

```
In [3]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
In [5]: who_df["Country"].value_counts()
```

```
Out[5]: China                210
Thailand                201
Japan                  200
Republic of Korea      195
United States of America 194
...
Saint Pierre and Miquelon 115
Yemen                  113
Comoros                92
Tajikistan             92
Lesotho                 79
Name: Country, Length: 216, dtype: int64
```

```
In [4]: who_df = pd.read_csv("who.csv")
who_df["Date_reported"] = pd.to_datetime(who_df['Date_reported'])
who_df
```

```
Out[4]:
```

	Date_reported	Country	New_cases	Cumulative_cases	New_deaths	Cumulative_deaths
0	2020-04-01	China	1	1	0	0
1	2020-05-01	China	0	1	0	0
2	2020-06-01	China	3	4	0	0
3	2020-07-01	China	0	4	0	0
4	2020-08-01	China	0	4	0	0
...
31871	2020-07-31	Panama	1046	63269	25	1374
31872	2020-07-31	Timor-Leste	0	24	0	0
31873	2020-07-31	Guatemala	1221	48826	32	1867
31874	2020-07-31	Saint Vincent and the Grenadines	0	52	0	0
31875	2020-07-31	Democratic Republic of the Congo	79	9009	2	214

31876 rows × 6 columns

```
In [ ]: y = who_df["Country"].value_counts().sort_values(ascending=False).head(10)
y
```

```
In [ ]: date_cum = ["Date_reported", "Cumulative_cases"]
who_df[date_cum].value_counts().sort_values()
```

```
In [6]: italy = who_df["Country"]
italy_bool = italy == "Italy"
italia = who_df[italy_bool]
italia
```

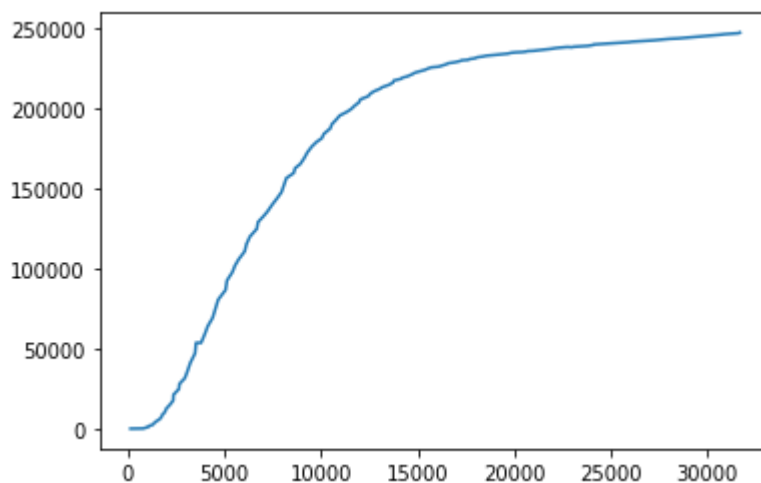
```
Out[6]:
```

	Date_reported	Country	New_cases	Cumulative_cases	New_deaths	Cumulative_deaths
123	2020-01-29	Italy	6	6	0	0
144	2020-01-30	Italy	0	6	0	0
158	2020-01-31	Italy	0	6	0	0
187	2020-01-02	Italy	0	6	0	0
215	2020-02-02	Italy	0	6	0	0
...
30870	2020-07-27	Italy	254	246118	5	35107
31097	2020-07-28	Italy	168	246286	5	35112
31333	2020-07-29	Italy	202	246488	11	35123
31629	2020-07-30	Italy	288	246776	6	35129
31686	2020-07-31	Italy	382	247158	3	35132

185 rows × 6 columns

```
In [264]: it = italia["Cumulative_cases"]
dt = pd.date_range(start = '2020-01-29', end = '2020-07-31')
plt.plot(dt, it)
```

```
Out[264]: [<matplotlib.lines.Line2D at 0x249c21c8>]
```



```
In [11]: india = who_df["Country"]
india_bool = india == "India"
indian = who_df[india_bool]
indian
```

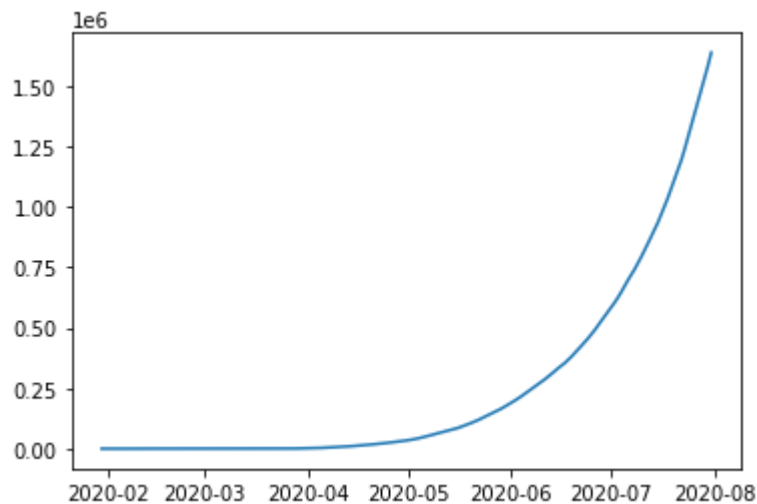
```
Out[11]:
```

	Date_reported	Country	New_cases	Cumulative_cases	New_deaths	Cumulative_deaths
136	2020-01-30	India	5	5	0	0
160	2020-01-31	India	0	5	0	0
178	2020-01-02	India	0	5	0	0
209	2020-02-02	India	1	6	0	0
243	2020-03-02	India	1	7	0	0
...
30900	2020-07-27	India	49931	1435453	708	32771
31200	2020-07-28	India	47703	1483156	654	33425
31441	2020-07-29	India	48513	1531669	768	34193
31624	2020-07-30	India	52123	1583792	775	34968
31812	2020-07-31	India	55078	1638870	779	35747

184 rows × 6 columns

```
In [13]: ind = indian["Cumulative_cases"]
dtm = pd.date_range(start = '2020-01-30', end = '2020-07-31')
plt.plot(dtm, ind)
```

```
Out[13]: [<matplotlib.lines.Line2D at 0x1e1ed708>]
```



```
In [ ]: feb = "2020-02-28"
march = "2020-03-30"
april = "2020-04-29"
may = "2020-05-30"
june = "2020-06-29"
july = "2020-07-30"
feb_july = (italia["Date_reported"] == feb) | (italia["Date_reported"] == march)
italia.loc[feb_july, "Cumulative_cases"]
```

```
In [ ]: italy_cum = italia.loc[feb_july]
italy_cum
```

```
In [ ]: plt.figure(figsize=(6, 4))
italian_cases = italy_cum["Cumulative_cases"]
italian_date = italy_cum["Date_reported"]
month_number = [1, 2, 3, 4, 5, 6, 7]
plt.plot(italian_date, italian_cases);
```

```
In [ ]: italy = who_df["Country"]
italy_bool = italy == "India"
india = who_df[italy_bool]
india
```

```
In [ ]: feb = "2020-02-29"
march = "2020-03-31"
april = "2020-04-30"
may = "2020-05-31"
june = "2020-06-30"
july = "2020-07-31"
feb_ju = (india["Date_reported"] == feb) | (india["Date_reported"] == march) | (i
feb_ju_df = india.loc[feb_ju]
```

```
In [ ]: plt.figure(figsize=(6, 4))
indiana_cases = feb_ju_df["Cumulative_cases"]
indiana_date = feb_ju_df["Date_reported"]
month_number = [1, 2, 3, 4, 5, 6, 7]
plt.plot(indiana_date, indiana_cases);
```

```
In [ ]: def plot_cumulative_cases(country_name):
    country = italy_cum[italy_cum['Country'] == country_name]
    plt.plot(country['Date_reported'], country['Cumulative_cases'])
    plt.title('{}: Cumulative Reported Cases'.format(country_name))
    plt.xlabel('Date')
    plt.ylabel('Number of Cases')
    plt.show()
plot_cumulative_cases("Italy")
```

```

In [ ]: france = who_df["Country"]
france_bool = france == "France"
franca = who_df[france_bool]
feb = "2020-02-29"
march = "2020-03-31"
april = "2020-04-30"
may = "2020-05-31"
june = "2020-06-30"
july = "2020-07-31"
french = (franca["Date_reported"] == feb) | (franca["Date_reported"] == march) |
franca_df = franca.loc[french]
plt.figure(figsize=(6, 4))
french_cases = franca_df["Cumulative_cases"]
french_date = franca_df["Date_reported"]
plt.plot(french_date, french_cases);

```

```

In [ ]: uk = who_df["Country"]
uk_bool = uk == "The United Kingdom"
England = who_df[uk_bool]
feb = "2020-02-29"
march = "2020-03-31"
april = "2020-04-30"
may = "2020-05-31"
june = "2020-06-30"
july = "2020-07-31"
Eng = (England["Date_reported"] == feb) | (England["Date_reported"] == march) |
England_df = England.loc[Eng]
plt.figure(figsize=(6, 4))
england_cases = England_df["Cumulative_cases"]
england_date = England_df["Date_reported"]
plt.plot(england_date, england_cases);

```

```

In [ ]: plt.figure(figsize=(6, 4))
england_cases = England_df["Cumulative_cases"]
england_date = England_df["Date_reported"]
french_cases = franca_df["Cumulative_cases"]
french_date = franca_df["Date_reported"]
plt.plot(england_date, england_cases)
plt.plot(french_date, french_cases)
plt.legend(["England", "France"])
plt.xlabel("years")
plt.ylabel("cases")
plt.title("England vs France cummulative cases");

```

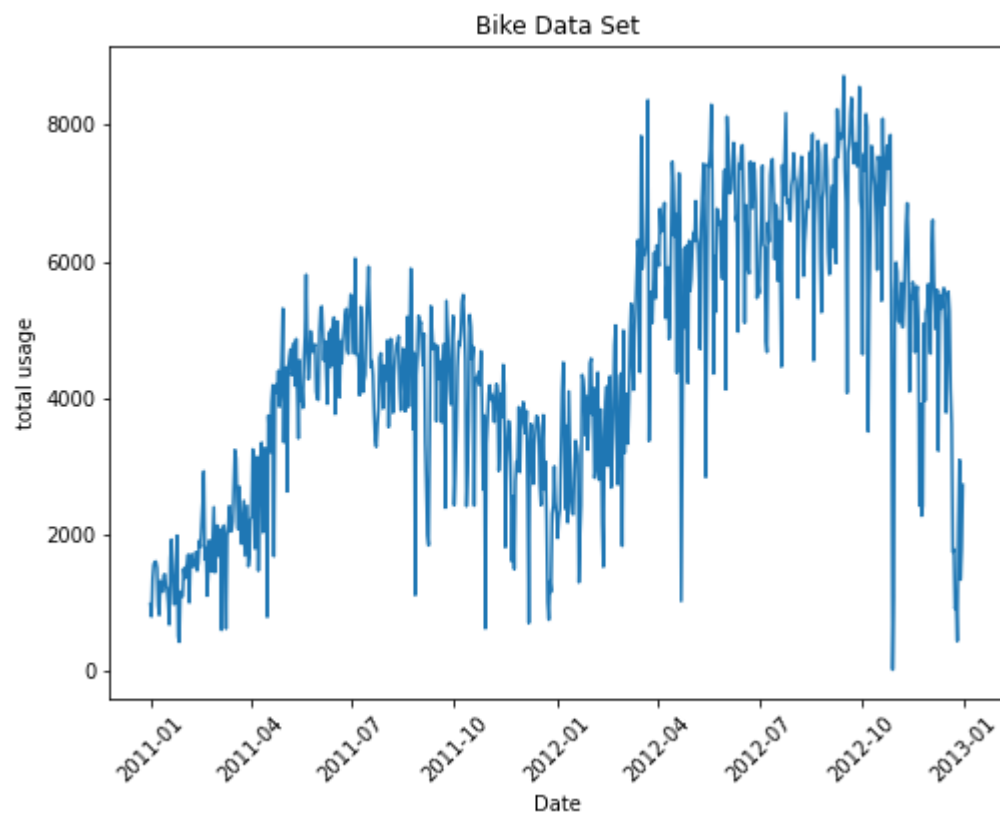
```

In [4]: bike_df = pd.read_csv("bike.csv")
bike_df["dteday"] = pd.to_datetime(bike_df["dteday"])
d_time = pd.date_range(start = '2011-01-01', end = '2012-12-31')

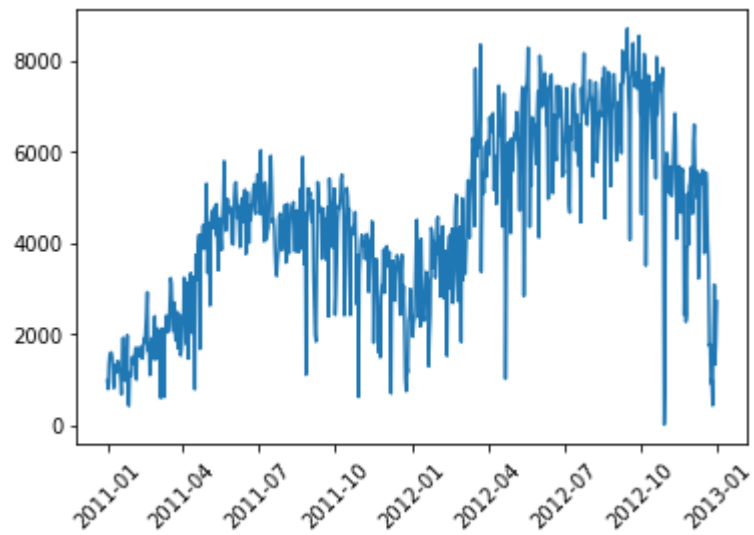
```

```
In [5]: total = bike_df["cnt"]  
plt.figure(figsize=(8, 6))  
plt.xticks(rotation=45)  
plt.title("Bike Data Set")  
plt.plot(d_time, total)  
plt.xlabel("Date")  
plt.ylabel("total usage")
```

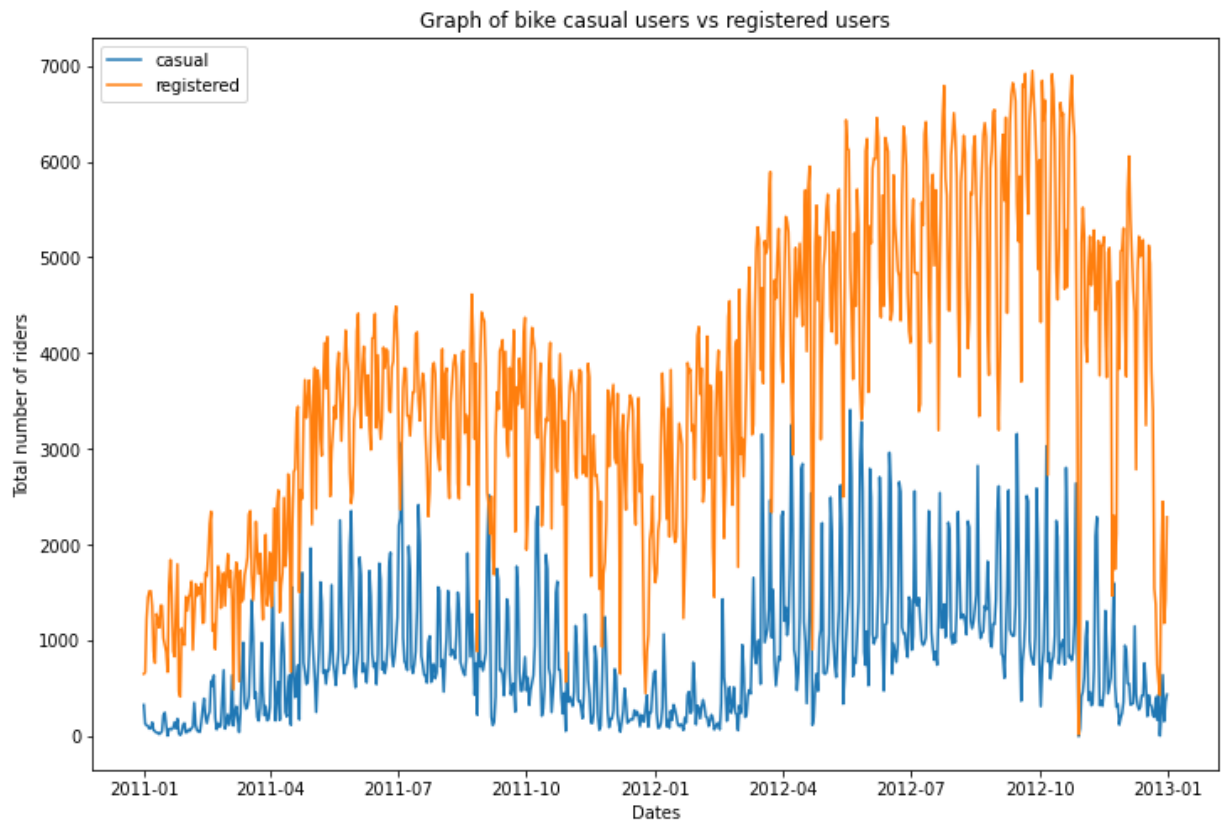
```
Out[5]: Text(0, 0.5, 'total usage')
```



```
In [67]: import matplotlib.pyplot as plt  
plt.plot(d_time, bike_df['cnt'])  
plt.xticks(rotation=45);
```



```
In [68]: casual = bike_df["casual"]
registered = bike_df["registered"]
date_range = pd.date_range(start="2011-01-01", end="2012-12-31")
plt.figure(figsize=(12, 8))
plt.plot(date_range, casual)
plt.plot(date_range, registered)
plt.xlabel("Dates")
plt.title("Graph of bike casual users vs registered users")
plt.ylabel("Total number of riders")
plt.legend(["casual", "registered"]);
```



In [69]:

bike_df

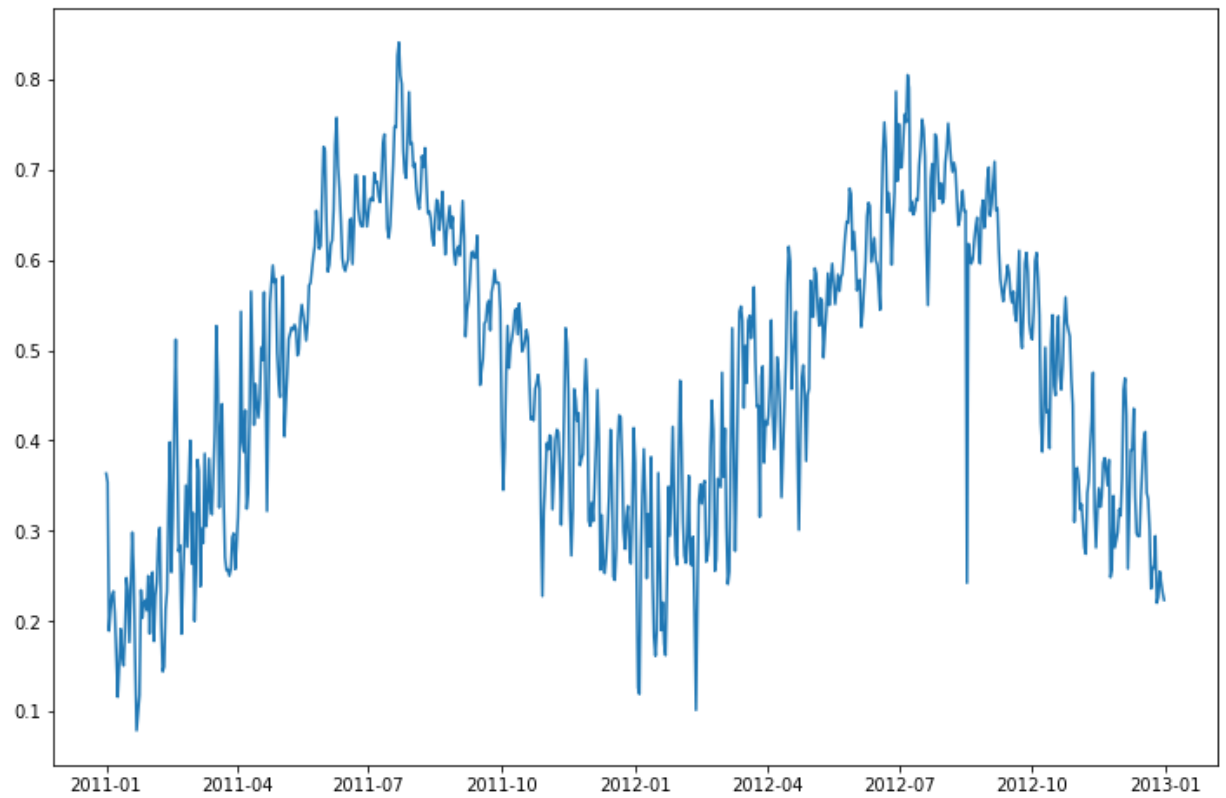
Out[69]:

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	at
0	1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.365
1	2	2011-02-01	1	0	1	0	0	0	2	0.363478	0.355
2	3	2011-03-01	1	0	1	0	1	1	1	0.196364	0.185
3	4	2011-04-01	1	0	1	0	2	1	1	0.200000	0.215
4	5	2011-05-01	1	0	1	0	3	1	1	0.226957	0.225
...
726	727	2012-12-27	1	1	12	0	4	1	2	0.254167	0.225
727	728	2012-12-28	1	1	12	0	5	1	2	0.253333	0.255
728	729	2012-12-29	1	1	12	0	6	0	2	0.253333	0.245
729	730	2012-12-30	1	1	12	0	0	0	1	0.255833	0.235
730	731	2012-12-31	1	1	12	0	1	1	2	0.215833	0.225

731 rows × 16 columns

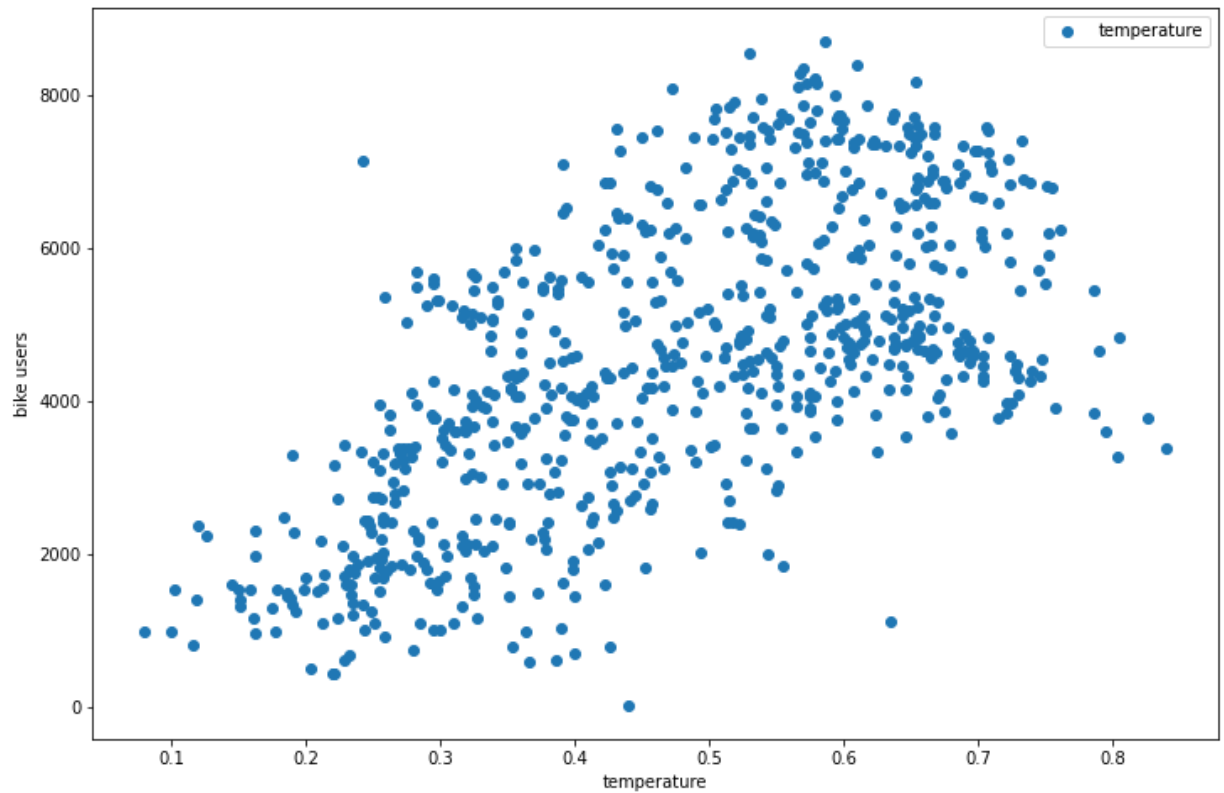


```
In [70]: temp = bike_df["atemp"]  
users = bike_df["cnt"]  
plt.figure(figsize=(12, 8))  
plt.plot(date_range, temp);
```



```
In [71]: temp = bike_df["atemp"]
users = bike_df["cnt"]
plt.figure(figsize=(12, 8))
plt.scatter(temp, users)
plt.xlabel("temperature")
plt.ylabel("bike users")
plt.legend(["temperature", "users"])
```

Out[71]: <matplotlib.legend.Legend at 0x22984588>

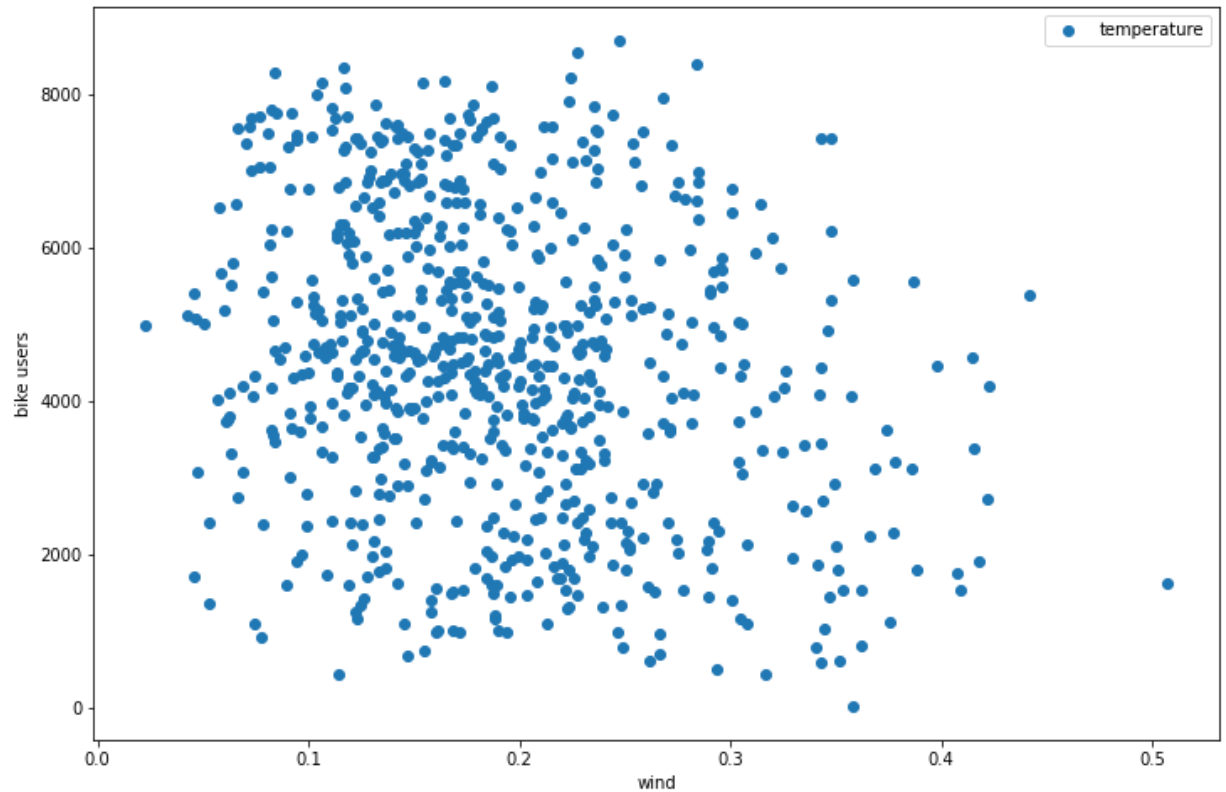


```
In [72]: bike_df["workingday"].corr(bike_df["registered"])
```

Out[72]: 0.30390711704591705

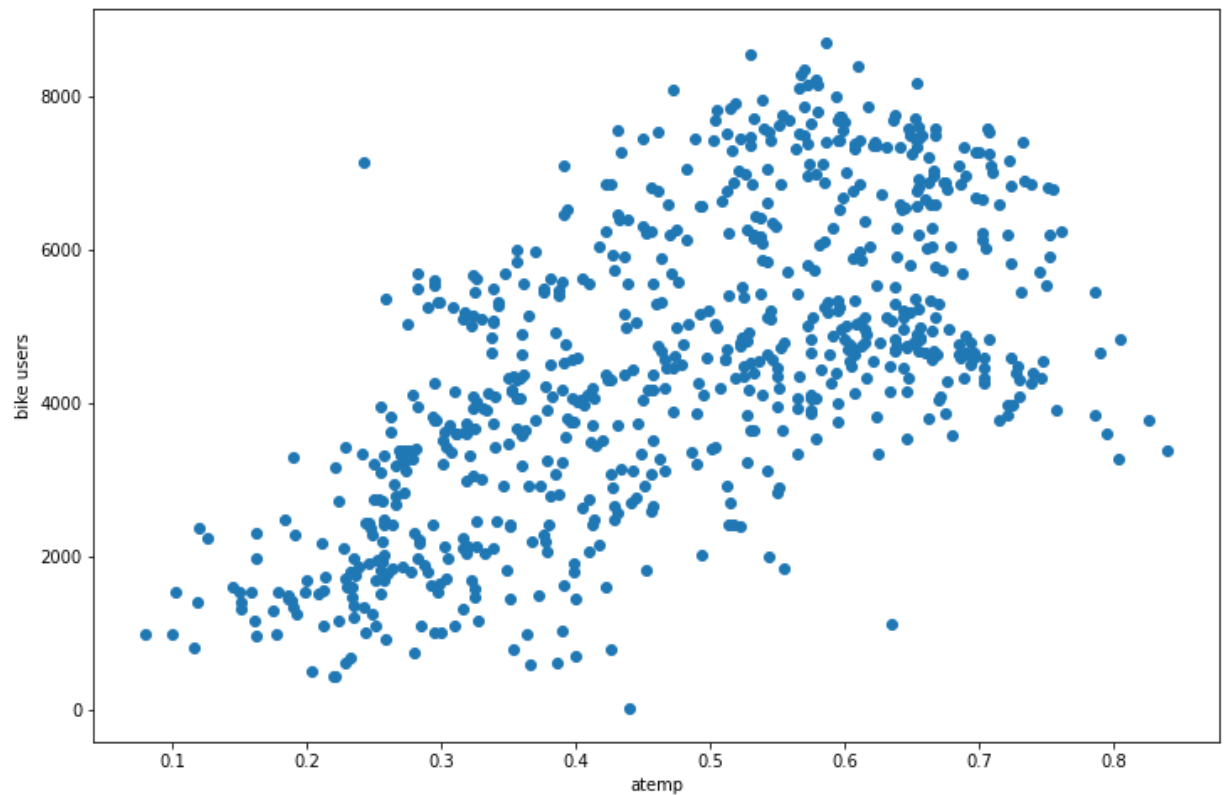
```
In [73]: wind = bike_df["windspeed"]
users = bike_df["cnt"]
plt.figure(figsize=(12, 8))
plt.scatter(wind, users)
plt.xlabel("wind")
plt.ylabel("bike users")
plt.legend(["temperature", "users"])
```

Out[73]: <matplotlib.legend.Legend at 0x22da3048>



```
In [74]: atemp = bike_df["atemp"]  
users = bike_df["cnt"]  
plt.figure(figsize=(12, 8))  
plt.scatter(atemp, users)  
plt.xlabel("atemp")  
plt.ylabel("bike users")
```

```
Out[74]: Text(0, 0.5, 'bike users')
```



```
In [75]: pearson = bike_df.corr()
pearson["temp_atemp"] = pearson["temp"] - pearson["atemp"]
pearson
```

```
Out[75]:
```

	instant	season	yr	mnth	holiday	weekday	workingday	weathers
instant	1.000000	0.412224	0.866025	0.496702	0.016145	-0.000016	-0.004337	-0.02147
season	0.412224	1.000000	-0.001844	0.831440	-0.010537	-0.003080	0.012485	0.01921
yr	0.866025	-0.001844	1.000000	-0.001792	0.007954	-0.005461	-0.002013	-0.04872
mnth	0.496702	0.831440	-0.001792	1.000000	0.019191	0.009509	-0.005901	0.04352
holiday	0.016145	-0.010537	0.007954	0.019191	1.000000	-0.101960	-0.253023	-0.03462
weekday	-0.000016	-0.003080	-0.005461	0.009509	-0.101960	1.000000	0.035790	0.03108
workingday	-0.004337	0.012485	-0.002013	-0.005901	-0.253023	0.035790	1.000000	0.06120
weathersit	-0.021477	0.019211	-0.048727	0.043528	-0.034627	0.031087	0.061200	1.00000
temp	0.150580	0.334315	0.047604	0.220205	-0.028556	-0.000170	0.052660	-0.12060
atemp	0.152638	0.342876	0.046106	0.227459	-0.032507	-0.007537	0.052182	-0.12158
hum	0.016375	0.205445	-0.110651	0.222204	-0.015937	-0.052232	0.024327	0.59104
windspeed	-0.112620	-0.229046	-0.011817	-0.207502	0.006292	0.014282	-0.018796	0.03951
casual	0.275255	0.210399	0.248546	0.123006	0.054274	0.059923	-0.518044	-0.24735
registered	0.659623	0.411623	0.594248	0.293488	-0.108745	0.057367	0.303907	-0.26038
cnt	0.628830	0.406100	0.566710	0.279977	-0.068348	0.067443	0.061156	-0.29739

In [76]:

bike_df

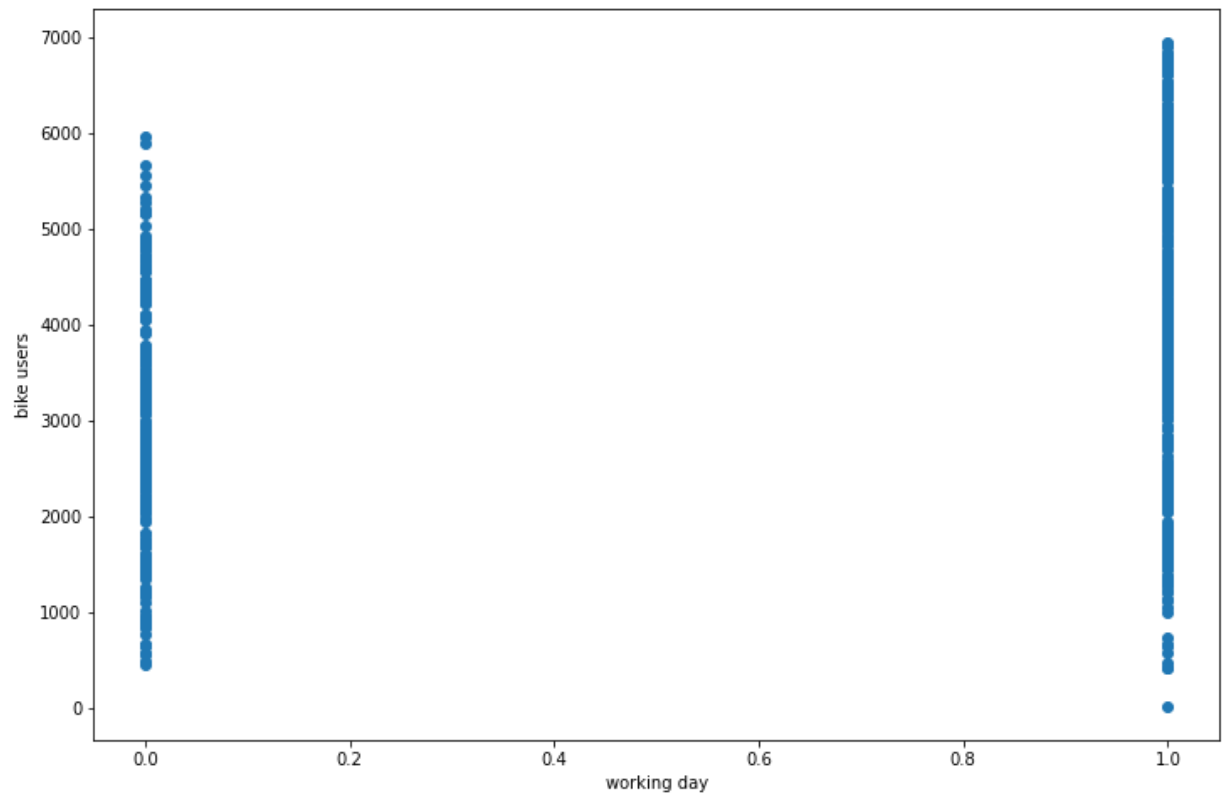
Out[76]:

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	at
0	1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.365
1	2	2011-02-01	1	0	1	0	0	0	2	0.363478	0.355
2	3	2011-03-01	1	0	1	0	1	1	1	0.196364	0.185
3	4	2011-04-01	1	0	1	0	2	1	1	0.200000	0.215
4	5	2011-05-01	1	0	1	0	3	1	1	0.226957	0.225
...
726	727	2012-12-27	1	1	12	0	4	1	2	0.254167	0.225
727	728	2012-12-28	1	1	12	0	5	1	2	0.253333	0.255
728	729	2012-12-29	1	1	12	0	6	0	2	0.253333	0.245
729	730	2012-12-30	1	1	12	0	0	0	1	0.255833	0.235
730	731	2012-12-31	1	1	12	0	1	1	2	0.215833	0.225

731 rows × 16 columns

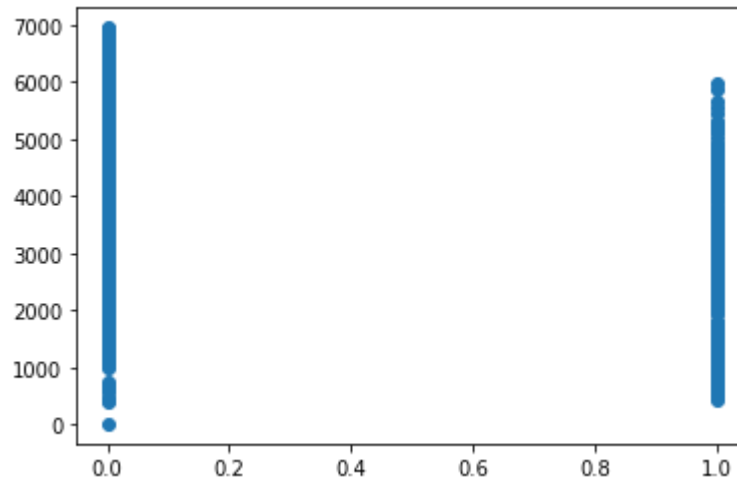


```
In [77]: wd = bike_df["workingday"]  
bk = bike_df["registered"]  
plt.figure(figsize=(12, 8))  
plt.scatter(wd, bk)  
plt.xlabel("working day")  
plt.ylabel("bike users");
```




```
In [84]: wday = bike_df["workingday"]  
casual = bike_df["casual"]  
reg = bike_df["registered"]  
plt.scatter(wday, reg)
```

Out[84]: <matplotlib.collections.PathCollection at 0x234fec48>



In [86]: bike_df

Out[86]:

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	at
0	1	2011-01-01	1	0	1	0	6	1	2	0.344167	0.36
1	2	2011-02-01	1	0	1	0	0	1	2	0.363478	0.35
2	3	2011-03-01	1	0	1	0	1	0	1	0.196364	0.18
3	4	2011-04-01	1	0	1	0	2	0	1	0.200000	0.21
4	5	2011-05-01	1	0	1	0	3	0	1	0.226957	0.22
...
726	727	2012-12-27	1	1	12	0	4	0	2	0.254167	0.22
727	728	2012-12-28	1	1	12	0	5	0	2	0.253333	0.25
728	729	2012-12-29	1	1	12	0	6	1	2	0.253333	0.24
729	730	2012-12-30	1	1	12	0	0	1	1	0.255833	0.23
730	731	2012-12-31	1	1	12	0	1	0	2	0.215833	0.22

731 rows × 16 columns



In [89]: lt = bike_df.shape[0]

In [90]: bike_df["casual"].sum() / lt

Out[90]: 848.1764705882352

In [91]: bike_df["registered"].sum() / lt

Out[91]: 3656.172366621067

In [111]: bike_df["workingday"].value_counts()

Out[111]: 1 500
0 231
Name: workingday, dtype: int64

```
In [112]: working_days = bike_df["workingday"]
working_days_bool = working_days == 1
casual_sum = bike_df.loc[working_days_bool, "casual"].sum()
casual_avg = casual_sum / 500
casual_avg
```

Out[112]: 606.57

```
In [122]: mf = bike_df["weekday"].value_counts().sort_index
mf
```

Out[122]: <bound method Series.sort_index of 6 105
0 105
1 105
2 104
3 104
4 104
5 104
Name: weekday, dtype: int64>

```
In [129]: week = bike_df["weekday"]
week_bool = week == 0
week_avg = bike_df.loc[week_bool, "casual"].mean()
week_avg
```

Out[129]: 1338.2952380952381

```
In [127]: wee = bike_df["weekday"]
wee_bool = wee == 1
wee_sum = bike_df.loc[wee_bool, "casual"].sum()
wee_avg = wee_sum / 105
wee_avg
```

Out[127]: 674.1333333333333

```
In [8]: weekday_averages = bike_df.groupby('weekday').mean()[['casual', 'registered']].re
```

```
In [9]: weekday_averages
```

Out[9]:

	weekday	casual	registered
0	0	1338.295238	2890.533333
1	1	674.133333	3663.990476
2	2	556.182692	3954.480769
3	3	551.144231	3997.394231
4	4	590.961538	4076.298077
5	5	752.288462	3938.000000
6	6	1465.257143	3085.285714

```
In [1]: wd_averages = bike_df.groupby('workingday').mean()[['casual', 'registered']]
```

```
-----
NameError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_8600\4268187822.py in <module>
----> 1 wd_averages = bike_df.groupby('workingday').mean()[['casual', 'registered']]

NameError: name 'bike_df' is not defined
```

```
In [149]: wd_averages
```

```
Out[149]:
```

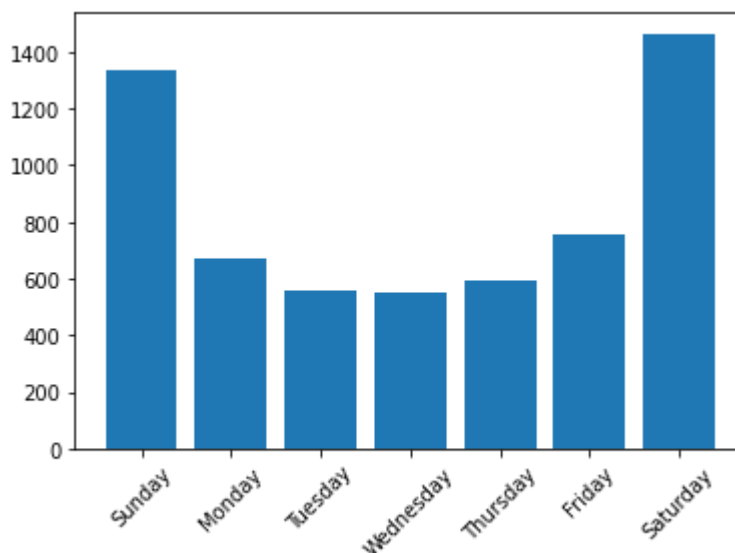
	casual	registered
workingday		
0	1371.134199	2959.034632
1	606.570000	3978.250000

```
In [166]: weekday_averages["casual"].index
```

```
Out[166]: RangeIndex(start=0, stop=7, step=1)
```

```
In [176]: axis = (weekday_averages["weekday"])
          axes = (weekday_averages["casual"])
```

```
In [189]: plt.bar(axis, axes)
          plt.xticks(ticks=[0, 1, 2, 3, 4, 5, 6], labels=["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"])
```



```
In [167]: list(weekday_averages["casual"])
```

```
Out[167]: [1338.2952380952381,  
674.1333333333333,  
556.1826923076923,  
551.1442307692307,  
590.9615384615385,  
752.2884615384615,  
1465.2571428571428]
```

```
In [172]: list(weekday_averages["weekday"])
```

```
Out[172]: [0, 1, 2, 3, 4, 5, 6]
```

```
In [190]: bike_df
```

```
Out[190]:
```

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	at
0	1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.36
1	2	2011-02-01	1	0	1	0	0	0	2	0.363478	0.35
2	3	2011-03-01	1	0	1	0	1	1	1	0.196364	0.18
3	4	2011-04-01	1	0	1	0	2	1	1	0.200000	0.21
4	5	2011-05-01	1	0	1	0	3	1	1	0.226957	0.22
...
726	727	2012-12-27	1	1	12	0	4	1	2	0.254167	0.22
727	728	2012-12-28	1	1	12	0	5	1	2	0.253333	0.25
728	729	2012-12-29	1	1	12	0	6	0	2	0.253333	0.24
729	730	2012-12-30	1	1	12	0	0	0	1	0.255833	0.23
730	731	2012-12-31	1	1	12	0	1	1	2	0.215833	0.22

731 rows × 16 columns



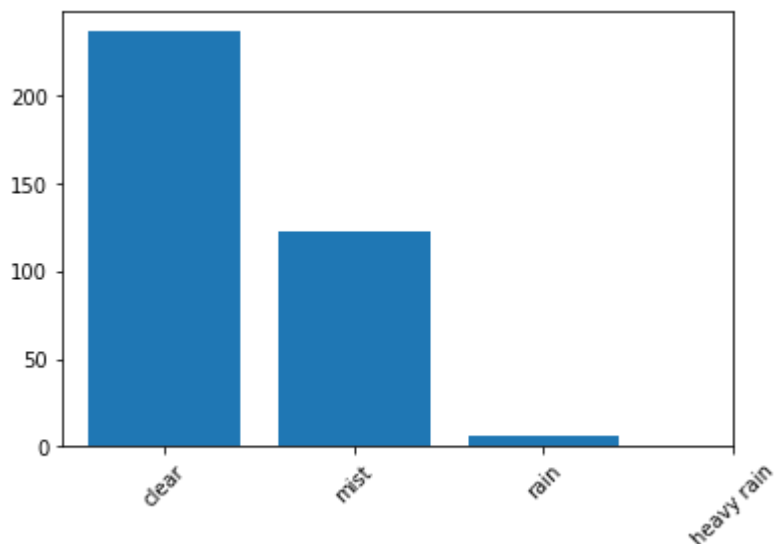
```
In [246]: start_date = '2012-01-01'
end_date = '2012-12-31'
# Select DataFrame rows between two dates
mask = (bike_df["dteday"] >= start_date) & (bike_df['dteday'] <= end_date)
year = bike_df.loc[mask]
weather = list(year["weathersit"].value_counts())
unique = list(year["weathersit"].unique())
height = year["weathersit"]
weather
```

Out[246]: [237, 123, 6]

```
In [216]: height
```

```
Out[216]: 365    1
366    1
367    1
368    2
369    1
..
726    2
727    2
728    2
729    1
730    2
Name: weathersit, Length: 366, dtype: int64
```

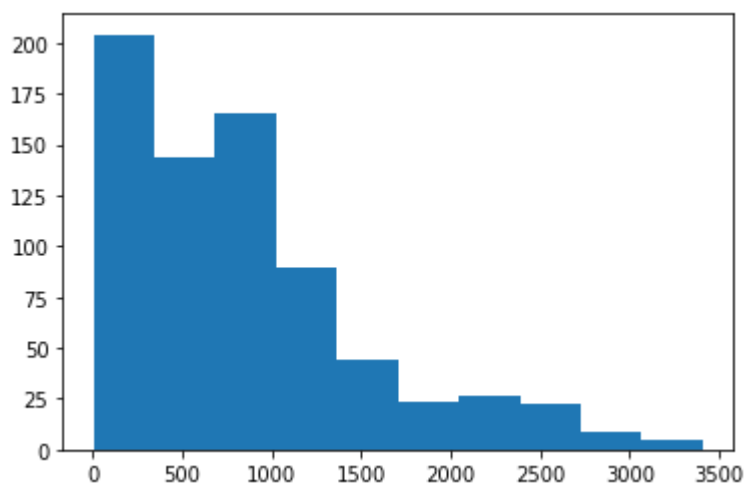
```
In [233]: plt.bar(unique, weather)
plt.xticks(ticks=[1, 2, 3, 4], labels=["clear", "mist", "rain", "heavy rain"], rot=45)
```



```
In [256]: fre = bike_df["casual"].value_counts(bins=10).sort_index()  
fre
```

```
Out[256]: (-1.4089999999999998, 342.8]    204  
(342.8, 683.6]    144  
(683.6, 1024.4]    165  
(1024.4, 1365.2]    89  
(1365.2, 1706.0]    44  
(1706.0, 2046.8]    23  
(2046.8, 2387.6]    26  
(2387.6, 2728.4]    22  
(2728.4, 3069.2]     9  
(3069.2, 3410.0]     5  
Name: casual, dtype: int64
```

```
In [265]: tot = bike_df["casual"]  
plt.hist(tot);
```



```
In [ ]:
```