Planned Analyses

Patrick Frenett

7/14/2016

Introduction

This document gives a brief overview of the current planned analyses that will be undertaken using the NYC taxicabs dataset. It includes a number of interesting questions to be answered as well as points that could be developed further in the future.

Questions to be answered.

1) Is there a 'Thanksgiving effect' on the taxi data in New York? This includes changes in numbers, frequency, tips, passengers.
2) For 2015, how do the average credit card tips vary in the dataset?

1. Is there a 'Thanksgiving effect' in the dataset? If there is, what is it and how does it affect the rides over the holiday period.

Finding what (if any) Thanksgiving effect there is on the data is a broad question with a number of different approaches (from the simpliest of ideas to more complex ones). This means that finding a definitive answer to the question is nigh on impossible; however a lot of interesting conclusions can be made by looking at how particular measures are influenced by the holiday period.

1.a Does the total number of rides on Thanksgiving significantly differ from other Thursdays around a similar time of the year?

To start with this question, we have to count the number of rides in a number of Thursdays in and around Thanksgiving day. To do this we will select the two Thursdays both before and after Thanksgiving day, as well as Thanksgiving day itself, and count the number of rides in those given days. If we do this for all 7 years ('09 - '15) then an ANOVA model can be constructed to test whether there is a signficance difference between the average tip on Thanksgiving and other Thursdays around that period.

PostgreSQL Queries

```
AVG(
SELECT
  tip_amount
FROM
  data_2015
WHERE
  tpep_pickup_datetime
BETWEEN
  11/26/2015 00:00 AND 11/27/2015 00:00
  );
```

R Code

```
Thanksgiving <- data.frame(day = day, avg_tip = avg_tip)
lmThanksgiving <- lm(avg_tip ~ day, data=Thanksgiving)

summary(lmThanksgiving)
bwplot(avg_tip ~ day, data=Thanksgiving)
```

1.b What do the hour by hour time plots of ride frequency look like on Thanksgiving compared to the weeks before and after?

1.c

2. How do the average credit card tips vary in the dataset?

This question, like the last will look at the variable in question, average credit card trip, in the context of a few questions that can be asked about tips in New York City.

2.a Does the average credit card tip given vary across different locations in NYC?

The NYC dataset provides latitudes and longitudes for the pickup and dropoff of the journeys in NYC from 2009-15. These are very precise spatial measurements, however this makes them hard to work with and interpret as they are not easily aggregated (something that is clearly neccersary due to the size of the dataset). To solve this issue, the latitudes and longitudes were converted to zip codes using PostGIS - an extention of the relational database software PostgreSQL.

With the spatial specificity decided, the data needs to be aggregated by zip code. Below is the SQL needed to take the average card tip for both pickups and dropoffs. The tips given in the dataset are only credit card tips (cash tips are not registered) so the `WHERE payment_type=1` query gives an accurate mean of the credit card tips, rather than having the means lowered by the apparent non tips when paying by cash.

PostgreSQL Queries

```
INSERT INTO pickup_tips
SELECT
 pickup_zip,
 AVG (tip_amount)
FROM
 data_2015
WHERE
  payment_type=1
GROUP BY
 pickup_zip
ORDER BY
 pickup_zip ASCN;

INSERT INTO dropoff_tips
SELECT
 dropoff_zip,
 AVG (tip_amount)
FROM
 data_2015
WHERE
  payment_type=1
GROUP BY
```

```
 dropoff_zip
ORDER BY
 dropoff_zip ASCN;
```

Once this data has been collected, there are a few things that can be done with it. A great visualisation of this data would be a chloropleth map showing zip codes coloured darker to indicate higher tips. The `leaflet` package can be used to achieve this.

Another approach would be to group each zip code by borough and then run an ANOVA analysis to test whether there was a siginificant borough effect on average credit card tips in the city.

2.b How closely correlated is the average tip given with demographic information such as average income or age?

2.c How does the average tip given vary across the day and year?

PostgreSQL Queries

Create table for day of the year against percentage tip.

```
CREATE TABLE public.doy_tips_2015
(
  pct_tip real,
  doy double precision
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.doy_tips_2015
  OWNER TO postgres;

INSERT INTO
    doy_tips_2015 (pct_tip, doy)
SELECT
    tip_amount/total_amount*100,
    EXTRACT(DOY FROM tpep_pickup_datetime)
FROM
    data_2015
WHERE
    payment_type = 1;
```

Create table for hour of the day against percentage tip.

```
CREATE TABLE public.hour_tips_2015
(
  pct_tip real,
  hour double precision
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.hour_tips_2015
  OWNER TO postgres;
```

```
INSERT INTO
    hour_tips_2015 (pct_tip, hour)
SELECT
    tip_amount/total_amount*100,
    EXTRACT(HOUR FROM tpep_pickup_datetime)
FROM
    data_2015
WHERE
    payment_type = 1;
```

Get average tip against day of the year.

```
SELECT
    doy,
    AVG (pct_tip)
FROM
    doy_tips_2015
GROUP BY
    doy
ORDER BY
    doy;
```

Get average tip against hour of the day.

```
SELECT
    hour,
    AVG (pct_tip)
FROM
    hour_tips_2015
GROUP BY
    hour
ORDER BY
    hour;
```

This leaves us with two tables : doy & pct_tip, hour & pct_tip. Once the dataframes (doy and hour) are loaded into R, we can execute these commands : XXX CHECK DOY = 1 is Jan 1st XXX REFERNCE http://www.r-bloggers.com/ggplot2-time-series-heatmaps/

R Code

This will create a calender heatmap of average tips.

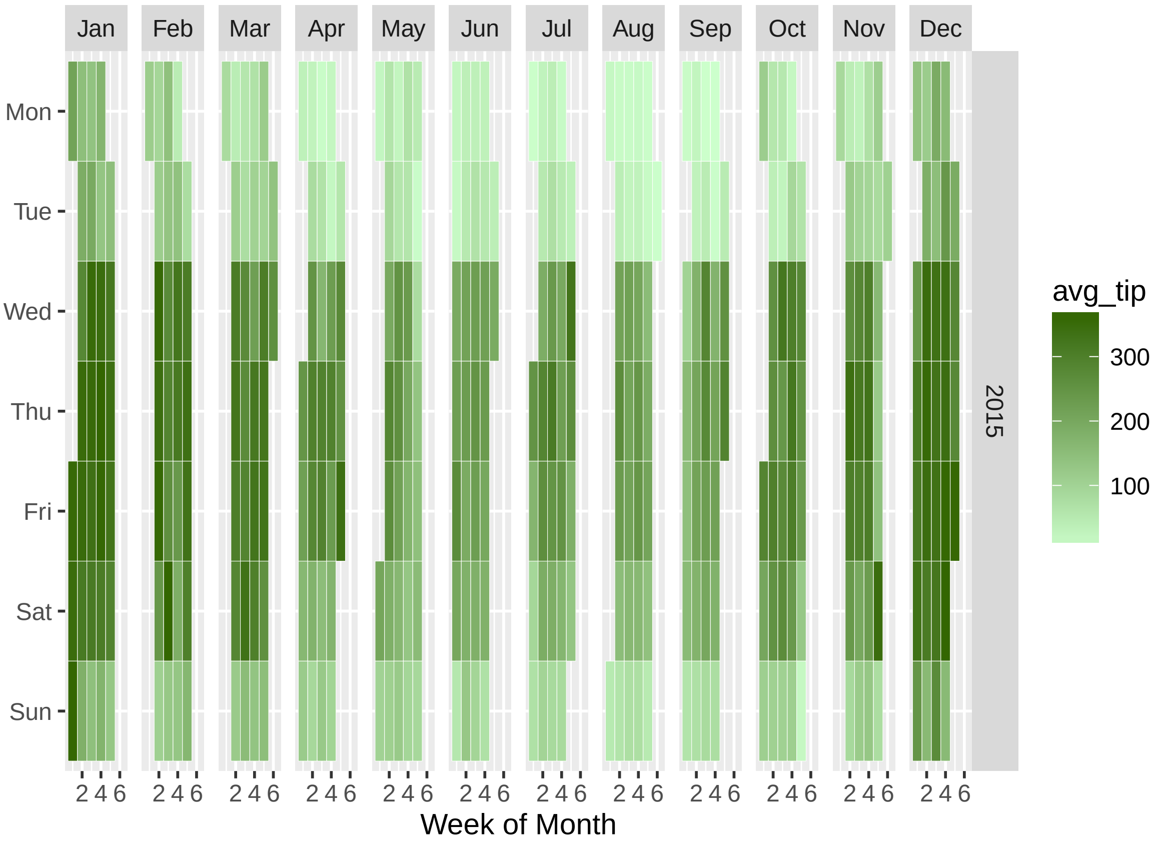
```r
library("plyr")
library("ggplot2")
library("zoo")
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library("mosaic")
```

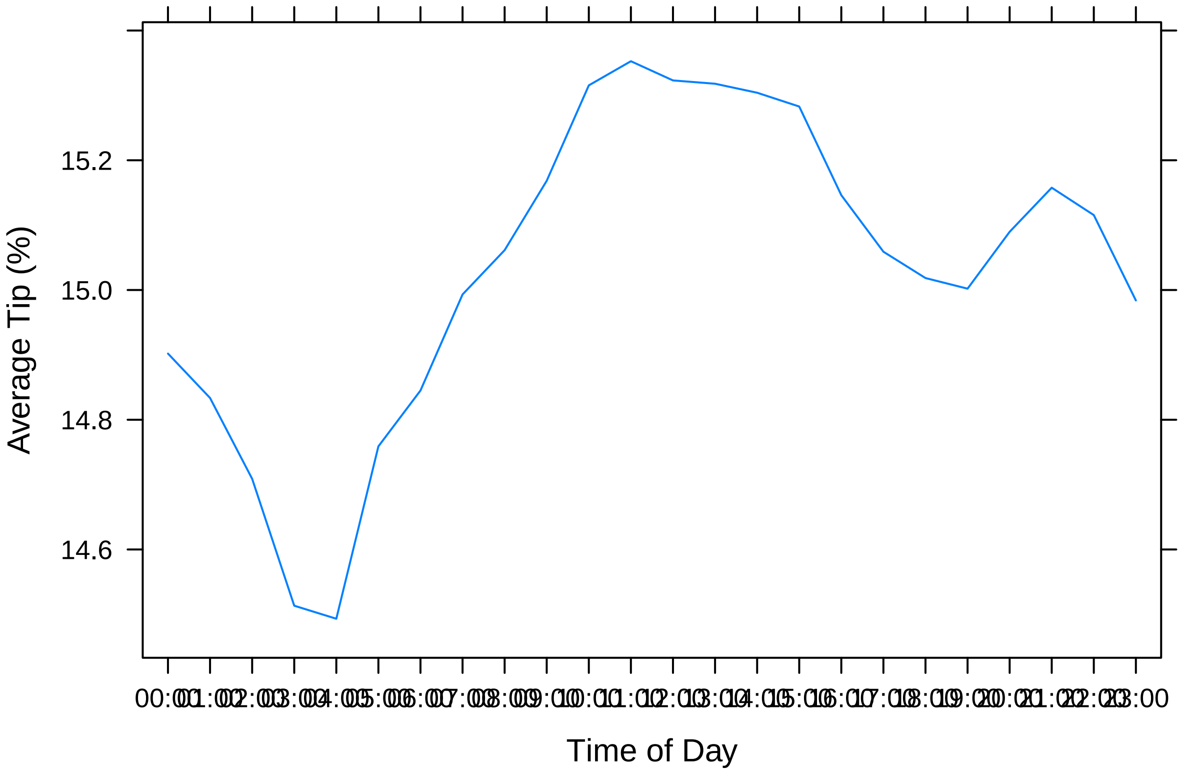
```
## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: lattice

## Loading required package: mosaicData

## Loading required package: Matrix

##
## The 'mosaic' package masks several functions from core packages in order to add additional features.
## The original behavior of these functions should not be affected by this.

##
## Attaching package: 'mosaic'

## The following object is masked from 'package:Matrix':
##
##     mean

## The following objects are masked from 'package:dplyr':
##
##     count, do, tally

## The following object is masked from 'package:plyr':
##
##     count

## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cov, D, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var

## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

```r
dat <- read.csv("data/doy_tips_2014.csv", sep = ";")
dat <- mutate(dat, X.doy.avg_tip = as.character(X.doy.avg_tip))
dat <- strsplit(dat$X.doy.avg_tip, ",")
dat <- unlist(dat)
dat <- dat[seq(3, length(dat), 3)]
dat <- data.frame(doy = 1:365, avg_tip = dat)
dat <- mutate(dat, avg_tip = as.numeric(avg_tip))

#turn doy to date
dat$date <- as.Date(dat$doy,"%Y-%m-%d", origin = "2014-12-31", tz = "EST")

#create a year column with 2015 for all entries
dat$year <- 2015

#get numeric month
dat$month<-as.numeric(as.POSIXlt(dat$date)$mon+1)

#create factor month
dat$monthf<-factor(dat$month,levels=as.character(1:12),labels=c("Jan","Feb","Mar","Apr","May","Jun","Jul

#get numeric weekday
dat$weekday = as.POSIXlt(dat$date)$wday

#get factor weekday
dat$weekdayf<-factor(dat$weekday,levels=rev(0:6),labels=rev(c("Mon","Tue","Wed","Thu","Fri","Sat","Sun"

#create variable IE 'March 2015'
dat$yearmonth<-as.yearmon(dat$date)

#above as factor
dat$yearmonthf<-factor(dat$yearmonth)

#week of the year for each day
dat$week <- as.numeric(format(dat$date,"%W"))

# and now for each monthblock we normalize the week to start at 1
dat<-ddply(dat,.(yearmonthf),transform,monthweek=1+week-min(week))

P<- ggplot(dat, height = 300, aes(monthweek, weekdayf, fill = avg_tip)) +
  geom_tile(colour = "white") + facet_grid(year~monthf) + scale_fill_gradient(low="#ccffcc", high="#336
  xlab("Week of Month") + ylab("")

P
```



This will create a line plot of average tip throughout the day.

```
hour <- read.csv("data/hour_tips_2014.csv", sep = ";")
tmp1 <- hour[1:10,]
tmp2 <- hour[11:24,]
tmp1$time <- paste0("0",tmp1$hour,":00")
tmp2$time <- paste0(tmp2$hour,":00")

hour <- rbind(tmp1, tmp2)
hour <- mutate(hour, time  = as.factor(time))

xyplot(avg_tip ~ time, data=hour, xlab = "Time of Day", ylab = "Average Tip (%)", width = 600, type = "
```



2.d Is there a suitable model that can be used to predict the average tip paid for a given journey?