2015-unzip and rename the csv files
unzip Indego_Trips_2015Q2.zip
unzip Indego_Trips_2015Q3.zip
unzip Indego_Trips_2015Q4.zip
mv Indego_Trips_2015Q2.csv q2_2015 # renaming for convenience later
mv Indego_Trips_2015Q3.csv q3_2015
mv Indego_Trips_2015Q4.csv q4_2015

2016-unzip and rename the csv files
unzip Indego_Trips_2016Q1.zip
unzip Indego_Trips_2016Q2.zip
unzip Q3_2016_trips.zip
unzip Indego_trips_Q4_2016.zip
mv Indego_Trips_2016Q1.csv q1_2016
mv Indego_Trips_2016Q2.csv q2_2016
mv Q3_2016_trips.zip q3_2016
mv indego_trips_Q4_2016.zip q4_2016

2017-unzip and rename the csv files
unzip indego_gbfs_trips_Q1_2017.zip
unzip indego_gbfs_trips_Q2_2017.csv.zip
unzip indego-trips-2017-q3.csv.zip
unzip indego-trips-2017-q4.csv.zip
mv indego_gbfs_trips_Q1_2017.csv q1_2017
mv indego_gbfs_trips_Q2_2017.csv q2_2017
mv indego-trips-2017-q3.csv q3_2017
mv indego-quarter-echo.bicycletransit.com-2018-01-19-9-57AM.csv q4_2017

2018-unzip and rename the csv files
unzip indego-trips-2018-q1.csv.zip
unzip indego-trips-2018-q2.csv.zip
unzip indego-trips-2018-q3.csv.zip
mv indego-trips-2018-q1.csv q1_2018
mv indego-trips-2018-q2.csv q2_2018
mv indego-trips-2018-q3.csv q3_2018

Script for manipulating the files of 2015-2017 [start_end.sh]

```
#!/bin/bash
#the script below is for 2015,2016 and Q1 of 2017.
files=(q[0-9]_2015 q[0-9]_2016 q1_2017) #array of file names
for f in ${files[@]}
do
echo "start_date,start_time" > header_start
cut -d"," -f3 $f | sed "1d" | sed 's/ /,/g'| sed 's./.-.g' | sed 's/$/:00/' > start_col #start_date and start_time are separated to two different columns, empty space is replaced with comma
cat header_start start_col > start$f #Header to the new start date & time columns are added
rm start_col header_start
echo "end_date,end_time" > header_end
cut -d"," -f4 $f | sed "1d" | sed 's/ /,/g'| sed 's./.-.g' | sed 's/$/:00/' > end_col #end_date and end_time are separated to two different columns
cat header_end end_col > end$f #Header to the new end date & time columns are added
rm header_end end_col
cut -d "," -f1,2 $f > beginning$f
cut -d "," -f5,6,7,8,9,10,11,12,13,14 $f > last$f
paste -d "," beginning$f start$f end$f last$f  > $f.csv # recreating the the original file with changes added
rm beginning$f start$f end$f last$f
done
```

Files from 2015 to Q1 2017 are in same format coming in group1, the files from Q2 2017 to Q3 2018 are in same format coming in group2.
The start date and time are merged in one column which is separated and so is end date and time column data. The date format is changed from MM/DD/YY format to MM-DD-YYYY format to maintain uniformity with ISO 8601 format. The seconds (":00") is added in time format for group1 data files to be maintain similarity with group2 data files which has time in HH:MM:SS format.

#the script below is for Q2,Q3,Q4 2017 [start_end.sh]

```
philes=(q2_2017 q3_2017 q4_2017)#array of files name
echo "start_date,start_time" > sdt
echo "end_date,end_time" > edt
for p in ${philes[@]}
do
cut -d "," -f3  $p | sed "1d" | sed 's/ /,/' | sed 's/"//g'> third_col#Header is removed from start date & time column, empty space is replaced with comma, after extraction the column seems to be within double quotes so it is been removed.
cut -d "," -f1 third_col | sed 's/[0-9][0-9][0-9][0-9]-//' | sed 's/$/-2017/' > date # YYYY 2017 is removed from the beginning of the new column and added to the end to be in MM-DD-YYYY format in line with ISO 8601 format.
cut -d "," -f2 third_col > clock
paste -d "," date clock > date_time
cat sdt date_time > startdt$p #Header to the new start date & time columns are added
rm third_col date clock date_time
cut -d "," -f4  $p | sed "1d" | sed 's/ /,/' | sed 's/"//g'> third_col #end date and time column is extracted
cut -d "," -f1 third_col | sed 's/[0-9][0-9][0-9][0-9]-//' | sed 's/$/-2017/' > date # YYYY 2017 is removed from the beginning of the new column and added to the end to be in MM-DD-YYYY format in line with ISO 8601 format.
cut -d "," -f2 third_col > clock
paste -d "," date clock > date_time
cat edt date_time > enddt$p
rm third_col date clock date_time
cut -d "," -f1,2 $p| sed 's/"//g' > beginning$p
cut -d "," -f5,6,7,8,9,10,11,12,13,14 $p | sed 's/"//g'> last$p
paste -d "," beginning$p startdt$p enddt$p last$p > $p.csv
rm startdt$p enddt$p beginning$p last$p
done
```

```
rm edt sdt
```

```
#the script below is for Q1,Q2,Q3 2018 [start_end.sh]
zhiles=(q1_2018 q2_2018 q3_2018)
echo "start_date,start_time" > sdt
echo "end_date,end_time" > edt
for z in ${zhiles[@]}
do
cut -d "," -f3  $z | sed "1d" | sed 's/ /,/' | sed 's/"//g'> third_col #after extraction the column seems to be within double quotes so it is been removed.
cut -d "," -f1 third_col | sed 's/[0-9][0-9][0-9][0-9]-//' | sed 's/$/-2018/' > date
cut -d "," -f2 third_col > clock
paste -d "," date clock > date_time
cat sdt date_time > startdt$z
rm third_col date clock date_time
cut -d "," -f4  $z | sed "1d" | sed 's/ /,/'| sed 's/"//g' > third_col #end date and time column is extracted, after extraction the column seems to be within double quotes so it is been removed.
cut -d "," -f1 third_col | sed 's/[0-9][0-9][0-9][0-9]-//' | sed 's/$/-2018/' > date# YYYY 2018 is removed from the beginning of the new column and added to the end to be in MM-DD-YYYY format in line with ISO 8601 format.
cut -d "," -f2 third_col > clock
paste -d "," date clock > date_time
cat edt date_time > enddt$z
rm third_col date clock date_time
cut -d "," -f1,2 $z| sed 's/"//g' > beginning$z
cut -d "," -f5,6,7,8,9,10,11,12,13,14 $z | sed 's/"//g'> last$z
paste -d "," beginning$z startdt$z enddt$z last$z > $z.csv
rm startdt$z enddt$z beginning$z last$z
done
rm edt sdt
```

Changing the year for 2015

All the data files have date format in MM-DD-YYYY, thus converted the 2015 Q2-Q4 years from 15 to 2015

```
sed 's/-15/-2015/g' q2_2015.csv | sed 's/-2015-/-15-/g' > q2
sed 's/-15/-2015/g' q3_2015.csv | sed 's/-2015-/-15-/g' > q3
sed 's/-15/-2015/g' q4_2015.csv | sed 's/-2015-/-15-/g' > q4
rm q2_2015.csv q3_2015.csv q4_2015.csv
mv q2  q2_2015.csv
mv q3  q3_2015.csv
mv q4  q4_2015.csv
```

Sorting the files to be Unique, and by start date and start time column in ascending numeric order [sort.sh]

```
#!/bin/bash
sorting=(q[0-9]_*.csv)
for s in ${sorting[@]}
do
head -n 1 $s > head$s
tail -n +2 $s | sort -t "," -u -k3 -k4 -V >> data$s # sorting the entire data with respect to start_date and start_time column
cat head$s data$s > q$s
rm head$s data$s
done
```

For Q2 and Q3 2018 Day Pass is changed to One Day Pass as it is in all other files

```
sed 's/Day Pass/One Day Pass/g' qq2_2018.csv > 2q2018
sed 's/Day Pass/One Day Pass/g' qq3_2018.csv > 3q2018
rm qq2_2018.csv qq3_2018.csv
mv 2q2018 qq2_2018.csv
mv 3q2018 qq3_2018.csv
```

Data for bar plots

Using wc -l for each file, the number of lines are derived as following.

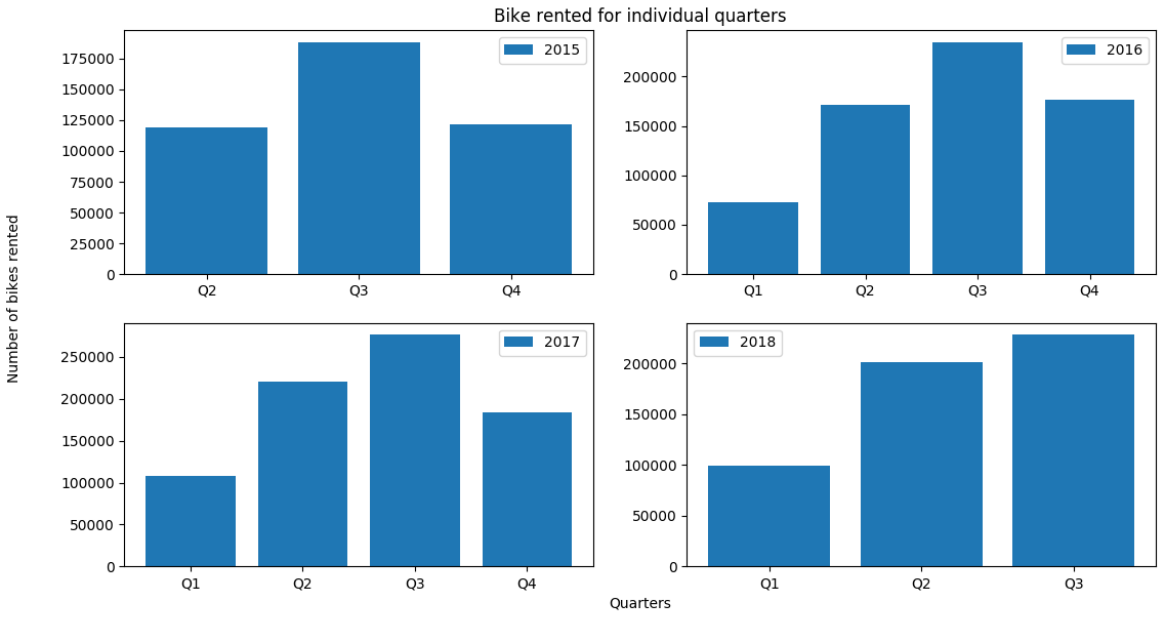
Quarter/Year	Number of observations
Q2-2015	118916
Q3-2015	188361
Q4-2015	121779
Q1-2016	73167
Q2-2016	170823
Q3-2016	234798
Q4-2016	176120
Q1-2017	107773
Q2-2017	220442
Q3-2017	276786
Q4-2017	183861
Q1-2018	98966
Q2-2018	201124
Q3-2018	228306

As per the Indego website, the trip_id are unique, thus finding the number of lines will give the number of rides.

Plot for which quarter gave highest business for Indego[bar.py]

```
#!/usr/bin/env python
import matplotlib as mpl
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure

data1 = {'Q2': 118916, 'Q3': 188361, 'Q4': 121779}
names1 = list(data1.keys())
values1 = list(data1.values())
data2 = {'Q1': 73167, 'Q2': 170823, 'Q3': 234798, 'Q4': 176120}
names2 = list(data2.keys())
values2 = list(data2.values())
data3 = {'Q1': 107773, 'Q2': 220442, 'Q3': 276786, 'Q4': 183861}
names3 = list(data3.keys())
values3 = list(data3.values())
data4 = {'Q1': 98966, 'Q2': 201124, 'Q3': 228306}
names4 = list(data4.keys())
values4 = list(data4.values())

# Initialize the plot
fig = plt.figure()
ax=fig.add_subplot(111)#adding this subplot for customizing the title and axis label which will be plotted over the 4 subplots
ax.set_xlabel('Quarters')
```

```
ax.set_ylabel('Number of bikes rented')
ax.yaxis.set_label_coords(-0.1,0.5)
ax.set_title('Bike rented for individual quarters')
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_color('none')
ax.spines['left'].set_color('none')
ax.tick_params(labelcolor='w',bottom='off', left='off', right='off', top='off')

ax1 = fig.add_subplot(221)
ax1.bar(names1, values1, label="2015")
plt.legend()

ax2 = fig.add_subplot(222)
ax2.bar(names2, values2, label="2016")
plt.legend()

ax3 = fig.add_subplot(223)
ax3.bar(names3, values3, label="2017")
plt.legend()

ax4 = fig.add_subplot(224)
ax4.bar(names4, values4, label="2018")
plt.legend()

plt.show()
```



The plot shows that in year 2017 for third quarter, Indego had good business ie. over 250k rides comparing to all other years, also from the plots above, it can be observed that for all third quarter of the year, which is from July to September people tend to ride more bikes.

Counting the discrete values of fields

In the csv files the three fields as below are discrete.

1. plan_duration having values among 0,1,30,365 in days
2. trip_route_category having values among “round trip” and “one way”
3.passholder_type having values among “one day pass”, “two day pass”,“walk-up”, “indego30”, “indego365” and “indegoflex”

Script for calculating the discrete values per file[count.sh]

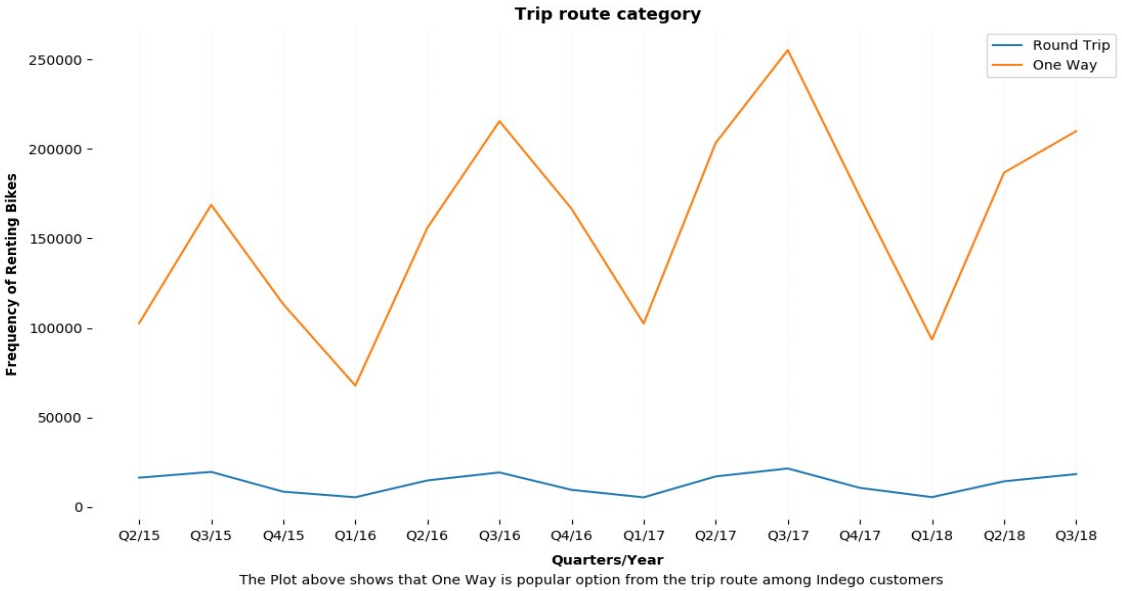
```
#!/bin/bash
files=(qq2_2015.csv qq3_2015.csv qq4_2015.csv qq1_2016.csv qq2_2016.csv qq3_2016.csv qq4_2016.csv qq1_2017.csv qq2_2017.csv qq3_2017.csv qq4_2017.csv qq1_2018.csv qq2_2018.csv qq3_2018.csv)
# could not use "qq*_.csv" because files alignment in my directory is not according to above order.
for f in ${files[@]}
do
cut -d"," -f14 $f > plan_col
grep -F -c "0" plan_col >> zero_plan.csv
grep -F -c "1" plan_col >> one_plan.csv
grep -F -c "2" plan_col >> two_plan.csv
grep -F -c "30" plan_col >> thirty_plan.csv
grep -F -c "365" plan_col >> threesix_plan.csv
rm plan_col
cut -d"," -f15 $f > route_category
grep -F -c "Round Trip" route_category >> round_trip.csv
grep -F -c "One Way" route_category >> one_way.csv
rm route_category
cut -d"," -f16 $f > passholder_type
grep -i -c "Walk-up" passholder_type >> walk.csv
grep -i -c "One Day Pass" passholder_type >> day.csv
grep -i -c "Two Day Pass" passholder_type >> two.csv
grep -i -c "indego30" passholder_type >> indegothirty.csv
grep -i -c "indego365" passholder_type >> indegosixtyfive.csv
grep -i -c "indegoFlex" passholder_type >> flex.csv
rm passholder_type
done
```

trip_route_category

Quarter/Year	Round Trip	One Way
Q2 2015	16318	102597
Q3 2015	19556	168804
Q4 2015	8483	113295
Q1 2016	5382	67784
Q2 2016	14786	156036
Q3 2016	19241	215556
Q4 2016	9503	166616
Q1 2017	5325	102447
Q2 2017	17018	203423
Q3 2017	21462	255323
Q4 2017	10616	173244
Q1 2018	5441	93524
Q2 2018	14301	186822
Q3 2018	18321	209984

Script for the plot of trip route category[trip_route.py]

```
#!/usr/bin/env python
import matplotlib as mpl
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
import numpy as np
txt="The Plot above shows that One Way is popular option from the trip route among Indego
customers" #caption
fig,ax=plt.subplots()
plt.grid(True, 'major', 'x', ls='-.', lw=.5, c='grey', alpha=.1)
#ax.xaxis.grid(alpha=0.1)for horizontal grids
plt.xlabel("Quarters/Year",fontweight='bold')
plt.ylabel("Frequency of Renting Bikes",fontweight='bold')
plt.title("Trip route category",fontweight='bold')
roun=[16318,19556,8483,5382,14786,19241,9503,5325,17018,21462,10616,5441,14301,18321]
one=[102597,168804,113295,67784,156036,215556,166616,102447,203423,255323,173244,93524,18
6822,209984]
ax.xaxis.set_label_coords(0.5,-0.07)
ax.yaxis.set_label_coords(-0.07,0.5)
# removing the spines
ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.spines['bottom'].set_visible(False)
ax.spines['left'].set_visible(False)
plt.xticks(np.arange(14),('Q2/15','Q3/15','Q4/15','Q1/16','Q2/16','Q3/16','Q4/16','Q1/17','Q2/17','Q3/17',
'Q4/17','Q1/18','Q2/18','Q3/18'))
plt.plot(roun, label='Round Trip')
plt.legend()
plt.plot(one, label='One Way')
fig.text(.5, .009, txt, ha='center')plt.legend()
plt.show()
```



It is very evident that Indego customers prefer more of one way trips than round way. One way trip exceeds 250000 rides alone in third quarter of 2017 and round trip is hardly crossing 3000 rides for any quarters from 2015 to 2018.

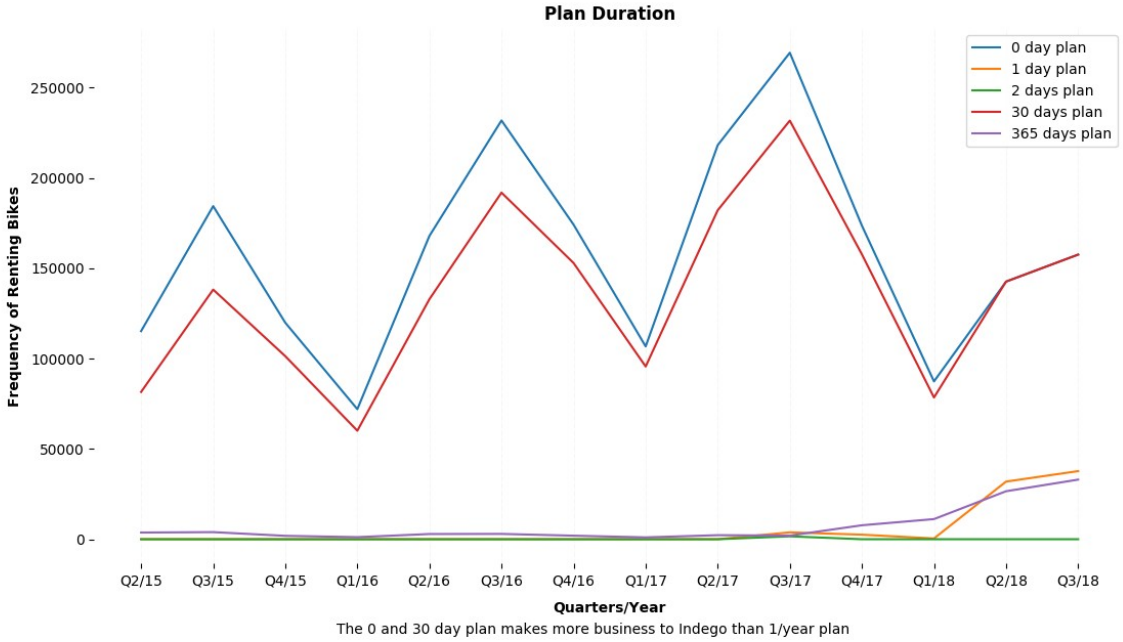
plan_duration

Quarter/Year	0 day plan	1 day plan	2 day plan	30 day plan	365 day plan
Q2 2015	115212	0	0	81554	3703
Q3 2015	184425	0	0	138209	3935
Q4 2015	119899	0	0	101369	1879
Q1 2016	72013	0	0	60088	1153
Q2 2016	167942	0	0	132910	2880
Q3 2016	231856	0	0	191906	2941
Q4 2016	174157	0	0	152918	1962
Q1 2017	106745	0	0	95626	1027
Q2 2017	218200	4	0	182154	2237
Q3 2017	269358	3805	1603	231773	2019
Q4 2017	173619	2517	0	157717	7724
Q1 2018	87380	439	0	78482	11146
Q2 2018	142643	31911	0	142643	26569
Q3 2018	157586	37713	0	157586	33006

Script plot of Plan Duration[plan-duration.py]

```
#!/usr/bin/env python
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
fig,ax=plt.subplots()
txt="The 0 and 30 day plan makes more business to Indego than 1 or year plan"
plt.grid(True, 'major', 'x', ls='-.', lw=.5, c='grey', alpha=.1)
#ax.xaxis.grid(alpha=0.1)for horizontal grids only
plt.xlabel("Quarters/Year",fontweight='bold')
plt.ylabel("Frequency of Renting Bikes",fontweight='bold')
plt.title("Plan Duration",fontweight='bold')
zero=[115212,184425,119899,72013,167942,231856,174157,106745,218200,269358,173619,87380,14
2643,157586]
one=[0,0,0,0,0,0,0,0,4,3805,2517,439,31911,37713]
two=[0,0,0,0,0,0,0,0,0,1603,0,0,0,0]
thirty=[81554,138209,101369,60088,132910,191906,152918,95626,182154,231773,157717,78482,14
2643,157586]
tresixtfiv=[3703,3935,1879,1153,2880,2941,1962,1027,2237,2019,7724,11146,26569,33006]
ax.xaxis.set_label_coords(0.5,-0.07)
ax.yaxis.set_label_coords(-0.07,0.5)
# removing the spines
ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.spines['bottom'].set_visible(False)
ax.spines['left'].set_visible(False)
plt.xticks(np.arange(14),('Q2/15','Q3/15','Q4/15','Q1/16','Q2/16','Q3/16','Q4/16','Q1/17','Q2/17','Q3/17',
'Q4/17','Q1/18','Q2/18','Q3/18'))
plt.plot(zero, label='0 day plan')
```

```
plt.legend()
plt.plot(one, label='1 day plan')
plt.legend()
plt.plot(two, label='2 days plan')
plt.legend()
plt.plot(thirty, label='30 days plan')
plt.legend()
plt.plot(tresixtfiv, label='365 days plan')
fig.text(.5, .009, txt, ha='center')
plt.legend()
plt.show()
```



From this plot it is evident that Indego customers prefer more of 0 and 30 day plan rather than 1 and 365 days plan. The two days plan was introduced in third quarter of 2017 and thereafter discontinued, the reason can be seen in plot, in 3Q of 2017 only 1603 passes were purchased by customers. The plot also shows that renting is more favorable in 3Q and drops down in 1Q of all the years. The trend also show that in 3Q of 2018 the renting of bikes must have been gone up but, may be because of in sufficient data we see that the renting is below than the 2015,2016 and 2017. One day and year plans doesnt seems like of an interest for indego customers. We can see its rise from 1Q of 2018 but will need data for few more years to conclude if could also bring more business for Indego.

pass holder type

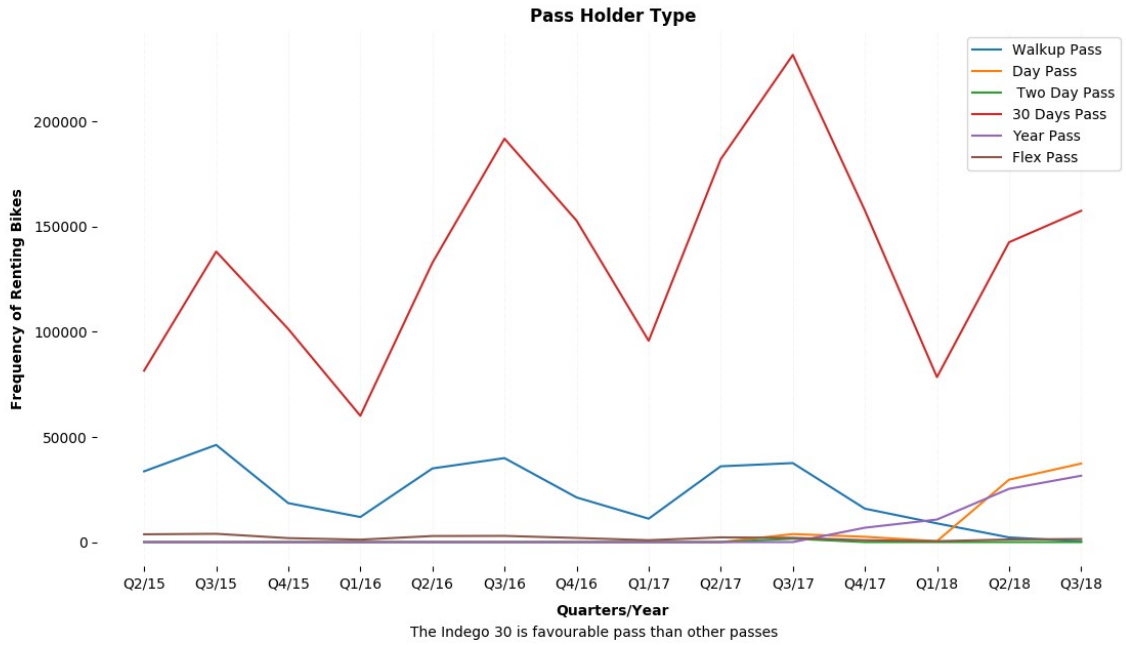
Quarter/Year	Walk up	Day Pass	2 day Pass	Indego 30	Indego365	IndegoFlex
Q2 2015	33658	0	0	81554	0	3703
Q3 2015	46216	0	0	138209	0	3935
Q4 2015	18530	0	0	101369	0	1879
Q1 2016	11925	0	0	60088	0	1153
Q2 2016	35032	0	0	132910	0	2880
Q3 2016	39950	0	0	191906	0	2941
Q4 2016	21239	0	0	152918	0	1962
Q1 2017	11123	0	0	95755	0	894
Q2 2017	36046	4	0	182154	0	2237
Q3 2017	37585	3805	1603	231773	0	2019
Q4 2017	15902	2517	0	157717	6840	884
Q1 2018	8898	439	0	78482	10726	420
Q2 2018	2216	29695	0	142643	25326	1243
Q3 2018	359	37354	0	157586	31561	1445

From the results above it is observed that IndegoFlex pass is distributed for both the customers who have bought plan duration 30 or 365 days. Although, it is unclear under which criteria IndegoFlex is provided for both of these plan durations.

Script for plotting Pass types [pass-type.py]

```
#!/usr/bin/env python
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
fig,ax=plt.subplots()
txt="The Indego 30 is favourable pass than other passes "
plt.grid(True, 'major', 'x', ls='-.', lw=.5, c='grey', alpha=.1)
#ax.xaxis.grid(alpha=0.1) for horizontal grids only
plt.xlabel("Quarters/Year",fontweight='bold')
plt.ylabel("Frequency of Renting Bikes",fontweight='bold')
plt.title("Pass Holder Type",fontweight='bold')
walkup=[33658,46216,18530,11925,35032,39950,21239,11123,36046,37585,15902,8898,2216,359]
daypass=[0,0,0,0,0,0,0,0,4,3805,2517,439,29695,37354]
twopass=[0,0,0,0,0,0,0,0,0,1603,0,0,0,0]
indego30=[81554,138209,101369,60088,132910,191906,152918,95755,182154,231773,157717,78482
,142643,157586]
indego365=[0,0,0,0,0,0,0,0,0,0,6840,10726,25326,31561]
indegoflex=[3703,3935,1879,1153,2880,2941,1962,894,2237,2019,884,420,1243,1445]
ax.xaxis.set_label_coords(0.5,-0.07)
ax.yaxis.set_label_coords(-0.07,0.5)
# removing the spines
ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.spines['bottom'].set_visible(False)
ax.spines['left'].set_visible(False)
plt.xticks(np.arange(14),('Q2/15','Q3/15','Q4/15','Q1/16','Q2/16','Q3/16','Q4/16','Q1/17','Q2/17','Q3/17',
'Q4/17','Q1/18','Q2/18','Q3/18'))
```

```
plt.plot(walkup, label='Walkup Pass')
plt.legend()
plt.plot(daypass, label='Day Pass')
plt.legend()
plt.plot(twopass, label=' Two Day Pass')
plt.legend()
plt.plot(indego30, label='30 Days Pass')
plt.legend()
plt.plot(indego365, label='Year Pass')
plt.legend()
plt.plot(indegoflex, label='Flex Pass')
plt.legend()
fig.text(.5, .009, txt, ha='center')
plt.show()
```



The plot shows that Indego customers prefer in majority to buy 30 days pass, but in reference to plan duration plot above, it seems that even after having 30 days pass, people choose more of 0 day plan.

Finding the number of bikes rented at the time of snow in philadelphia

I downloaded the weather data from noaa.gov website. The effects on renting of bikes at the day of snow.

I had to request for data from year 2015 to 2018 which was delivered to my e-mail, so curling was not an option. I saved the file, naming it snow.csv

Script for sorting the snow.csv[snowdata.sh]

```
#!/bin/bash
for i in snow.csv
do
```

```
cat snow.csv | grep "2015" | cut -d"," -f4,8 | sed 's/"//g' | sort -t"," -k2 -r > 2015.csv
cat snow.csv | grep "2016" | cut -d"," -f4,8 | sed 's/"//g' | sort -t "," -k2 -r > 2016.csv
cat snow.csv | grep "2017" | cut -d"," -f4,8 | sed 's/"//g' | sort -t "," -k2 -r > 2017.csv
cat snow.csv | grep "2018" | cut -d"," -f4,8 | sed 's/"//g' | sort -t "," -k2 -r > 2018.csv
done
```

As per the observation

In 2015 – (from April 23rd 2015 till end of the year) – there was no snow
In 2016- 5 days of snow
In 2017-10 days of snow
In 2018-(from beginning of the year till 30th september 2018)- there was no snow

The number rides are counted and saved in csv files for each day[date.sh]

The start date of every ride is calculated and saved for individual years

```
#!/bin/bash
f15=(qq2_2015.csv qq3_2015.csv qq4_2015.csv)
for f in ${f15[@]}
do
cut -d"," -f3 $f | sed "1d" | uniq -c | sed 's/^[ \t]*//'| sed 's/ /,/g'>> 15date.csv
done
f16=(qq1_2016.csv qq2_2016.csv qq3_2016.csv qq4_2016.csv)
for q in ${f16[@]}
do
cut -d"," -f3 $q | sed "1d" | uniq -c | sed 's/^[ \t]*//'| sed 's/ /,/g' >> 16date.csv
done
f17=(qq1_2017.csv qq2_2017.csv qq3_2017.csv qq4_2017.csv)
for a in ${f17[@]}
do
cut -d"," -f3 $a | sed "1d" | uniq -c | sed 's/^[ \t]*//'| sed 's/ /,/g' >> 17date.csv
done
f18=(qq1_2018.csv qq2_2018.csv qq3_2018.csv)
for z in ${f18[@]}
do
cut -d"," -f3 $z | sed "1d" | uniq -c | sed 's/^[ \t]*//'| sed 's/ /,/g' >> 18date.csv
done
```

Every start date column's observation share unique ride id, by counting the number of observations we get the number of rides customers rented on every unique date.

For 2016 and 2017 , philadelphia snow experience dates are stored for plotting purpose

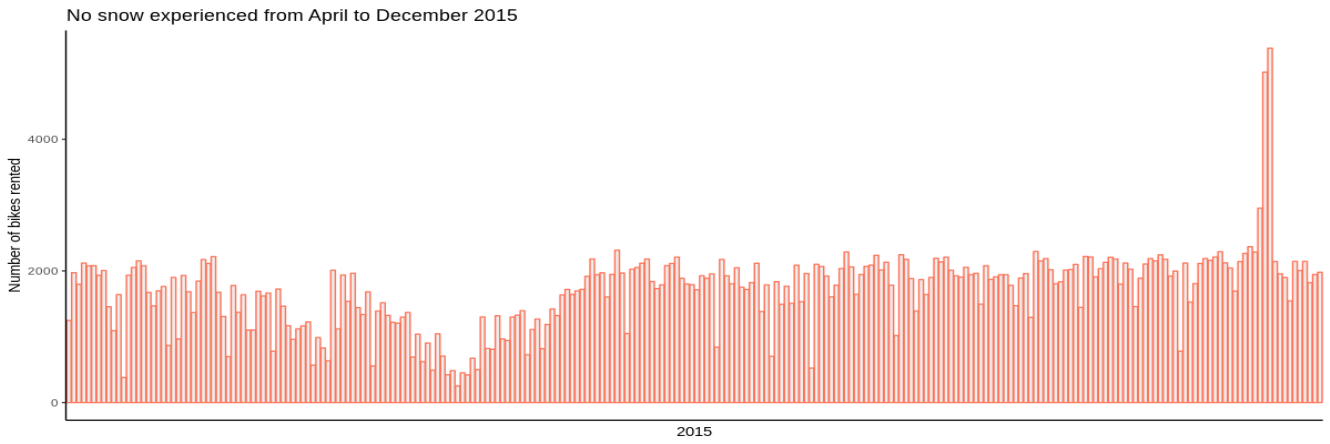
Since there are very few days associated with the snow in year 2016 and 2017 manually entered the dates filtered the corresponding rows for plotting purpose. The dates were collected from 2016.csv and 2017.csv as mentioned in page above

```
grep -w "1-18-2016" 16date.csv > 16snow.csv
grep -w "2-10-2016" 16date.csv >> 16snow.csv
grep -w "3-4-2016" 16date.csv >> 16snow.csv
grep -w "12-15-2016" 16date.csv >> 16snow.csv
grep -w "12-17-2016" 16date.csv >> 16snow.csv
```

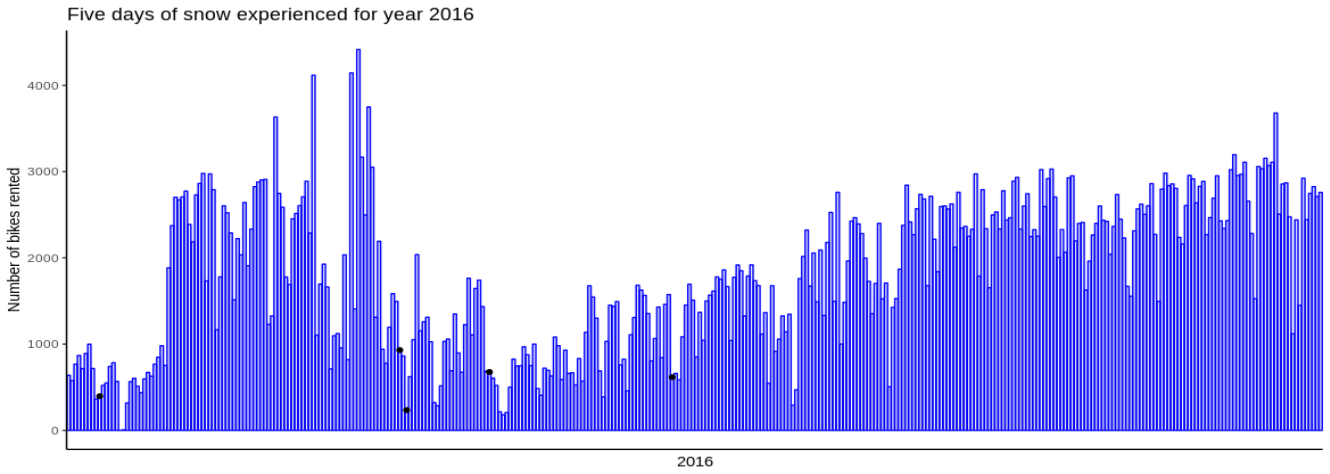
```
grep -w "1-6-2017" 17date.csv >> 17snow.csv
grep -w "1-7-2017" 17date.csv >> 17snow.csv
grep -w "1-8-2017" 17date.csv >> 17snow.csv
grep -w "2-9-2017" 17date.csv >> 17snow.csv
grep -w "2-10-2017" 17date.csv >> 17snow.csv
grep -w "3-14-2017" 17date.csv >> 17snow.csv
grep -w "12-10-2017" 17date.csv >> 17snow.csv
grep -w "12-14-2017" 17date.csv >> 17snow.csv
grep -w "12-16-2017" 17date.csv >> 17snow.csv
grep -w "12-30-2017" 17date.csv >> 17snow.csv
```

The script in R studio to plot the effects of renting of bikes on the day of snow

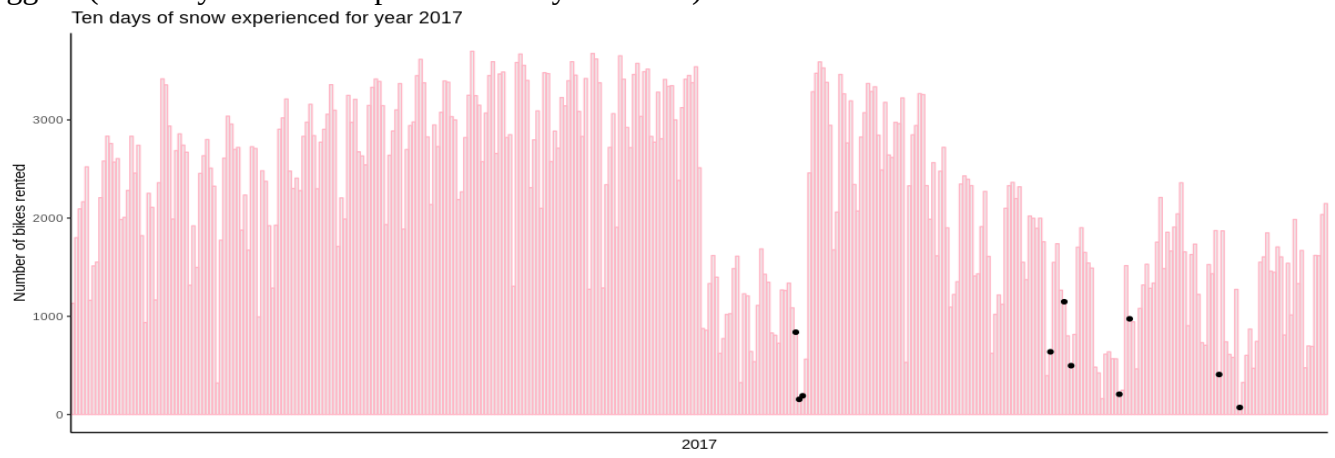
```
>fifteen <- read.csv("15date.csv")
>ggplot()+
geom_col(data=fifteen, aes(x=date,y=rate), color="coral1", alpha=0.1)+
theme(axis.text.x = element_blank(),axis.ticks.x = element_blank(),panel.grid.major =
element_blank(), panel.grid.minor = element_blank(),
panel.background = element_blank(), axis.line = element_line(colour = "black"))+xlab("2015")
+ylab("Number of bikes rented")+
ggtitle("No snow experienced from April to December 2015")
```



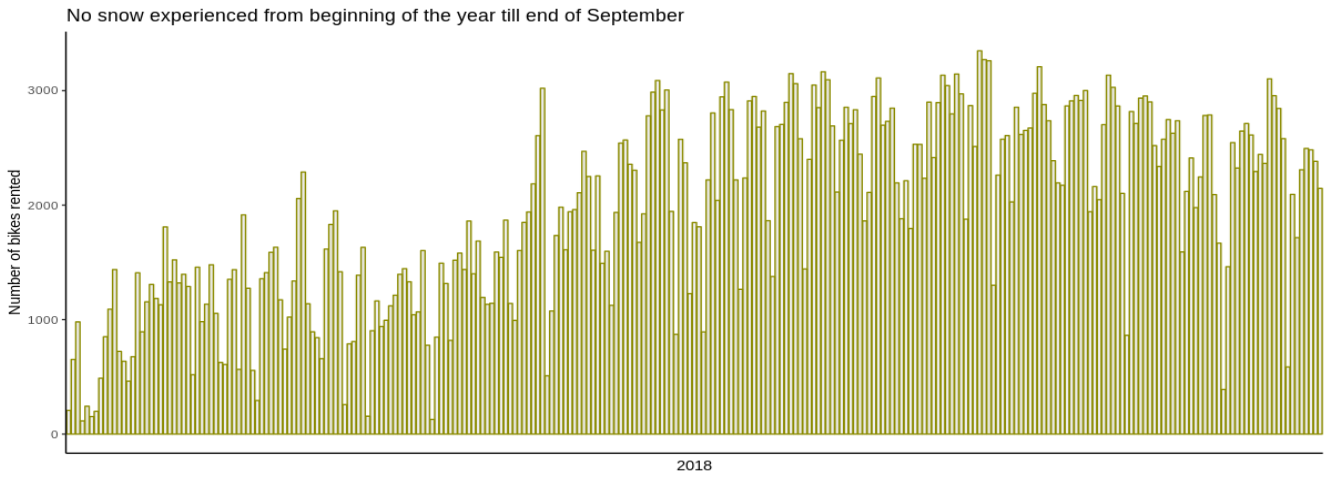

```
>sixteen <- read.csv("16date.csv")
>snow <- read.csv("16snow.csv")
>ggplot()+
geom_col(data=sixteen, aes(x=date, y=rate), color="blue", alpha=0.1)+
geom_point(data=snow, aes(x=day, y=num), color="black")+
theme(axis.text.x = element_blank(),axis.ticks.x = element_blank(),panel.grid.major =
element_blank(), panel.grid.minor = element_blank(),
panel.background = element_blank(), axis.line = element_line(colour = "black"))+xlab("2016")
+ylab("Number of bikes rented")+
ggtitle("Five days of snow experienced for year 2016")
```




```
>seventeen <- read.csv("17date.csv")
>snow <- read.csv("17snow.csv")
>ggplot()+
geom_col(data=seventeen, aes(x=date, y=rate, color="pink1", alpha=0.1)+
geom_point(data=snow, aes(x=day, y=num), color="black")+
theme(axis.text.x = element_blank(),axis.ticks.x = element_blank(),panel.grid.major =
element_blank(), panel.grid.minor = element_blank(),
panel.background = element_blank(), axis.line = element_line(colour = "black"))+xlab("2017")
+ylab("Number of bikes rented")+
ggtitle("Ten days of snow experienced for year 2017")
```




```
>eighteen <- read.csv("18date.csv")
>ggplot()+
geom_col(data=eighteen, aes(x=date, y=rate), color="yellow4", alpha=0.1)+
theme(axis.text.x = element_blank(),axis.ticks.x = element_blank(),panel.grid.major =
element_blank(), panel.grid.minor = element_blank(),
panel.background = element_blank(), axis.line = element_line(colour = "black"))+xlab("2018")
+ylab("Number of bikes rented")+
ggtitle("No snow experienced from beginning of the year till end of September")
```



To convert all the plot as GIF

convert -delay 100 -loop 0 -scale 480x270 *.png Indego.gif