RecSys / ara218 Private

Code Issues Pull requests Projects Wiki Security Insights Sett

main

ara218 / hw1.ipynb

 ara218 adding HW to repo History

1 contributor

1210 lines (1210 sloc) | 239 KB

In [2]:
```
import pandas as pd
import numpy as np
import seaborn as sns
```

In [3]:
```
data_directory = "C:\\Users\\alsae\\Desktop\\recmond\\ara218\\data\\"

bikes = pd.read_csv(f"{data_directory}day.csv")

bikes
```

Out[3]:

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	ter
0	1	2011-01-01	1	0	1	0	6	0	2	0.3441
1	2	2011-01-02	1	0	1	0	0	0	2	0.3634
2	3	2011-01-03	1	0	1	0	1	1	1	0.1963
3	4	2011-01-04	1	0	1	0	2	1	1	0.2000
4	5	2011-01-05	1	0	1	0	3	1	1	0.2269
...	...	...	...	...	...	...	...	...	...	
726	727	2012-12-27	1	1	12	0	4	1	2	0.2541
727	728	2012-12-28	1	1	12	0	5	1	2	0.2533
728	729	2012-12-29	1	1	12	0	6	0	2	0.2533
729	730	2012-12-30	1	1	12	0	0	0	1	0.2558
730	731	2012-12-31	1	1	12	0	1	1	2	0.2158

731 rows × 16 columns

In [4]:
```
first_rows = bikes.head()
first_rows
```

Out[4]:

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp
0	1	2011-01-01	1	0	1	0	6	0	2	0.344167
1	2	2011-01-02	1	0	1	0	0	0	2	0.363478

2	3	2011-01-03	1	0	1	0	1	1	1	0.196364
3	4	2011-01-04	1	0	1	0	2	1	1	0.200000
4	5	2011-01-05	1	0	1	0	3	1	1	0.226957

In [5]:
```
data_info = bikes.info()
data_info
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 731 entries, 0 to 730
Data columns (total 16 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   instant     731 non-null    int64  
 1   dteday      731 non-null    object 
 2   season      731 non-null    int64  
 3   yr          731 non-null    int64  
 4   mnth        731 non-null    int64  
 5   holiday     731 non-null    int64  
 6   weekday     731 non-null    int64  
 7   workingday  731 non-null    int64  
 8   weathersit  731 non-null    int64  
 9   temp        731 non-null    float64
 10  atemp       731 non-null    float64
 11  hum         731 non-null    float64
 12  windspeed   731 non-null    float64
 13  casual      731 non-null    int64  
 14  registered  731 non-null    int64  
 15  cnt         731 non-null    int64  
dtypes: float64(4), int64(11), object(1)
memory usage: 91.5+ KB
```

In [6]:
```
data_stats = bikes.describe()
data_stats
```

Out[6]:

	instant	season	yr	mnth	holiday	weekday	workingday
count	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000
mean	366.000000	2.496580	0.500684	6.519836	0.028728	2.997264	0.683995
std	211.165812	1.110807	0.500342	3.451913	0.167155	2.004787	0.465233
min	1.000000	1.000000	0.000000	1.000000	0.000000	0.000000	0.000000
25%	183.500000	2.000000	0.000000	4.000000	0.000000	1.000000	0.000000
50%	366.000000	3.000000	1.000000	7.000000	0.000000	3.000000	1.000000
75%	548.500000	3.000000	1.000000	10.000000	0.000000	5.000000	1.000000
max	731.000000	4.000000	1.000000	12.000000	1.000000	6.000000	1.000000

In [7]:
```
mean_value = data_stats["temp"]["mean"]
filtered_dataframe = bikes[bikes['temp'] > mean_value]
row_count = len(filtered_dataframe)
#row_count = filtered_dataframe.count()
print(row_count)
```

```
367
```

In [8]:
```
num_array = bikes.to_numpy()
print(num_array.shape)
print(num_array[99:105])
```

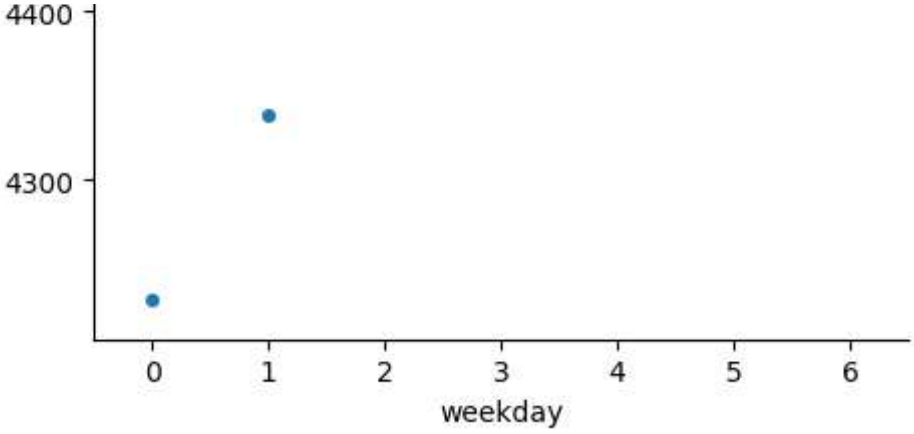
```
(731, 16)
[[100 '2011-04-10' 2 0 4 0 0 0 2 0.426667 0.426737 0.8575 0.146767 1188
  1707 2895]
 [101 '2011-04-11' 2 0 4 0 1 1 2 0.595652 0.565217 0.716956 0.324474 855
  2493 3348]
 [102 '2011-04-12' 2 0 4 0 2 1 2 0.5025 0.493054 0.739167 0.274879 257
  1777 2034]
 [103 '2011-04-13' 2 0 4 0 3 1 2 0.4125 0.417283 0.819167 0.250617 209
  1953 2162]
 [104 '2011-04-14' 2 0 4 0 4 1 1 0.4675 0.462742 0.540417 0.1107 529 2738
  3267]
 [105 '2011-04-15' 2 0 4 1 5 0 1 0.446667 0.441913 0.67125 0.226375 642
  2484 3126]]
```

In [9]:
```
new_array = num_array[:,9:13]
sorted_array = new_array[new_array[:,1].argsort()]
print(sorted_array[:5])
```

```
[[0.0591304 0.0790696 0.4 0.17197]
 [0.0965217 0.0988391 0.436522 0.2466]
 [0.1275 0.101658 0.464583 0.409212]
 [0.138333 0.116175 0.434167 0.36195]
 [0.0973913 0.11793 0.491739 0.15833]]
```

In [10]:
```
mean_riders_per_day = bikes.groupby("weekday")["cnt"].mean().reset_index()

mean_riders = sns.catplot(x="weekday", y="cnt", data=mean_riders_per_day)
```



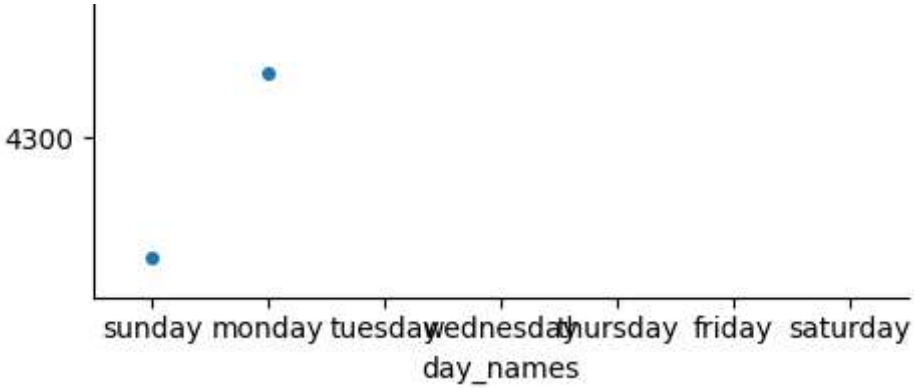
In [11]:
```
codes = pd.CategoricalDtype(categories=['sunday','monday','tuesday','wednesd
bikes['day_names'] = pd.Categorical.from_codes(bikes['weekday'], dtype=codes
bikes.head()
```

Out[11]:

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp
0	1	2011-01-01	1	0	1	0	6	0	2	0.344167
1	2	2011-01-02	1	0	1	0	0	0	2	0.363478
2	3	2011-01-03	1	0	1	0	1	1	1	0.196364
3	4	2011-01-04	1	0	1	0	2	1	1	0.200000
4	5	2011-01-05	1	0	1	0	3	1	1	0.226957

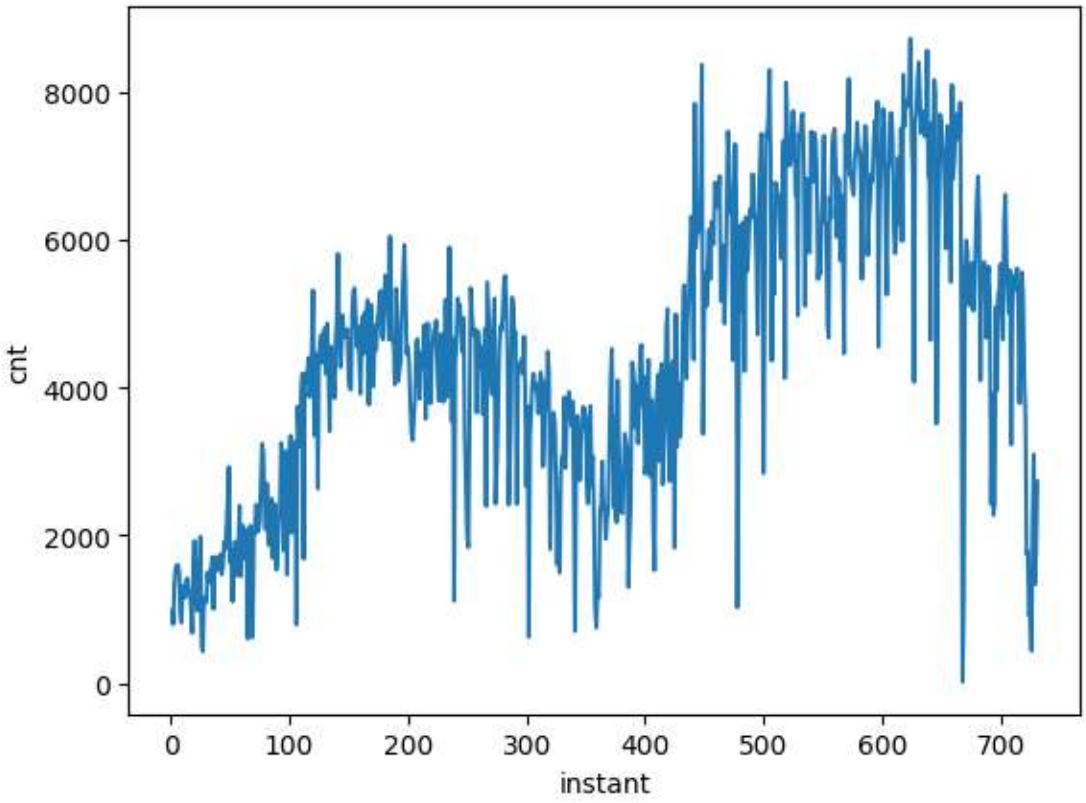
In [12]:
```
mean_riders_per_day = bikes.groupby("day_names")["cnt"].mean().reset_index()

mean_riders = sns.catplot(x="day_names", y="cnt", data=mean_riders_per_day)
```



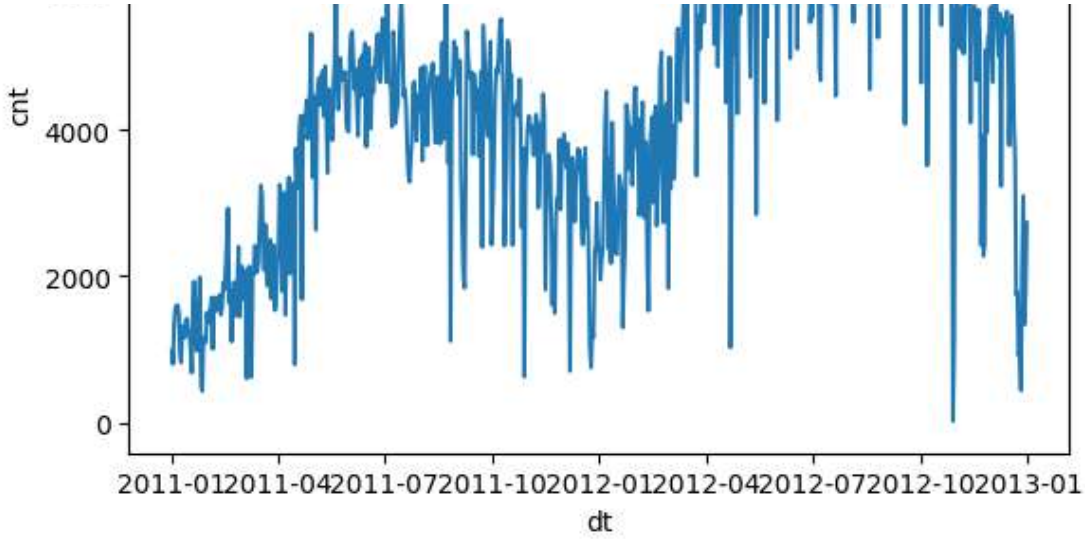
In [13]:
```
line_plot = sns.lineplot(data=bikes, x=bikes["instant"], y=bikes["cnt"])
line_plot
```

Out[13]: <Axes: xlabel='instant', ylabel='cnt'>



In [14]:
```
bikes['dt'] = pd.to_datetime(bikes['dteday'])
_plot = sns.lineplot(data=bikes, x=bikes["dt"], y=bikes["cnt"])
_plot
```

Out[14]: <Axes: xlabel='dt', ylabel='cnt'>



In [15]:
```
bikes.set_index('dt')['cnt'].resample('1W').sum().plot()
```

Out[15]: <Axes: xlabel='dt'>