Data_curation_project | Projects | Branches | Notebook | Dataset | Caveats | Settings

On branch **default**

[1]

Console | Timing | Datasets | Charts

Group_15 Divvy Biking Beyond Boundaries

[2]

Console | Timing | Datasets | Charts

Data Collection and Preparation

Divvy Trip Data

The datasets were downloaded from this link. A total of 12 csv files (1 file per month) were upload into Python as Pandas Dataframes. The files were combi into 1 file using `.concat()` method.

```
[3] LOAD DATASET 202211-divvy-tripdata AS csv FROM 202211-divvy-tripdata.csv @ artifact file 32
```

Console | Timing | Datasets | Charts

202211-divvy-tripdata (**337735** rows)

Views

	ride_id (string)	rideable_type (string)	started_at (date)	ended_at (date)	start_station_name (string)	start_station_id (string)	end_station_name (string)
0	BCC66FC6FAB27CC7	electric_bike	2022-10-10	2022-10-10	Canal St & Adams St	13011	St. Clair St & Erie St
1	772AB67E902C180F	classic_bike	2022-10-04	2022-10-04	Canal St & Adams St	13011	St. Clair St & Erie St
2	585EAD07FDEC0152	classic_bike	2022-10-21	2022-10-21	Indiana Ave & Roosevelt Rd	SL-005	St. Clair St & Erie St
3	91C4E7ED3C262FF9	classic_bike	2022-10-25	2022-10-25	Indiana Ave & Roosevelt Rd	SL-005	St. Clair St & Erie St
4	709206A3104CABC8	classic_bike	2022-10-29	2022-10-29	Indiana Ave & Roosevelt Rd	SL-005	St. Clair St & Erie St
5	11DE62E16D1A6BD1	classic_bike	2022-10-04	2022-10-04	Streeter Dr & Grand Ave	13022	Desplaines St & Kinzie St
6	0F05C92DE9981154	classic_bike	2022-10-06	2022-10-06	Halsted St & Clybourn Ave	331	Desplaines St & Kinzie St
7	CBEB7068C91BB672	classic_bike	2022-10-08	2022-10-08	Halsted St & Clybourn Ave	331	Orleans St & Chestnut St (NEXT Ap
8	C79980F697C514FF	electric_bike	2022-10-13	2022-10-13	Franklin St & Adams St (Temp)	TA1309000008	Desplaines St & Kinzie St
9	A65691E4D563258E	classic_bike	2022-10-01	2022-10-01	Halsted St & Clybourn Ave	331	Orleans St & Chestnut St (NEXT Ap
10	DE804DEF5E506AF6	classic_bike	2022-10-07	2022-10-07	Halsted St & Clybourn Ave	331	Orleans St & Chestnut St (NEXT Ap
11	53706BF327A91FF2	classic_bike	2022-10-06	2022-10-06	Streeter Dr & Grand Ave	13022	Orleans St & Chestnut St (NEXT Ap
12	94D1BE7DE259ACD5	electric_bike	2022-10-02	2022-10-02	Halsted St & Clybourn Ave	331	Desplaines St & Kinzie St
13	5608C656DE64F4B2	classic_bike	2022-10-15	2022-10-15	Larrabee St & Division St	KA1504000079	Orleans St & Chestnut St (NEXT Ap
14	1423D0A1B5F36926	electric_bike	2022-10-30	2022-10-30	Kedzie Ave & Bryn Mawr Ave	KA1504000167	Western Ave & Ardmore Ave
15	6E1AA92E7988B202	electric_bike	2022-10-24	2022-10-24	Larrabee St & Division St	KA1504000079	Desplaines St & Kinzie St
16	FB7DA0A2E2380E1C	classic_bike	2022-10-10	2022-10-10	Larrabee St & Division St	KA1504000079	Orleans St & Chestnut St (NEXT Ap
17	11AECEB7779D78EA	electric_bike	2022-10-21	2022-10-21	Franklin St & Adams St (Temp)	TA1309000008	Desplaines St & Kinzie St
18	9718191F824DF2D7	classic_bike	2022-10-10	2022-10-10	Streeter Dr & Grand Ave	13022	Desplaines St & Kinzie St
19	0F50E772CD3143A9	electric_bike	2022-10-07	2022-10-07	Franklin St & Adams St (Temp)	TA1309000008	Desplaines St & Kinzie St

```
[4] LOAD DATASET mar AS csv FROM 202303-divvy-tripdata.csv @ artifact file 34
```

Console | Timing | Datasets | Charts

mar (**258678** rows)

Views

	ride_id (string)	rideable_type (string)	started_at (date)	ended_at (date)	start_station_name (string)	start_station_id (string)	end_station_name (str
0	6842AA605EE9FBB3	electric_bike	2023-02-16	2023-02-16	Clark St & Armitage Ave	13146	Larrabee St & Webster Ave
1	F984267A75B99A8C	electric_bike	2023-02-04	2023-02-04	Public Rack - Kedzie Ave & Argyle St	491	
2	FF7CF57CFE026D02	classic_bike	2023-02-31	2023-02-31	Orleans St & Chestnut St (NEXT Apts)	620	Clark St & Randolph St
3	6B61B916032CB6D6	classic_bike	2023-02-22	2023-02-22	Desplaines St & Kinzie St	TA1306000003	Sheffield Ave & Kingsbury St
4	E55E61A5F1260040	electric_bike	2023-02-09	2023-02-09	Walsh Park	18067	Sangamon St & Lake St

Data_curation_project | Projects | Branches | Notebook | Dataset | Caveats | Settings

6	5929D3080983AF4F	classic_bike	2023-02-08	2023-02-08	Rush St & Hubbard St	KA1503000044	Wells St & Huron St
7	B2624BAEDDDA3FB1	docked_bike	2023-02-22	2023-02-22	Adler Planetarium	13431	Michigan Ave & Ida B Wells
8	979C41EAC356278F	classic_bike	2023-02-16	2023-02-16	Broadway & Wilson Ave	13074	Ravenswood Ave & Irving Pa
9	6C1DCA9593CA8F5F	classic_bike	2023-02-16	2023-02-16	Stetson Ave & South Water St	TA1308000029	Clinton St & Washington Blv
10	74FA89B21DC5856D	docked_bike	2023-02-16	2023-02-16	Clark St & Wellington Ave	TA1307000136	DuSable Lake Shore Dr & M
11	FBDE4914FD9CBB92	electric_bike	2023-02-13	2023-02-13	Desplaines St & Kinzie St	TA1306000003	Green St & Madison St
12	C8A1A1A8CD23C9EF	electric_bike	2023-02-24	2023-02-24	Sheridan Rd & Lawrence Ave	TA1309000041	Broadway & Argyle St
13	66E9299F769DA880	classic_bike	2023-02-05	2023-02-05	Sheridan Rd & Lawrence Ave	TA1309000041	Michigan Ave & Washington
14	B783A22C8C33115C	classic_bike	2023-02-26	2023-02-26	Clark St & Armitage Ave	13146	Clark St & Elm St
15	106A5EAF32F1B4F5	classic_bike	2023-02-15	2023-02-15	Clark St & Elm St	TA1307000039	Larrabee St & Kingsbury St
16	100EE4CDE868CEBC	classic_bike	2023-02-02	2023-02-02	Sheffield Ave & Wellington Ave	TA1307000052	Racine Ave & Belmont Ave
17	E6B41D39CBA26BA7	classic_bike	2023-02-02	2023-02-02	Sheffield Ave & Wellington Ave	TA1307000052	Racine Ave & Belmont Ave
18	824392AF00330966	classic_bike	2023-02-04	2023-02-04	Orleans St & Chestnut St (NEXT Apts)	620	State St & Chicago Ave
19	147943A31717D1CA	electric_bike	2023-02-10	2023-02-10	Desplaines St & Jackson Blvd	15539	N Green St & W Lake St

```
[5]  LOAD DATASET feb AS csv FROM 202302-divvy-tripdata.csv @ artifact file 36
```

Console | Timing | Datasets | Charts

feb (**190445** rows)

Views

	ride_id (string)	rideable_type (string)	started_at (date)	ended_at (date)	start_station_name (string)	start_station_id (string)	end_station_name (string)	enc
0	CBCD0D7777F0E45F	classic_bike	2023-01-14	2023-01-14	Southport Ave & Clybourn Ave	TA1309000030	Clark St & Schiller St	TA1
1	F3EC5FCE5FF39DE9	electric_bike	2023-01-15	2023-01-15	Clarendon Ave & Gordon Ter	13379	Sheridan Rd & Lawrence Ave	TA1
2	E54C1F27FA9354FF	classic_bike	2023-01-19	2023-01-19	Southport Ave & Clybourn Ave	TA1309000030	Aberdeen St & Monroe St	131
3	3D561E04F739CC45	electric_bike	2023-01-26	2023-01-26	Southport Ave & Clybourn Ave	TA1309000030	Franklin St & Adams St (Temp)	TA1
4	0CB4B4D53B2DBE05	electric_bike	2023-01-20	2023-01-20	Prairie Ave & Garfield Blvd	TA1307000160	Cottage Grove Ave & 63rd St	KA
5	C67EB62172C472EB	classic_bike	2023-01-24	2023-01-24	Wells St & Concord Ln	TA1308000050	Clybourn Ave & Division St	TA1
6	08A1E9326F68ACF7	classic_bike	2023-01-28	2023-01-28	Wells St & Concord Ln	TA1308000050	Clybourn Ave & Division St	TA1
7	904C61FB3984A60E	classic_bike	2023-01-27	2023-01-27	Wells St & Concord Ln	TA1308000050	Clybourn Ave & Division St	TA1
8	A96A6DA2D96544E6	classic_bike	2023-01-08	2023-01-08	Wells St & Concord Ln	TA1308000050	Clybourn Ave & Division St	TA1
9	DA895AE47787D208	classic_bike	2023-01-21	2023-01-21	Wells St & Concord Ln	TA1308000050	Clybourn Ave & Division St	TA1
10	C00FEDDDD22036FC	classic_bike	2023-01-07	2023-01-07	Franklin St & Illinois St	RN-	Clybourn Ave & Division St	TA1
11	C44FB10BA3BB81A6	classic_bike	2023-01-23	2023-01-23	Wells St & Concord Ln	TA1308000050	Clybourn Ave & Division St	TA1
12	19034AD09CB28231	classic_bike	2023-01-20	2023-01-20	Wells St & Concord Ln	TA1308000050	Clybourn Ave & Division St	TA1
13	AE8D0EF0F724CD16	electric_bike	2023-01-06	2023-01-06	Clarendon Ave & Gordon Ter	13379	Southport Ave & Irving Park Rd	TA1
14	2BDFB72683246DE2	electric_bike	2023-01-17	2023-01-17	Southport Ave & Clybourn Ave	TA1309000030	Ritchie Ct & Banks St	KA
15	E24AAE6D0AEFBDEE	classic_bike	2023-01-05	2023-01-05	Southport Ave & Clybourn Ave	TA1309000030	Sheffield Ave & Webster Ave	TA1
16	13C104BD0BE3C777	classic_bike	2023-01-01	2023-01-01	Southport Ave & Clybourn Ave	TA1309000030	Sheffield Ave & Webster Ave	TA1
17	DAAFF461BBFC9B51	classic_bike	2023-01-26	2023-01-26	Indiana Ave & 31st St	TA1308000036	Indiana Ave & 31st St	TA1
18	0BD0F9320649C18B	classic_bike	2023-01-09	2023-01-09	Calumet Ave & 33rd St	13217	Indiana Ave & 26th St	TA1
19	B56FB3AA56CD1F5B	classic_bike	2023-01-12	2023-01-12	Broadway & Thorndale Ave	15575	Sheridan Rd & Lawrence Ave	TA1

```
[6]  LOAD DATASET 202304-divvy-tripdata AS csv FROM 202304-divvy-tripdata.csv @ artifact file 38
```

Console | Timing | Datasets | Charts

202304-divvy-tripdata (**426590** rows)

Views

	ride_id (string)	rideable_type (string)	started_at (date)	ended_at (date)	start_station_name (string)	start_station_id (string)	end_station_name (string)	end_st
0	8FE8F7D9C10E88C7	electric_bike	2023-03-02	2023-03-02				
1	34E4ED3ADF1D821B	electric_bike	2023-03-19	2023-03-19				
2	5296BF07A2F77CB5	electric_bike	2023-03-19	2023-03-19				
3	40759916B76D5D52	electric_bike	2023-03-19	2023-03-19				
4	77A96F460101AC63	electric_bike	2023-03-19	2023-03-19				

Data_curation_project | Projects | Branches | Notebook | Dataset | Caveats | Settings

6	C97BBA66E07889F9	electric_bike	2023-03-19	2023-03-19				
7	6687AD4C575FF734	electric_bike	2023-03-11	2023-03-11				
8	A8FA4F73B22BC11F	electric_bike	2023-03-11	2023-03-11				
9	81E158FE63D99994	electric_bike	2023-03-19	2023-03-19				
10	23825895B7494035	electric_bike	2023-03-20	2023-03-20				
11	D0851F6357674EA9	electric_bike	2023-03-20	2023-03-20				
12	B4A58C92320522A7	electric_bike	2023-03-20	2023-03-20				
13	2FD726F06E1AB12F	electric_bike	2023-03-20	2023-03-20				
14	AF1EB9BF06F96747	electric_bike	2023-03-20	2023-03-20				
15	65C5A699A9E24A11	electric_bike	2023-03-20	2023-03-20				
16	E61C962970871D04	electric_bike	2023-03-11	2023-03-11				
17	178B87D025D70DD7	electric_bike	2023-03-11	2023-03-11				
18	47CCDCDB305A3F79	electric_bike	2023-03-11	2023-03-11				
19	058F26BBB79B8E36	electric_bike	2023-03-27	2023-03-27				

[7]

```
select * from feb union select * from mar
```

Console | Timing | Datasets | Charts

df (**449123** rows)

Views

	ride_id (string)	rideable_type (string)	started_at (date)	ended_at (date)	start_station_name (string)	start_station_id (string)	end_station_name (st
0	BDF1870A9E24D1CD	classic_bike	2023-01-21	2023-01-21	Canal St & Jackson Blvd	13138	Stetson Ave & South Wat
1	96758F299BD962BA	classic_bike	2023-01-13	2023-01-13	Clark St & Bryn Mawr Ave	KA1504000151	Marine Dr & Ainslie St
2	8D3F3500A10BC669	electric_bike	2023-01-19	2023-01-19	Eckhart Park	13289	California Ave & Cortez S
3	3ABDCD0D55AD3D33	classic_bike	2023-01-02	2023-01-02	Calumet Ave & 33rd St	13217	State St & 33rd St
4	0B3264B472C5CBE3	classic_bike	2023-02-12	2023-02-12	Kimbark Ave & 53rd St	TA1309000037	Woodlawn Ave & Lake Pa
5	C0773416A1E63FB7	electric_bike	2023-02-16	2023-02-16	State St & Randolph St	TA1305000029	Michigan Ave & Oak St
6	EE784792847E094E	classic_bike	2023-02-27	2023-02-27	Michigan Ave & Jackson Blvd	TA1309000002	State St & Chicago Ave
7	829B762CECD9BFCF	classic_bike	2023-02-17	2023-02-17	Desplaines St & Jackson Blvd	15539	Clinton St & Madison St
8	26C669769E87AF2F	electric_bike	2023-02-01	2023-02-01	Damen Ave & Wellington Ave	13268	Southport Ave & Wellingt
9	1B073A58A64D41A6	classic_bike	2023-02-20	2023-02-20	Broadway & Sheridan Rd	13323	Wilton Ave & Belmont Av
10	E91F551514C9E570	electric_bike	2023-01-05	2023-01-05			
11	23A4A7BFE97D1DA5	electric_bike	2023-01-01	2023-01-01			Throop St & Taylor St
12	72B7906E02A6F833	electric_bike	2023-02-16	2023-02-16			
13	6945E7D2160ADE41	electric_bike	2023-02-10	2023-02-10			
14	2DC570EDD622F397	electric_bike	2023-02-27	2023-02-27			
15	809656331F09DA90	electric_bike	2023-01-03	2023-01-03	Lincoln Ave & Fullerton Ave	TA1309000058	Sedgwick St & Schiller St
16	CE7E5BBE418C7B9A	electric_bike	2023-01-19	2023-01-19	DuSable Lake Shore Dr & Diversey Pkwy	TA1309000039	Wabash Ave & Adams St
17	3C2DAE0D7F0436C2	electric_bike	2023-01-07	2023-01-07	Halsted St & Polk St	TA1307000121	Racine Ave & 18th St
18	FBEE084301966F5C	classic_bike	2023-01-17	2023-01-17	Kingsbury St & Kinzie St	KA1503000043	Peoria St & Jackson Blvd
19	B48A5A869B70B608	electric_bike	2023-01-18	2023-01-18	Lakeview Ave & Fullerton Pkwy	TA1309000019	Lakeview Ave & Fullerton

[8]

```
#import packages
import pandas as pd
import math
import numpy as np
import matplotlib.pyplot as plt
#import seaborn as sns

#read each csv file
#nov_22 = vizierdb.get_data_frame('202211-divvy-tripdata')
#feb_23 = vizierdb.get_data_frame('202302-divvy-tripdata')
#mar_23 = vizierdb.get_data_frame('202303-divvy-tripdata')
#apr_23 = vizierdb.get_data_frame('202304-divvy-tripdata')
```

```
#dec_22 = pd.read_csv("202212-divvy-tripdata.csv")
#jan_23 = pd.read_csv("202301-divvy-tripdata.csv")
#feb_23 = pd.read_csv("202302-divvy-tripdata.csv")
#mar_23 = pd.read_csv("202303-divvy-tripdata.csv")
#apr_23 = pd.read_csv("202304-divvy-tripdata.csv")
#may_23 = pd.read_csv("202305-divvy-tripdata.csv")
#jun_23 = pd.read_csv("202306-divvy-tripdata.csv")
#jul_23 = pd.read_csv("202307-divvy-tripdata.csv")
#aug_23 = pd.read_csv("202308-divvy-tripdata.csv")
#sep_23 = pd.read_csv("202309-divvy-tripdata.csv")
#oct_23 = pd.read_csv("202310-divvy-tripdata.csv")
#print('import done')

#use concat to combine 12 csv
#df=pd.concat([feb_23,mar_23], ignore_index=True)

#drop unnecessary columns
#df.drop(df.columns[[5, 7]], axis=1, inplace = True)

#df = pd.read_csv("df.csv")
df = vizierdb.get_data_frame('df')


#inspect dataframe
#df.head()
print("Merged Dataset having November-2022, February-2023, March-2023, April-2023")
df.info()

#df.describe()
print('The combined dataframe has 1.2 million records (1,213,447) and has 13 columns (attributes). Upon closer inspection it is obse
df['started_at'] = pd.to_datetime(df['started_at'])
df['ended_at'] = pd.to_datetime(df['ended_at'])
df.info()

print(' Calculate % unique values per column')

duplicates = df.nunique().reset_index()
duplicates.columns = ['column', 'unique_values']
duplicates['unique%'] = round((duplicates['unique_values'] / len(df)) * 100, 2)

print(' Calculate % missing values per column')

missing = df.isna().sum().reset_index()
missing.columns = ['column', 'missing_values']
missing['missing%'] = round((missing['missing_values'] / len(df)) * 100, 2)

print(' Combine the dataframes duplicates and missing to get a clear picture of the data')
combined_df = pd.merge(duplicates, missing, on='column')
print(combined_df)

print('checking if all the ride_id have exactly 16 charaters')
ride_id_len = (df['ride_id'].str.len()==16).all()
print(ride_id_len)

print('count duplicates on unique column')
print('this is done to find if there are any dupicate records present in the data as ride_id is the primary key, it is chosen')
print('Total duplicates in ride_id column: ',df['ride_id'].duplicated().sum())

print('rideable_type, this column is like the ID badge for each bike, telling us what model was used for the ride. We\'ve got three

print('finding the unique values in rideable_type column')
print(df['rideable_type'].unique())

print('counting the number of records that contained docked_bike')
count_before_change = (df['rideable_type'] == 'docked_bike').sum()
print('The number of records having docked_bike type before:', count_before_change)
print('changing docked_bike to classic_bike')
df['rideable_type'] = df['rideable_type'].replace('docked_bike','classic_bike')
print('counting the number of records after change')
count_after_change = (df['rideable_type'] == 'docked_bike').sum()
print('The number of records having docked_bike type after the change:', count_after_change)
print('checking for null values or empty values in rideable_type')
print('Total number of empty values or null values in rideable_type : ',df['rideable_type'].isna().sum())

print("We have already converted the started_at and ended_at columns to datetime format and hence we can proceed to find the duratic
df['date'] = df['started_at'].dt.date
```

```
df.head()

print("*************************************Dealing with outliers *********************************************** ")

print(" Count rows before filtering")
count_before = len(df)

print("First, create a DataFrame of outliers")
df_duration_noise = df[(df['ride_duration'] < 1) | (df['ride_duration'] > 24*60)]

print(" Then, filter df to remove outliers")
df = df[(df['ride_duration'] >= 1) & (df['ride_duration'] <= 24*60)]

print(" Count rows after filtering")
count_after = len(df)

print(" Print the number of rows deleted ")
print('This change has resulted in deleting', count_before - count_after, 'rows')

print("Check the shape of df_duration_noise")
df_duration_noise.shape

print("*************************************start_station_name and end_station_name ***********************************************

print("We've noticed a bit of a puzzle: quite a few trips are missing either their starting or ending station names. Now, here's whe

print("*************************************Identify classic bike records missing station names ***********************************

classic_missing_stations = (df['rideable_type'] == 'classic') & (df['start_station_name'].isna() | df['end_station_name'].isna())

print("*************************************Create df_station_noise DataFrame *********************************************** ")
df_station_noise = df[classic_missing_stations].copy()

print("*************************************Update the main DataFrame by removing these records ***********************************
df = df[~classic_missing_stations]

print("*************************************storing the outliers or noise in a dataframe called df_noise, this is done to preserve t
df_noise = pd.concat([df_duration_noise, df_station_noise])
df_noise.head()

print("*************************************Shape of the outliers dataframe that has been isolated ********************************
df_noise.shape
print("*************************************checking member_casual column for possible values ************************************
print((df['member_casual']).unique())
df['member_casual'].isna().sum()

print("************************************************************************************************************************
print("In our journey through the data, we've come across a neat little detail about the member_casual column. It turns out, it's pr
print("********************************************************************************* ****************************************
print("finding the number of missing values and their percentage after removing the outliers and noise")
missing = df.isna().sum().reset_index()
missing.columns = ['column', 'missing_values']
missing['missing%'] = round((missing['missing_values'] / len(df)) * 100, 2)
print(missing)
print("*************************************Columns to replace missing values with 'unknown' *************************************
columns_to_replace = ['start_station_name', 'start_station_id', 'end_station_name', 'end_station_id']

print("*************************************Replace NaN values in each specified column with 'unknown' ***************************
for column in columns_to_replace:
    df[column] = df[column].fillna('unknown')
print('****************************************DATA ANALYSIS**************************************')
ride_counts = df['member_casual'].value_counts()

# Data for the pie chart
labels = ride_counts.index
sizes = ride_counts.values

# Plotting the pie chart
plt.figure(figsize=(4, 3))
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140)
plt.axis('equal')  # Ensures the pie chart is circular
plt.title('Distribution of Rides: Members vs. Casual Riders')
plt.show()
df.to_csv('cleaned_divvydata.csv')
```

Data_curation_project Projects Branches Notebook Dataset Caveats Settings

Console Timing Datasets Charts

```
Merged Dataset having November-2022, February-2023, March-2023, April-2023
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 449123 entries, 0 to 449122
Data columns (total 13 columns):
 #   Column              Non-Null Count   Dtype  
---  ------              --------------   -----  
 0   ride_id             449123 non-null  object 
 1   rideable_type       449123 non-null  object 
 2   started_at          449123 non-null  object 
 3   ended_at            449123 non-null  object 
 4   start_station_name  387740 non-null  object 
 5   start_station_id    387608 non-null  object 
 6   end_station_name    383947 non-null  object 
 7   end_station_id      383806 non-null  object 
 8   start_lat           449123 non-null  float32
 9   start_lng           449123 non-null  float32
 10  end_lat             448824 non-null  float32
 11  end_lng             448824 non-null  float32
 12  member_casual       449123 non-null  object 
dtypes: float32(4), object(9)
memory usage: 37.7+ MB
The combined dataframe has 1.2 million records (1,213,447) and has 13 columns (attributes). Upon closer inspection it is observed
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 449123 entries, 0 to 449122
Data columns (total 13 columns):
 #   Column              Non-Null Count   Dtype         
---  ------              --------------   -----         
 0   ride_id             449123 non-null  object        
 1   rideable_type       449123 non-null  object        
 2   started_at          449123 non-null  datetime64[ns]
 3   ended_at            449123 non-null  datetime64[ns]
 4   start_station_name  387740 non-null  object        
 5   start_station_id    387608 non-null  object        
 6   end_station_name    383947 non-null  object        
 7   end_station_id      383806 non-null  object        
 8   start_lat           449123 non-null  float32       
 9   start_lng           449123 non-null  float32       
 10  end_lat             448824 non-null  float32       
 11  end_lng             448824 non-null  float32       
 12  member_casual       449123 non-null  object        
dtypes: datetime64[ns](2), float32(4), object(7)
memory usage: 37.7+ MB
Calculate % unique values per column
Calculate % missing values per column
Combine the dataframes duplicates and missing to get a clear picture of the data
column  unique_values  unique%  missing_values  missing%
0              ride_id        449123   100.00               0      0.00
1        rideable_type             3     0.00               0      0.00
2           started_at            59     0.01               0      0.00
3             ended_at            62     0.01               0      0.00
4   start_station_name          1060     0.24           61383     13.67
5     start_station_id          1032     0.23           61515     13.70
6     end_station_name          1066     0.24           65176     14.51
7       end_station_id          1038     0.23           65317     14.54
8            start_lat         21351     4.75               0      0.00
9            start_lng         11172     2.49               0      0.00
10             end_lat           761     0.17             299      0.07
11             end_lng           731     0.16             299      0.07
12       member_casual             2     0.00               0      0.00

checking if all the ride_id have exactly 16 charaters
True
count duplicates on unique column
this is done to find if there are any dupicate records present in the data as ride_id is the primary key, it is chosen
Total duplicates in ride_id column:  0
rideable_type, this column is like the ID badge for each bike, telling us what model was used for the ride. We've got three types
finding the unique values in rideable_type column
['classic_bike' 'electric_bike' 'docked_bike']
counting the number of records that contained docked_bike
The number of records having docked_bike type before: 5215
```

Data_curation_project | Projects | Branches | Notebook | Dataset | Caveats | Settings

```
The number of records having docked_bike type after the change: 0
checking for null values or empty values in rideable_type
Total number of empty values or null values in rideable_type :  0
We have already converted the started_at and ended_at columns to datetime format and hence we can proceed to find the duration of
*************************************Dealing with outliers ***********************************************
Count rows before filtering
First, create a DataFrame of outliers
Then, filter df to remove outliers
Count rows after filtering
Print the number of rows deleted
This change has resulted in deleting 447910 rows
Check the shape of df_duration_noise
*************************************start_station_name and end_station_name ***********************************************
We've noticed a bit of a puzzle: quite a few trips are missing either their starting or ending station names. Now, here's where t
*************************************Identify classic bike records missing station names ****************************************
*************************************Create df_station_noise DataFrame ***********************************************
*************************************Update the main DataFrame by removing these records ****************************************
*************************************storing the outliers or noise in a dataframe called df_noise, this is done to preserve the d
*************************************Shape of the outliers dataframe that has been isolated **************************************
*************************************checking member_casual column for possible values *******************************************
['member' 'casual']
*****************************************************************************************************************************
In our journey through the data, we've come across a neat little detail about the member_casual column. It turns out, it's pretty
********************************************************************************** ***********************************************
finding the number of missing values and their percentage after removing the outliers and noise
```

[9]

```sql
select count(*) as rides, member_casual from df
group by member_casual
```

Console | Timing | Datasets | Charts

temporary_dataset (2 rows)

Views

	rides (long)	member_casual (string)
0	105217	casual
1	343906	member

Connected to vizier @ http://localhost:5001/vizier-db/api/v1/