# CSE 578 : Data Visualization

## VAST Mini Challenge 1 - 2015

## “Mayhem at DinoFun World”


**Group Members:**

**Adersh Ganesh**

**Bhagyasri Musunuru**

**Krushali Shah**

**Monisha Gopinath**

**Radhika Ganapathy**

**Sruthi Sathyamoorthy**

# 1. Introduction

Data is ubiquitous and is being generated in abundance. This has led to its exponential growth. With the onslaught of various modern technologies, it is imperative for us to leverage on the profusion of data to make calculated decisions for the future.

Exploratory data analysis is an approach of analyzing data sets to summarize their main characteristics by using statistical graphics and other data visualizations. Data visualization is the representation of data in a pictorial or graphical format. It enables decision makers or humans to see analytics presented visually, so they can grasp difficult concepts or identify new patterns and outliers. It is used to translate large data sets and metrics into charts, graphs and other visuals. The resulting visual representation of data makes it easier to identify and share real-time trends, outliers, and new insights about the information represented in the data. The term is often used interchangeably with others, including information graphics, information visualization and statistical graphics. Data visualization is one of the steps of the data science process, which states that after data has been collected, processed and modeled, it must be visualized for conclusions to be made.

In this project, we have taken up the Mini Challenge -1 presented by the IEEE Visual Analytics Science and Technology (VAST) Challenge in the year 2015. Through this, we wish to explore and implement the theories and concepts we have learnt in the course CSE 578. Our main aim is to build interactive and unique visualizations from the datasets provided in the challenge which will help us deduce certain trends, outliers and enable us to make decisions. This in turn will help us solve some of the questions posed in the challenge. All the visualizations that we have built are based on D3.js which is a JavaScript library for producing dynamic, interactive data visualizations in web browsers. It makes use of Scalable Vector Graphics, HTML5, and Cascading Style Sheets standards.

## 2. Description of Problem Statement

In the VAST Mini Challenge 1, we are dealing with a typical modest-sized amusement park named the DinoFun World. One event last year was a weekend tribute to Scott Jones, an internationally renowned football ("soccer," in US terminology) star. Scott Jones is from a town nearby DinoFun World. He was a classic hometown hero, with thousands of fans who cheered his success as if he were a beloved family member. To celebrate his years of stardom, DinoFun World declared "Scott Jones Weekend", where Scott was scheduled to appear in two stage shows each on Friday, Saturday, and Sunday to talk about his life and career. In addition to this, a show of memorabilia related to his illustrious career were displayed in the park's Creighton Pavilion. However, the event did not go as planned. Scott Jones Weekend was marred by crime and mayhem perpetrated by a poor, misguided and disgruntled figure from Scott's past.

While the crimes were rapidly solved, park officials and law enforcement figures are interested in understanding just what happened during that weekend to better prepare themselves for future events. They are interested in understanding how people move and communicate in the park, as well as how patterns change and evolve over time, and what can be understood about motivations for changing patterns.

The problem statement deals with  a robbery that takes place in the park. We have to access the movement tracking information for all of the paying park visitors over the three days of the Scott Jones celebration. This data contains many patterns that are useful for planning park operations. On this particular weekend a crime occurred and the data likely contains information pertinent to that crime.

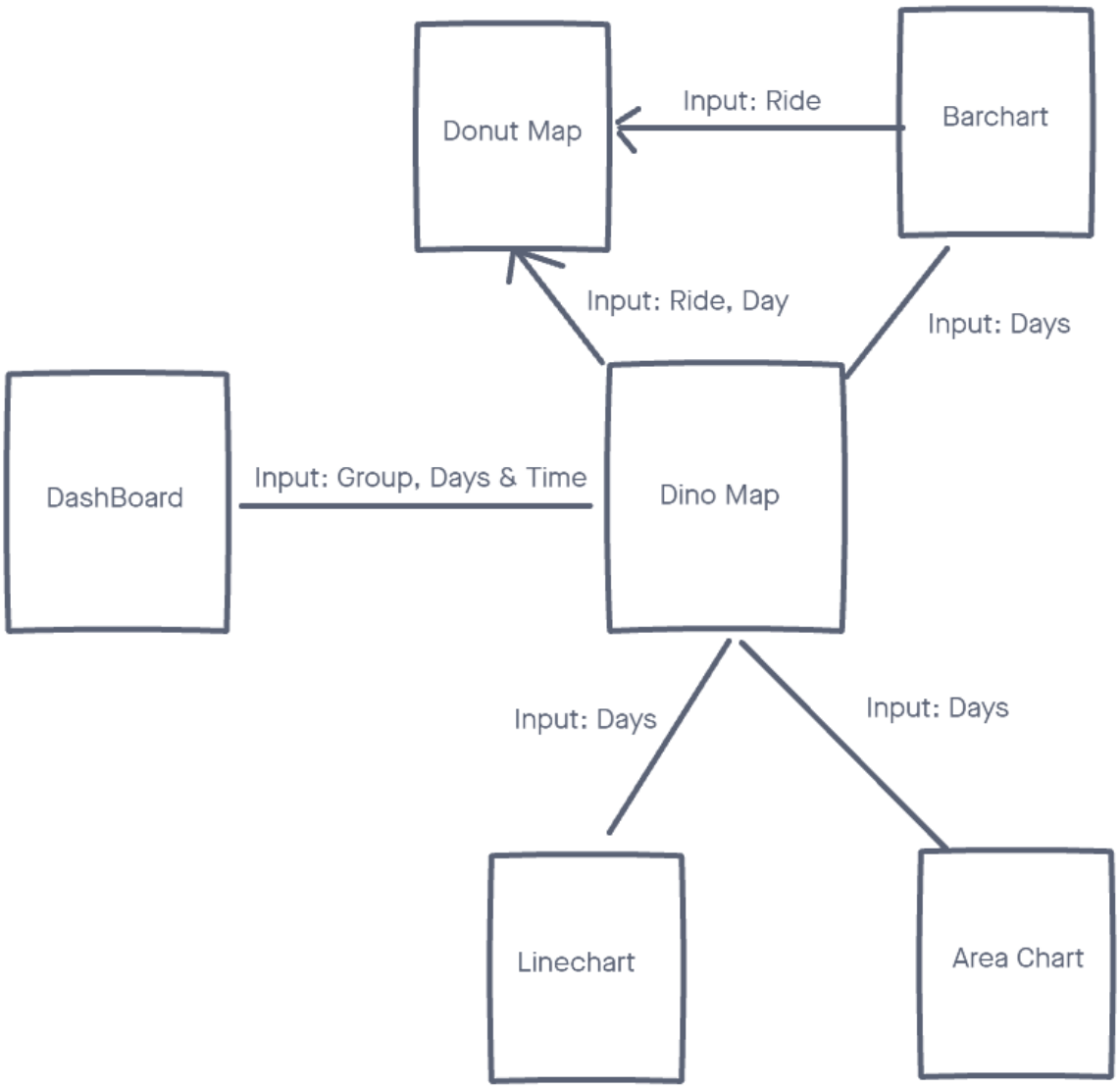
We have to use visual analytics to analyze the available data and develop responses to the questions below. We have to characterize the attendance at the park on that weekend and describe different types of groups at the park. Some of the questions to be answered pertaining to the groups are:

- How big is the group type?
- Where does this type of group like to go in the park?

- How common is this type of group?
- What are your other observations about this type of group?
- What can you infer about the group?
- If you were to make one improvement to the park to better meet this group's needs, what would it be?

We also need to infer notable differences in the patterns of activity in the park across the three days. We also need to hypothesize any anomalies or unusual patterns that we deem crucial in solving this robbery.

## 3. Visualization Design



## 4. Dataset Description

For this challenge, the original data was provided for three days: Friday, Saturday, and Sunday. Each day's dataset includes a timestamp, an id (a unique identifier for a person), and a type (check-in or movement). If the individual is checking in for a ride, the type is check-in, and if the person is moving around the park, the type is movement. The coordinates X and Y are used to track the movement of the person and the rides that they have checked in for.

Using the given movement data, we have extracted the following features for each person-

1) Total Time spent in the park.
2) Time of Entry.
3) Time of Exit.
4) Total Number of check-ins.
5) Total number of movements.

Using the following features, we created a feature matrix which allows us to classify different people into different clusters/groups using K-Means Classification. Each person's group details are stored in the *groups.csv* file.

For the Map, we have assumed a person to be the center of a cluster/group and have animated the movement around it. Since only one person per group's movement data is required, we remove the other person's row. The three days' data after compressing and deleting the row will be stored in fri.csv, sat.csv, and sun.csv. Which person in the group is assumed as the center of the cluster can be identified from the identifier_group.csv.

For Line chart, the above data is pre-processed for each day to build the dataset that is used to plot count of movements against timestamp. The pre-processed data includes Group no, count of movements, timestamp.

For the Bar chart, we first filtered the data day wise i.e., Friday, Saturday and Sunday. We then built an auxiliary dataset to hold the ride name and ride number. Using this, we accessed the

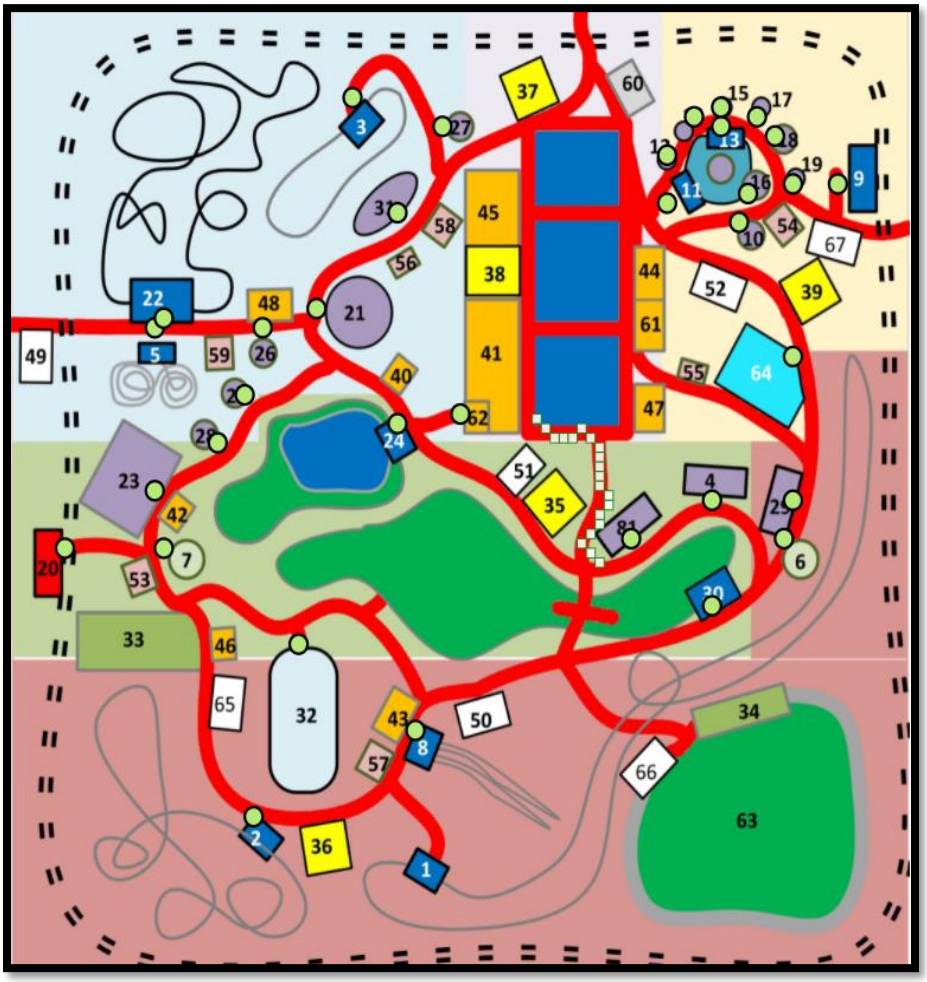
original check in data and counted the number of check-ins for each ride day wise. All the data wrangling for the Bar chart was done using Python (Pandas Library).
For the innovative chart, day wise data with timestamp, person id, their coordinates, Group ID, and ride id was created using pandas in a csv format and then a json file was created for each day using lodash library as discussed below.

- We saved the pre-calculated data in json format to avoid re-calculating the data for the donut chart.
- We have used the lodash library to maintain the code readability access to create.
  Process:
    1. Fetched unique groups and rides because our i/p is rides and dates.
    2. Filtered the data using outliers and false statements such as null and undefined values.
    3. By iterating through the unique rides, we have created a json of rides for a particular day. This JSON has a ride name as key and values as time frame object.
    4. Now our task is to create/save groups frequencies for corresponding rides and time frames together.
    5. As the time frame is not given in the csv we have calculated it using the moment js library for dates manipulation.
    6. And lastly updated the group count to create the inner donut layer.

## 5. Visualization Descriptions

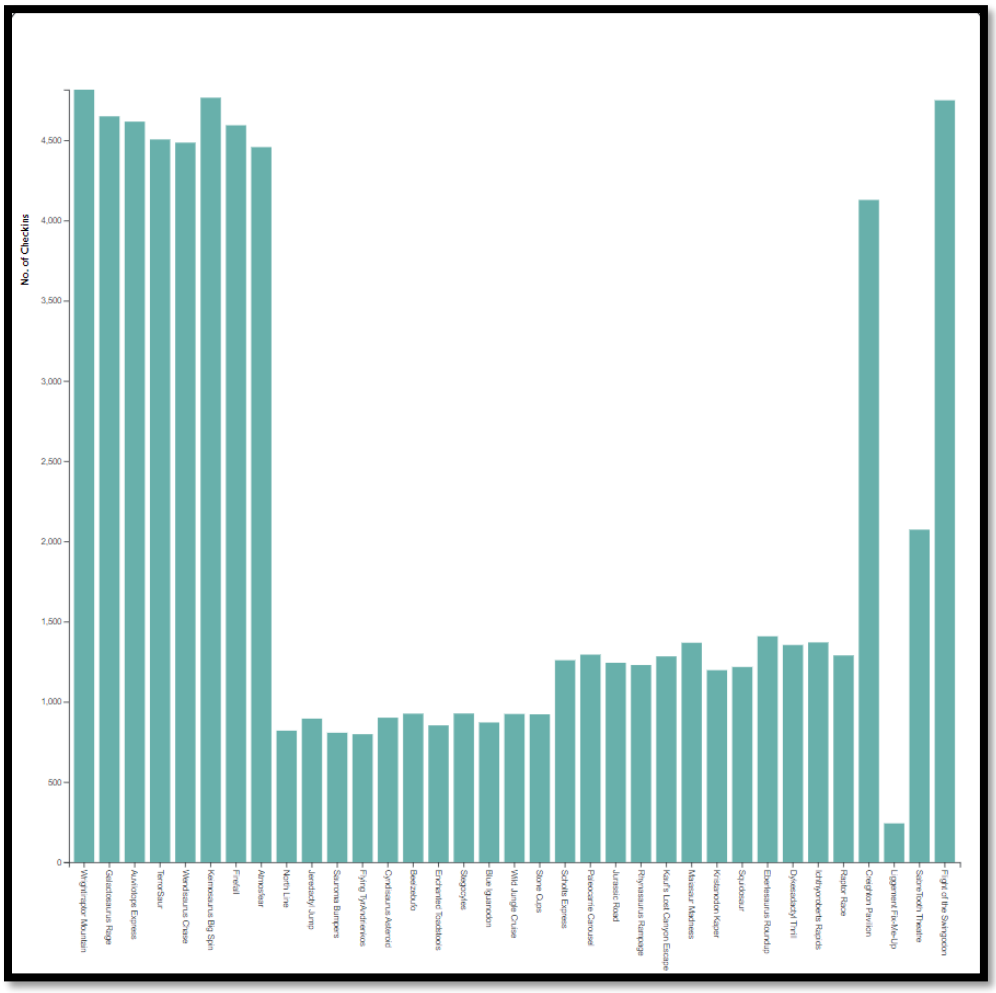
### 5.1 Map Visualization

The Map allows us to visualize the movement trajectory of a particular group on a particular day. The user can click the play button to run the animation of a particular group on the map. The user can apply time filters on the movement visualization to effectively visualize the trajectory between particular time. The Map also includes the Ride details which serve as the input to the innovative chart when clicked.



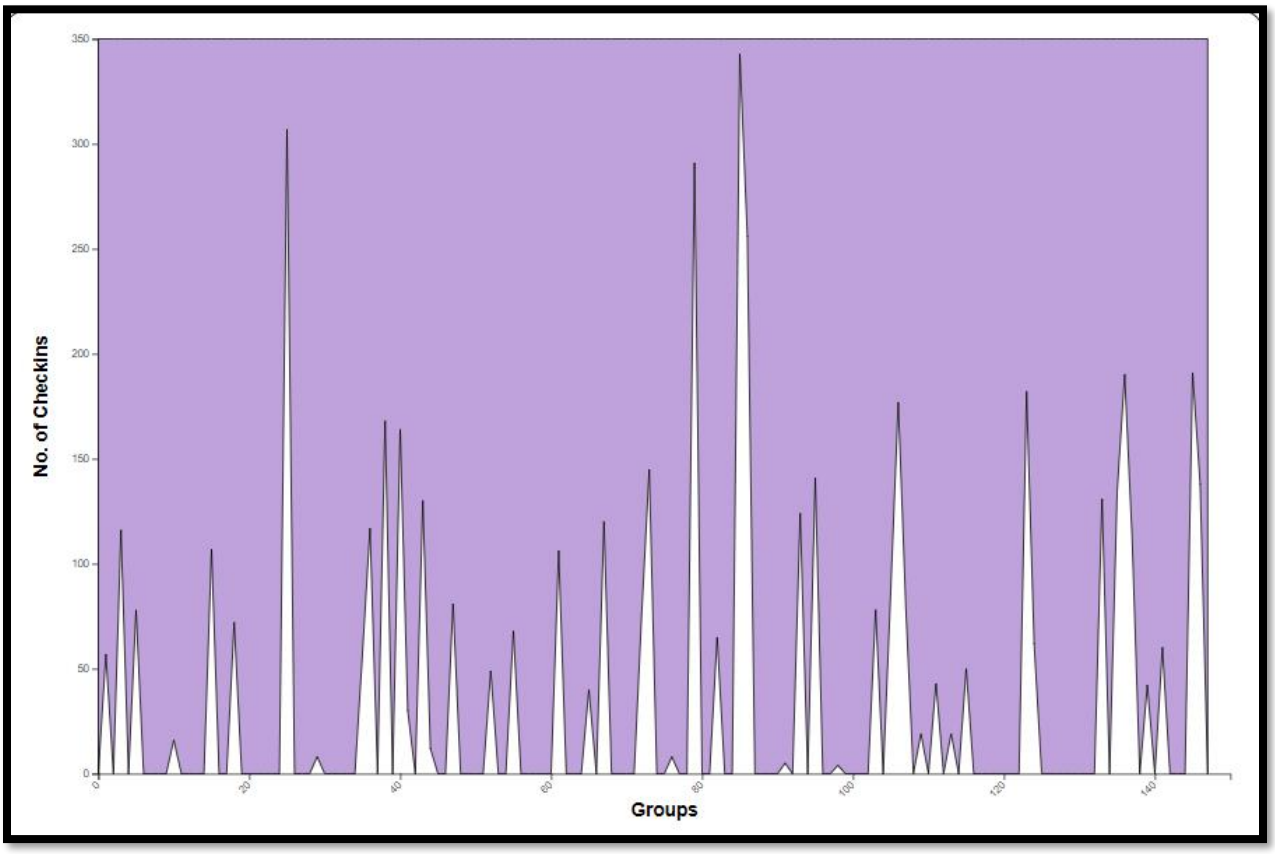
### 5.2 Bar Chart

In the Bar Chart, we wish to visualize the number of check-ins that occur on each day in the thrill rides. For this we first conducted data wrangling to filter out only the check in coordinates from the original dataset. We then built an auxiliary comma separated value file that had a list of rides and the count of check-ins of that ride for each day i.e. Friday, Saturday and Sunday. This was done by simultaneously comparing the check in coordinates and the coordinates of the entry points of the rides. This auxiliary data file was used to build the bar chart with the x axis holding the name of the thrill rides and the y axis representing the check in counts. The interactive segment of this chart is that it updates every time the day is changed. It also invokes a change in the innovative chart on clicking the bar. On clicking the bar, the innovative chart updates for that particular ride.



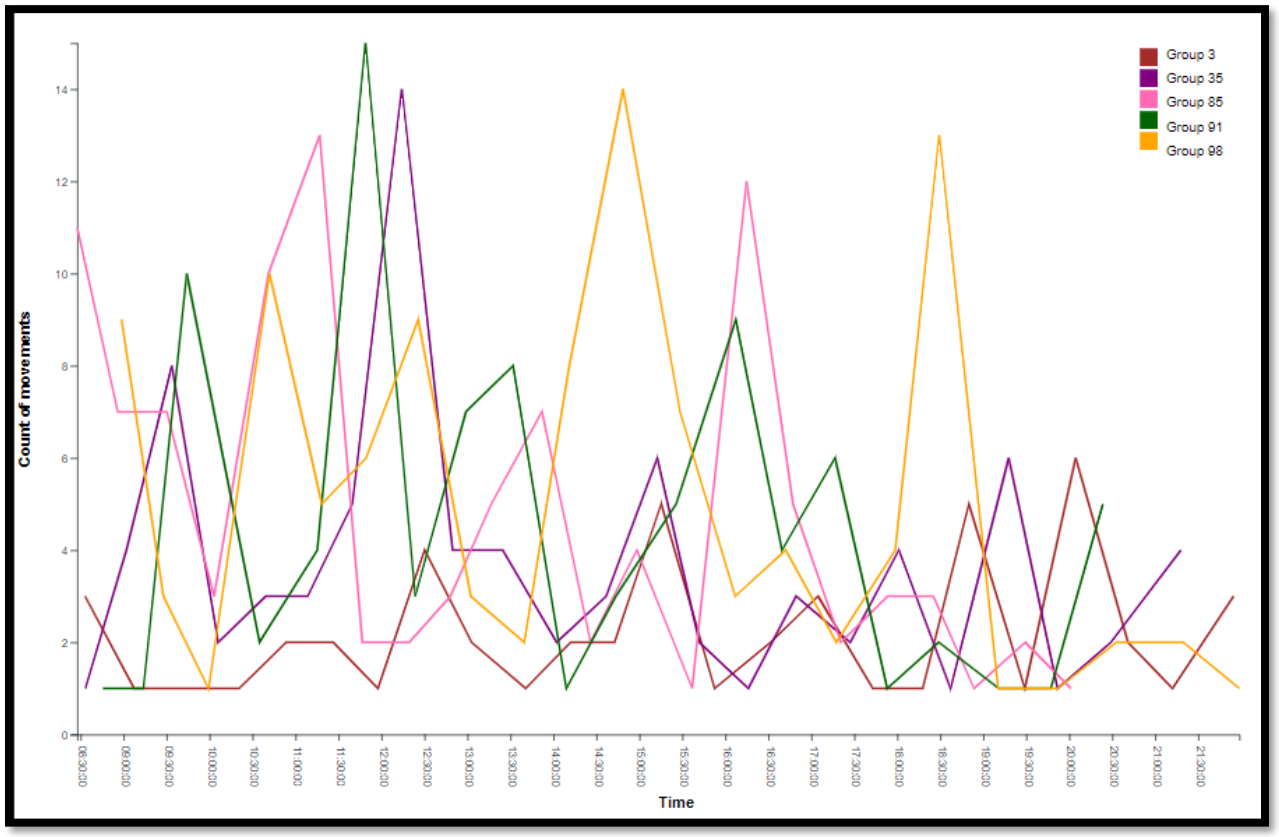
### 5.3 Floating Area Chart

The Area chart helps in visualizing the groups and their corresponding number of checkins for a particular day. The original data has been preprocessed to retrieve the groups a person belongs to and the total number of check-ins by all people in a group. This data has been calculated separately for three days: Friday, Saturday and Sunday. The data, which contains group ids and number of check-ins for 3 days, is stored in a csv. The area chart uses this data, with x-axis holding the group ids and number of check-ins in y-axis. We can observe some patterns from the chart, the total check ins on Friday is much lesser than other days and the groups which have excessively high check-ins or very low check ins can be investigated for suspicious movements further.



### 5.4 Line Chart

In the Line Chart, we can visualize the count of movements in the park for a set of groups in a day. We pre-processed the original data from three days (Friday, Saturday, Sunday) to plot this chart. For each group, we gathered data by grouping the date, timestamp, and type as movement. This yielded data containing date, timestamp, and count of movements, which was stored to a comma separated file. We created a line chart for a set of groups using this data, with timestamp on the x-axis and count of movements on the y-axis. This chart's interactive feature is that it updates every time the day is changed. A colored line is used to indicate each group. A tooltip with the group number is displayed for each line. We may compare the count of movements of different groups in a day using this chart.



### 5.5 Innovative Chart

Innovative charts can help visualize the hourly check in for a selected ride and day. It is a 2 layered donut chart with time frames and total check-in displayed in the outer layer and group wise number of check in for that particular time frame in the inner layer.

#### 5.5.1 Making of the donut chart:

For making the donut chart, we made a pie chart and subtracted an inner circle of a certain radius from the outer circle to create a donut.

1. We created a heat map for the outer layer of the donut - We get the sum of group counts for each time slot and divide that by 500 as that is the average value of check in on days per time slot to get the intensity of the pink color for outer donut chart.
2. Fetched the start and end angles from outer donut layer to make the inner donut layer segment for the particular time frame(from outer layer)

We created the timeframe split for the outer donut as follows :

- We defined all the timestamps as a list and created a list of json with timestamp, angle and frequency.
- We created pie for the outer circle and then segregated the data for the inner layer of the donut.
- We created and exported a function (updateDonutChart) from file donutChart.js to dynamically update the donut chart when a date and ride is selected as input.
- We are calling the updateDonutChart function from three input sources by using event listeners.
    1. Map.js (on clicking the ride circle in the map)
    2. Index.js (on day change event listener from the main dropdown)
    3. Barchart (upon clicking a particular bar as the ride name is passed as input)
- Mapping color for groups to color pallet - passing index as groupId so that color for corresponding group will be the same throughout the donut.

### 5.5.2 Innovative chart updation with change of input value:

Used JQuery library to manipulate the DOM - emptying the svg before creating it again.

## 6. Discussions

Our Visualization helps us narrow down from a group of 150 to a group of 10 from which we will be able to come up with a solution. The map helps us visualize the trajectory of a group. We can use the map and innovative chart, to find any anomaly in their activities throughout their time at the park. The Innovative charts can help visualize the hourly check in (total check in and group wise check in) for a selected ride and day. From the map and innovative chart, we can also investigate on an hourly basis if a particular group has visited a certain ride in a one-hour time frame. If they haven't visited that ride, then their activity around the rides can be marked as suspicious.

The bar chart provides insight into check-ins. If and when there is unusually high check-in or low check-in on a particular ride, we can find out the groups that attended the ride through the Area chart. The line chart provides us with details about the movement of groups at the park. If there is high movement from a group with low check-ins, it will surely raise a red flag and we can further monitor the group's activity to come to a conclusion.

## References


[1] http://www.vacommunity.org/VAST+Challenge+2015#Mini-Challenge_1

[2] https://www.sas.com/en_us/insights/big-data/data-visualization.html

[3] https://www.d3-graph-gallery.com/index.html

[4] https://d3js.org/