



UChicago | MSCA 31012
Data Engineering Platforms for Analytics

June 6, 2019

Annie Qurat ul ain
WooJong Choi
Tam Nguyen
Markus Wehr

Outline

- Executive Summary
- Business Use Case
- Relational database and tools
- Data Analysis and Visualization
- Tableau Visualization
- Summary

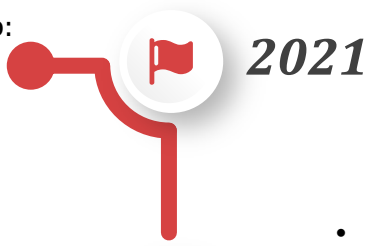




INTRODUCTION

Goal
Invest US\$50 million to:

- Expand stations to all **50 city wards**
- Add **175 stations** and **10,500 bikes**



2019 - 2020



- 2019: More than **20k rides** per day in peak seasons.
- March 2019, **Lyft** took over Divvy
- Early 2020: Plan to pass **20 millionth rides** mark.

Second expansion
(107 new stations)
Provided its 15 millionth rides in 2018



2017 - 2018

2015 - 2016



First expansion
(175 new stations)

Officially launched
in June 2013
(75 stations and 750 bikes)



2013

Bikeshare system



6,000 bikes



608 stations

Chicagoans' regular mode of transportation

RESEARCH OBJECTIVES

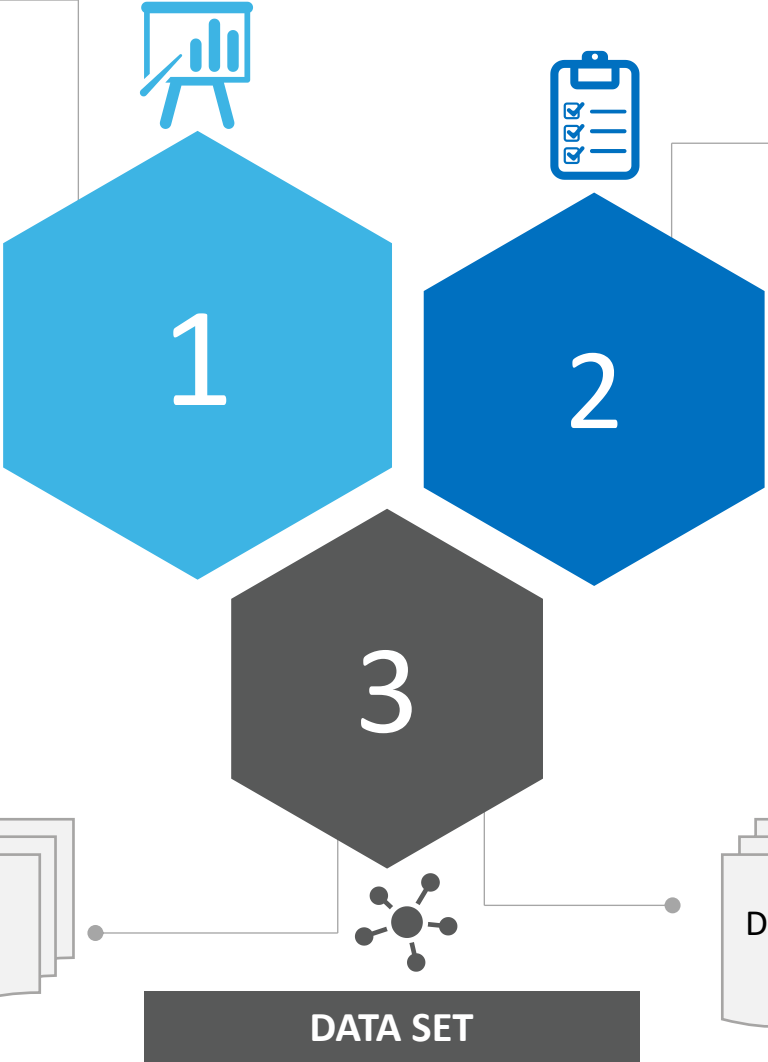
- To assist with the expansion plan, our team developed a relational database that will enable quick response and analysis on the current state Divvy operations in regard to ridership, station locations and various other factors affecting them. And:

- Provide methodologies and various tools used in the process
- Provide data analysis and visualization
- Put forward a future state blueprint for the new stations and bikes allocation process



PROPOSED FINDING

- Our final deliverables will enable Divvy leadership to:**
- Understand current ridership and station locations
 - Understand various factors that impact ridership. i.e
 - Demographic
 - Traffic volume
 - Bike racks / lanes
 - Weather
 - Develop dashboards and KPIs to gauge overall business / operation performance
 - Plan for future station & bikes allocation



METHODOLOGY

- Develop a scoring model to determine optimal number of stations and bikes by zip codes based on various factors
- Visualize findings from analysis - trends, outliers, patterns and predictions



Dataset	Source		File Format	Size
Trip	Divvy	https://www.divvybikes.com/system-data	CSV	> 1mil rows
Station	City of Chicago	https://data.cityofchicago.org/Transportation/Divvy-Bicycle-Stations/bbyy-e7gq	CSV	> 600 rows
Station_zip	Divvy	https://feeds.divvybikes.com/stations/stations.json	JSON	> 600 rows
Weather	National Weather Service Forecast Office	https://w2.weather.gov/climate/xmacis.php?wfo=lot	CSV	> 12k rows
Bike racks	City of Chicago	https://data.cityofchicago.org/Transportation/Bike-Racks/cbyb-69xx	CSV	> 5k rows
Population	City of Chicago	https://catalog.data.gov/dataset?res_format=CSV&organization=city-of-chicago	CSV	< 100 rows
Bike route	City of Chicago	https://data.cityofchicago.org/Transportation/Bike-Routes/3w5d-sru8	CSV	< 1k rows
Zip code	Chicago Data Type	http://robparal.blogspot.com/2013/07/chicago-community-area-and-zip-code.html	CSV	< 100 rows



Relational Database and Tools

Fact and dimensional table



Table Name	Table Type	Cardinality	Additional Details
fact_trip	Fact Table	M:1 Relationship with Station and Weather Table	Contains information about each trip including the start/end station, total time, age, gender of the customer
dim_station	Dimensional Table	1:M relationship with Fact Table	Contains information like station address, total number of docks available, date the station became available.
dim_weather	Dimensional Table	1:M relationship with Fact Table	Contains temperature, rain/snow, wind information in hourly format. Also, contains the sunset and sunrise time.
dim_population	Dimensional Table	1:M relationship with Location Table	Contains information about the population (age, gender) demographics zip wise.
dim_location	Dimensional Table	M:1 relationship with Population Table	Contains the location of all the stations, traffic routes, bike routes. Zip code is a must have for each address.
dim_traffic	Dimensional Table	1:M relationship with Location Table	Contains the traffic flow information daily including the direction (Northbound, Southbound, Westward, Eastward) on streets.
dim_bike_racks	Dimensional Table	1:M relationship with Location Table	Contains information about the non-divvy bike racks scattered across Chicago city
dim_bike_lane	Dimensional Table	1:1 relationship with Location Table	Contains information about the bike routes in the city, including their length and the streets they run on.

Fact table joined with Dimension tables provides interesting insights into how variables interact. Fact Table can be sliced by time and diced by stations, gender and age variables.

Database Design: Enhanced Entity Relational Diagram



Dimensional Schema: SNOWFLAKE

DDL

```

--- MySQL: Michael Ford Engineering
1
2
3 * SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
4 * SET @OLD_SQL_MODE=@SQL_MODE, @OLD_SQL_MODE_NAME=@SQL_MODE_NAME;
5
6
7 -- Schema: divvy
8
9
10 * CREATE SCHEMA IF NOT EXISTS 'divvy' DEFAULT CHARACTER SET utf8
11 * USE 'divvy' ;
12
13
14 -- Table 'divvy`.`dim_weather`
15
16 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_weather` (
17   `weather_date_hour` DATETIME NOT NULL,
18   `date` DATETIME NOT NULL,
19   `temperature` INT NOT NULL,
20   `wind` INT NOT NULL,
21   `precipitation` INT NOT NULL,
22   PRIMARY KEY (`weather_date_hour`))
23 ENGINE = InnoDB;
24
25
26 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_population` (
27   `zip` INT NOT NULL,
28   `0_19_M` INT NOT NULL,
29   `20_29_M` INT NOT NULL,
30   `30_39_M` INT NOT NULL,
31   `40_49_M` INT NOT NULL,
32   `50plus_M` INT NOT NULL,
33   `0_19_F` INT NOT NULL,
34   `20_29_F` INT NOT NULL,
35   `30_39_F` INT NOT NULL,
36   `40_49_F` INT NOT NULL,
37   `50plus_F` INT NOT NULL,
38   PRIMARY KEY (`zip`))
39 ENGINE = InnoDB;
40
41
42 -- Table 'divvy`.`dim_location`
43
44 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_location` (
45   `location_id` BIGINT NOT NULL,
46   `location_address` VARCHAR(60) NOT NULL,
47   `latitude` DOUBLE NOT NULL,
48   `longitude` DOUBLE NOT NULL,
49   `lat_long` MULTILINESTRING NOT NULL,
50   `zip` INT NOT NULL,
51   `location_type` VARCHAR(45) NOT NULL,
52   PRIMARY KEY (`location_id`))
53 ENGINE = InnoDB;
54
55
56 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_bike_racks` (
57   `rack_id` INT NOT NULL,
58   `location_id` BIGINT NOT NULL,
59   PRIMARY KEY (`rack_id`))
60 ENGINE = InnoDB;
61
62
63 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_station` (
64   `station_id` INT NOT NULL,
65   `station_name` VARCHAR(45) NOT NULL,
66   `total_docks` INT NOT NULL,
67   `location_id` BIGINT NOT NULL,
68   PRIMARY KEY (`station_id`))
69 ENGINE = InnoDB;
70
71
72 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_traffic` (
73   `traffic_id` INT NOT NULL,
74   `date` DATE NOT NULL,
75   `count_from` INT NOT NULL,
76   `count_to` INT NOT NULL,
77   `direction_from` VARCHAR(45) NOT NULL,
78   `direction_to` VARCHAR(45) NOT NULL,
79   `location_id` BIGINT NOT NULL,
80   PRIMARY KEY (`traffic_id`))
81 ENGINE = InnoDB;
82
83
84 * CREATE TABLE IF NOT EXISTS 'divvy`.`fact_trip` (
85   `trip_id` BIGINT NOT NULL,
86   `start_time` DATETIME NOT NULL,
87   `end_time` DATETIME NOT NULL,
88   `bike_id` INT NOT NULL,
89   `trip_duration` FLOAT NOT NULL,
90   `from_station_id` INT NOT NULL,
91   `to_station_id` INT NOT NULL,
92   `user_type` VARCHAR(45) NOT NULL,
93   `gender` VARCHAR(45) NOT NULL,
94   `age_group` VARCHAR(45) NOT NULL,
95   `weather_date_hour` BIGINT NOT NULL,
96   PRIMARY KEY (`trip_id`))
97 ENGINE = InnoDB;
98
99
100 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_bike_lane` (
101   `lane_id` INT NOT NULL,
102   `lane_type` VARCHAR(45) NOT NULL,
103   `from_street` VARCHAR(60) NOT NULL,
104   `to_street` VARCHAR(60) NOT NULL,
105   `length` FLOAT NOT NULL,
106   `location_id` BIGINT NOT NULL,
107   PRIMARY KEY (`lane_id`))
108 ENGINE = InnoDB;
109
110
111 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_weather` (
112   `weather_date_hour` DATETIME NOT NULL,
113   `date` DATETIME NOT NULL,
114   `temperature` INT NOT NULL,
115   `wind` INT NOT NULL,
116   `precipitation` INT NOT NULL,
117   PRIMARY KEY (`weather_date_hour`))
118 ENGINE = InnoDB;
119
120
121 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_population` (
122   `zip` INT NOT NULL,
123   `0_19_M` INT NOT NULL,
124   `20_29_M` INT NOT NULL,
125   `30_39_M` INT NOT NULL,
126   `40_49_M` INT NOT NULL,
127   `50plus_M` INT NOT NULL,
128   `0_19_F` INT NOT NULL,
129   `20_29_F` INT NOT NULL,
130   `30_39_F` INT NOT NULL,
131   `40_49_F` INT NOT NULL,
132   `50plus_F` INT NOT NULL,
133   PRIMARY KEY (`zip`))
134 ENGINE = InnoDB;
135
136
137 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_location` (
138   `location_id` BIGINT NOT NULL,
139   `location_address` VARCHAR(60) NOT NULL,
140   `latitude` DOUBLE NOT NULL,
141   `longitude` DOUBLE NOT NULL,
142   `lat_long` MULTILINESTRING NOT NULL,
143   `zip` INT NOT NULL,
144   `location_type` VARCHAR(45) NOT NULL,
145   PRIMARY KEY (`location_id`))
146 ENGINE = InnoDB;
147
148
149 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_bike_racks` (
150   `rack_id` INT NOT NULL,
151   `location_id` BIGINT NOT NULL,
152   PRIMARY KEY (`rack_id`))
153 ENGINE = InnoDB;
154
155
156 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_station` (
157   `station_id` INT NOT NULL,
158   `station_name` VARCHAR(45) NOT NULL,
159   `total_docks` INT NOT NULL,
160   `location_id` BIGINT NOT NULL,
161   PRIMARY KEY (`station_id`))
162 ENGINE = InnoDB;
163
164
165 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_traffic` (
166   `traffic_id` INT NOT NULL,
167   `date` DATE NOT NULL,
168   `count_from` INT NOT NULL,
169   `count_to` INT NOT NULL,
170   `direction_from` VARCHAR(45) NOT NULL,
171   `direction_to` VARCHAR(45) NOT NULL,
172   `location_id` BIGINT NOT NULL,
173   PRIMARY KEY (`traffic_id`))
174 ENGINE = InnoDB;
175
176
177 * CREATE TABLE IF NOT EXISTS 'divvy`.`fact_trip` (
178   `trip_id` BIGINT NOT NULL,
179   `start_time` DATETIME NOT NULL,
180   `end_time` DATETIME NOT NULL,
181   `bike_id` INT NOT NULL,
182   `trip_duration` FLOAT NOT NULL,
183   `from_station_id` INT NOT NULL,
184   `to_station_id` INT NOT NULL,
185   `user_type` VARCHAR(45) NOT NULL,
186   `gender` VARCHAR(45) NOT NULL,
187   `age_group` VARCHAR(45) NOT NULL,
188   `weather_date_hour` BIGINT NOT NULL,
189   PRIMARY KEY (`trip_id`))
190 ENGINE = InnoDB;
191
192
193 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_bike_lane` (
194   `lane_id` INT NOT NULL,
195   `lane_type` VARCHAR(45) NOT NULL,
196   `from_street` VARCHAR(60) NOT NULL,
197   `to_street` VARCHAR(60) NOT NULL,
198   `length` FLOAT NOT NULL,
199   `location_id` BIGINT NOT NULL,
200   PRIMARY KEY (`lane_id`))
201 ENGINE = InnoDB;
202
203
204 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_weather` (
205   `weather_date_hour` DATETIME NOT NULL,
206   `date` DATETIME NOT NULL,
207   `temperature` INT NOT NULL,
208   `wind` INT NOT NULL,
209   `precipitation` INT NOT NULL,
210   PRIMARY KEY (`weather_date_hour`))
211 ENGINE = InnoDB;
212
213
214 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_population` (
215   `zip` INT NOT NULL,
216   `0_19_M` INT NOT NULL,
217   `20_29_M` INT NOT NULL,
218   `30_39_M` INT NOT NULL,
219   `40_49_M` INT NOT NULL,
220   `50plus_M` INT NOT NULL,
221   `0_19_F` INT NOT NULL,
222   `20_29_F` INT NOT NULL,
223   `30_39_F` INT NOT NULL,
224   `40_49_F` INT NOT NULL,
225   `50plus_F` INT NOT NULL,
226   PRIMARY KEY (`zip`))
227 ENGINE = InnoDB;
228
229
230 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_location` (
231   `location_id` BIGINT NOT NULL,
232   `location_address` VARCHAR(60) NOT NULL,
233   `latitude` DOUBLE NOT NULL,
234   `longitude` DOUBLE NOT NULL,
235   `lat_long` MULTILINESTRING NOT NULL,
236   `zip` INT NOT NULL,
237   `location_type` VARCHAR(45) NOT NULL,
238   PRIMARY KEY (`location_id`))
239 ENGINE = InnoDB;
240
241
242 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_bike_racks` (
243   `rack_id` INT NOT NULL,
244   `location_id` BIGINT NOT NULL,
245   PRIMARY KEY (`rack_id`))
246 ENGINE = InnoDB;
247
248
249 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_station` (
250   `station_id` INT NOT NULL,
251   `station_name` VARCHAR(45) NOT NULL,
252   `total_docks` INT NOT NULL,
253   `location_id` BIGINT NOT NULL,
254   PRIMARY KEY (`station_id`))
255 ENGINE = InnoDB;
256
257
258 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_traffic` (
259   `traffic_id` INT NOT NULL,
260   `date` DATE NOT NULL,
261   `count_from` INT NOT NULL,
262   `count_to` INT NOT NULL,
263   `direction_from` VARCHAR(45) NOT NULL,
264   `direction_to` VARCHAR(45) NOT NULL,
265   `location_id` BIGINT NOT NULL,
266   PRIMARY KEY (`traffic_id`))
267 ENGINE = InnoDB;
268
269
270 * CREATE TABLE IF NOT EXISTS 'divvy`.`fact_trip` (
271   `trip_id` BIGINT NOT NULL,
272   `start_time` DATETIME NOT NULL,
273   `end_time` DATETIME NOT NULL,
274   `bike_id` INT NOT NULL,
275   `trip_duration` FLOAT NOT NULL,
276   `from_station_id` INT NOT NULL,
277   `to_station_id` INT NOT NULL,
278   `user_type` VARCHAR(45) NOT NULL,
279   `gender` VARCHAR(45) NOT NULL,
280   `age_group` VARCHAR(45) NOT NULL,
281   `weather_date_hour` BIGINT NOT NULL,
282   PRIMARY KEY (`trip_id`))
283 ENGINE = InnoDB;
284
285
286 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_bike_lane` (
287   `lane_id` INT NOT NULL,
288   `lane_type` VARCHAR(45) NOT NULL,
289   `from_street` VARCHAR(60) NOT NULL,
290   `to_street` VARCHAR(60) NOT NULL,
291   `length` FLOAT NOT NULL,
292   `location_id` BIGINT NOT NULL,
293   PRIMARY KEY (`lane_id`))
294 ENGINE = InnoDB;
295
296
297 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_weather` (
298   `weather_date_hour` DATETIME NOT NULL,
299   `date` DATETIME NOT NULL,
300   `temperature` INT NOT NULL,
301   `wind` INT NOT NULL,
302   `precipitation` INT NOT NULL,
303   PRIMARY KEY (`weather_date_hour`))
304 ENGINE = InnoDB;
305
306
307 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_population` (
308   `zip` INT NOT NULL,
309   `0_19_M` INT NOT NULL,
310   `20_29_M` INT NOT NULL,
311   `30_39_M` INT NOT NULL,
312   `40_49_M` INT NOT NULL,
313   `50plus_M` INT NOT NULL,
314   `0_19_F` INT NOT NULL,
315   `20_29_F` INT NOT NULL,
316   `30_39_F` INT NOT NULL,
317   `40_49_F` INT NOT NULL,
318   `50plus_F` INT NOT NULL,
319   PRIMARY KEY (`zip`))
320 ENGINE = InnoDB;
321
322
323 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_location` (
324   `location_id` BIGINT NOT NULL,
325   `location_address` VARCHAR(60) NOT NULL,
326   `latitude` DOUBLE NOT NULL,
327   `longitude` DOUBLE NOT NULL,
328   `lat_long` MULTILINESTRING NOT NULL,
329   `zip` INT NOT NULL,
330   `location_type` VARCHAR(45) NOT NULL,
331   PRIMARY KEY (`location_id`))
332 ENGINE = InnoDB;
333
334
335 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_bike_racks` (
336   `rack_id` INT NOT NULL,
337   `location_id` BIGINT NOT NULL,
338   PRIMARY KEY (`rack_id`))
339 ENGINE = InnoDB;
340
341
342 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_station` (
343   `station_id` INT NOT NULL,
344   `station_name` VARCHAR(45) NOT NULL,
345   `total_docks` INT NOT NULL,
346   `location_id` BIGINT NOT NULL,
347   PRIMARY KEY (`station_id`))
348 ENGINE = InnoDB;
349
350
351 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_traffic` (
352   `traffic_id` INT NOT NULL,
353   `date` DATE NOT NULL,
354   `count_from` INT NOT NULL,
355   `count_to` INT NOT NULL,
356   `direction_from` VARCHAR(45) NOT NULL,
357   `direction_to` VARCHAR(45) NOT NULL,
358   `location_id` BIGINT NOT NULL,
359   PRIMARY KEY (`traffic_id`))
360 ENGINE = InnoDB;
361
362
363 * CREATE TABLE IF NOT EXISTS 'divvy`.`fact_trip` (
364   `trip_id` BIGINT NOT NULL,
365   `start_time` DATETIME NOT NULL,
366   `end_time` DATETIME NOT NULL,
367   `bike_id` INT NOT NULL,
368   `trip_duration` FLOAT NOT NULL,
369   `from_station_id` INT NOT NULL,
370   `to_station_id` INT NOT NULL,
371   `user_type` VARCHAR(45) NOT NULL,
372   `gender` VARCHAR(45) NOT NULL,
373   `age_group` VARCHAR(45) NOT NULL,
374   `weather_date_hour` BIGINT NOT NULL,
375   PRIMARY KEY (`trip_id`))
376 ENGINE = InnoDB;
377
378
379 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_bike_lane` (
380   `lane_id` INT NOT NULL,
381   `lane_type` VARCHAR(45) NOT NULL,
382   `from_street` VARCHAR(60) NOT NULL,
383   `to_street` VARCHAR(60) NOT NULL,
384   `length` FLOAT NOT NULL,
385   `location_id` BIGINT NOT NULL,
386   PRIMARY KEY (`lane_id`))
387 ENGINE = InnoDB;
388
389
390 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_weather` (
391   `weather_date_hour` DATETIME NOT NULL,
392   `date` DATETIME NOT NULL,
393   `temperature` INT NOT NULL,
394   `wind` INT NOT NULL,
395   `precipitation` INT NOT NULL,
396   PRIMARY KEY (`weather_date_hour`))
397 ENGINE = InnoDB;
398
399
400 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_population` (
401   `zip` INT NOT NULL,
402   `0_19_M` INT NOT NULL,
403   `20_29_M` INT NOT NULL,
404   `30_39_M` INT NOT NULL,
405   `40_49_M` INT NOT NULL,
406   `50plus_M` INT NOT NULL,
407   `0_19_F` INT NOT NULL,
408   `20_29_F` INT NOT NULL,
409   `30_39_F` INT NOT NULL,
410   `40_49_F` INT NOT NULL,
411   `50plus_F` INT NOT NULL,
412   PRIMARY KEY (`zip`))
413 ENGINE = InnoDB;
414
415
416 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_location` (
417   `location_id` BIGINT NOT NULL,
418   `location_address` VARCHAR(60) NOT NULL,
419   `latitude` DOUBLE NOT NULL,
420   `longitude` DOUBLE NOT NULL,
421   `lat_long` MULTILINESTRING NOT NULL,
422   `zip` INT NOT NULL,
423   `location_type` VARCHAR(45) NOT NULL,
424   PRIMARY KEY (`location_id`))
425 ENGINE = InnoDB;
426
427
428 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_bike_racks` (
429   `rack_id` INT NOT NULL,
430   `location_id` BIGINT NOT NULL,
431   PRIMARY KEY (`rack_id`))
432 ENGINE = InnoDB;
433
434
435 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_station` (
436   `station_id` INT NOT NULL,
437   `station_name` VARCHAR(45) NOT NULL,
438   `total_docks` INT NOT NULL,
439   `location_id` BIGINT NOT NULL,
440   PRIMARY KEY (`station_id`))
441 ENGINE = InnoDB;
442
443
444 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_traffic` (
445   `traffic_id` INT NOT NULL,
446   `date` DATE NOT NULL,
447   `count_from` INT NOT NULL,
448   `count_to` INT NOT NULL,
449   `direction_from` VARCHAR(45) NOT NULL,
450   `direction_to` VARCHAR(45) NOT NULL,
451   `location_id` BIGINT NOT NULL,
452   PRIMARY KEY (`traffic_id`))
453 ENGINE = InnoDB;
454
455
456 * CREATE TABLE IF NOT EXISTS 'divvy`.`fact_trip` (
457   `trip_id` BIGINT NOT NULL,
458   `start_time` DATETIME NOT NULL,
459   `end_time` DATETIME NOT NULL,
460   `bike_id` INT NOT NULL,
461   `trip_duration` FLOAT NOT NULL,
462   `from_station_id` INT NOT NULL,
463   `to_station_id` INT NOT NULL,
464   `user_type` VARCHAR(45) NOT NULL,
465   `gender` VARCHAR(45) NOT NULL,
466   `age_group` VARCHAR(45) NOT NULL,
467   `weather_date_hour` BIGINT NOT NULL,
468   PRIMARY KEY (`trip_id`))
469 ENGINE = InnoDB;
470
471
472 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_bike_lane` (
473   `lane_id` INT NOT NULL,
474   `lane_type` VARCHAR(45) NOT NULL,
475   `from_street` VARCHAR(60) NOT NULL,
476   `to_street` VARCHAR(60) NOT NULL,
477   `length` FLOAT NOT NULL,
478   `location_id` BIGINT NOT NULL,
479   PRIMARY KEY (`lane_id`))
480 ENGINE = InnoDB;
481
482
483 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_weather` (
484   `weather_date_hour` DATETIME NOT NULL,
485   `date` DATETIME NOT NULL,
486   `temperature` INT NOT NULL,
487   `wind` INT NOT NULL,
488   `precipitation` INT NOT NULL,
489   PRIMARY KEY (`weather_date_hour`))
490 ENGINE = InnoDB;
491
492
493 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_population` (
494   `zip` INT NOT NULL,
495   `0_19_M` INT NOT NULL,
496   `20_29_M` INT NOT NULL,
497   `30_39_M` INT NOT NULL,
498   `40_49_M` INT NOT NULL,
499   `50plus_M` INT NOT NULL,
500   `0_19_F` INT NOT NULL,
501   `20_29_F` INT NOT NULL,
502   `30_39_F` INT NOT NULL,
503   `40_49_F` INT NOT NULL,
504   `50plus_F` INT NOT NULL,
505   PRIMARY KEY (`zip`))
506 ENGINE = InnoDB;
507
508
509 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_location` (
510   `location_id` BIGINT NOT NULL,
511   `location_address` VARCHAR(60) NOT NULL,
512   `latitude` DOUBLE NOT NULL,
513   `longitude` DOUBLE NOT NULL,
514   `lat_long` MULTILINESTRING NOT NULL,
515   `zip` INT NOT NULL,
516   `location_type` VARCHAR(45) NOT NULL,
517   PRIMARY KEY (`location_id`))
518 ENGINE = InnoDB;
519
520
521 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_bike_racks` (
522   `rack_id` INT NOT NULL,
523   `location_id` BIGINT NOT NULL,
524   PRIMARY KEY (`rack_id`))
525 ENGINE = InnoDB;
526
527
528 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_station` (
529   `station_id` INT NOT NULL,
530   `station_name` VARCHAR(45) NOT NULL,
531   `total_docks` INT NOT NULL,
532   `location_id` BIGINT NOT NULL,
533   PRIMARY KEY (`station_id`))
534 ENGINE = InnoDB;
535
536
537 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_traffic` (
538   `traffic_id` INT NOT NULL,
539   `date` DATE NOT NULL,
540   `count_from` INT NOT NULL,
541   `count_to` INT NOT NULL,
542   `direction_from` VARCHAR(45) NOT NULL,
543   `direction_to` VARCHAR(45) NOT NULL,
544   `location_id` BIGINT NOT NULL,
545   PRIMARY KEY (`traffic_id`))
546 ENGINE = InnoDB;
547
548
549 * CREATE TABLE IF NOT EXISTS 'divvy`.`fact_trip` (
550   `trip_id` BIGINT NOT NULL,
551   `start_time` DATETIME NOT NULL,
552   `end_time` DATETIME NOT NULL,
553   `bike_id` INT NOT NULL,
554   `trip_duration` FLOAT NOT NULL,
555   `from_station_id` INT NOT NULL,
556   `to_station_id` INT NOT NULL,
557   `user_type` VARCHAR(45) NOT NULL,
558   `gender` VARCHAR(45) NOT NULL,
559   `age_group` VARCHAR(45) NOT NULL,
560   `weather_date_hour` BIGINT NOT NULL,
561   PRIMARY KEY (`trip_id`))
562 ENGINE = InnoDB;
563
564
565 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_bike_lane` (
566   `lane_id` INT NOT NULL,
567   `lane_type` VARCHAR(45) NOT NULL,
568   `from_street` VARCHAR(60) NOT NULL,
569   `to_street` VARCHAR(60) NOT NULL,
570   `length` FLOAT NOT NULL,
571   `location_id` BIGINT NOT NULL,
572   PRIMARY KEY (`lane_id`))
573 ENGINE = InnoDB;
574
575
576 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_weather` (
577   `weather_date_hour` DATETIME NOT NULL,
578   `date` DATETIME NOT NULL,
579   `temperature` INT NOT NULL,
580   `wind` INT NOT NULL,
581   `precipitation` INT NOT NULL,
582   PRIMARY KEY (`weather_date_hour`))
583 ENGINE = InnoDB;
584
585
586 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_population` (
587   `zip` INT NOT NULL,
588   `0_19_M` INT NOT NULL,
589   `20_29_M` INT NOT NULL,
590   `30_39_M` INT NOT NULL,
591   `40_49_M` INT NOT NULL,
592   `50plus_M` INT NOT NULL,
593   `0_19_F` INT NOT NULL,
594   `20_29_F` INT NOT NULL,
595   `30_39_F` INT NOT NULL,
596   `40_49_F` INT NOT NULL,
597   `50plus_F` INT NOT NULL,
598   PRIMARY KEY (`zip`))
599 ENGINE = InnoDB;
600
601
602 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_location` (
603   `location_id` BIGINT NOT NULL,
604   `location_address` VARCHAR(60) NOT NULL,
605   `latitude` DOUBLE NOT NULL,
606   `longitude` DOUBLE NOT NULL,
607   `lat_long` MULTILINESTRING NOT NULL,
608   `zip` INT NOT NULL,
609   `location_type` VARCHAR(45) NOT NULL,
610   PRIMARY KEY (`location_id`))
611 ENGINE = InnoDB;
612
613
614 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_bike_racks` (
615   `rack_id` INT NOT NULL,
616   `location_id` BIGINT NOT NULL,
617   PRIMARY KEY (`rack_id`))
618 ENGINE = InnoDB;
619
620
621 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_station` (
622   `station_id` INT NOT NULL,
623   `station_name` VARCHAR(45) NOT NULL,
624   `total_docks` INT NOT NULL,
625   `location_id` BIGINT NOT NULL,
626   PRIMARY KEY (`station_id`))
627 ENGINE = InnoDB;
628
629
630 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_traffic` (
631   `traffic_id` INT NOT NULL,
632   `date` DATE NOT NULL,
633   `count_from` INT NOT NULL,
634   `count_to` INT NOT NULL,
635   `direction_from` VARCHAR(45) NOT NULL,
636   `direction_to` VARCHAR(45) NOT NULL,
637   `location_id` BIGINT NOT NULL,
638   PRIMARY KEY (`traffic_id`))
639 ENGINE = InnoDB;
640
641
642 * CREATE TABLE IF NOT EXISTS 'divvy`.`fact_trip` (
643   `trip_id` BIGINT NOT NULL,
644   `start_time` DATETIME NOT NULL,
645   `end_time` DATETIME NOT NULL,
646   `bike_id` INT NOT NULL,
647   `trip_duration` FLOAT NOT NULL,
648   `from_station_id` INT NOT NULL,
649   `to_station_id` INT NOT NULL,
650   `user_type` VARCHAR(45) NOT NULL,
651   `gender` VARCHAR(45) NOT NULL,
652   `age_group` VARCHAR(45) NOT NULL,
653   `weather_date_hour` BIGINT NOT NULL,
654   PRIMARY KEY (`trip_id`))
655 ENGINE = InnoDB;
656
657
658 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_bike_lane` (
659   `lane_id` INT NOT NULL,
660   `lane_type` VARCHAR(45) NOT NULL,
661   `from_street` VARCHAR(60) NOT NULL,
662   `to_street` VARCHAR(60) NOT NULL,
663   `length` FLOAT NOT NULL,
664   `location_id` BIGINT NOT NULL,
665   PRIMARY KEY (`lane_id`))
666 ENGINE = InnoDB;
667
668
669 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_weather` (
670   `weather_date_hour` DATETIME NOT NULL,
671   `date` DATETIME NOT NULL,
672   `temperature` INT NOT NULL,
673   `wind` INT NOT NULL,
674   `precipitation` INT NOT NULL,
675   PRIMARY KEY (`weather_date_hour`))
676 ENGINE = InnoDB;
677
678
679 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_population` (
680   `zip` INT NOT NULL,
681   `0_19_M` INT NOT NULL,
682   `20_29_M` INT NOT NULL,
683   `30_39_M` INT NOT NULL,
684   `40_49_M` INT NOT NULL,
685   `50plus_M` INT NOT NULL,
686   `0_19_F` INT NOT NULL,
687   `20_29_F` INT NOT NULL,
688   `30_39_F` INT NOT NULL,
689   `40_49_F` INT NOT NULL,
690   `50plus_F` INT NOT NULL,
691   PRIMARY KEY (`zip`))
692 ENGINE = InnoDB;
693
694
695 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_location` (
696   `location_id` BIGINT NOT NULL,
697   `location_address` VARCHAR(60) NOT NULL,
698   `latitude` DOUBLE NOT NULL,
699   `longitude` DOUBLE NOT NULL,
700   `lat_long` MULTILINESTRING NOT NULL,
701   `zip` INT NOT NULL,
702   `location_type` VARCHAR(45) NOT NULL,
703   PRIMARY KEY (`location_id`))
704 ENGINE = InnoDB;
705
706
707 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_bike_racks` (
708   `rack_id` INT NOT NULL,
709   `location_id` BIGINT NOT NULL,
710   PRIMARY KEY (`rack_id`))
711 ENGINE = InnoDB;
712
713
714 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_station` (
715   `station_id` INT NOT NULL,
716   `station_name` VARCHAR(45) NOT NULL,
717   `total_docks` INT NOT NULL,
718   `location_id` BIGINT NOT NULL,
719   PRIMARY KEY (`station_id`))
720 ENGINE = InnoDB;
721
722
723 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_traffic` (
724   `traffic_id` INT NOT NULL,
725   `date` DATE NOT NULL,
726   `count_from` INT NOT NULL,
727   `count_to` INT NOT NULL,
728   `direction_from` VARCHAR(45) NOT NULL,
729   `direction_to` VARCHAR(45) NOT NULL,
730   `location_id` BIGINT NOT NULL,
731   PRIMARY KEY (`traffic_id`))
732 ENGINE = InnoDB;
733
734
735 * CREATE TABLE IF NOT EXISTS 'divvy`.`fact_trip` (
736   `trip_id` BIGINT NOT NULL,
737   `start_time` DATETIME NOT NULL,
738   `end_time` DATETIME NOT NULL,
739   `bike_id` INT NOT NULL,
740   `trip_duration` FLOAT NOT NULL,
741   `from_station_id` INT NOT NULL,
742   `to_station_id` INT NOT NULL,
743   `user_type` VARCHAR(45) NOT NULL,
744   `gender` VARCHAR(45) NOT NULL,
745   `age_group` VARCHAR(45) NOT NULL,
746   `weather_date_hour` BIGINT NOT NULL,
747   PRIMARY KEY (`trip_id`))
748 ENGINE = InnoDB;
749
750
751 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_bike_lane` (
752   `lane_id` INT NOT NULL,
753   `lane_type` VARCHAR(45) NOT NULL,
754   `from_street` VARCHAR(60) NOT NULL,
755   `to_street` VARCHAR(60) NOT NULL,
756   `length` FLOAT NOT NULL,
757   `location_id` BIGINT NOT NULL,
758   PRIMARY KEY (`lane_id`))
759 ENGINE = InnoDB;
760
761
762 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_weather` (
763   `weather_date_hour` DATETIME NOT NULL,
764   `date` DATETIME NOT NULL,
765   `temperature` INT NOT NULL,
766   `wind` INT NOT NULL,
767   `precipitation` INT NOT NULL,
768   PRIMARY KEY (`weather_date_hour`))
769 ENGINE = InnoDB;
770
771
772 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_population` (
773   `zip` INT NOT NULL,
774   `0_19_M` INT NOT NULL,
775   `20_29_M` INT NOT NULL,
776   `30_39_M` INT NOT NULL,
777   `40_49_M` INT NOT NULL,
778   `50plus_M` INT NOT NULL,
779   `0_19_F` INT NOT NULL,
780   `20_29_F` INT NOT NULL,
781   `30_39_F` INT NOT NULL,
782   `40_49_F` INT NOT NULL,
783   `50plus_F` INT NOT NULL,
784   PRIMARY KEY (`zip`))
785 ENGINE = InnoDB;
786
787
788 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_location` (
789   `location_id` BIGINT NOT NULL,
790   `location_address` VARCHAR(60) NOT NULL,
791   `latitude` DOUBLE NOT NULL,
792   `longitude` DOUBLE NOT NULL,
793   `lat_long` MULTILINESTRING NOT NULL,
794   `zip` INT NOT NULL,
795   `location_type` VARCHAR(45) NOT NULL,
796   PRIMARY KEY (`location_id`))
797 ENGINE = InnoDB;
798
799
800 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_bike_racks` (
801   `rack_id` INT NOT NULL,
802   `location_id` BIGINT NOT NULL,
803   PRIMARY KEY (`rack_id`))
804 ENGINE = InnoDB;
805
806
807 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_station` (
808   `station_id` INT NOT NULL,
809   `station_name` VARCHAR(45) NOT NULL,
810   `total_docks` INT NOT NULL,
811   `location_id` BIGINT NOT NULL,
812   PRIMARY KEY (`station_id`))
813 ENGINE = InnoDB;
814
815
816 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_traffic` (
817   `traffic_id` INT NOT NULL,
818   `date` DATE NOT NULL,
819   `count_from` INT NOT NULL,
820   `count_to` INT NOT NULL,
821   `direction_from` VARCHAR(45) NOT NULL,
822   `direction_to` VARCHAR(45) NOT NULL,
823   `location_id` BIGINT NOT NULL,
824   PRIMARY KEY (`traffic_id`))
825 ENGINE = InnoDB;
826
827
828 * CREATE TABLE IF NOT EXISTS 'divvy`.`fact_trip` (
829   `trip_id` BIGINT NOT NULL,
830   `start_time` DATETIME NOT NULL,
831   `end_time` DATETIME NOT NULL,
832   `bike_id` INT NOT NULL,
833   `trip_duration` FLOAT NOT NULL,
834   `from_station_id` INT NOT NULL,
835   `to_station_id` INT NOT NULL,
836   `user_type` VARCHAR(45) NOT NULL,
837   `gender` VARCHAR(45) NOT NULL,
838   `age_group` VARCHAR(45) NOT NULL,
839   `weather_date_hour` BIGINT NOT NULL,
840   PRIMARY KEY (`trip_id`))
841 ENGINE = InnoDB;
842
843
844 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_bike_lane` (
845   `lane_id` INT NOT NULL,
846   `lane_type` VARCHAR(45) NOT NULL,
847   `from_street` VARCHAR(60) NOT NULL,
848   `to_street` VARCHAR(60) NOT NULL,
849   `length` FLOAT NOT NULL,
850   `location_id` BIGINT NOT NULL,
851   PRIMARY KEY (`lane_id`))
852 ENGINE = InnoDB;
853
854
855 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_weather` (
856   `weather_date_hour` DATETIME NOT NULL,
857   `date` DATETIME NOT NULL,
858   `temperature` INT NOT NULL,
859   `wind` INT NOT NULL,
860   `precipitation` INT NOT NULL,
861   PRIMARY KEY (`weather_date_hour`))
862 ENGINE = InnoDB;
863
864
865 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_population` (
866   `zip` INT NOT NULL,
867   `0_19_M` INT NOT NULL,
868   `20_29_M` INT NOT NULL,
869   `30_39_M` INT NOT NULL,
870   `40_49_M` INT NOT NULL,
871   `50plus_M` INT NOT NULL,
872   `0_19_F` INT NOT NULL,
873   `20_29_F` INT NOT NULL,
874   `30_39_F` INT NOT NULL,
875   `40_49_F` INT NOT NULL,
876   `50plus_F` INT NOT NULL,
877   PRIMARY KEY (`zip`))
878 ENGINE = InnoDB;
879
880
881 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_location` (
882   `location_id` BIGINT NOT NULL,
883   `location_address` VARCHAR(60) NOT NULL,
884   `latitude` DOUBLE NOT NULL,
885   `longitude` DOUBLE NOT NULL,
886   `lat_long` MULTILINESTRING NOT NULL,
887   `zip` INT NOT NULL,
888   `location_type` VARCHAR(45) NOT NULL,
889   PRIMARY KEY (`location_id`))
890 ENGINE = InnoDB;
891
892
893 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_bike_racks` (
894   `rack_id` INT NOT NULL,
895   `location_id` BIGINT NOT NULL,
896   PRIMARY KEY (`rack_id`))
897 ENGINE = InnoDB;
898
899
900 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_station` (
901   `station_id` INT NOT NULL,
902   `station_name` VARCHAR(45) NOT NULL,
903   `total_docks` INT NOT NULL,
904   `location_id` BIGINT NOT NULL,
905   PRIMARY KEY (`station_id`))
906 ENGINE = InnoDB;
907
908
909 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_traffic` (
910   `traffic_id` INT NOT NULL,
911   `date` DATE NOT NULL,
912   `count_from` INT NOT NULL,
913   `count_to` INT NOT NULL,
914   `direction_from` VARCHAR(45) NOT NULL,
915   `direction_to` VARCHAR(45) NOT NULL,
916   `location_id` BIGINT NOT NULL,
917   PRIMARY KEY (`traffic_id`))
918 ENGINE = InnoDB;
919
920
921 * CREATE TABLE IF NOT EXISTS 'divvy`.`fact_trip` (
922   `trip_id` BIGINT NOT NULL,
923   `start_time` DATETIME NOT NULL,
924   `end_time` DATETIME NOT NULL,
925   `bike_id` INT NOT NULL,
926   `trip_duration` FLOAT NOT NULL,
927   `from_station_id` INT NOT NULL,
928   `to_station_id` INT NOT NULL,
929   `user_type` VARCHAR(45) NOT NULL,
930   `gender` VARCHAR(45) NOT NULL,
931   `age_group` VARCHAR(45) NOT NULL,
932   `weather_date_hour` BIGINT NOT NULL,
933   PRIMARY KEY (`trip_id`))
934 ENGINE = InnoDB;
935
936
937 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_bike_lane` (
938   `lane_id` INT NOT NULL,
939   `lane_type` VARCHAR(45) NOT NULL,
940   `from_street` VARCHAR(60) NOT NULL,
941   `to_street` VARCHAR(60) NOT NULL,
942   `length` FLOAT NOT NULL,
943   `location_id` BIGINT NOT NULL,
944   PRIMARY KEY (`lane_id`))
945 ENGINE = InnoDB;
946
947
948 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_weather` (
949   `weather_date_hour` DATETIME NOT NULL,
950   `date` DATETIME NOT NULL,
951   `temperature` INT NOT NULL,
952   `wind` INT NOT NULL,
953   `precipitation` INT NOT NULL,
954   PRIMARY KEY (`weather_date_hour`))
955 ENGINE = InnoDB;
956
957
958 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_population` (
959   `zip` INT NOT NULL,
960   `0_19_M` INT NOT NULL,
961   `20_29_M` INT NOT NULL,
962   `30_39_M` INT NOT NULL,
963   `40_49_M` INT NOT NULL,
964   `50plus_M` INT NOT NULL,
965   `0_19_F` INT NOT NULL,
966   `20_29_F` INT NOT NULL,
967   `30_39_F` INT NOT NULL,
968   `40_49_F` INT NOT NULL,
969   `50plus_F` INT NOT NULL,
970   PRIMARY KEY (`zip`))
971 ENGINE = InnoDB;
972
973
974 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_location` (
975   `location_id` BIGINT NOT NULL,
976   `location_address` VARCHAR(60) NOT NULL,
977   `latitude` DOUBLE NOT NULL,
978   `longitude` DOUBLE NOT NULL,
979   `lat_long` MULTILINESTRING NOT NULL,
980   `zip` INT NOT NULL,
981   `location_type` VARCHAR(45) NOT NULL,
982   PRIMARY KEY (`location_id`))
983 ENGINE = InnoDB;
984
985
986 * CREATE TABLE IF NOT EXISTS 'divvy`.`dim_bike_racks` (
987   `rack_id` INT NOT NULL,
988   `location_id` BIGINT NOT NULL,
989   PRIMARY KEY (`rack_id`))
99
```



1

Data



Data from various sources



2

ETL Process



3

Data Warehouse



Google Cloud Platform

4

Data Analysis



5

Data Visualization



6

Findings and Recommendations



Data extraction, Cleaning, Normalization



- Create and load database
- Produce queries to support project's analysis purpose

Number of trips by hour by weekday and weekend.

```
SELECT
CASE WHEN dayname(start_time) IN ("Saturday", "Sunday") THEN "Weekend" ELSE "Weekday" END AS DateType,
COUNT(trip_id) AS NoOfTrips
FROM fact_trip
GROUP BY TimeOfDay, DateType
ORDER BY TimeOfDay DESC;
```

```
#. Number of TripIn per zip
SELECT
dl.zip,
COUNT(ft.to_station_id) AS TripIn
FROM
fact_trip ft
INNER JOIN dim_station ds
ON ds.station_id = ft.to_station_id
LEFT JOIN dim_location dl
ON dl.location_id = ds.location_id
GROUP BY zip
ORDER BY TripIn DESC;
```

- Clean all dimensional tables to import to mySQL
- Analyze descriptive data: customer profiling, zip, stations
- Build the scoring system for research objectives' purpose: add more stations and bikes.

community_id	community_area	Male	Female	0-19	20-29	30-39	40-49	50-59	60-69	70-79	80+	White	Black	Hispanic	Asian	Other	Hispanic	Asian	Other	
1	Rogers Park	27754	27239	12314	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	
2	West Ridge	35522	36420	20005	16111	16111	16111	16111	16111	16111	16111	16111	16111	16111	16111	16111	16111	16111	16111	
3	Uptown	29512	26550	8106	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	
4	Lincoln Square	19309	20184	7168	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	
5	North Center	15557	16310	6289	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	
6	Lake View	47000	47368	9334	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	
7	Lincoln Park	30430	33686	10135	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	
8	Near North Side	37337	43147	14247	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	
9	Edison Park	5367	5820	2556	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	
10	Norwood Park	17522	19501	8237	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	
11	Jefferson Park	12293	13155	5707	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	
12	Forest Glen	8534	9574	4850	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	
13	North Park	8649	9282	4429	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	
14	Albany Park	26407	25135	14526	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	
15	Portage Park	31337	32787	16227	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	
16	Irving Park	26674	26885	13426	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	
17	Dunning	20504	21428	9403	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	
18	Montclare	6592	6834	3646	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	
19	Belmont Crown	39609	39134	25998	12899	12366	10232	17248	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111
20	Hermosa	12566	12444	8416	4237	3932	3252	5173	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111
21	Arden/Alex	20119	19143	10710	7755	7235	5082	8480	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111
22	Logan Square	36805	35886	16711	17952	16386	8912	12830	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111
23	Humboldt Park	27273	29050	19317	9315	8177	6985	12529	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111
24	West Town	42058	40178	13909	22925	21992	9493	13917	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111
25	Austin	45189	53325	30332	19900	12315	12761	29206	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111	11111



- Import, clean, and extract real-time station data from Divvy to get the zip code for each station.



- Get the zipcode using longitude and latitude for dim_location table
- Estimated the distance between trips
- Stack the distance data to produce an adaptable format for tableau visualization purpose
- Conduct some correlation between trips and other factors: weekday, bike racks, weather...

```
1 install.packages("rgeos")
2 library(rgeos)
3
4
5 dataPath <- "C:/Users/amy/family/ChicagoData/Engineering_Pipeline/FinalProject/Processing/"
6
7
8 #Rack1
9 rack1_zip <- read.csv(paste(dataPath, "rack1.csv", sep="/"))
10 rack1_zip <- rgeos(longitude=rack1$longitude, latitude=rack1$latitude,
11 output="frame", item="zip")
12 write.table(rack1_zip, file = paste(dataPath, "rack1_zip.csv", sep="/"), row.names = F)
13
14 #Rack2
15 rack2_zip <- read.csv(paste(dataPath, "rack2.csv", sep="/"))
16 rack2_zip <- rgeos(longitude=rack2$longitude, latitude=rack2$latitude,
17 output="frame", item="zip")
18 write.table(rack2_zip, file = paste(dataPath, "rack2_zip.csv", sep="/"), row.names = F)
19
20 #Rack3
21 rack3_zip <- read.csv(paste(dataPath, "rack3.csv", sep="/"))
22 rack3_zip <- rgeos(longitude=rack3$longitude, latitude=rack3$latitude,
23 output="frame", item="zip")
24 write.table(rack3_zip, file = paste(dataPath, "rack3_zip.csv", sep="/"), row.names = F)
25
26 # Average distance per station
27 # [distance]
28 # [distance]
29 # [distance]
30 # [distance]
31 # [distance]
32 # [distance]
33 # [distance]
34 # [distance]
35 # [distance]
36 # [distance]
37 # [distance]
38 # [distance]
39 # [distance]
40 # [distance]
41 # [distance]
42 # [distance]
43 # [distance]
44 # [distance]
45 # [distance]
46 # [distance]
47 # [distance]
48 # [distance]
49 # [distance]
50 # [distance]
51 # [distance]
52 # [distance]
53 # [distance]
54 # [distance]
55 # [distance]
56 # [distance]
57 # [distance]
58 # [distance]
59 # [distance]
60 # [distance]
61 # [distance]
62 # [distance]
63 # [distance]
64 # [distance]
65 # [distance]
66 # [distance]
67 # [distance]
68 # [distance]
69 # [distance]
70 # [distance]
```



- Construct fact_trip table to import to my SQL:
 - Calculate the age group of Divvy users
 - Add in new column as a foreign key using in mySQL.

```
10 [10]: def get_group_age:
11     def get_group_age:
12         return '0-19'
13     else:
14         return '20-29'
15     else:
16         return '30-39'
17     else:
18         return '40-49'
19     else:
20         return '50-59'
21     else:
22         return '60-69'
23     else:
24         return '70-79'
25     else:
26         return '80+'
27     else:
28         return 'Other'
29     else:
30         return 'Other'
31     else:
32         return 'Other'
33     else:
34         return 'Other'
35     else:
36         return 'Other'
37     else:
38         return 'Other'
39     else:
40         return 'Other'
41     else:
42         return 'Other'
43     else:
44         return 'Other'
45     else:
46         return 'Other'
47     else:
48         return 'Other'
49     else:
50         return 'Other'
51     else:
52         return 'Other'
53     else:
54         return 'Other'
55     else:
56         return 'Other'
57     else:
58         return 'Other'
59     else:
60         return 'Other'
61     else:
62         return 'Other'
63     else:
64         return 'Other'
65     else:
66         return 'Other'
67     else:
68         return 'Other'
69     else:
70         return 'Other'
71     else:
72         return 'Other'
73     else:
74         return 'Other'
75     else:
76         return 'Other'
77     else:
78         return 'Other'
79     else:
80         return 'Other'
81     else:
82         return 'Other'
83     else:
84         return 'Other'
85     else:
86         return 'Other'
87     else:
88         return 'Other'
89     else:
90         return 'Other'
91     else:
92         return 'Other'
93     else:
94         return 'Other'
95     else:
96         return 'Other'
97     else:
98         return 'Other'
99     else:
100        return 'Other'
```





Net influx per station and hour

```

5 • SELECT
6     TripFrom.station_id,
7     TripFrom.stationName AS stationName,
8     TripFrom.TimeOfDay AS tripTime,
9     TripFrom.tripFrom,
10    TripTo.tripTo,
11    (TripFrom.tripFrom - TripTo.tripTo) AS NetTrip
12 FROM
13     (SELECT
14         ds.station_id,
15         ds.station_name AS stationName,
16         ds.total_docks AS totalDocks,
17         HOUR(ft.start_time) AS TimeOfDay,
18         COUNT(ft.from_station_id) as tripFrom
19     FROM
20         fact_trip ft
21         INNER JOIN
22         dim_station ds ON ds.station_id = ft.from_station_id
23     GROUP BY
24         ds.station_id, TimeOfDay
25     ORDER BY
26         ds.station_id,TimeOfDay ASC) AS TripFrom
27     INNER JOIN
28     (SELECT
29         ds.station_id,
30         ds.station_name AS stationName,
31         ds.total_docks AS totalDocks,
32         HOUR(ft.end_time) AS TimeOfDay,
33         COUNT(ft.to_station_id) as tripTo
34     FROM
35         fact_trip ft
36         INNER JOIN
37         dim_station ds ON ds.station_id = ft.to_station_id
38     GROUP BY
39         ds.station_id, TimeOfDay
40     ORDER BY ds.station_id,TimeOfDay ASC) AS TripTo ON TripFrom.station_id = TripTo.station_id
41 WHERE TripFrom.TimeOfDay = TripTo.TimeOfDay;
    
```

Average distance travelled per station and zip code

```

91 • SELECT
92
93     FrS.station_id,
94     FrS.trip_id,
95     FrS.latitude AS lat1,
96     FrS.longitude AS long1,
97     TrS.station_id,
98     TrS.trip_id,
99     TrS.latitude AS lat2,
100    TrS.longitude AS long2
101 FROM
102     (SELECT
103         ds.station_id,
104         ft.trip_id,
105         dl.latitude,
106         dl.longitude
107     FROM
108         dim_location dl
109         INNER JOIN
110         dim_station ds ON dl.location_id=ds.location_id
111         INNER JOIN
112         fact_trip ft ON ds.station_id=ft.from_station_id) AS FrS
113     INNER JOIN
114     (SELECT
115         ds.station_id,
116         ft.trip_id,
117         dl.latitude,
118         dl.longitude
119     FROM
120         dim_location dl
121         INNER JOIN
122         dim_station ds ON dl.location_id=ds.location_id
123         INNER JOIN
124         fact_trip ft ON ds.station_id=ft.to_station_id) AS TrS ON FrS.trip_id=TrS.trip_id
125 WHERE
126     FrS.station_id != TrS.station_id;
    
```

Data Analysis and Visualization



Customer Profiling

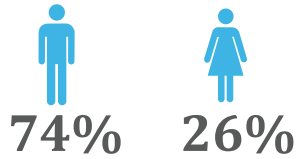


Users Type

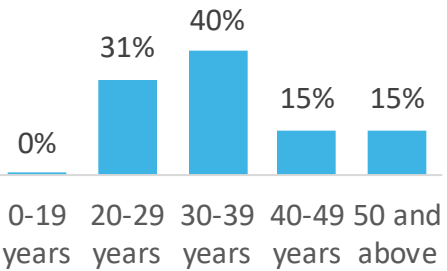


- Subscriber
- Non-subscriber

Gender



Age Group



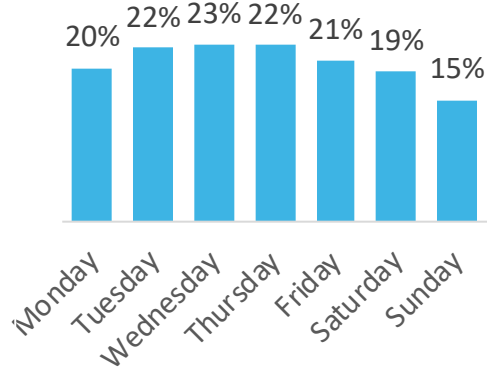
Average Distance Travel



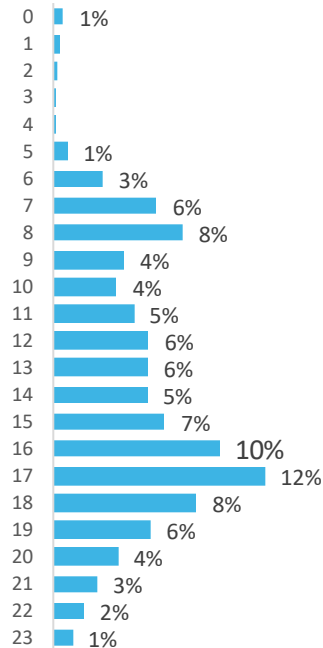
Average Trip Duration



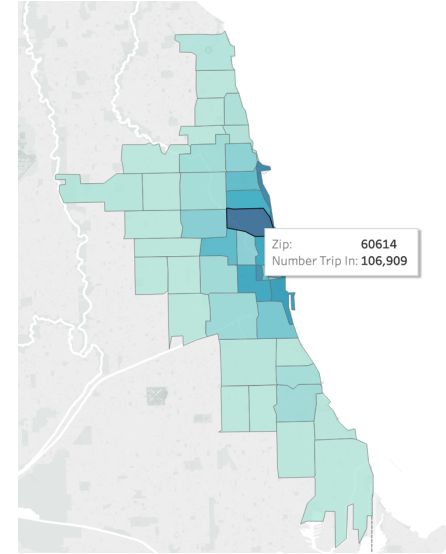
Trip by day



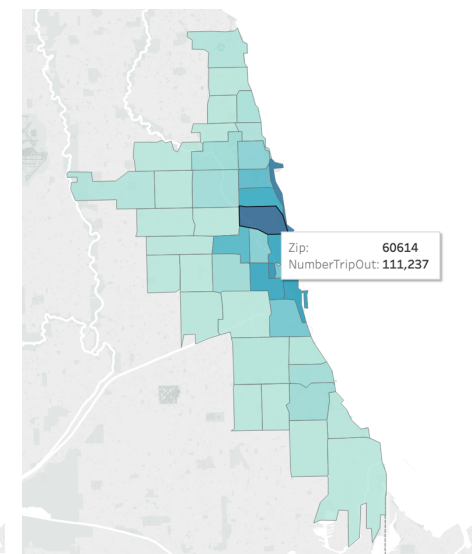
Trip by hour



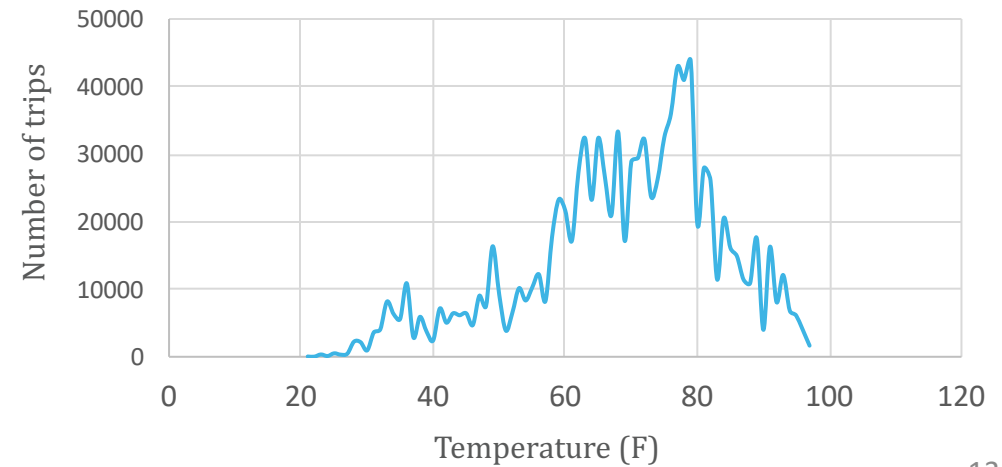
Trip-in by area



Trip-out by area



Trip by weather



Findings by zip code



Zipcode Analysis	Demographic													Traffic Vehicle Volume	BIKE Number of Bike racks	Number of stations	Dwvy Stations		Avg of Avg Distance of stations from other stations (miles)	Dwvy Trips				Subscriber %
	Population		Gender		Age												Total # of docks	Avg # of docks		Trips Out	Trips In	Net	% of total trips	
	Total	Male%	Female%	0_19	20_29	30_39	40_49	50plus	0_19	20_29	30_39	40_49	50plus											
60605	24672	48.5%	51.5%	0.0678	0.138	0.304	0.0585	0.173	0.0719	0.1451	0.1789	0.0603	0.1206	8,300	358	13	550	28.9	4.75	68,902	65,243	(3,659)	6.3%	60.72%
60601	1115	49.4%	50.6%	0.0699	0.1702	0.093	0.0595	0.103	0.0716	0.174	0.0953	0.0571	0.113	23,800	1	11	364	11.2	4.58	68,506	63,594	(4,912)	5.2%	72.4%
60609	94911	50.2%	49.8%	0.1763	0.078	0.046	0.0623	0.11	0.173	0.0789	0.0743	0.062	0.11	8,700	28	34	112	6.43	2,953	2,998	45	0.2%	67.6%	
60649	46654	43.8%	56.2%	0.1172	0.0632	0.0563	0.0539	0.1434	0.1503	0.0785	0.0723	0.0769	0.184	18,300	31	14	170	12.1	3.73	1,522	1,462	(60)	4.6%	62.0%
60614	66623	47.6%	52.4%	0.0761	0.1642	0.0696	0.0527	0.0951	0.0927	0.161	0.0975	0.0901	0.1943	21,700	195	34	638	16.8	4.62	106,309	111,237	4,928	10.2%	73.3%
60608	67143	52.4%	47.6%	0.1639	0.0394	0.0693	0.0691	0.1071	0.1053	0.0969	0.0939	0.0598	0.104	19,000	339	27	370	13.7	5.03	14,409	15,289	739	14.4%	67.6%
60622	52583	51.1%	48.9%	0.0882	0.1414	0.1354	0.0532	0.0668	0.0845	0.153	0.1285	0.0566	0.0831	34,400	354	25	445	16.6	4.57	45,114	46,260	1,146	4.3%	66.0%
60606	2314	49.2%	50.8%	0.0843	0.1517	0.1108	0.0519	0.0938	0.0869	0.164	0.1141	0.0536	0.0969	10,200	62	6	170	26.3	4.40	37,608	34,087	(3,521)	3.3%	90.9%
60607	23932	49.2%	50.8%	0.0887	0.1405	0.1131	0.0522	0.0938	0.0907	0.1535	0.1178	0.054	0.0936	29,300	82	26	453	17.4	4.40	61,365	61,617	252	5.6%	91.1%
60642	34495	51.1%	48.9%	0.0958	0.1424	0.1361	0.059	0.0674	0.0821	0.164	0.1303	0.0565	0.084	11,000	3	12	217	16.1	4.37	24,416	24,523	107	2.3%	90.8%
60610	37730	48.4%	51.6%	0.0397	0.1339	0.0975	0.0533	0.1458	0.0459	0.1547	0.1067	0.0675	0.1882	22,300	122	18	403	22.4	4.52	61,694	62,402	708	5.6%	81.9%
60611	28722	48.4%	51.6%	0.0394	0.1337	0.0975	0.0533	0.146	0.0458	0.1545	0.1068	0.0676	0.1881	16,100	149	8	372	23.3	4.67	69,207	65,133	4,074	8.7%	64.4%
60654	4680	46.7%	53.3%	0.0421	0.1342	0.0941	0.0536	0.1426	0.0476	0.1534	0.1072	0.0673	0.1876	23,000	115	14	356	25.4	4.40	69,187	65,381	(3,806)	6.3%	69.9%
60604	575	49.4%	50.6%	0.0636	0.1704	0.0939	0.0504	0.1096	0.0713	0.1739	0.0957	0.0522	0.113	23,000	3	85	28.3	4.58	5,680	5,230	(450)	14.6%	60.4%	
60603	497	49.3%	50.7%	0.0704	0.169	0.0926	0.0503	0.1107	0.0724	0.1731	0.0946	0.0523	0.112	13,700	38	5	195	27.0	4.58	27,542	24,721	(2,821)	24.7%	60.4%
60616	48437	49.3%	50.7%	0.0958	0.0594	0.0776	0.0627	0.1499	0.1029	0.1036	0.0939	0.0669	0.1026	6,700	87	23	447	15.4	5.35	32,058	32,470	412	3.0%	91.9%
60602	1210	49.3%	50.7%	0.0702	0.1702	0.0926	0.0504	0.1099	0.0719	0.1744	0.0938	0.0521	0.112	13,700	3	77	25.7	4.58	17,152	17,130	(22)	16.2%	60.4%	
60661	7738	49.2%	50.8%	0.0682	0.1464	0.1157	0.0524	0.0987	0.0913	0.154	0.1198	0.0542	0.0939	15,800	102	12	364	29.5	4.34	78,847	76,413	(2,434)	7.3%	90.5%
60637	4959	44.9%	55.1%	0.1343	0.0776	0.081	0.0545	0.121	0.0893	0.094	0.0748	0.0672	0.1494	20,300	12	17	264	15.5	8.35	12,684	12,423	(261)	11.9%	61.1%
60608	1210	49.3%	50.7%	0.0702	0.1702	0.0926	0.0504	0.1099	0.0719	0.1744	0.0938	0.0521	0.112	13,700	3	77	25.7	4.58	17,152	17,130	(22)	16.2%	60.4%	
60662	7738	49.2%	50.8%	0.0682	0.1464	0.1157	0.0524	0.0987	0.0913	0.154	0.1198	0.0542	0.0939	15,800	102	12	364	29.5	4.34	78,847	76,413	(2,434)	7.3%	90.5%
60637	4959	44.9%	55.1%	0.1343	0.0776	0.081	0.0545	0.121	0.0893	0.094	0.0748	0.0672	0.1494	20,300	12	17	264	15.5	8.35	12,684	12,423	(261)	11.9%	61.1%
60657	66601	49.7%	50.3%	0.0525	0.1862	0.1163	0.0589	0.0875	0.0532	0.188	0.1163	0.0585	0.0885	12,300	201	20	371	19.6	5.28	54,456	57,328	2,872	5.2%	80.5%
60647	67297	50.4%	49.6%	0.1877	0.1217	0.112	0.0616	0.0911	0.112	0.1193	0.108	0.0606	0.0997	10,600	204	24	376	15.7	5.43	26,516	27,351	835	2.5%	62.6%
60612	23478	48.4%	51.6%	0.1148	0.1631	0.0933	0.0559	0.0979	0.1249	0.127	0.1041	0.0599	0.1054	29,500	76	19	335	17.8	4.88	15,200	12,852	(2,348)	12.2%	93.3%
60615	46068	44.9%	55.1%	0.0931	0.0974	0.0893	0.0593	0.1236	0.1193	0.1193	0.0809	0.0658	0.1448	10,600	64	11	177	16.1	7.54	10,623	10,307	(316)	10.3%	76.7%
60618	32093	50.2%	49.8%	0.1238	0.0956	0.1071	0.0701	0.1107	0.1223	0.096	0.1044	0.0697	0.1077	18,700	193	21	308	14.7	5.35	10,973	11,778	805	10.7%	65.2%
60613	46295	50.4%	49.6%	0.0682	0.164	0.1156	0.0628	0.109	0.0825	0.1339	0.0612	0.0992	0.1160	76	23	435	16.5	5.30	43,437	44,266	829	4.4%	73.5%	
60653	29932	43.9%	56.1%	0.1143	0.0672	0.0641	0.0549	0.1375	0.1157	0.0971	0.0846	0.0725	0.1121	19,900	59	9	123	13.7	6.82	3,101	2,777	(324)	0.2%	73.3%
60625	78854	50.0%	50.0%	0.119	0.0859	0.1004	0.0676	0.1173	0.1173	0.0958	0.1007	0.0677	0.1178	23,800	288	19	285	15.0	6.85	11,857	12,247	390	1.4%	87.8%
60640	65796	51.8%	48.2%	0.0779	0.1019	0.1029	0.0783	0.1359	0.0721	0.1031	0.1032	0.0729	0.1001	16,700	205	14	282	20.1	8.71	22,815	23,090	275	2.6%	80.4%
60619	63830	43.8%	56.2%	0.1177	0.0921	0.0997	0.0572	0.1056	0.1056	0.1062	0.074	0.0963	0.1033	17,800	31	17	193	19.8	3.89	838	894	46	0.6%	71.6%
60634	74302	49.1%	50.9%	0.1233	0.0697	0.0711	0.0683	0.1089	0.0738	0.1073	0.0687	0.0708	0.105	21,800	112	1	19	19.0	5.33	3,164	3,693	529	0.3%	73.8%
60626	50144	50.4%	49.6%	0.1198	0.1078	0.0991	0.0715	0.1165	0.1198	0.0971	0.0703	0.0995	0.1145	7,400	145	15	233	15.5	8.32	7,235	7,295	60	0.9%	64.5%
60621	35319	44.9%	55.1%	0.1517	0.0654	0.0625	0.0543	0.1259	0.107	0.0906	0.0623	0.067	0.1053	15,000	28	12	130	19.8	8.26	395	381	(14)	0.4%	68.3%
60624	38109	46.6%	53.4%	0.161	0.0633	0.0607	0.0589	0.1183	0.1444	0.0732	0.0634	0.0674	0.1334	11,800	35	8	88	11.0	5.71	339	320	(19)	0.3%	68.1%
60623	32112	53.7%	46.3%	0.1648	0.0991	0.0891	0.0674	0.1001	0.1038	0.0834	0.0785	0.0579	0.0998	13,700	109	6	66	11.0	5.88	560	588	28	0.6%	62.0%
60645	45200	43.8%	56.2%	0.1341	0.0763	0.076	0.0680	0.1106	0.1077	0.1177	0.0744	0.0696	0.1433	11,600	55	7	85	12.1	8.39	112	115	3	0.2%	77.2%
60659	38109	49.1%	50.9%	0.1325	0.0719	0.0704	0.0681	0.1007	0.1172	0.0973	0.0929	0.0685	0.1065	44,000	19	4	44	11.0	7.79	601	687	86	0.6%	75.5%
60660	42757	51.2%	48.8%	0.0844	0.1033	0.0953	0.0791	0.1496	0.0906	0.0962	0.0922	0.0753	0.1424	35,200	42	5	19	18.2	7.57	5,126	5,686	471	0.5%	73.7%
60641	7868	48.4%	51.6%	0.1068	0.1097	0.0923	0.0777	0.1062	0.0895	0.1098	0.0798	0.0938	0.1039	8,100	81	6	78	13.0	7.21	1,174	1,110	(63)	0.1%	77.2%
60630	54089	49.2%	50.8%	0.1221	0.0795	0.0768	0.0715	0.1499	0.1259	0.1033	0.0799	0.0741	0.1063	12,300	27	2	28	15.0	7.28	230	239	9	0.02%	72.5%
60651	64273	47.3%	52.7%	0.1652	0.0734	0.0646	0.0588	0.1201	0.1224	0.0816	0.0717	0.0667	0.1348	8,300	31	4	44	11.0	6.25	804	795	(9)	0.08%	70.0%
60644	48652	45.3%	54.7%	0.1412	0.0647	0.0572	0.0594	0.138																

Score based approach



Current station locations (Before expansion plan)

Where are the stations?

- CTA, Metra stations
- employment centers, shopping districts, medical centers, schools
- other popular destinations.

How were the locations chosen?

- population density
- business permits
- other stations in the surrounding network.

Our scoring methodology

- When Divvy first launched, it focused more on the popular destinations (tourist attraction areas, shopping centers, offices etc.)
- The expansion plan is focused more on expanding to the areas where there are currently no Divvy stations
- Priority = underserved communities (in terms of number of Divvy stations).
- Score based system for the allocation of the stations and the bikes taking into consideration the below factors. New station allocation determined based on overall score (i.e. higher score = more stations)

Category	Score Description		Weight	Comments
Divvy Stations (existing)	less number of stations = more points	↓ ↑	20%	More weight assigned to zip codes with no stations. Points deducted to zip codes with stations
Trips (Trips Out)	more number of trips = more points	↑ ↑	10%	-
Net (Trip From - Trip To)	lower Net value = more points	↓ ↑	5%	Points only added to zip codes with a negative net value
Subscriber%	higher % of subscribers = more points	↑ ↑	15%	-
Population Total	higher population = more points	↑ ↑	15%	-
Male%	higher male % = more points	↑ ↑	5%	-
20_39 Age Group	higher % of 20_39 age group = more points	↑ ↑	10%	-
Average Distance to other stations	higher avg distance to other stations = more points	↑ ↑	10%	-
Traffic	higher vehicle volume = more points	↑ ↑	5%	-
Bike racks	more number of bike racks (bike friendliness score) = more points	↑ ↑	5%	-

Scores by zip code



Scoring by zip calculation

Weight	Demographic										Traffic		Bike Racks		Divvy Stations			Divvy Trips			Total Points	New stations	New Bikes
	Population		Gender		20_39		Vehicle Volume		Number of Bike racks		Number of stations		Avg of Avg Distance of station from other stations (miles)		Trips (Trips Out)		Avg (Trip From - Trip To)		Subscriber %				
	Total	Male%	Total	Male%	1000	500	1000	500	5%	20%	1000	500	10%	5%	10%	5%	15%	5%	15%	15%			
60605	24672	48.5%	0.5034	8,300	356	19	4.75	68,902	(3,859)	60.7%	13.7	8.7	23.9	3.9	31.9	(31.25)	15.4	65.0	84.3	24.4	240.0	5	308
60601	11115	49.4%	0.5229	23,800	1	11	4.59	68,506	(4,912)	72.4%	6.2	8.8	25.0	11.1	0.1	(18.09)	4.3	103.2	64.7	103.2	255.0	5	328
60609	64911	50.2%	0.3066	8,100	75	28	6.49	2,553	(45)	87.5%	36.1	3.0	14.4	3.8	6.7	(46.05)	211	2.4	-	35.2	82.6	2	106
60649	46654	43.8%	0.2693	18,300	31	14	9.79	1,522	(60)	62.0%	26.0	7.8	12.6	8.6	2.8	(23.03)	318	14	14	25.0	94.3	2	121
60614	66623	47.8%	0.5213	21,700	195	34	4.82	106,909	4,328	79.3%	37.1	8.5	24.3	10.2	17.5	(55.92)	157	100.3	-	31.9	190.7	4	245
60608	82743	52.4%	0.3789	18,900	109	27	5.03	14,489	799	87.8%	46.0	9.4	17.8	8.8	3.8	(44.41)	163	13.7	-	35.4	112.7	2	145
60622	52593	51%	0.5418	34,400	354	25	4.57	45,114	(146)	86.0%	29.2	9.1	25.4	16.1	31.1	(41.12)	14.8	42.6	-	34.7	162.6	3	209
60606	234	49.2%	0.5239	10,200	227	8	4.40	37,508	(242)	90.9%	1.3	8.8	25.0	4.8	20.3	(3.87)	14.3	35.4	78.8	36.5	215.4	5	271
60607	23902	49.2%	0.5239	29,300	62	28	4.40	61,395	232	91.2%	12.3	8.8	25.0	12.7	5.6	(42.78)	14.3	57.9	-	36.7	132.5	3	170
60642	18495	51%	0.5492	11,900	9	12	4.37	24,416	107	90.8%	10.3	9.1	25.6	5.2	0.8	(19.74)	14.2	23.0	-	36.6	105.1	2	135
60610	37730	46.4%	0.4858	22,300	122	18	4.52	61,694	708	81.9%	21.0	8.3	22.8	10.3	10.9	(29.61)	14.7	58.2	-	33.0	149.6	3	192
60611	28722	46.4%	0.4858	18,100	149	16	4.67	89,207	5,926	58.4%	16.0	8.3	22.8	8.5	13.3	(26.32)	15.2	84.2	-	23.5	165.4	4	213
60654	14880	46.7%	0.4889	23,000	115	14	4.40	68,187	(2,806)	88.9%	8.3	8.3	22.8	10.8	10.3	(23.03)	14.3	64.4	64.6	35.8	216.7	5	278
60604	575	49.4%	0.5239	11,000	227	3	4.55	15,680	(450)	80.4%	0.3	8.8	25.1	5.1	20.3	(4.93)	14.8	14.8	10.4	32.4	127.1	3	163
60603	497	49.3%	0.5213	13,700	98	5	4.56	27,542	(2,821)	85.5%	0.3	8.8	24.9	6.4	8.8	(8.22)	14.8	60.0	22.4	14.2	169.2	4	217
60616	48437	48.1%	0.3604	6,100	87	29	5.35	32,058	412	77.8%	27.0	8.6	16.9	2.9	7.8	(47.70)	17.4	30.3	-	31.2	94.3	2	121
60602	1210	49.2%	0.5222	18,750	50	3	4.53	17,152	(22)	79.3%	0.7	8.8	25.0	8.8	4.4	(4.93)	14.7	16.2	0.5	31.9	106.1	2	136
60661	7798	49.2%	0.5231	15,600	102	12	4.34	78,847	(2,434)	83.3%	4.3	8.8	25.0	7.3	9.1	(19.74)	14.1	74.4	56.1	37.6	217.0	5	279
60637	49508	44.9%	0.3076	20,900	72	17	8.15	12,684	(205)	81.4%	27.5	8.0	14.4	8.8	6.4	(27.96)	26.5	12.0	5.9	32.8	110.4	2	148
60657	68001	43.7%	0.508	12,900	201	20	5.28	54,456	2,872	80.5%	36.7	8.9	28.4	6.0	18.0	(32.89)	17.1	51.4	-	32.4	166.2	4	218
60647	87291	50.4%	0.4402	10,600	204	24	5.43	26,516	835	82.6%	48.8	9.0	21.6	5.0	16.3	(38.47)	17.6	25.0	-	32.3	138.9	3	174
60612	33478	48.4%	0.4414	29,500	78	19	4.88	13,200	(840)	89.2%	18.9	8.7	20.8	10.8	6.8	(32.25)	16.2	13.8	14.9	30.5	116.7	2	150
60615	40606	44.3%	0.3601	10,900	84	11	7.94	10,822	84	76.7%	22.8	8.0	16.9	5.1	6.7	(18.09)	24.6	10.2	-	30.9	105.8	2	136
60618	92089	50.2%	0.3917	18,700	193	21	5.95	10,973	805	85.2%	51.2	9.0	18.5	8.7	17.3	(24.54)	19.3	10.4	-	34.3	134.2	3	172
60613	48285	50.4%	0.556	11,600	76	23	5.90	43,431	884	79.5%	26.9	9.0	26.1	5.4	6.8	(37.83)	19.2	41.0	-	32.0	128.6	3	165
60653	29912	43.2%	0.3031	19,900	59	9	6.62	2,101	176	73.3%	16.6	7.7	14.2	9.3	5.3	(14.80)	21.5	2.0	-	29.5	91.4	2	117
60625	78954	50.0%	0.3828	23,800	286	19	6.85	11,657	390	87.8%	43.8	8.9	18.4	11.1	25.6	(31.25)	22.2	11.2	-	35.4	145.5	3	187
60640	65796	51.8%	0.4291	16,700	205	14	6.71	22,613	447	80.4%	36.6	9.3	20.1	7.8	18.4	(23.03)	21.8	21.3	-	32.4	144.6	3	186
60619	63830	43.8%	0.2334	17,800	31	17	9.86	638	46	77.8%	35.5	7.8	11.0	8.3	2.8	(27.86)	32.0	0.6	-	31.2	101.3	2	130
60634	74302	49.1%	0.2885	21,800	112	1	5.33	3,164	528	79.8%	11.2	8.8	13.4	10.2	10.0	(1.64)	17.3	3.0	-	32.1	134.6	3	173
60626	50144	50.4%	0.4019	7,100	165	15	8.92	7,235	60	80.5%	27.9	9.0	18.9	3.3	13.0	(24.67)	29.0	6.8	-	34.0	117.3	3	151
60621	35916	44.8%	0.2688	15,000	28	12	8.28	395	(14)	86.3%	21.2	8.0	12.1	7.0	2.5	(19.74)	26.9	0.4	0.3	34.8	92.3	2	119
60624	38109	46.8%	0.2786	11,800	35	8	5.71	339	(15)	88.3%	21.2	8.3	13.1	5.5	3.1	(13.16)	18.5	0.3	0.4	27.4	84.8	2	109
60623	92112	53.7%	0.3291	13,700	109	6	5.88	560	28	82.0%	51.3	8.6	15.1	6.4	9.8	(9.97)	19.1	0.5	-	33.0	135.7	3	174
60645	45280	49.9%	0.3074	11,400	95	7	9.99	1286	-	77.2%	25.2	8.9	14.4	5.3	4.9	(1.59)	29.2	1.2	-	31.1	108.7	2	140
60659	39108	49.1%	0.2895	14,400	19	4	7.79	691	86	76.5%	21.2	8.8	13.5	20.6	1.7	(6.59)	25.3	0.6	-	30.4	115.5	2	148
60660	42787	51.2%	0.3886	35,200	42	5	7.57	5,216	471	79.7%	23.8	9.2	18.2	16.5	3.8	(8.23)	24.6	4.9	-	32.1	124.8	3	160
60641	78668	49.8%	0.3211	39,900	91	6	7.21	1,173	(63)	77.2%	39.9	8.9	15.1	18.7	7.3	(9.87)	23.4	11	1.5	31.1	131.0	3	176
60630	54099	49.2%	0.3003	12,300	37	2	7.28	230	9	79.5%	30.1	8.8	14.1	5.8	3.3	(3.29)	23.6	0.2	-	29.6	112.2	2	144
60651	64273	47.3%	0.2911	34,900	31	4	6.25	894	(9)	70.0%	35.8	8.5	13.7	16.3	2.8	(6.58)	20.3	0.8	0.2	28.2	119.9	3	154
60644	48852	45.9%	0.2681	9,800	49	13	6.89	278	(19)	69.4%	27.1	8.2	12.5	4.6	4.4	(21.38)	22.4	0.3	0.4	28.0	86.4	2	111
60636	40923	46.8%	0.257	18,200	20	8	8.10	105	4	76.2%	22.8	8.3	12.1	8.5	1.8	(13.16)	26.3	0.1	-	30.7	97.4	2	125
60617	84161	46.5%	0.2504	8,600	190	6	10.84	191	(2)	80.5%	46.8	8.3	11.8	4.0	17.0	(9.87)	35.2	0.2	0.0	36.3	149.8	3	192
60201	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	9	11.62	3,783	(54)	77.9%	#N/A	#N/A	#N/A	#N/A	(14.80)	37.7	3.6	1.2	31.4	134.6	3	173
60202	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	3	10.36	1,141	102	80.5%	#N/A	#N/A	#N/A	#N/A	(4.93)	33.6	11	-	32.4	137.0	3	176
60208	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	2	11.80	1,876	(26)	80.5%	#N/A	#N/A	#N/A	#N/A	(3.29)	38.3	1.8	0.6	32.4	91.4	2	117
60639	90411	49.5%	0.3133	12,500	45	1	6.04	200	(9)	75.5%	50.3	8.8	14.7	5.8	4.0	(1.64)	19.6	0.2	0.2	30.4	132.5	3	170
60628	72206	44.9%	0.2238	12,500	103	-	-	-	-	-	40.2	8.0	11.0	5.8	9.2	88.89	-	-	-	-	163.1	3	210
60643	49997	49.9%	0.2177	27,800	93	-	-	-	-	-	27.8	9.2	10.2	13.0	9.3	80.89	-	-	-	-	148.4	3	191
60620	72220	44.0%	0.2235	12,300	95	-	-	-	-	-	40.2	11.0	5.8	4.8	4.								

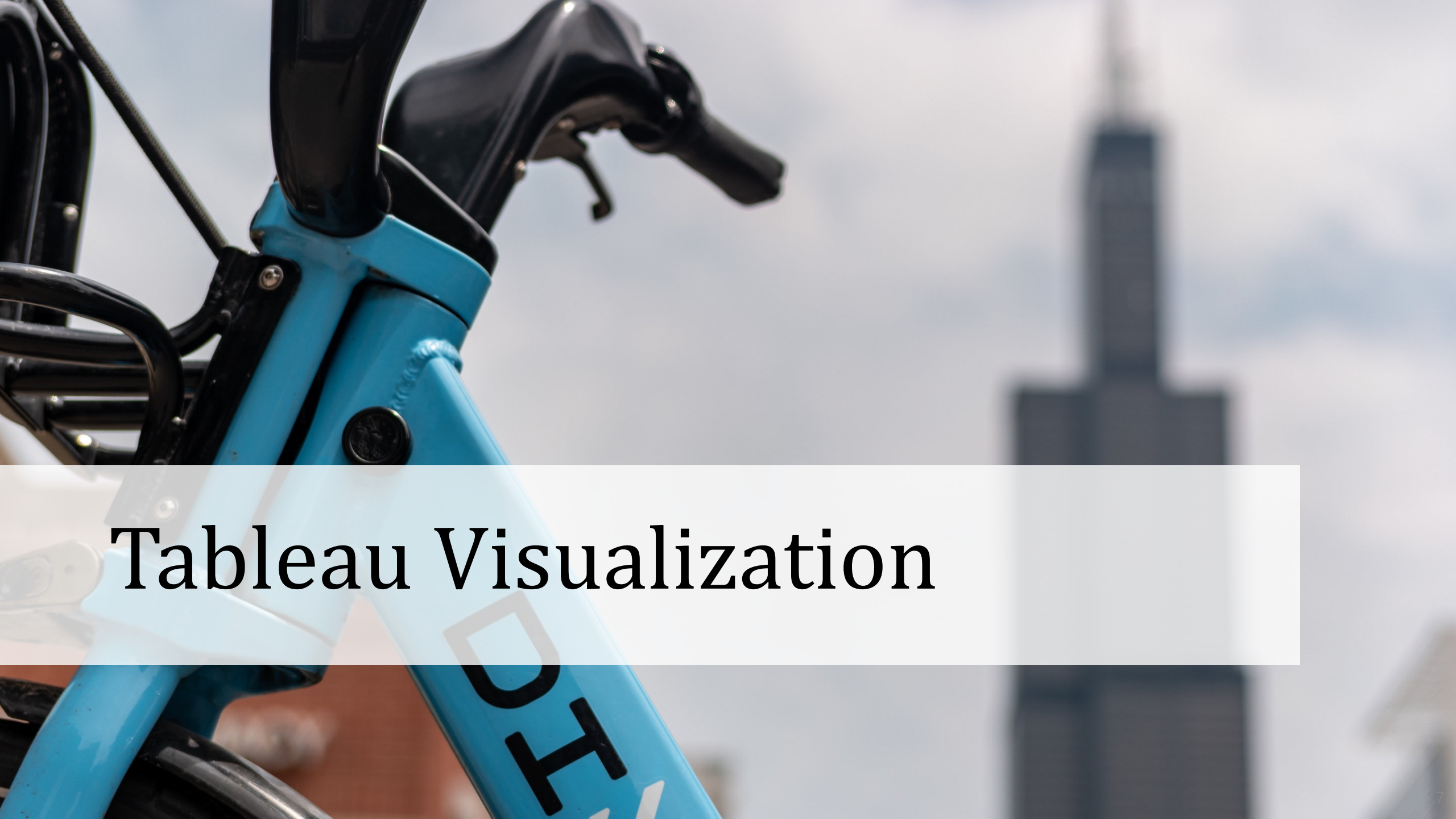
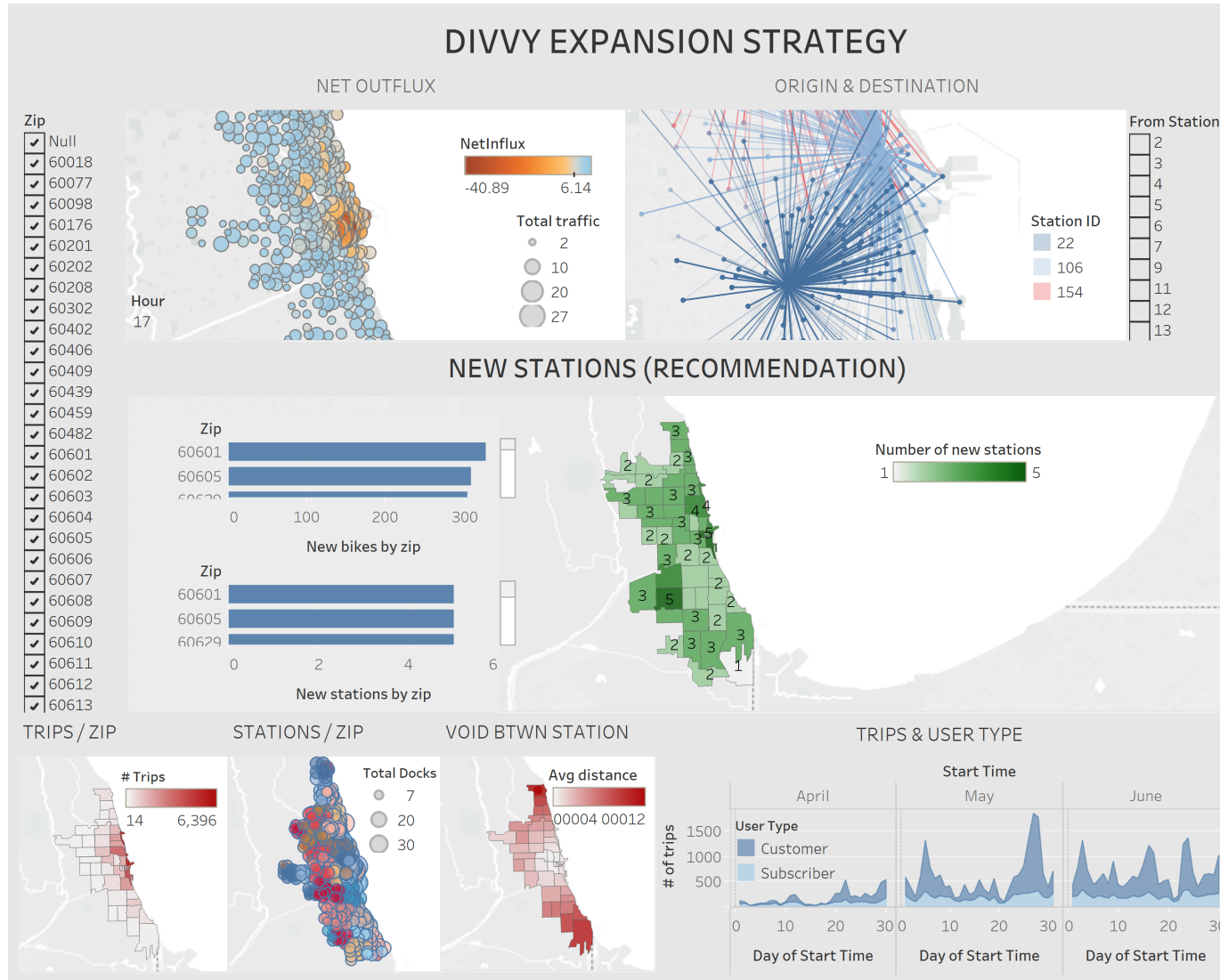


Tableau Visualization



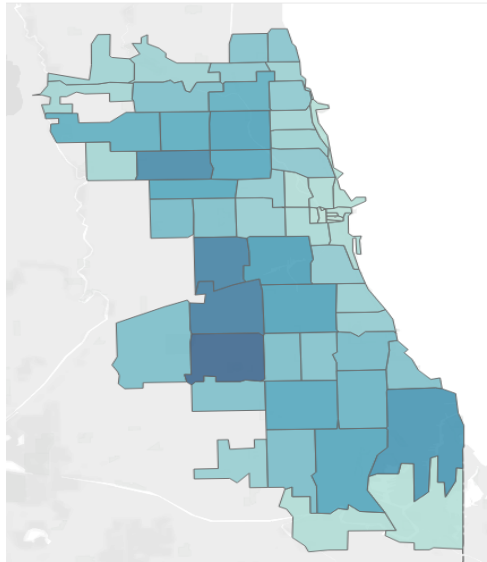
Derived recommendation from trip and zip demographics:

- **Net Outflux:** Number of bikes stalled minus number of bikes taken for each station and filtered by hour
- **Origin & Destination:** All destinations of the trips taken from a respective station
- **New Stations (Recommendation):** Suggested number of new stations per zip code, based on the previously described scoring methodology (+ Number of suggested new bikes and stations per zip code as bar chart)
- **Trips / Zip:** Average number of trips started in a respective zip code
- **Stations / Zip:** All divvy stations filtered by zip code (color wise) and number of docks (bubble size)
- **Void Btw Station:** Average distance in 100 meters between stations within one zip code
- **Trips & User Type:** Number of trips taken filtered by subscribers and non-subscribers ('customers')

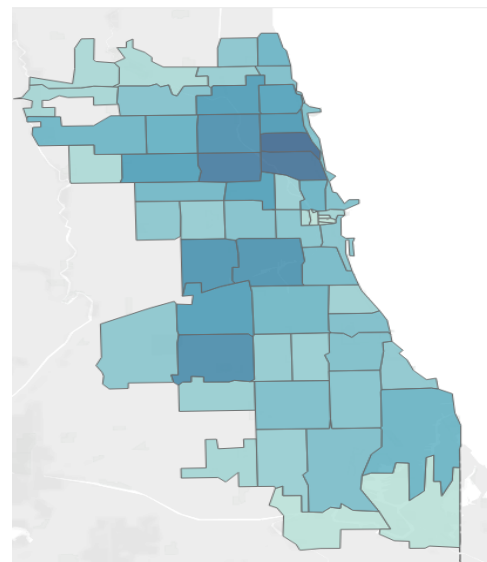
Demographics by Zip Code



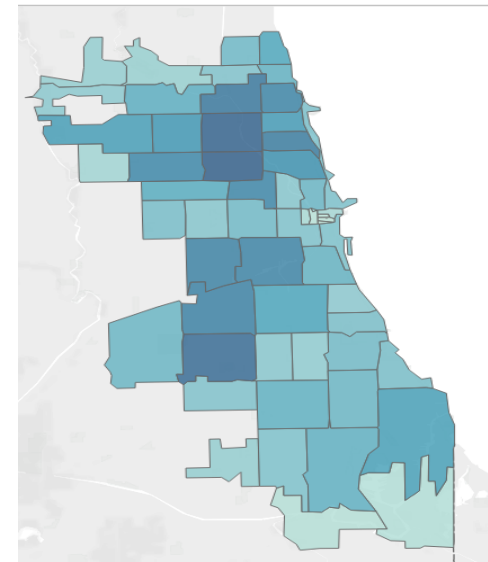
0-19 ZIP



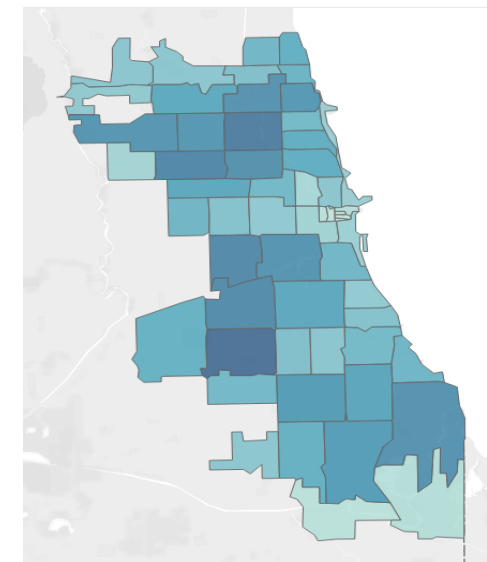
20-29



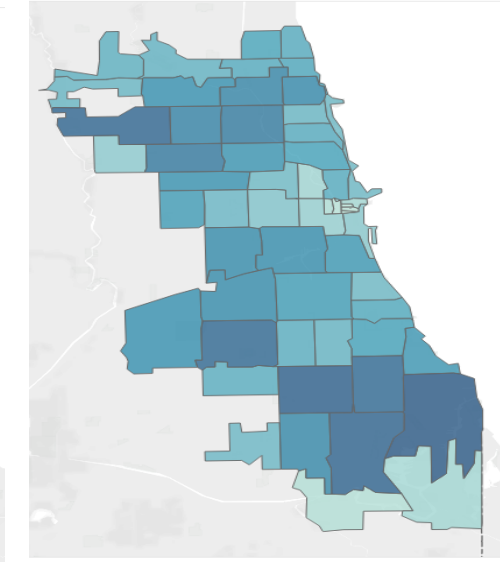
30-39



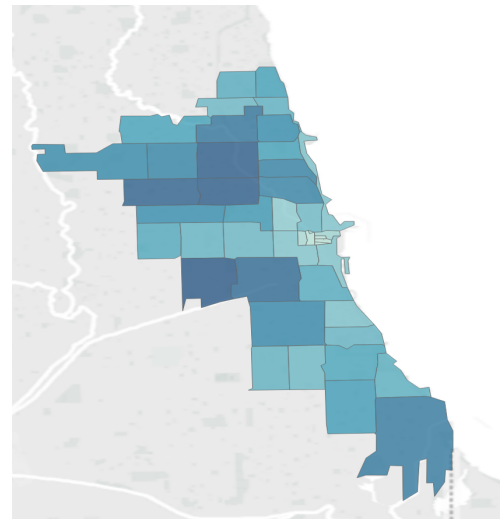
40-49



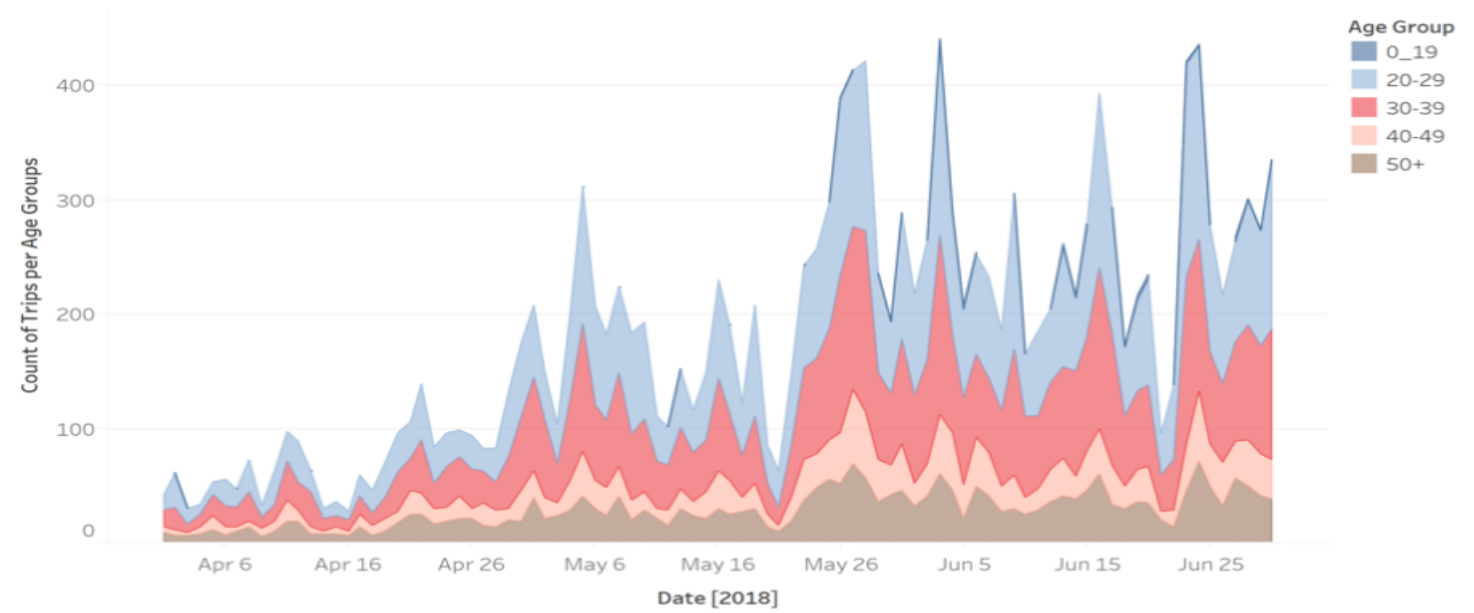
50+



Total population by zip



Number of trips taken by age groups





Summary



Recommendations and Future Vision:

- Increase stations in ZIPs further from downtown Chicago based on scoring variables to serve the needs of local residents better
- Allocate more bikes to stations with higher net outflux (especially during summer)
- More advanced analysis based on variables like customer feedback, commercial footprints, real estate bike scores etc.
- Capitalize on the existing bike rack network in Chicago
- Expand to OLTP framework to support real time trip information.
- Scaling out to support the ever increasing data repository.

Lessons Learned:

- Choose your data sources carefully, every data source has its own conventions and business case.
- Make sure geographic data from different sources is coherent.
- Don't over normalize for OLAP - keep it simple!
- Split up data sources / use views for faster processing in tableau.
- Excel is a very powerful tool.



THANK YOU!
