# Yelp Challenge Data Set - Variogram Modeling of Location and Checkins

*Kate Stohr*

*November 2, 2015*

### Introduction

This research explores the relationship between checkins and location in the Yelp Data Set Challenge using variogram modeling with the "gstats" package in R.

Often people need to know the most active areas of a city. This is useful for planning canvasing routes, such as for sales outreach, advocacy campaigns or political campaigns, identifing the best locations for an event, or evaluating foot traffic and commercial real estate values.

Because checkins are a reflection of an individual at a specific location at a time, it's a reasonable proxy for foot traffic. In this, project I asked the question: Is there a correlation between location and the number of checkins such that it is possible to reasonably predict the number of checkins at a given location?

Note: Time did not permit analysing time series models of the same data. Although, that might be a logical next step.

Data source: http://www.yelp.com/dataset_challenge

## Methods and Data

Exploring the data, it is not clear over what period of time the checkin data was collected. There is no date time stamp associated with the data.

The other datasets contain cumulative data. The reviews data was collected between the period of Oct. 10, 2004 and Jan. 8, 2015 (10 years and 3 months). Tip data starts in April 15, 2009 and ends on Jan 22, 2015. This roughly corresponds to the period that yelp made its mobile apps available (2008 onwards). However, Yelp didn't introduce it's checkin feature on the iphone until Jan 15, 2010.

For the purposes of this analysis, we're going to assume that the checkin data is cumulative over a nearly five year period. The data is stochastic, and binned into time windows (hour/day).

### Mapping the data

After loading and processing the data (view .rmd file for code), I first explored the data by using k means clustering to seperate the locations by city. This resulted in a clean subset of data. I select Pittsburgh, as a city to model as it had approximately an average amount of data as compared to the other cities.

```
saveRDS(geo.cluster, file.path(dir.data, "geo.cluster.rds"))
```

After merging the Pittsburgh checkin data with the business data by "business_id", I mapped the business information using leaflet, which allows the mapped data to be more readily explored than other mapping tools.

Overlaying businesses with checkins against all Pittsburgh businesses in the data set, one can see that checkins tend to align with traffic arteries. This suggests that there is a correlation between location and checkins.

```
#Map of Pittsburgh checkin data overlaid on all businesses
m5_checkins <- leaflet() %>%
  addTiles() %>%  # Add default OpenStreetMap map tiles
  addCircleMarkers(lng=bus_cl_5$longitude, lat=bus_cl_5$latitude, popup=bus_cl_5$name, color = c('blue'
  addCircleMarkers(lng=merge_cl_5$longitude, lat=merge_cl_5$latitude, popup=merge_cl_5$name, color = c(
print(m5_checkins)
```

## Exploratory Modeling

To establish a baseline, I used some common models to get a sense of the relationships in the data.

*GLM and Random Forests Models*

```
## note: only 1 unique complexity parameters in default grid. Truncating the grid to 1 .
```

The GLM marginally better, but with only 0.217% accuracy. The model is producing results that have a much lower standard deviation than the original set. In addition, it skews high on the low end of the scale, giving a false sense of potential traffic at a location.

Including all factors in the model, shows that a few times do result in statistically significant outcomes, such as midnight on Saturday, for example. So the low accuracy could also be result of a lack of datapoints for certain time periods.

```
#summary of model results
#summary(prediction6)
#summary(training$total)
percent(acc_test6)
```

```
## [1] "0.217%"
```

```
percent(acc_test7)
```

```
## [1] "1.59%"
```

```
sd(prediction6)
```

```
## [1] 10.33
```

```
sd(training$total)
```

```
## [1] 132.1
```

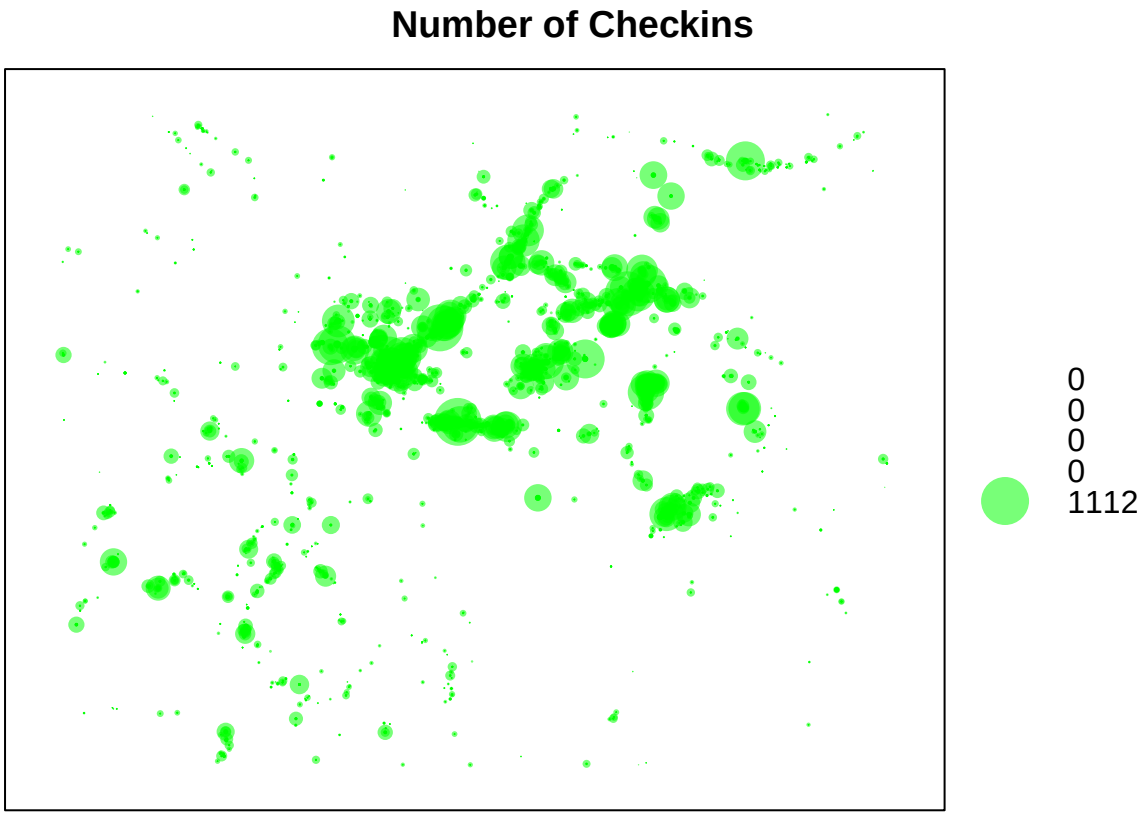
Neither Random Forest or GLM models perform well. The p-values are very high indicating that the results may as well be random. This is likely because the model is not recognizing the spatial/time nature of the data.

*Variogram Model*

To better model the spatial nature of the data, I turned to a variogram model which allows the points (locations) to be projected onto a grid and then values for the variable to be assigned to areas on the grid accordingly.

```
##Convert time period to POSIXlt NOTE: NOT PERFORMED. RESULTED IN ERRORS. WILL TEST SPATIAL CORRELATION
##times<-dt_melt$time
##times<-strptime(times, format = "%H.%w")
#NOTE although the format is specified correctly, the timestamp is missing day info and time info is in
```

With the data projected, it's possible to plot a 'bubble' map which shows the value of the checkin variable at various places on the map. High checkin numbers appear to be clusetered. In other words, the number of checkins at a location appears inversely proportional to the distance between nearby points.




Next I create a 'grid' against which to plot the data. Coordinates of the grid are obtained from the boundary points of the data set. A map of the points against the boundary diagonal shows the data fits the grid.

To build the model, first create a variogram of the projected data.

```
auto.vgm = autofitVariogram(checkins~1, training) #variogram of checkins
plot(auto.vgm)
#nugget = 114, #sill=114, range=2.7
```

The variogram shows that most locations have a variance of 2-4 checkins within an 8km distance of each other. The number of checkins taper off the further the points are away from each other. Given this, the model needs to be adjusted to take into account that the checkins are clustered around the city center. Taking the log of the checkin values and the squareroot of the distance value will result in a more linear model that better reflects this inverse relationship.

```
#create model using lof of checkins and sqrt of distance to city center.
log.vgm = variogram(log(checkins)~sqrt(dist), training, cloud=TRUE) #variogram of log
plot(log.vgm)
```

## Results

While not a strong model, the updated model comparing distances to the center of town performs better. Sill the results as measured by comparing the variogram model to the training sample do not show strong correlation after 2km, and high variability at short distances. This indicates that data is not being modeled well or may be too sparse in locations to accurately reflect a trend.

## Discussion

Modeling the data using cluster anlaysis (kmeans, etc.) may improve the results. However, businesses next to each other often have very different levels of checkins, indicating that location alone is not the driving factor influencing checkins.

To improve the model, it may be good to path the traffic arteries and use distance from those line paths as a factor in the model.

In addition, currently the time data is being read as a factor. Converting that to a time series, and modeling a time variogram as well as spatial variogram would likely improve performance and *may* allow for some predictions of time series data.