Quantifying crowd size with mobile phone and Twitter data - Final Report

Ogi Moore and Connor Smith

12/5/2016

Introduction

We elected to replicate the findings of Federico Botta, Helen Susannah Moat, and Tobias Preis's paper on Quantifying crowd size with mobile phone and *Twitter* data. In the paper, they look at a number of soccer games with a known attendence and known phone, internet and twitter acitivity; and they evaluate the similar phone and internet and twitter acitivty in comparison to a number of flights over a several week period.

Data Import

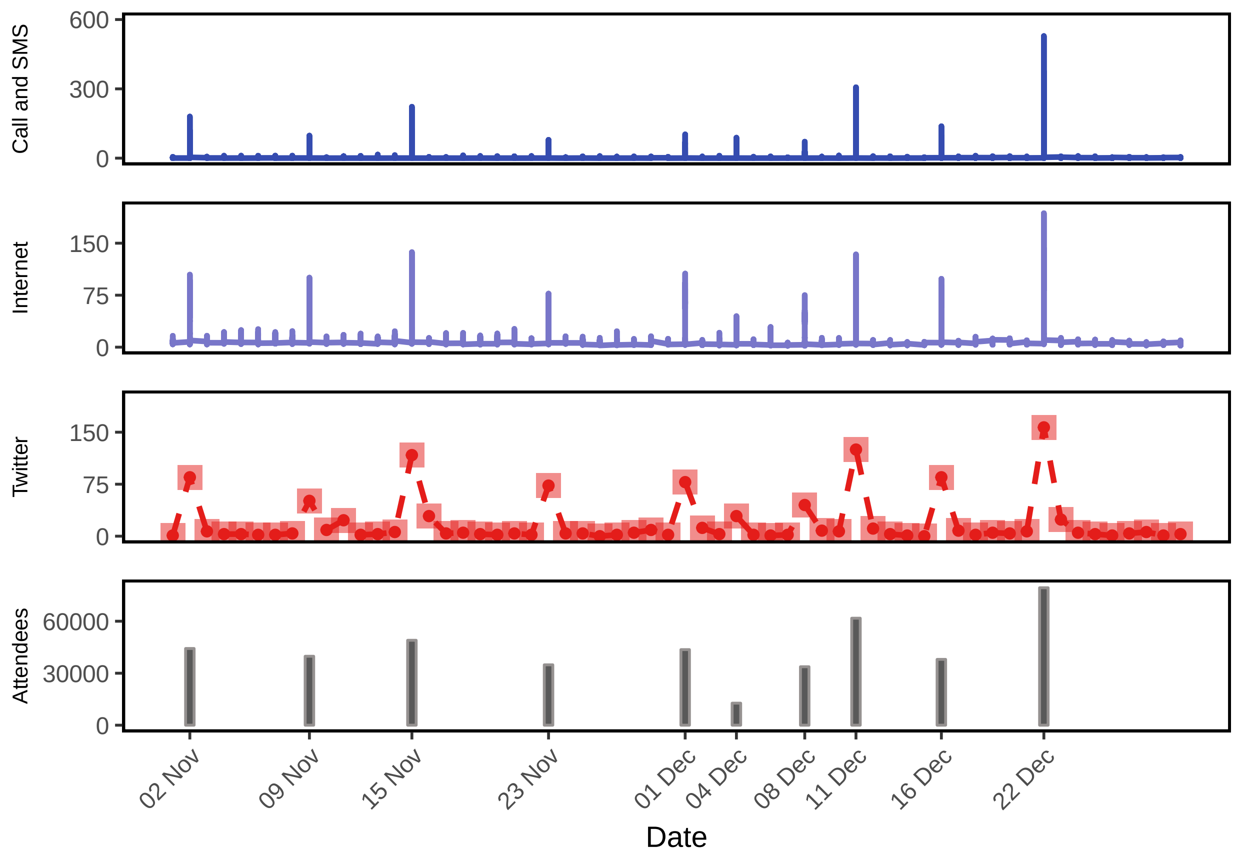
The data is in very good shape, but we do need to tell `R` that the timestamps are in-fact date-time objects, and not just generic strings. The source of the data is explained in the Appendix.

```r
san_siero.attendees = read.csv('./data/Attendees_San_Siro.csv')
san_siero.phone_data = read.csv('./data/San_Siro_Mobile_Phone_Data.csv')
san_siero.twitter_data = read.csv('./data/San_Siro_Twitter_Data.csv')
linate.data = read.csv('./data/Linate_Data.csv')
linate.flight_schedule = read.csv('./data/Linate_Flights_Schedule.csv')

# Converting to dates
san_siero.phone_data$Timestamp = as.Date(strptime(san_siero.phone_data$Timestamp,
                                                  "%Y-%m-%d %H:%M:%S"))
san_siero.twitter_data$Timestamp = as.Date(san_siero.twitter_data$Timestamp)
san_siero.attendees$Date = as.Date(san_siero.attendees$Date)
```

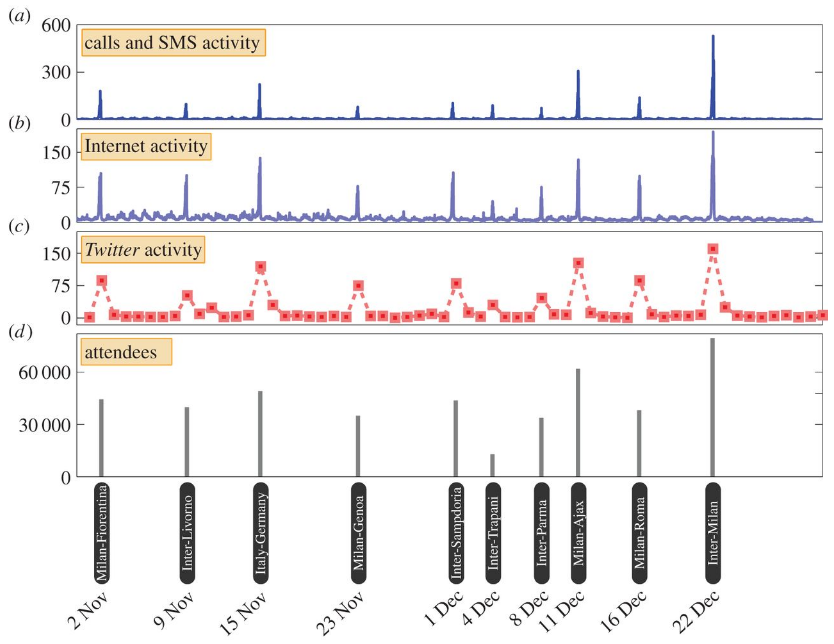
Soccer Game Attendence

The soccer data includes 3 raw files, one containing the dates of the soccer games as well as the attendence. The phone data includes a measure of calls / SMS activity and Internet activity with a timestamp. We will show later that we are only interested in the date and have no interest in the time. The third file includes a measure of twitter activity with an associated date.

First, the authors created a plot showing the various measures (calls and SMS data, internet activity, *Twitter* activity, and attendance) vs date. We are able to replicate this plot.



This chart shows that the activity data tends to peak in relation to the attendee data. For comparison, here is the plot attached to the paper:



Grouping

To analyze this data, we need to merge the data into a common dataframe, where we are interested only in the maximum value of the calls / SMS activity and Internet activity numbers for only days with a soccer match. The authors assume that the highest mobile phone activity on the day of a soccer match would occur during the soccer match. Because of this assumption, I can disregard the time component of the raw data, as I am looking for the maximum value in a given day.

```r
san_siero.daily_data <- aggregate(san_siero.phone_data$Calls.and.SMS.Activity,
                                  by=list(Category=san_siero.phone_data$Timestamp),
                                  FUN=max)
san_siero.daily_data <- plyr::rename(san_siero.daily_data, c("x"="Calls.and.SMS.Activity"))
san_siero.daily_data$Internet.Activity = aggregate(san_siero.phone_data$Internet.Activity,
                                         by=list(Cateogry=san_siero.phone_data$Timestamp),
                                         FUN=max)$x
san_siero.daily_data$Twitter.Activity = san_siero.twitter_data$Twitter.Activity
san_siero.daily_data <- plyr::rename(san_siero.daily_data, c("Category"="Date"))
soccer_data <- merge(san_siero.daily_data, san_siero.attendees,
                                        by="Date")

kable(head(soccer_data),
            format='pandoc',
            caption='Soccer Game Data',
            centering=TRUE)
```

Table 1: Soccer Game Data

Date	Calls.and.SMS.Activity	Internet.Activity	Twitter.Activity	Attendees.at.San.Siro
2013-11-02	180.050	104.640	85	44261
2013-11-09	97.693	100.350	51	39775
2013-11-15	222.520	137.080	117	49000
2013-11-23	79.276	77.290	73	34848
2013-12-01	102.930	106.180	78	43607
2013-12-04	88.803	44.783	29	12714

Linear Modeling

The authors then performed a linear regression comparing calls and SMS activity, Internet activity, *Twitter* activity to the number of attendees. With `R` we are able to perform the same linear regression analysis with ease.

```r
attendees_v_phone <- lm(soccer_data$Attendees.at.San.Siro ~
                          soccer_data$Calls.and.SMS.Activity)
attendees_v_internet <- lm(soccer_data$Attendees.at.San.Siro ~
                             soccer_data$Internet.Activity)
attendees_v_twitter <- lm(soccer_data$Attendees.at.San.Siro ~
                            soccer_data$Twitter.Activity)
```

Now we can compare the values we calculated vs. those in the paper.

```r
lm_paper_results <- c(0.771, 0.937, 0.855)
lm_duplication_results <- c(round(summary(attendees_v_phone)$adj.r.squared, 3),
                            round(summary(attendees_v_internet)$adj.r.squared, 3),
                            round(summary(attendees_v_twitter)$adj.r.squared, 3))
```

```
lm_results <- data.frame(lm_paper_results,
                         lm_duplication_results,
                         row.names=c('Calls and SMS Data',
                                     'Internet Activity',
                                     'Twitter Activity'))
kable(lm_results,
            format='pandoc',
            centering=TRUE,
            caption='Linear Regression R^2^ Values',
            col.names = c('Published Results', 'Duplication Results'))
```

Table 2: Linear Regression R^2 Values

	Published Results	Duplication Results
Calls and SMS Data	0.771	0.771
Internet Activity	0.937	0.937
Twitter Activity	0.855	0.855

We can see that our R^2 values match up exactly. The results show that the internet activity mobile phone data was the most accurate predictor of crowd size, having the highest R^2 value.

Spearman Correlations

Next, the authors evaluate the correlation to see how the relationship holds up to a non-parametric analysis.

```
cor_paper_results <- c(0.927, 0.976, 0.924)
cor_duplication_results <- c(round(cor(soccer_data$Attendees.at.San.Siro,
                                       soccer_data$Calls.and.SMS.Activity,
                                       method='spearman'), 3),
                             round(cor(soccer_data$Attendees.at.San.Siro,
                                       soccer_data$Internet.Activity,
                                       method='spearman'), 3),
                             round(cor(soccer_data$Attendees.at.San.Siro,
                                       soccer_data$Twitter.Activity,
                                       method='spearman'), 3))
cor_results <- data.frame(cor_paper_results,
                          cor_duplication_results,
                          row.names=c('Calls and SMS Data',
                                      'Internet Activity',
                                      'Twitter Activity'))
kable(cor_results,
            format='pandoc',
            caption='Spearman Correlation Values',
            col.names = c('Published Results', 'Duplication Results'))
```

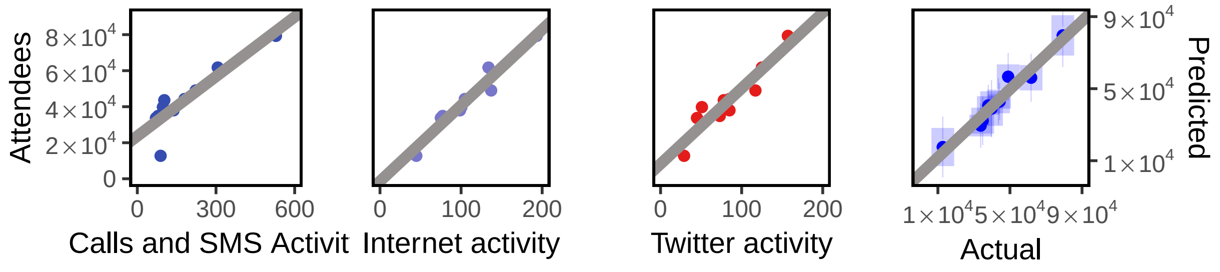
Table 3: Spearman Correlation Values

	Published Results	Duplication Results
Calls and SMS Data	0.927	0.927
Internet Activity	0.976	0.976
Twitter Activity	0.924	0.924

Our spearman correlation values match up with the published values as well. These results also indicate that Internet activity is the best guage for crowd size at the soccer games. Since we have come to the same conclusions, we can replicate their plot.

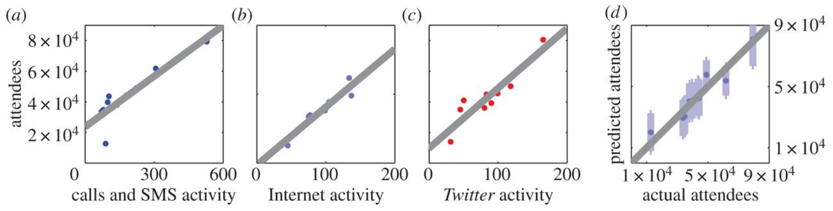
Results

Four plots were made by the authors to show the linear relationship that each of the measures of activity has with the amount of people in the area. The first three plots were very straight forward, comparing values for activity to attendance at soccer games. The fourth plot was a little tricky.



For the fourth plot, the authors said they performed a *leave-one-out cross-validation* analysis, which we are able to do in `R`. Unfortunately we were unable to get confidence interval results based on the 'LOOCV' model method, so we then resorted to doing a generic linear model. When we did the standard linear model (`lm`) method, we noticed the predicted numbers were the same as the predicted output from the *leave-one-out cross-validation*, but `R` provided confidence intervals. The confidence intervals match up with the plot published in the paper.

For comparison, here is the plot attached to the paper:



Conclusion

From looking at calls/sms data, internet activity and *Twitter* activity, it appears that internet activity is the best predictor to crowd sizes, although all methods show strong linear relationships and correlations. Also, when we compare the projected crowd size vs. the actual crowd size, and evaluate the 95% confidence interval of what the actual crowd size is based on the predicted value, we notice that the best fit curve falls within the 95% confidence interval for all the points.

Airport Crowd Approximations

In the airport dataset the authors took a different method to approximating the crowd size. They approximated the number of people at the airport based on the number of arriving and departing flights. More specifically, they summed the flights departing in the two hour window following the time of interest and the number of incoming flights in the hour leading up to the time of interest, and used the result as a relative approximation of crowd size. The raw data provides the number of flights arriving and departing the airport on an hour by hour basis over a 1 week period.

```
kable(head(linate.flight_schedule),
           format='pandoc',
           caption="Linate Flight Schedule Data",
           centering=TRUE)
```

Table 4: Linate Flight Schedule Data

Timestamp	Departures	Arrivals
2014-05-05 00:00:00	0	0
2014-05-05 01:00:00	0	0
2014-05-05 02:00:00	0	0
2014-05-05 03:00:00	0	0
2014-05-05 04:00:00	0	0
2014-05-05 05:00:00	0	0

The authors also provide a relative quantity of calls and SMS activity and internet activity, as well as *Twitter* activity

```
kable(head(linate.data),
           format='pandoc',
           caption='Linate Phone Data',
           centering=TRUE)
```

Table 5: Linate Phone Data

Timestamp	Calls.and.SMS.Activity	Internet.Activity	Twitter.Activity
2013-11-01 00:00:00	133.940	1599.8	0
2013-11-01 01:00:00	87.867	1247.0	0
2013-11-01 02:00:00	134.630	1210.1	0
2013-11-01 03:00:00	41.017	1159.6	0
2013-11-01 04:00:00	100.430	1575.1	2
2013-11-01 05:00:00	463.340	3730.6	0

The reader may notice here that the dates of the time-stamps do not match up (they are off by 6 months).

The authors explain that the way they compensate for this is that they line up the days of the week from the flights data, and assume that the flight schedule remains fairly consistent for each day of the week (the airport activity at any given time on a Monday in November will be similar to the same time on a Monday in May). They excluded November 1st, 2nd, and 3rd, as well as December 30th and 31st as they were holidays.

Grouping

As the authors decided to look at the number of incoming flights up to an hour before, and the number of departing flights for two hours following, this made for having to modify the raw data and doing a calculation for 'total flights'. Furthermore, the authors then decided to average the calls and sms activity, internet activity and twitter activity associated with any given hour and weekday over the two month span. This kind of data wrangling is outside of our skill set in `R`, however we were able to make the modifications necessary in `Python`. Should a reviewer wish to run this python code, they will need the Pandas library installed. The `Python` code outputs a file titled 'Linate_wrangled.csv' which we will import into `R` to generate our statistics with, the notebook is configured such that the following code cell will not be executed (in case the reviewer does not have Python with the Pandas library installed). The python code outputs a file titled `Linate_wrangled.csv`, which we then import into `R` further along.

```python
import numpy as np
import pandas as pd
import datetime as dt
linate_sched_data = pd.read_csv('./data/Linate_Flights_Schedule.csv',
                                parse_dates=[0],
                                infer_datetime_format=True,
                                index_col=0)
linate_sched_data['Day'] = linate_sched_data.index.weekday_name
linate_sched_data['Hour'] = linate_sched_data.index.hour

linate_sched_data['Flights'] = np.roll(linate_sched_data['Departures'], -2) + \
                               np.roll(linate_sched_data['Departures'], -1) + \
                               np.roll(linate_sched_data['Arrivals'], 1)
linate_flight_data = linate_sched_data.groupby(['Day', 'Hour']).sum()
linate_flight_data.drop(['Arrivals', 'Departures'], inplace=True, axis=1)
linate_phone_data = pd.read_csv('./data/Linate_Data.csv',
                               parse_dates=[0],
                               infer_datetime_format=True)

days_to_skip = pd.to_datetime(['2013-11-01',
                               '2013-11-02',
                               '2013-11-03',
                               '2013-12-30',
                               '2013-12-31']).date
linate_phone_data = \
    linate_phone_data[linate_phone_data['Timestamp'].dt.date.isin(days_to_skip) == False]
linate_phone_data.set_index('Timestamp', drop=True, inplace=True)
linate_phone_data['Day'] = linate_phone_data.index.weekday_name
linate_phone_data['Hour'] = linate_phone_data.index.hour
linate_avg_phone_data = pd.DataFrame(linate_phone_data.groupby(['Day', 'Hour'],
                                                               sort=True).mean())
result = pd.concat([linate_flight_data, linate_avg_phone_data], axis=1)
result.to_csv('./data/Linate_wrangled.csv')
```

We can now import the wrangled CSV file that our python code generated and move on with our analysis. Here is what the head of that dataframe looks like.

```r
flight_data <- read.csv('./data/Linate_wrangled.csv')
kable(head(flight_data),
            format='pandoc',
            caption='Linate Flight Data Cleaned Up',
            centering=TRUE)
```

Table 6: Linate Flight Data Cleaned Up

Day	Hour	Flights	Calls.and.SMS.Activity	Internet.Activity	Twitter.Activity
Friday	0	2	1296.475	5226.762	0.125
Friday	1	0	2104.547	6965.100	0.125
Friday	2	0	2974.243	8148.863	0.000
Friday	3	0	3546.717	9635.212	0.000
Friday	4	10	4371.842	10568.325	0.375
Friday	5	22	4768.887	11925.612	2.000

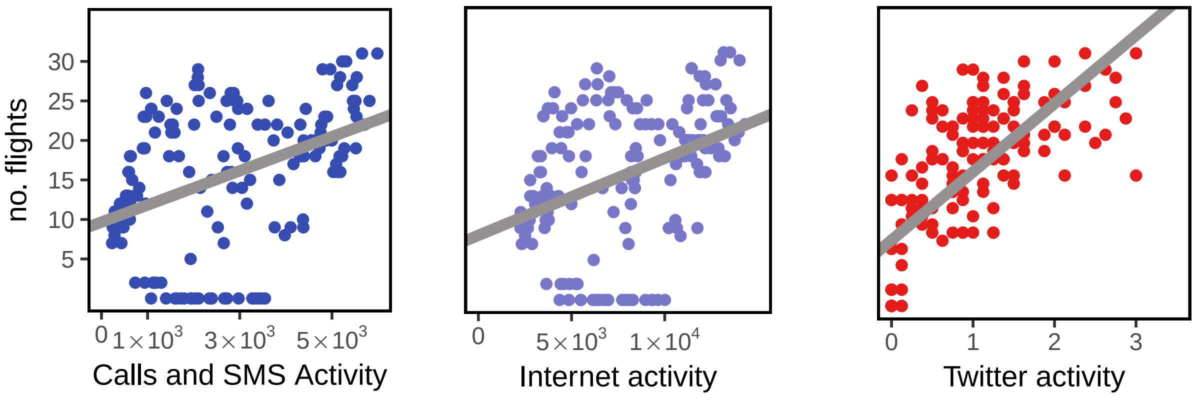
Linear Analysis

```r
lm_paper_results <- c(0.175, 0.143, 0.510)
flights_v_phone <- lm(flight_data$Flights ~
                      flight_data$Calls.and.SMS.Activity)
flights_v_internet <- lm(flight_data$Flights ~
                         flight_data$Internet.Activity)
flights_v_twitter <- lm(flight_data$Flights ~
                        flight_data$Twitter.Activity)

lm_duplication_results <- c(round(summary(flights_v_phone)$adj.r.squared, 3),
                            round(summary(flights_v_internet)$adj.r.squared, 3),
                            round(summary(flights_v_twitter)$adj.r.squared, 3))

lm_results <- data.frame(lm_paper_results,
                         lm_duplication_results,
                         row.names=c('Calls and SMS Data',
                                     'Internet Activity',
                                     'Twitter Activity'))

kable(lm_results,
            format='pandoc',
            centering=TRUE,
            caption='Linear Regression R^2^ Values',
            col.names = c('Published Results', 'Duplication Results'))
```

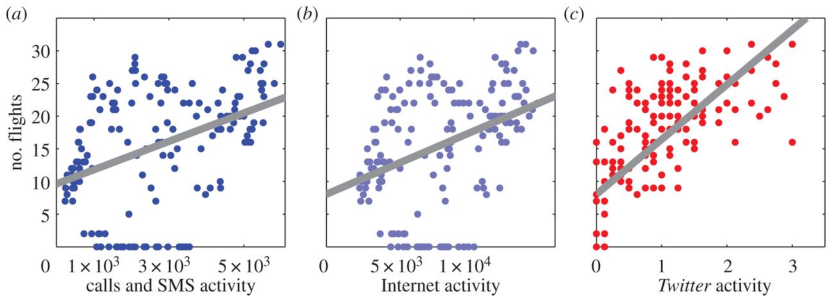
Table 7: Linear Regression R^2 Values

	Published Results	Duplication Results
Calls and SMS Data	0.175	0.175
Internet Activity	0.143	0.143
Twitter Activity	0.510	0.510

After wrangling the data and analyzing, we can see that the same values are obtained. Now, we can replicate the plots.



For comparison, here is the plot attached to the paper



Conclusion

As it can be seen, our plots line up exactly with the ones published. It should be noted that the authors of this paper made some assumptions that we do not agree with. For one, they used flight data from a period 5-6 months after the data of cell phone activity and assumed that the flight schedule would remain consistent on a day by day schedule for the period over the phone data. Given a lack of raw data for flights at the appropriate window, this assumption would need to be made, but it is one that could definitely be a major source of error. The second issue is that the period of cell phone and internet activity recorded includes the Christmas holidays, which we would assume the number of passengers at the airport during this time would be different enough from outside the holiday period that it may be a source for error. This potential source for error is minimized due to the way the grouping is done (summing all the recorded values for the same day of the week and hour of the day).

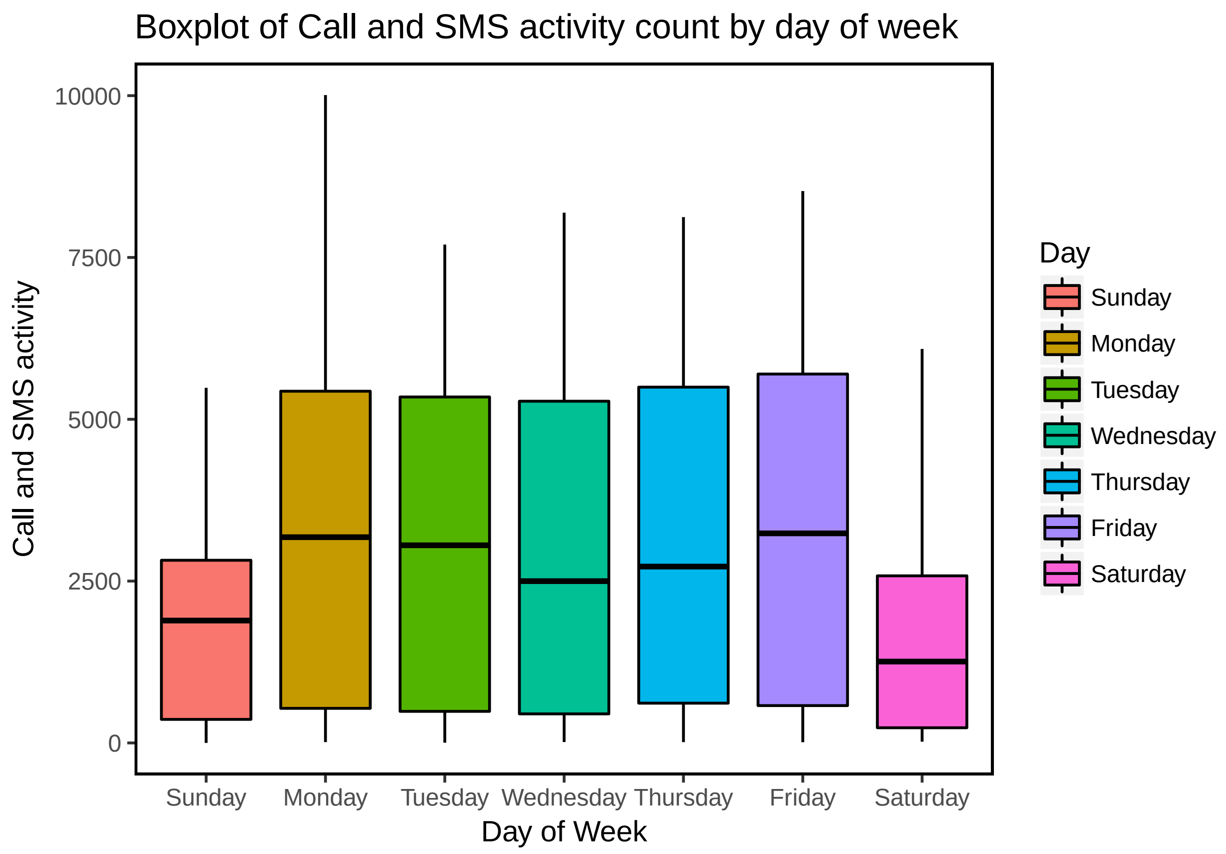
Bonus Section

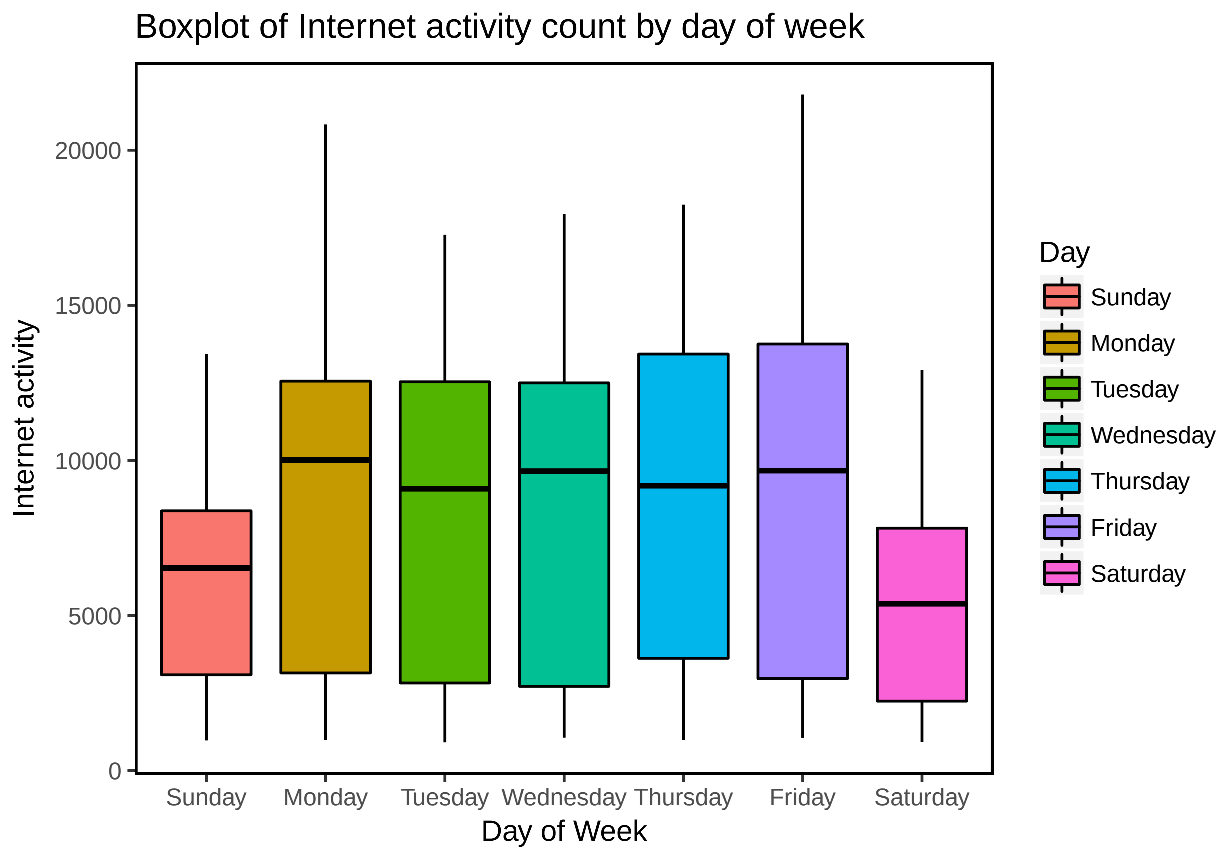
We decides for our bonus analysis, we would look at the mobile phone and *Twitter* activity at the Linates airport, on a per-day basis over the 2 month span, and see if there is a noticable difference day to day. To do that, we re-use a segment of our `Python` code earlier.

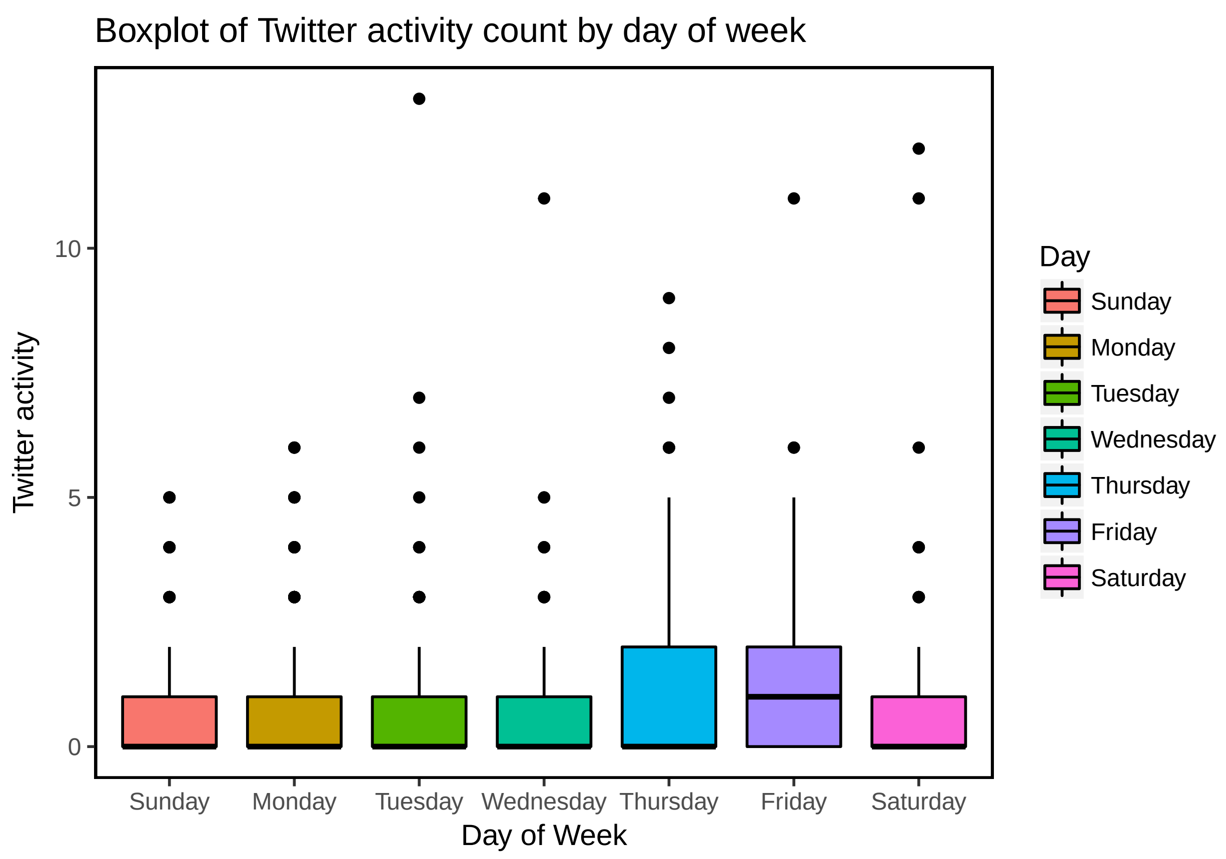
```
linate_phone_data = pd.read_csv('./data/Linate_Data.csv',
                               parse_dates=[0],
                               infer_datetime_format=True)

days_to_skip = pd.to_datetime(['2013-11-01',
                               '2013-11-02',
                               '2013-11-03',
                               '2013-12-30',
                               '2013-12-31']).date
linate_phone_data = \
    linate_phone_data[linate_phone_data['Timestamp'].dt.date.isin(days_to_skip) == False]
linate_phone_data.set_index('Timestamp', drop=True, inplace=True)
linate_phone_data['Day'] = linate_phone_data.index.weekday_name
linate_phone_data['Hour'] = linate_phone_data.index.hour
linate_phone_data.to_csv('Linate_bonus.csv')
```

We first read in the data our python code generated and generate a linear model with all the mobile phone related input factors.

Here is the data expressed in boxplots for each factor by the day of the week.




Boxplot of Internet activity count by day of week
Internet activity
0
5000
10000
15000
20000
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Day of Week
Day



```r
# Plot residuals of total model
linate.lm <- lm(Calls.and.SMS.Activity + Internet.Activity + Twitter.Activity ~ Day, data=bonus_data)
model_data <- augment(linate.lm)
```

The Calls/SMS and Internet data look very consistent, but the *Twitter* data looks slightly odd. This is due to the *Twitter* data being measured in much smaller numbers and is expected. Since we are looking at the effect of all factors on the attendance, this should be masked for the most part.

First, to make sure that an ANOVA is appropriate for this data-set, we need to meet three conditions.

1. The data must be iid
2. We must have the data come from a normal distrubition
3. The difference between variances should be no more than a factor of 5

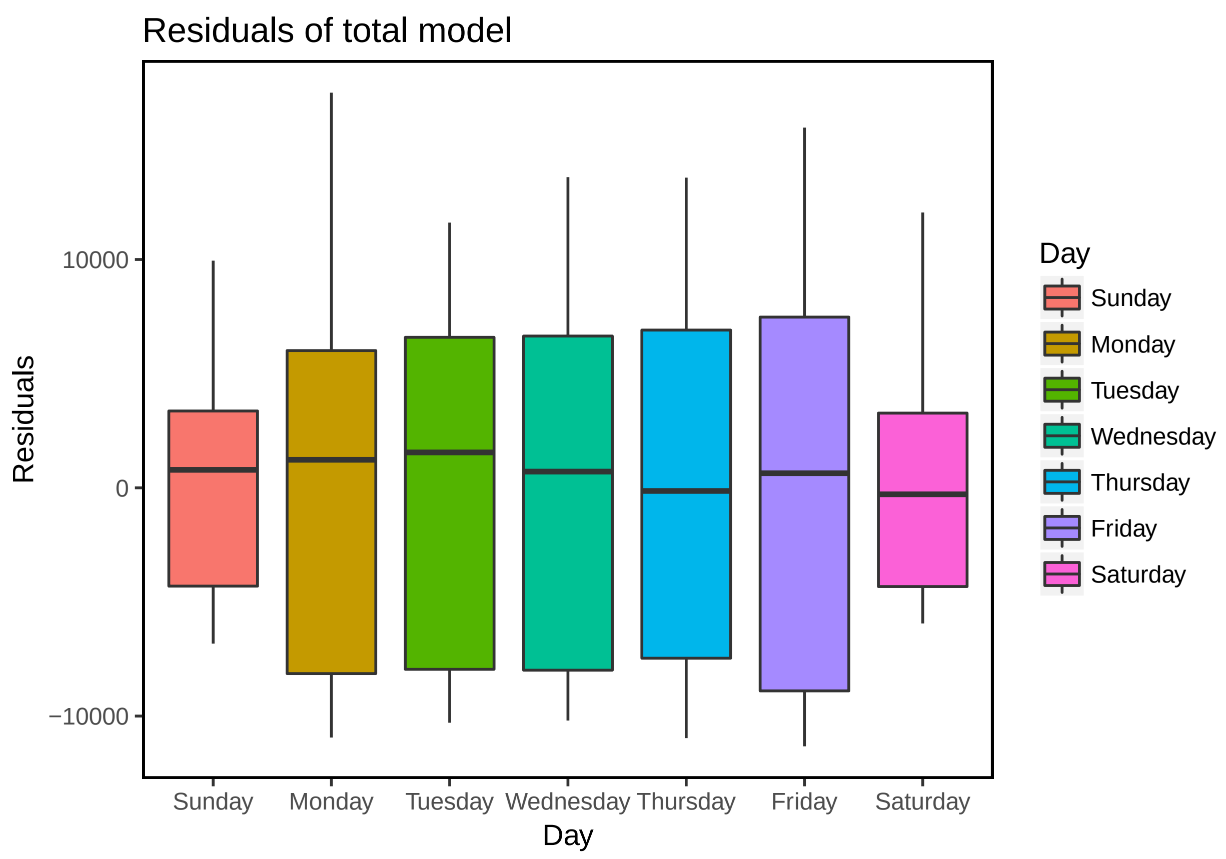
We meet the data must be iid requirement. We believe the data does come from a normal distribution, as given enough samples, calls/sms activity, internet activity and twitter activity should eventually converge to a normal-like distribution. To evaluate the differences in vairances, we can run the following code:

```r
max(aggregate(.resid ~ Day, model_data, var)$.resid) / min(aggregate(.resid ~ Day, model_data, var)$.res
```

```
## [1] 3.486544
```

We then plot the residuals

```r
ggplot(model_data, aes(Day, .resid)) + geom_boxplot(aes(fill=Day)) + ylab('Residuals') +
  ggtitle('Residuals of total model') +
  theme(panel.background = element_blank(),
        panel.border = element_rect(colour='black', fill=NA, size=1))
```



The null hypothesis here is that any day is equivalent to any other day of the week based on mobile phone activity. The alternative hypothesis is that not all days are equivalent.

```
anova(linate.lm)
```

```
## Analysis of Variance Table
##
## Response: Calls.and.SMS.Activity + Internet.Activity + Twitter.Activity
##             Df     Sum Sq   Mean Sq F value    Pr(>F)
## Day          6 5.6027e+09 933785451  19.274 < 2.2e-16 ***
## Residuals 1337 6.4776e+10  48448607
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Here, we can see that with a very high F-value and a very low P-value we can reject the null hypothesis.

From looking at the data, it's apparent that the weekend days look different than the week-days, so let's exclude the weekend days and do the ANOVA again.

```
# Subset the data to only weekdays
weekdays <- c('Monday','Tuesday','Wednesday','Thursday','Friday')
bonus_data_weekday <- bonus_data %>%
  filter(Day %in% weekdays)

# Linear Model for Weekdays data
linate_weekday.lm <- lm(Calls.and.SMS.Activity +
                          Internet.Activity +
                          Twitter.Activity ~ Day, data=bonus_data_weekday)
```

```r
# ANOVA
anova(linate_weekday.lm)
```

```
## Analysis of Variance Table
##
## Response: Calls.and.SMS.Activity + Internet.Activity + Twitter.Activity
##            Df     Sum Sq  Mean Sq F value Pr(>F)
## Day         4 1.9097e+08 47741836  0.8019  0.524
## Residuals 955 5.6856e+10 59535404
```

Here we get a low F-value and a high P-value, indicating that we should accept the null hypothesis that each weekday is eqauivalent.

Appendix

Data Sources

SMS, Call, and Internet Data

Data collected for internet, phone calls, SMS, and Twitter data was provided by the Telecom Italia Big Data Challenge. The data was acquired and anonymized by Telecom Italia. The data originates from Milan and surrounding areas between 1 November 2013 and 31 December 2013.

All interactions on the mobile network generate Call Detail Records (CDRs). These are acquired by the following parameters:
- SMS Data
- CDR is generated for each SMS sent and recieved
- Call Data
- CDR is generated for each call sent and recieved
- Internet Access: CDR is generated for the following events:
- Internet connection is opened
- Internet Connection is closed
- Internet Connection is open and 15 minutes has passed since last CDR
- Internet Connection is open and 5 MB have been transferred since last CDR was generated

After being collected, the data was rescaled for privacy reasons. The SMS and call data were scaled using the same factor, while internet data was scaled using a different factor.

Twitter Data

Similarly to the SMS, Call, and Internet Data, geo-localised tweets were collected from the Big Data Challenge (same time and location). It is not indicated that any data rescaling was completed. The paper does not state that the Twitter data is “normalized,” so it is assumed to be untouched.

Football Match Attendees

Football match attendance was retrieved from the Italian National Football League ‘Serie A’ official website. The final three games attendance data was obtained from two online newspapers (Calciomercato, Milan News 1, and Milan News 2).

Airport Data

The airport data was retrieved from the Linate Airport website. This data is only available for the current date + 4 days, so the data collected was for Monday 5 May 2014 through 11 May 2014. The measures used for estimating each hour of passenger amounts was calculated by summing the number of flights departing in the next 2 hours and the number of flights arriving in the past hour.