Untitled

Sruthi Kizhakathra

April 27, 2018

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

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

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

```
library(readxl)
library(ggplot2)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
instadata <- read_excel("Instacart_Data_Analyst_Challenge.xlsx", sheet="Raw Data (disguised)")
colnames(instadata)= c("delivery_time", "order_id", "ratings", "issue", "region" )
```

The dataset includes information about order delivery time, order id, customer order ratings, customer issue reported and the region of the order. For the ease of analysis I am renaming the columns to delivery_time, order_id, ratings, issue and region respectively.

```
summary(instadata)
```

```
##  delivery_time         order_id            ratings         issue
##  Length:14957       Min.   :   208056   Min.   :0.000   Length:14957
##  Class :character   1st Qu.:   232982   1st Qu.:5.000   Class :character
##  Mode  :character   Median :   245829   Median :5.000   Mode  :character
##                     Mean   :104111839   Mean   :4.558
##                     3rd Qu.:233588985   3rd Qu.:5.000
##                     Max.   :233614681   Max.   :5.000
##     region
##  Length:14957
##  Class :character
##  Mode  :character
##
##
##
```

```
glimpse(instadata)
```

```
## Observations: 14,957
## Variables: 5
```

```
## $ delivery_time <chr> "2014-06-02 04:23:16 UTC", "2014-06-02 03:57:50 ...
## $ order_id      <dbl> 233599337, 233599376, 233599328, 233599070, 2335...
## $ ratings       <dbl> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, ...
## $ issue         <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ region        <chr> "chi", "chi", "chi", "chi", "chi", "chi", "chi",...
```

The delivery_time is in character format. So lets convert thta into date time format. Inaddition, I am creating new columns such as month, weekday, hour(time when order delivery took place).

```
instadata$delivery_time <- as.POSIXct(strptime(instadata$delivery_time, "%Y-%m-%d %H:%M:%S"))
# creating month, weekday, time from order delivery time
instadata$month <- month(as.POSIXlt(instadata$delivery_time, format="%d/%m/%Y"))
instadata$day <-wday(instadata$delivery_time, label = TRUE)
instadata$date <- as.Date(instadata$delivery_time)
instadata$hour <- substr(instadata$delivery_time, 12, 13) %>% as.numeric()
```

Now lets have a glimpse of data. Now delivery time is in date format

```
glimpse(instadata)
```

```
## Observations: 14,957
## Variables: 9
## $ delivery_time <dttm> 2014-06-02 04:23:16, 2014-06-02 03:57:50, 2014-...
## $ order_id      <dbl> 233599337, 233599376, 233599328, 233599070, 2335...
## $ ratings       <dbl> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, ...
## $ issue         <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ region        <chr> "chi", "chi", "chi", "chi", "chi", "chi", "chi",...
## $ month         <dbl> 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, ...
## $ day           <ord> Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon...
## $ date          <date> 2014-06-02, 2014-06-02, 2014-06-02, 2014-06-02,...
## $ hour          <dbl> 4, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, ...
```

```
summary(instadata)
```

```
##  delivery_time                  order_id           ratings
##  Min.   :2014-05-01 08:54:00   Min.   :   208056   Min.   :0.000
##  1st Qu.:2014-05-09 01:12:55   1st Qu.:   232982   1st Qu.:5.000
##  Median :2014-05-17 03:24:46   Median :   245829   Median :5.000
##  Mean   :2014-05-17 08:13:28   Mean   :104111839   Mean   :4.558
##  3rd Qu.:2014-05-25 17:31:48   3rd Qu.:233588985   3rd Qu.:5.000
##  Max.   :2014-06-02 06:28:37   Max.   :233614681   Max.   :5.000
##
##     issue              region              month        day
##  Length:14957       Length:14957       Min.   :5.000   Sun:2588
##  Class :character   Class :character   1st Qu.:5.000   Mon:2381
##  Mode  :character   Mode  :character   Median :5.000   Tue:2051
##                                        Mean   :5.056   Wed:1729
##                                        3rd Qu.:5.000   Thu:1877
##                                        Max.   :6.000   Fri:2072
##                                                        Sat:2259
##       date                 hour
##  Min.   :2014-05-01   Min.   : 0.00
##  1st Qu.:2014-05-09   1st Qu.: 2.00
##  Median :2014-05-17   Median :17.00
##  Mean   :2014-05-17   Mean   :13.17
##  3rd Qu.:2014-05-25   3rd Qu.:21.00
##  Max.   :2014-06-02   Max.   :23.00
```

##

From the summary its clear that we have a total of 14957 records, without any missing values or outliers. Now lets go through these columns one by one.

1. We have records starting from May (2014-05-01 08:54:00) to June (2014-06-02 06:28:37).
2. Ratings are between 0-5 with average rating of 4.55
3. Sunday followed by Monday and Saturday are the busiest days.

Also, there are some order ids's repeating. Lets remove those as there is no info regarding these order_id's either duplicate or data entry error. These records appear on the same dat within a short time difference.

```
instadata <- instadata %>% unique()
```

Since we have only records of 2 days in June lets exclude those from our analysis. Now we have 14834 records total and 14006 from May 2014 and 828 records from June 2014.

```
instadata_may <- instadata %>% filter(month==5)
instadata_june <- instadata %>% filter(month==6)
```

since time is in UTC lets change it according to PST, CST and EDT for sf, chi and nyc assuming sf- san francisco PST chi- chicago CST nyc- new york city EDT

Lets look into average number of orders per day.

```
avg_order_per_day <- nrow(instadata_may)/31
avg_order_per_day
```

```
## [1] 451.8065
```

```
avg_order_rating <- sum(instadata$ratings)/nrow(instadata_may)
avg_order_rating
```

```
## [1] 4.843781
```

Lets look inot average issues per day. 903 issue sreported in 31 days which averages to 29 issues reported per day.

```
avg_issues_per_day <- instadata_may %>% group_by(issue) %>%
                                        summarise(n=n_distinct(order_id))
```

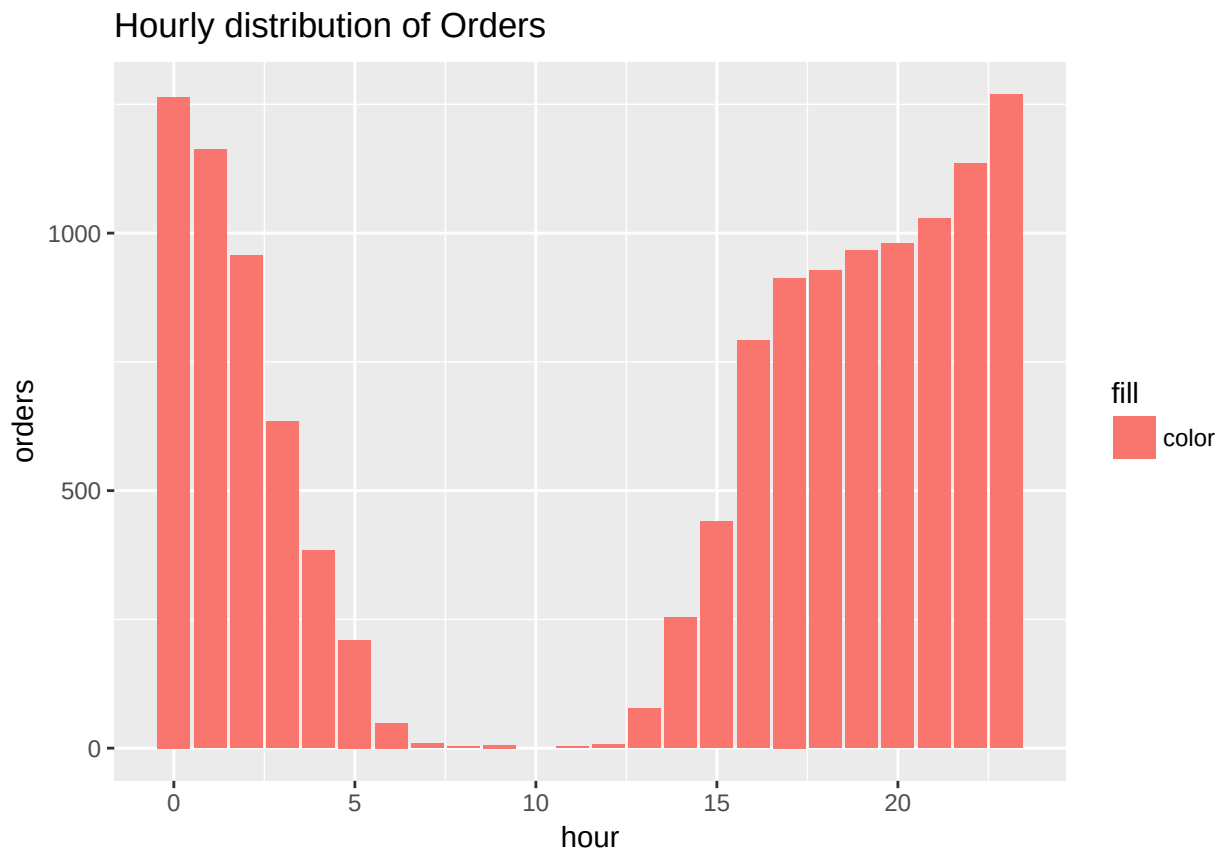
Lets look days with maximum number of orders- Friday, Saturday and Monday. It makes sense as people tend to shop more on weekends.

```
peak_order_days <- instadata_may %>% group_by(day) %>%
                summarise(orders= n_distinct(order_id)) %>% arrange(desc(orders))
```

Lets see what hours are most busy in the data. The data shows 5 pm - 2 am UTC as peak hours.

```
peak_order_hours <- instadata_may %>% group_by(hour) %>% summarise(orders= n_distinct(order_id)) %>% ar

ggplot(peak_order_hours, aes(x=hour,y=orders,fill="color")) + geom_col() + ggtitle("Hourly distribution
```



Looking into data we realize that insta serves 3 disntinct regions: nyc, sf, chi

```
regions_served <- unique(instadata_may$region)
regions_served
```

```
## [1] "chi" "nyc" "sf"
```

Lets see the percent distribution of orders in different markets. SF- 48 % chi - 43 % nyc - 8.8 %

```
percent_distribution_orders <- instadata_may %>% group_by(region) %>%
  summarise(orders = n_distinct(order_id)) %>% mutate ( percent_market =orders/sum(orders) )

ggplot(percent_distribution_orders, aes(x=region,y=percent_market,fill="color")) + geom_col() +ggtitle(
```

Lets breakdown summary statistics calculated on overall level into region wise.

```
avg_orders_region <- instadata_may %>% group_by(region) %>% summarise(orders = n_distinct(order_id)) %>%
```

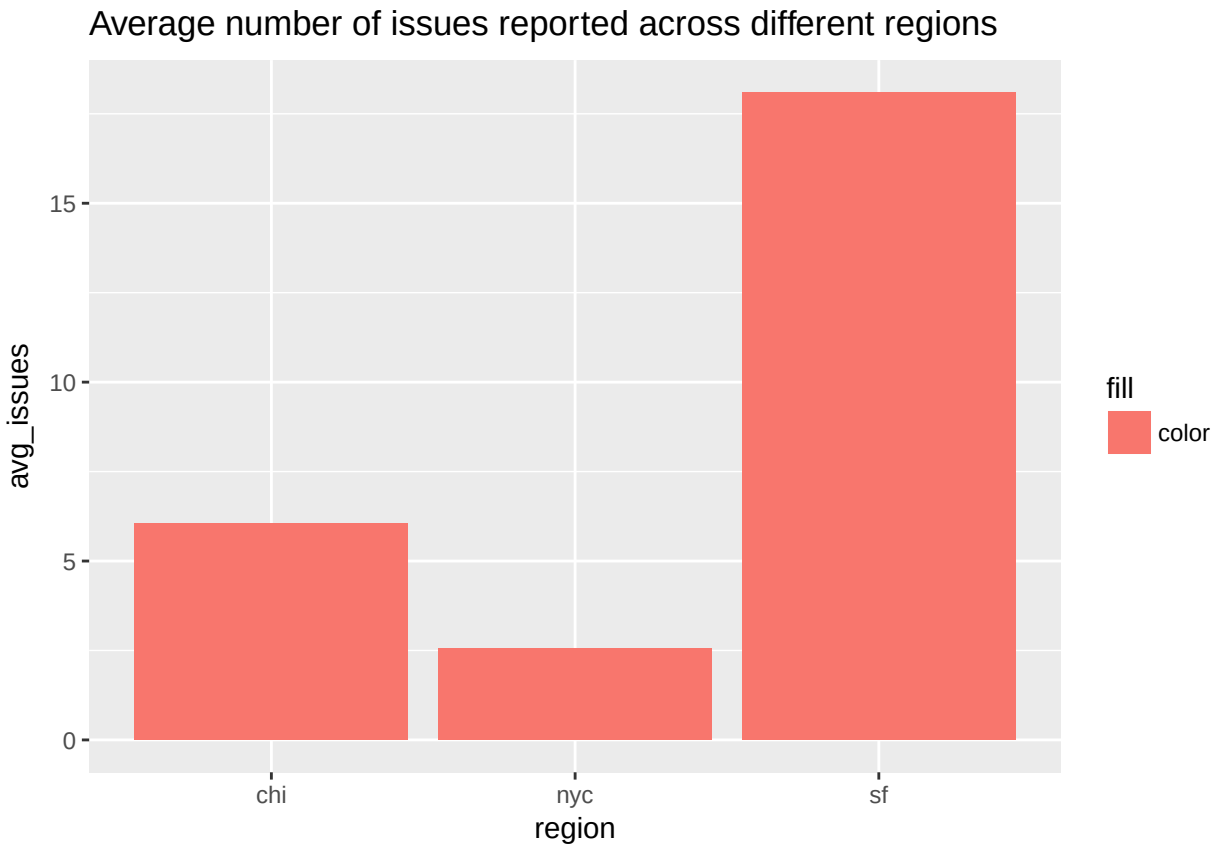
Now lets see average ratings per region. Chi have a better rating of 4.72 followed by sf and nyc with 4.46.

```
avg_rating_region <- instadata_may %>% group_by(region) %>% summarise(avg_ratings = mean(ratings))
ggplot(avg_rating_region, aes(x=region,y=avg_ratings,fill="color")) + geom_col() +ggtitle("Average Orde
```

Now lets look into issues across regions.SF (8.22%) market shows issues reported more in comparison with other markets (chi (3.10%) and nyc(6.34%)). This explains relatively lesser rating in sf market. Extra resorces and precautions to avoid these issues in this market needs to be implemented. # Reccomendations

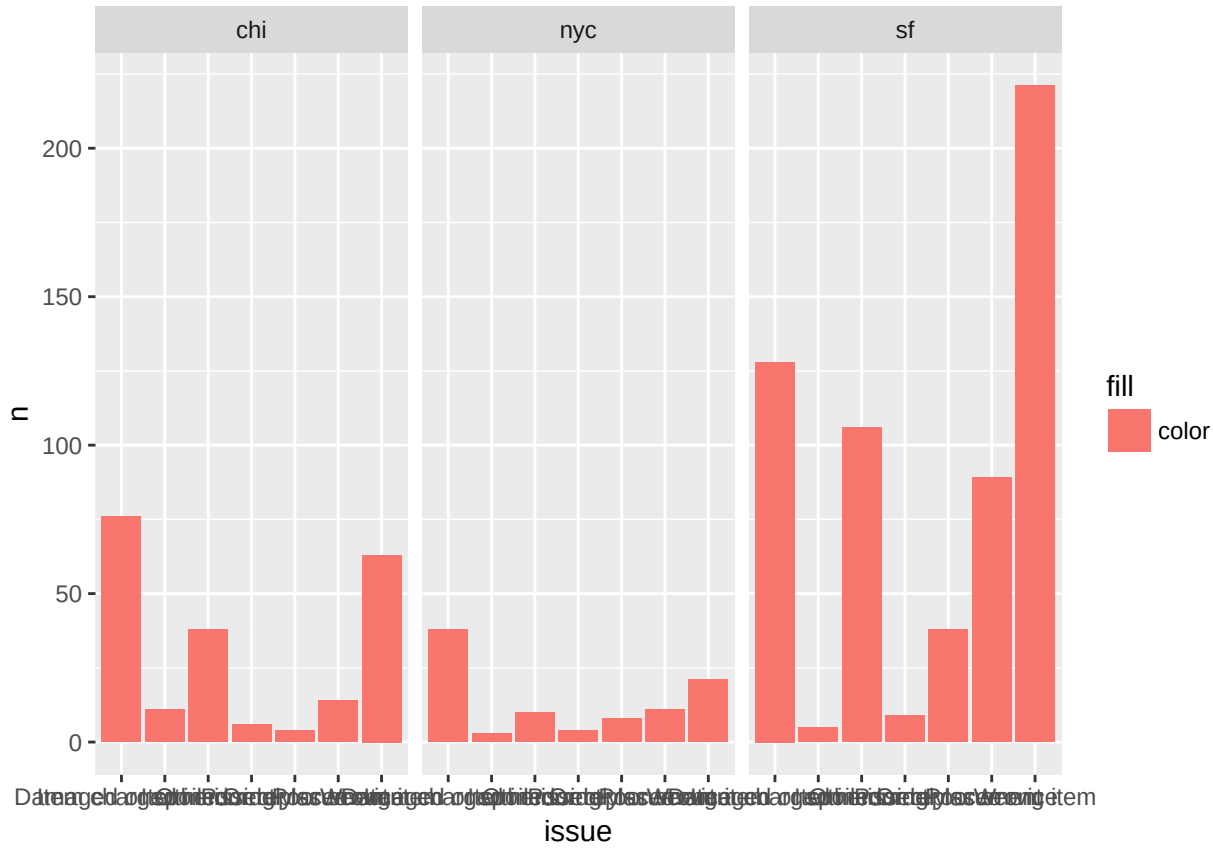
```
# issues reported across regions
issues_cnt_across_regions<- instadata_may %>% filter(issue != 'NA') %>% group_by(region) %>%
                                              summarise(n=n_distinct(order_id))  %>% mutate(avg_issues =

ggplot(issues_cnt_across_regions, aes(x=region,y=avg_issues,fill="color")) + geom_col() + ggtitle("Avera
```



Major issue across regions are wrong item and damaged/spoiled item. SF and nyc has damaged/spoiled and Chi has wrong item as major issue reported.

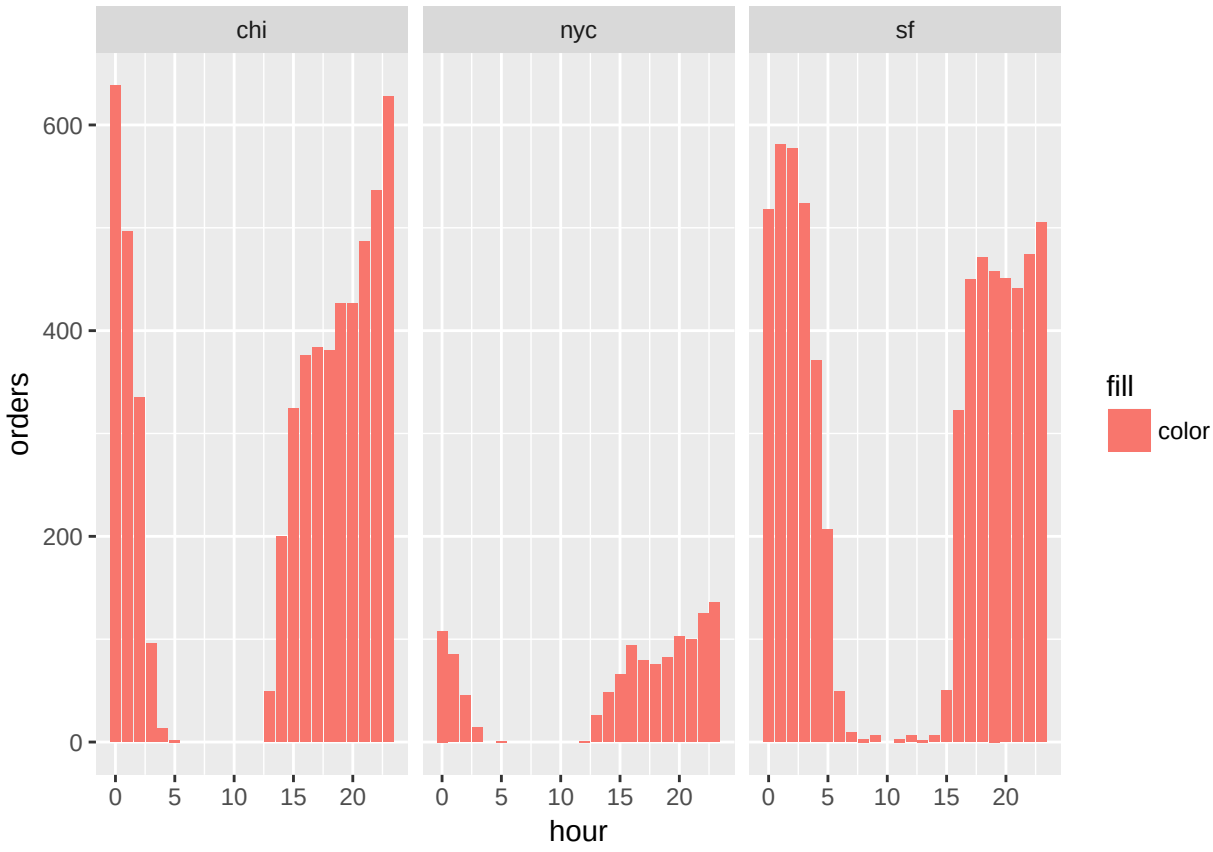
```
#issues
issues_across_regions<- instadata_may %>% filter(issue!='NA') %>% group_by(region,issue) %>% summarise(n
ggplot(issues_across_regions, aes(x=issue,y=n, fill="color")) + geom_col() + facet_wrap(~region)
```



Now lets see the peak hours regional level. The peak timing across regions are : SF - 9 am - 7 am Chi - 3 pm - 7 pm NYC - 5 pm - 8 pm

The plot shows sf is busy across the major portion of delivery timings.

```
#Peak time
hour_trends <- instadata_may %>% group_by(hour, region) %>% summarise(orders = n_distinct(order_id)) %>%
ggplot(hour_trends, aes(x=hour,y=orders, fill="color")) + geom_col() + facet_wrap(~region)
```



Based on these above mentioned data analysis, I have come up with my reccomendations and next steps.