EDA：

Red: Woochan

* **Transformation (2019 & 2020 separately)**
  + Add van reshuffling rows
  + Add capacity
  + Add january start (starting number of bikes) (add number of bikes that arrive that day)

Blue: Cecilia

**Explore top 10 high demand stations -> see trends in location / region**

**Explore bottom 10 low demand stations -> see trends in location / region**

* **Decrease or increase number of rides over time, remove or expand**
* **Compare average number of rides before 2020 and after to see effects of COVID**
* **Weekday / Weekend**
* **Commute time - Yes / No (6am~9am / 4pm~6pm)**
* **Get the final dataset out**
* **Random forest**
* **xgboost**

Brown: Yueni

* **Group stations into regions (urban / residential / office)(looking at the map)**

**K-means clustering using longitudes and latitudes(try 5 to 10)**

**Limitation: Electric\_bike, interval cross-validation selection,**

* **Threshold rule: more than 2 bikes will be at the bike station for more than 30 minutes.**
* **Days vs Total rides**
* **Hours vs Total rides**
* **Grouping the data by weekdays/locations**

Other Things:

* Add rules for charging fees for rides, member or casual
* Using Google Earth(?)/Google Map to check the locations of stations and see if there’s specific pattern in routes of users
  + Also check for the bus stops/subway stations near high/low demand bike stations-The difference in demand could be because of public transportation construction.(Future plans for construction?

Model:

|  | Station 1 | Station 2 |
| --- | --- | --- |
| 00:00 - 00:30 | Average no. of bikes -> more than 2 -> yes |  |
| 00:30 - 01:00 |  |  |

Availability = the original value+ **avg(sum(drop off))**

**Input**

Station, the time interval assigned / All the other categorical variables we mentioned in EDA

(Should we consider the lat&log?)

| Other variables | Station | Time interval | response/availability |
| --- | --- | --- | --- |
|  |  |  | Yes/No |
|  |  |  |  |

**Output**: availability

Hint： increase threshold

Find the available bikes during the same intervals

January 1st all the bikes are reshuffled,

the starting amount of bikes-all the bikes ending at that station

Assumptions: the sche

Potential variables

* Weather (precipitation, wind strength):

<https://data.noaa.gov/onestop/collections?q=%22NOAA%20Climate%20Data%20Record%22>

* Group stations into regions (urban / residential / office)
  + Or group by demand (number of rides per station cluster), get the number of data points for each station and cluster them to “high demand”, “median demand”, or “low demand”
* Weekday / Weekend
* Commute time - Yes / No (6am~9am / 4pm~6pm)
* auto.arima()

Response variable

Assumptions

* Remote work -> people don't commute to work (COVID)
  + Focus only on 2020~current (COVID time)
* Public holidays = weekend, Thanksgiving or Christmas

Questions to ask

* Factors limited to the dataset in the instructions? Weather?

Things to do:

1. Format data into input form
   1. Delete ID

**2021-Nov-12**

* Think about what observations to delete
  + Stations that have a large average negative availability value
    - Delete if smaller than -10?
  + Station\_id == 0
* Some problems
  + Some station\_ids have multiple station\_names
    - This is mostly due to spelling differences in station names over time
    - # of unique station\_ids = 582
    - # of unique station\_names = 640
    - Solution: use only station\_id for analysis, grouping
  + Cannot use station capacity since we have to join tables by station name. (capacity dataset does not have correct station\_ids to join with)
  + Remapped station 0 to 31127 (possibly due to station\_id change or relocation of station?)
  + Station 31209 is the National Park. On days with baseball games the number of bikes increased a lot. (2019 Oct 1, 2019 Oct 7, etc.)
* Modeling
  + First, group the dataset to every 30 mins
    - Weekday vs weekend (holidays included)
    - Seasonal (Spring, Summer, Winter, etc.)
    - Time of day (rush hour, group by time of day)
  + Goal: predict whether bike will be available the next 30 mins

-Recalibrate starting bike numbers every month

- Add minimum number of availability each month

- Lower negative -> we calibrate -> keep the higher cap -> to do proportion

Divide samples (trainng, test set -> how? seasonal?)