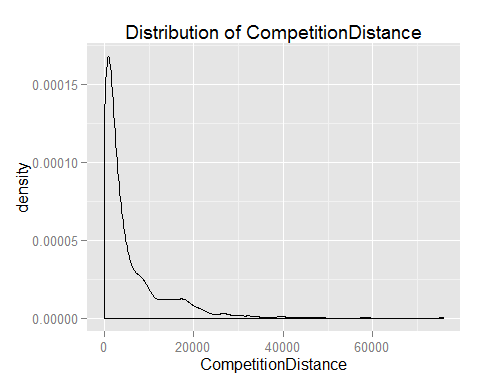
There are then several ways that R facilitiates easily to inspect the data, using the the functions head(), str() and summary(). On first glance, there there are NA values in store data which may require attention.

|  |  |
| --- | --- |
| Variable | NAs |
| CompetitionDistance | 3 |
| CompetitionOpenSinceMonth | 354 |
| CompetitionOpenSinceYear | 354 |
| Promo2SinceWeek | 544 |
| Promo2SinceYear | 544 |
| PromoInterval | 544 |

## Exploring NAs

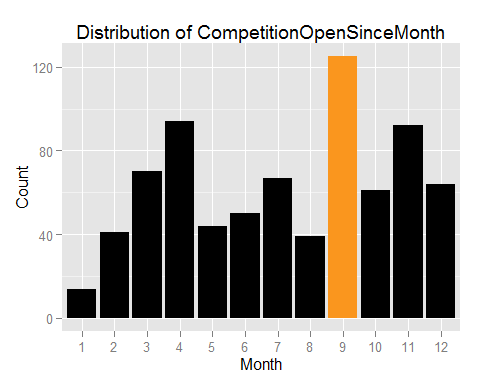
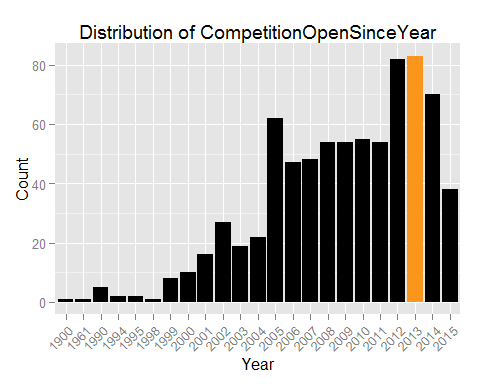
### CompetitionDistance

There are 3 NAs in the CompetitionDistance variable. Below is the distribution of ComptetitionDistance. Since there are only 3 missing values, assigning the mean of CompetitionDistance may be a better alternative than assigning it a zero (a common approach to dealing with missing values).



### CompetitionOpenSinceMonth and CompetitionOpenSinceYear

Both CompetitionOpenSinceMonth and CompetitionOpenSinceYear have 354 NAs. If a date was available, then both Year and Month would be available! From the bargraphs below, the most common month in the train data is September, and the most common year is 2013.

A simple approach could be to fill in NAs in each variable with the most common one, namely September and 2013. However, there are many other options available! NAs could be replaced by:

* finding out what the most common year-month combination is in the training set.
* given that the most common year was 2013, finding out what the most common month is.
* randomly assigning a year and month based on stratification of years and months.

What other ways could be used?

### Promo2SinceWeek and Promo2SinceYear

Similar to the CompetitionOpen variables, these two variables equally have 544 NAs. The NAs correspond to the 544 (48.7892377%) stores flagged as not participating in a "continuous and consecutive promotion".

table(store$Promo2)

##   
## 0 1   
## 544 571

When a store starts a promotion, they probably expect sales to increase (why have a promotion if you didn't?). Therefore, the indicator for Promo2 does not contain as much information as we would like. That's where we could use *Promo2SinceWeek* and *Promo2SinceYear*

##   
## 2009 2010 2011 2012 2013 2014 2015  
## 1 0 0 0 19 13 3 0  
## 5 0 3 3 0 33 0 0  
## 6 0 0 0 0 0 0 1  
## 9 0 0 14 0 0 0 0  
## 10 0 0 0 0 10 32 0  
## 13 0 34 0 0 0 0 0  
## 14 0 0 63 6 7 1 4  
## 18 0 4 14 1 2 6 2  
## 22 0 0 8 25 0 0 0  
## 23 0 0 0 0 1 1 3  
## 26 0 1 0 0 0 0 0  
## 27 0 0 2 5 4 0 0  
## 28 0 0 0 1 0 0 0  
## 31 7 0 0 0 37 0 0  
## 35 0 14 4 7 0 0 0  
## 36 0 0 0 0 10 0 0  
## 37 35 0 0 0 0 0 0  
## 39 1 5 0 0 0 0 0  
## 40 0 0 16 11 2 48 0  
## 44 0 1 0 2 0 0 0  
## 45 30 0 0 0 1 3 0  
## 48 0 1 4 4 0 0 0  
## 49 0 0 0 0 0 1 0  
## 50 0 1 0 0 0 0 0

The month promos began weren't supplied, unlike *CompetitionOpenSince* variables. But IS provided is even better! Where we assumed that CompetitionOpenDays were all at the start of the MONTH, we now only assume that promotions start at the beginning of the WEEK.

## Promo2SinceWeek Promo2SinceYear DaysSinceYearStart Promo2YearStart  
## 1 NA NA NA <NA>  
## 2 13 2010 85 2010-01-01  
## 3 14 2011 92 2011-01-01  
## 4 NA NA NA <NA>  
## 5 NA NA NA <NA>  
## 6 NA NA NA <NA>  
## Promo2StartDate  
## 1 <NA>  
## 2 2010-03-27  
## 3 2011-04-03  
## 4 <NA>  
## 5 <NA>  
## 6 <NA>

(There's a bit of data munging and feature extraction here, but I've included it in this section because it was initiated as part of examining features with NA values.)

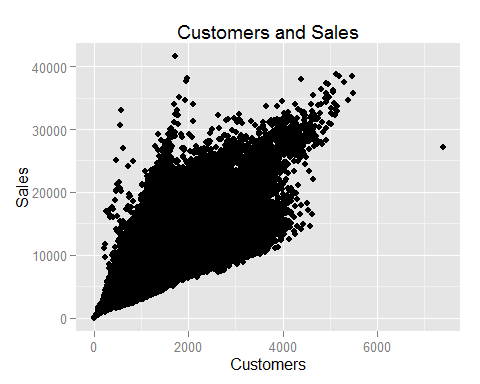
## Exploring relationships between Sales and input variables

Store data and variables are also input variables when we build the models. These can be combined before we explore the data. There are also many instances where sales is 0 because stores have not been open.

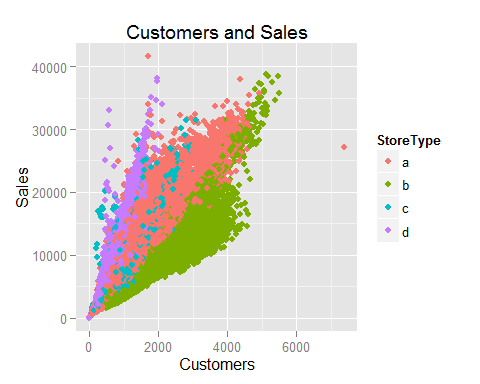
train <- left\_join(train, store, by = "Store") # dplyr  
train <- filter(train, Open == 1) # dplyr

### Sales and Customers

Only the train data set contains the Customers variable. Understandably, the Customers would explain a huge amount of the variance in Sales, but it's also a variable that wouldn't be available until that date. We can still have a look at the data though. The graph below shows the expected positive correlation between the number of customers and sales. It also shows some outliers. We can look at those later.

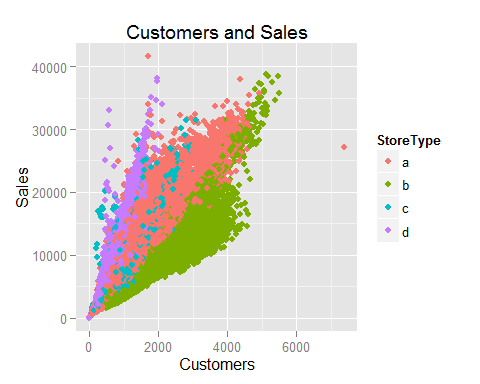


So, there's a positive correlation, but there appears more to this than meets the eye. If we see the points as a cone, there appears to be a sub-group that sticks close to the top-left line, and a sub-group that sticks close to the bottom-right line, and one that runs through the middle of the cone, stretching farther than two aforementioned lines. We could investigate whether there is a pattern through some of the categorical variables available to us.



The segregation of sales and customers by the StoreType variable appears to explain some of the observation. StoreType *d* tends to make up the top-left line, and StoreType *b* makes up the bottom-right collection of sales and customers. This means that for the same level of Sales, StoreType *d* requires fewer customers than StoreType *b*. Through the middle of the cone is a little less clear. We can try to separate this out by putting some lines through the graph.

## geom\_smooth: method="auto" and size of largest group is >=1000, so using gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the smoothing method.  
## geom\_smooth: method="auto" and size of largest group is >=1000, so using gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the smoothing method.  
## geom\_smooth: method="auto" and size of largest group is >=1000, so using gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the smoothing method.  
## geom\_smooth: method="auto" and size of largest group is >=1000, so using gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the smoothing method.



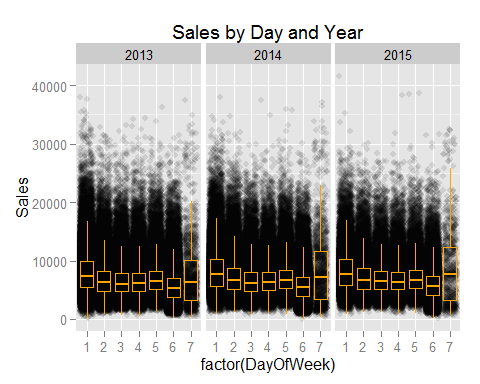
So this completes the analysis of StoreType. It's possible to say that StoreType is an important variable, but this appears only true when there are more Customers, as StoreType *a* and *c* appear quite mixed until around 2000 Customers.

But customers can't be used as an input variable anyway. So should we be predicting Customers as from the data and then subsequently use that as an input variable?

### Sales and DayOfWeek

What are sales like on different days of the week?

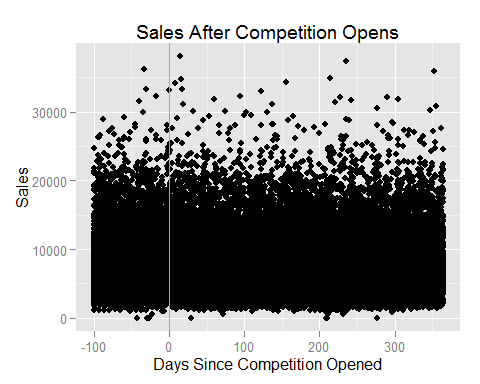
The variability of Sales on Sunday appears to be larger across all years.



### Sales and days since competition opened

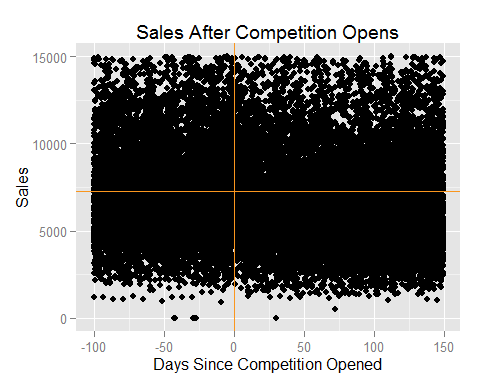
From an economic perspective, the month or a year that a competitor opens may not individuallt have a persistent impact on sales, *per se*. For example, if a competitor opened in January (of any year, or even January of 1989), it would not make much economic sense if it had an impact on a store on 13th February 2015. Even if there were for some stores, this would be diluted over 1,115 stores. What may make more economic sense is the number of days *since* a competitor opened.

The graph below shows the sales of each store prior to and after competition begins for on year. The vertical line at CompetitionDays = 0 denotes the day that competition opens. The smooth line running at approximately Sales = 7,200 is the average sales.



The graph does show some drop in Sales for about the first 50 days after competition opens, but it is visually hard to discern whether the effect is persistant. We could try to examine this a little closer.

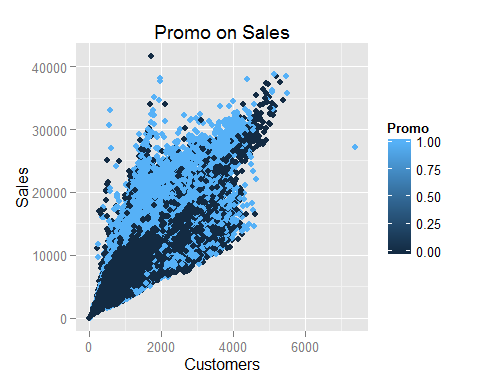
Based on the smooth line, representing the average Sales, having a competition opening does seem to have a negative effect on Sales.



Is this a good variable to include? Should a bivariate be created for before and after Competition begins? Should we create a factor variable for prior to competition beginning, between 0 and 100 days, and after 100 days?

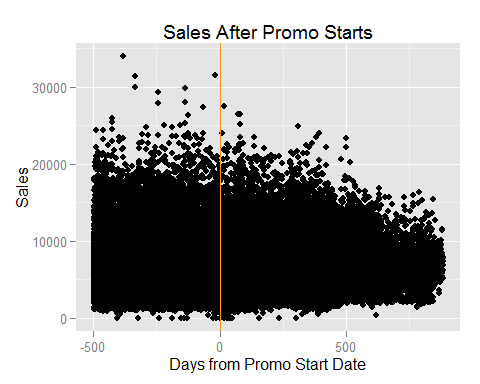
### Sales and Promo

Sales and Promo is quite straight forward. There are no missing values, so we can just throw this into a model as is.

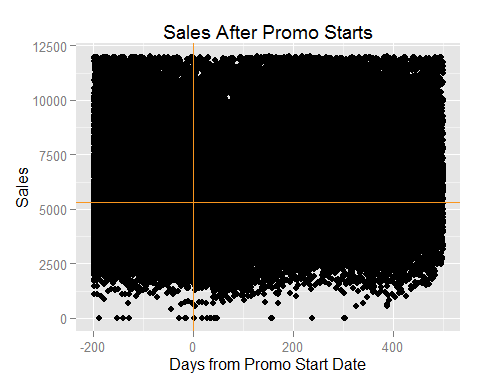


### Sales and Promo2

A date for when Promo2 started was created when NAs were examined. Now the question can be asked, "do Sales improve after promotions begin?"



It appears that Promo improves Sales. Zooming in again shows this a little clearer. Interestingly, there's a dip in sales just prior to promo, which persists for about 15 days before rising again.



What feature can be created from this information? Should a bivariate be created for before and after promo date? Should a window be used again to create three factors?

## Exploring relationships between input variables

### Store Type and Assortment

We're not provided information on what Store Type and Assortment are, but they're definitely not the same:

##   
## a b c  
## a 286053 0 171024  
## b 6409 8212 942  
## c 58561 0 54417  
## d 93886 0 164888

##   
## a b c  
## a 0.338768013 0.000000000 0.202541000  
## b 0.007590077 0.009725341 0.001115596  
## c 0.069352860 0.000000000 0.064445187  
## d 0.111187695 0.000000000 0.195274233

Should we combine these to make a new StoreTypeAssortment variable? If so, does this have an effect on modeling?

## Summary

Some questions raised are summarised below:

* Should we be predicting Customers as from the data and then subsequently use that as an input variable?
* How should we deal with missing CompetitionOpenSince Years and Months?
  + Finding out what the most common year-month combination is in the training set?
  + Given that the most common year was 2013, finding out what the most common month is?
  + Randomly assigning a year and month based on stratification of years and months?
* Should we combine StoreType and Assortment to make a new StoreTypeAssortment variable?
* Should a new variable be created for days after competition opens?
  + Should a bivariate be created for before and after Competition begins?
  + Should we create a factor variable for prior to competition beginning, between 0 and 100 days, and after 100 days?
* What feature can be created from Promo2SalesDate?
  + Should a bivariate be created for before and after promo date?
  + Should a window be used again to create three factors?