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

Ogi Moore and Connor Smith

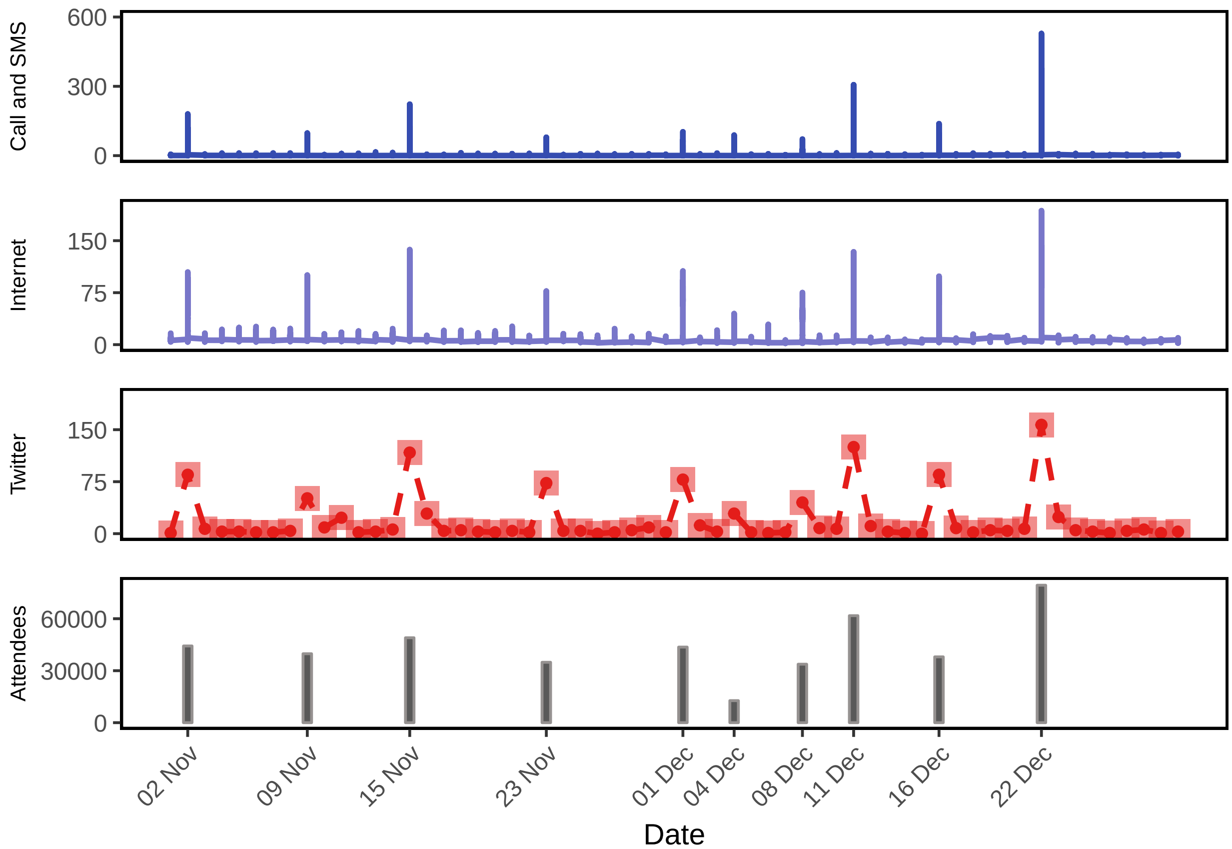
12/5/2016

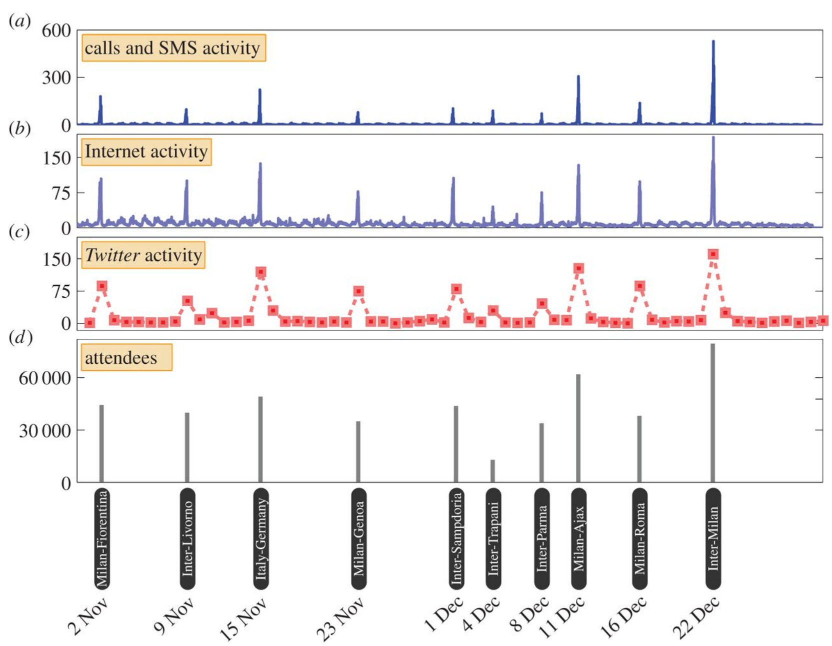
Introduction

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

```
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





Grouping

```
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 Mod	eling			

```
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)
```

```
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

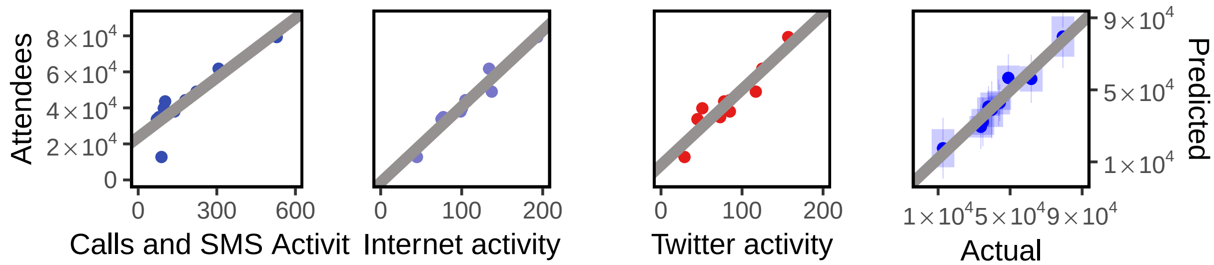
Spearman Correlations

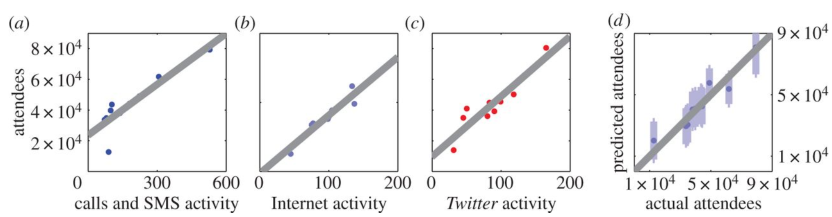
```r
cor_paper_results <- c(0.927, 0.976, 0.924)
cor_duplication_results <- c(round(cor(soccer_data$Attendees.at.San.Siro,
                                       soccer_data$Calls.and.SMS.Activity,
                                       method='spearman'), 3),
                             round(cor(soccer_data$Attendees.at.San.Siro,
                                       soccer_data$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

Results





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

```
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

```
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

Grouping

```
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')
```

```
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

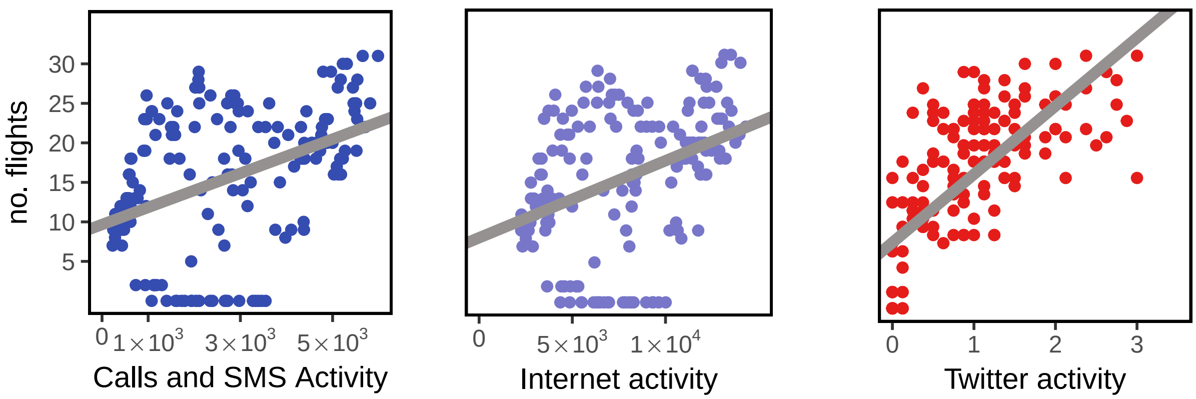
Linear Analysis

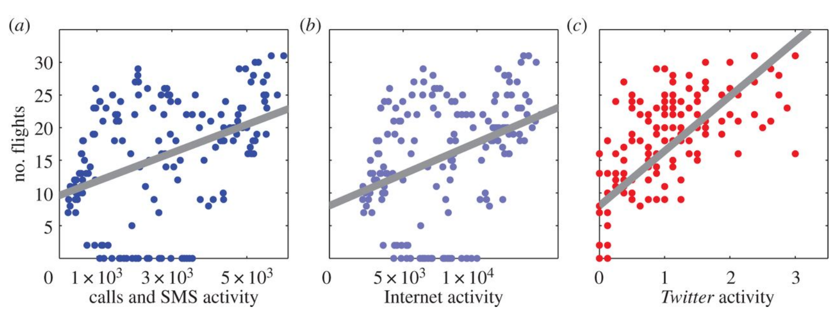
```
lm_paper_results <- c(0.175, 0.143, 0.510)
flights_v_phone <- lm(flight_data$Flights ~
                      flight_data$Calls.and.SMS.Activity)
flights_v_internet <- lm(flight_data$Flights ~
                         flight_data$Internet.Activity)
flights_v_twitter <- lm(flight_data$Flights ~
  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





Conclusion

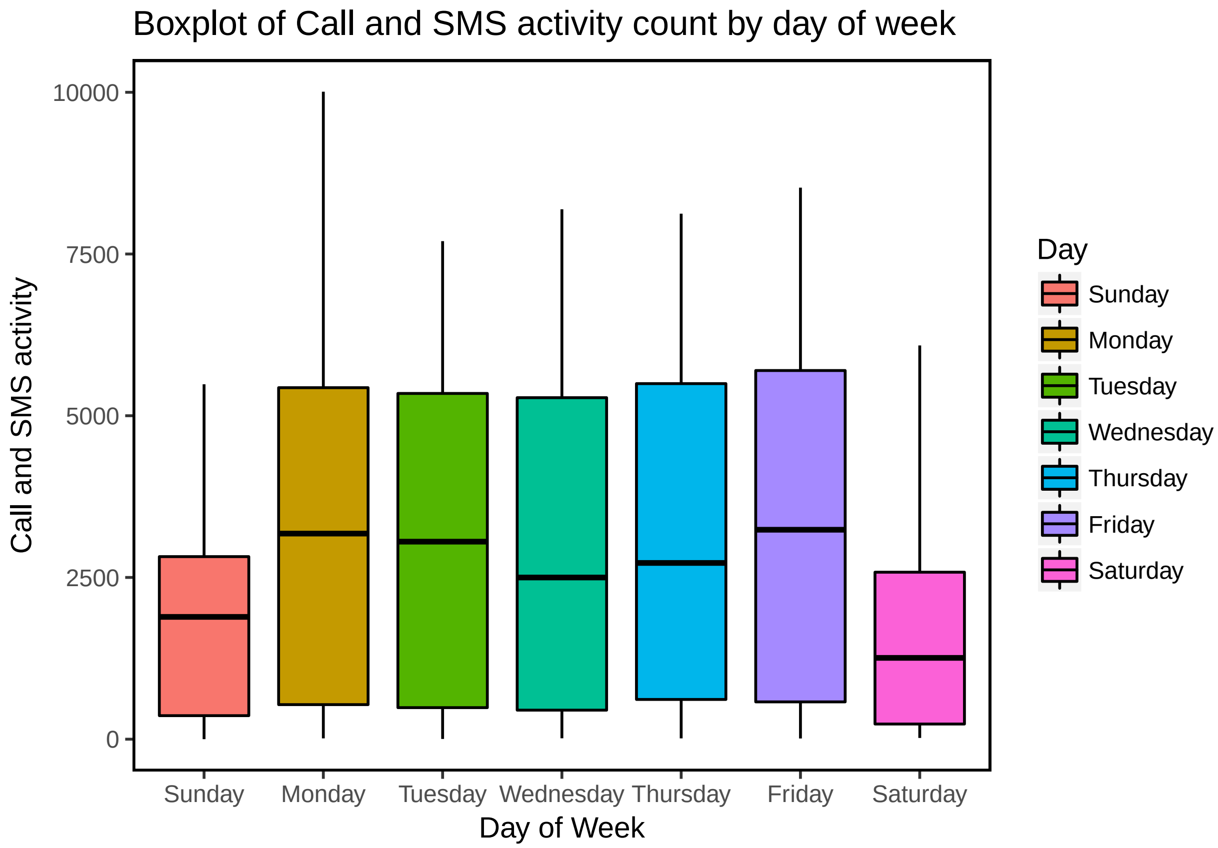
As it can be seen, our plots line up exactly with the ones published. It should be noted that the authors of this paper made some assumptions that we do not agree with. For one, they used flight data from a period

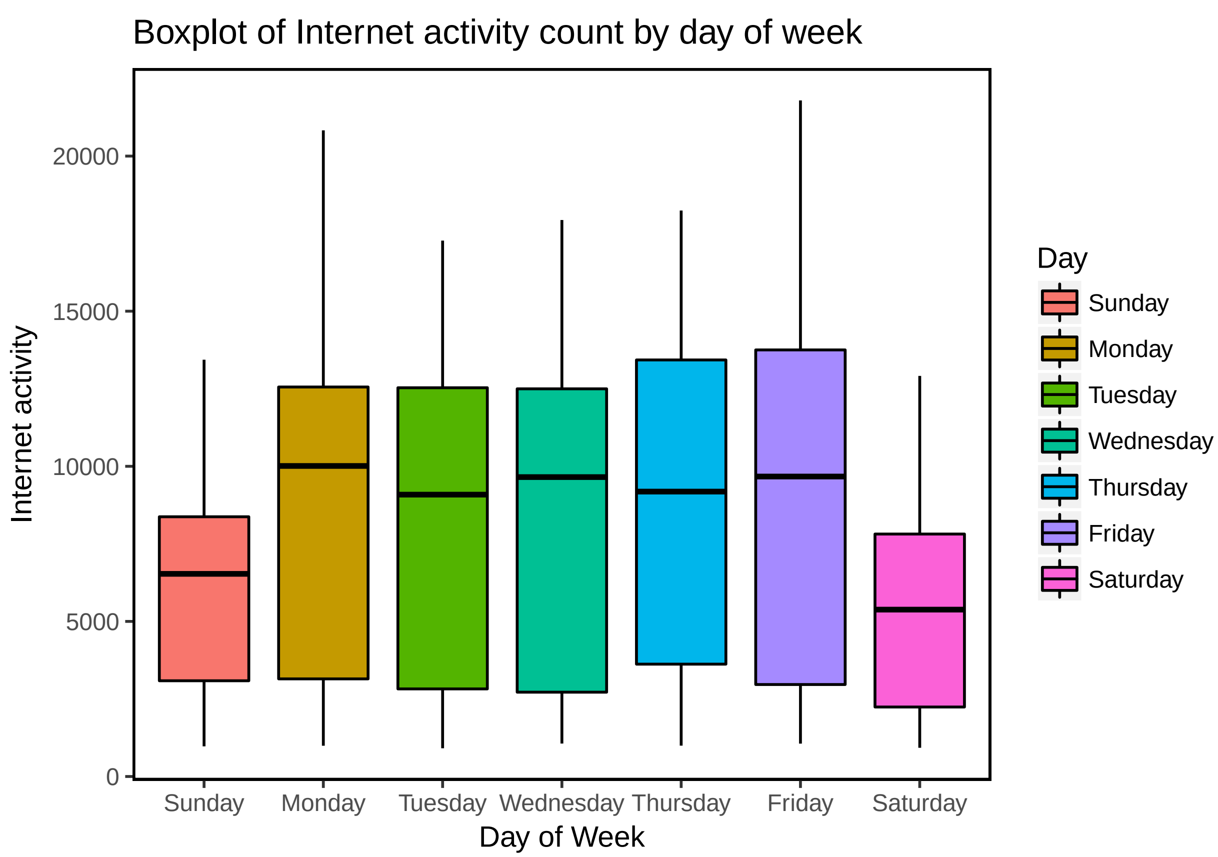
5-6 months after the data of cell phone activity and assumed that the flight schedule would remain consistent on a day by day schedule for the period over the phone data. Given a lack of raw data for flights at the appropriate window, this assumption would need to be made, but it is one that could definitely 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).

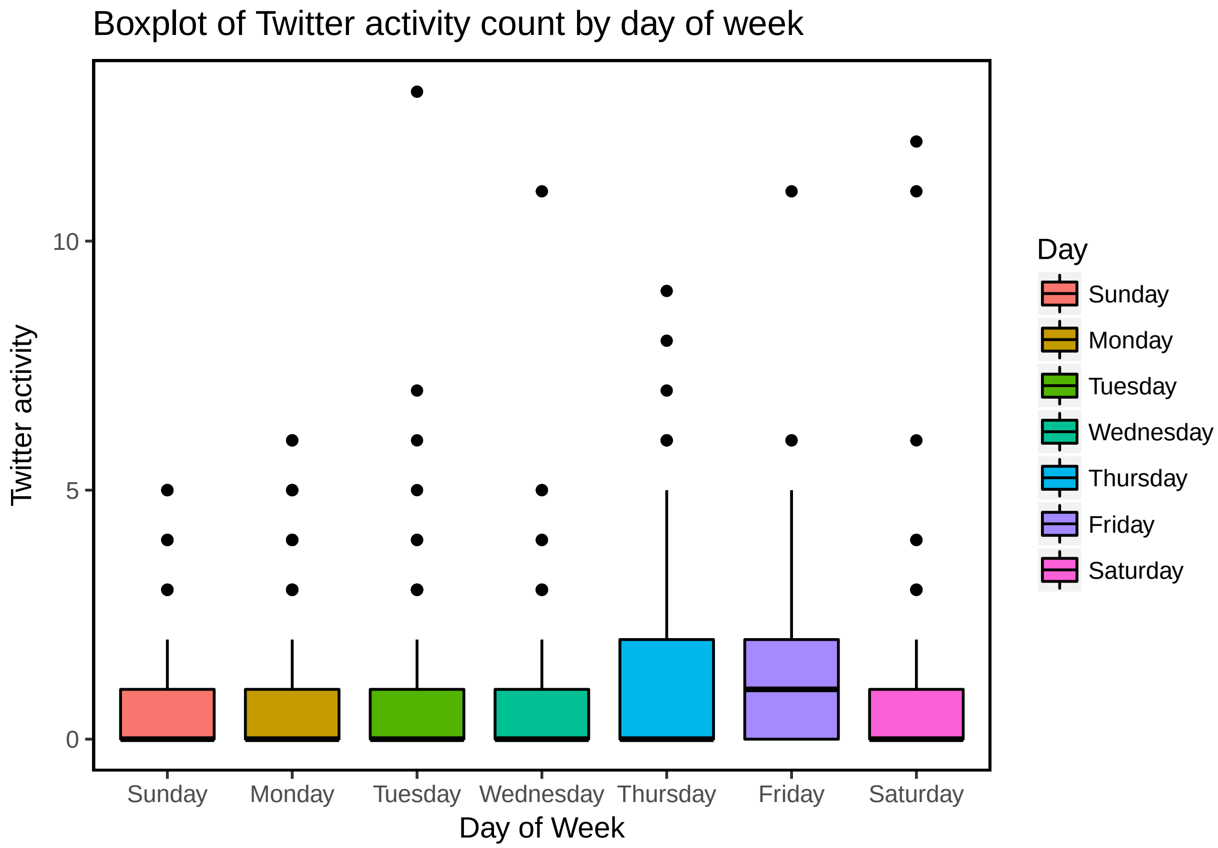
Bonus Section

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')
```


Boxplot of Call and SMS activity count by day of week
Call and SMS activity
10000
7500
5000
2500
0
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Day of Week
Day
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday


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

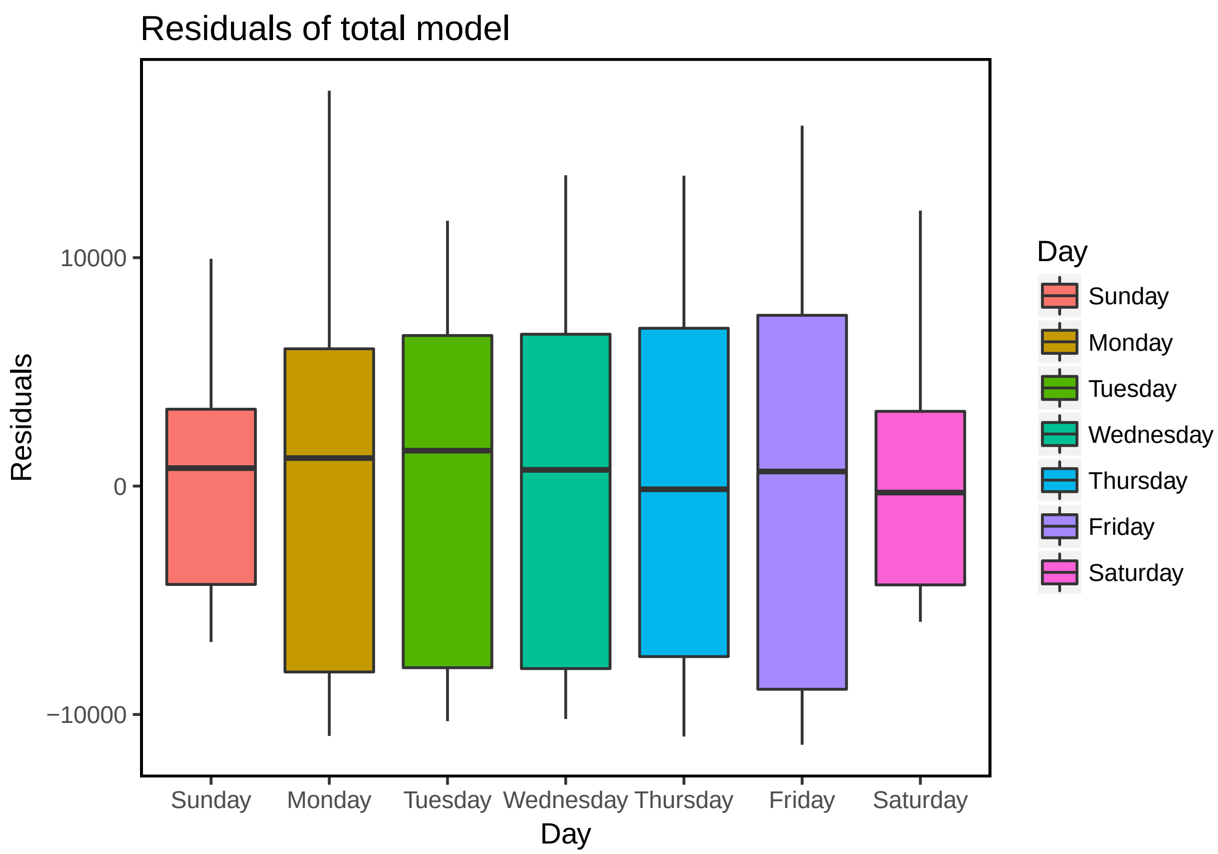



```
# Plot residuals of total model
linate.lm <- lm(Calls.and.SMS.Activity + Internet.Activity + Twitter.Activity ~ Day, data=bonus_data)
model_data <- augment(linate.lm)
```

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

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

```
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))
```




```r
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
```

```r
# 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)

# 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
```

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.