Electric Vehicle Data

Objective: To create a dataframe with "Region" and "Number of Vehicles"

Workflow_1:

1. From "vehicle_reg", extract "County" column.
2. Use groupby and .agg, count, .sort_values() to create a new column in the dataframe which gives the # of EV registered in each county
3. Upload "region" data. Take only the "region" and "county". Convert all the "county" values to uppercase.
4. Merge "count_county" and "region". Output: a dataframe called "veh_count_reg" with "region" "county" and "number of vehicles" >>This method does not work, the table is riddled with NaN. **Question1: How might I be able to join this dataframe and get rid of NaN values?**
5. Create a new dataframe with "region" and "number of vehicles"
6. Use groupby and .agg, count, .sort_values() to create a new column in "veh_count_reg" which gives the # of EV registered in each region

Edit - I make some changes to this workflow as I work through the problem.
Workflow_2:

1. From "vehicle_reg", extract "County" column.
2. Upload "region" data. Take only the "region" and "county". Convert all the "county" values to uppercase.
3. Merge "County" and "region". Output: ideally, a dataframe called "count_reg" although at the moment, "Albany" occurs numerous times, but "Capital" occurs under region just once, and NaN occurs every time after that. **Question 2: how do I return a region for every row that county appears?**
4. Once I work out how to have a region for every county entry, I can do a groupby in a similar fashion to my method in Workflow_1, but this time with regions. Output: # of vehicles and regions

```
In [80]: %matplotlib inline
         import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt
         import seaborn as sns; sns.set()
         sns.set(style="darkgrid")
```

In [81]:
```
#upload data
vehicle_reg = pd.read_csv('Vehicle__Snowmobile__and_Boat_Registrations.c
sv')
vehicle_reg[:5]
```

Out[81]:

	Record Type	VIN	Registration Class	City	State	Zip	County	M
0	VEH	8995	PAS	BUFFALO	NY	14207	ERIE	1
1	VEH	607SR2131A	PAS	N SYRACUSE	NY	13212	ONONDAGA	1
2	VEH	537LS7D46CT083476	PAS	NEWFANE	NY	14108	NIAGARA	2
3	VEH	53TBH2MC1BE900166	PAS	ROCKVILLE CTR	NY	11570	NASSAU	2
4	VEH	53G1B4A47DB000347	PAS	BROOKLYN	NY	11223	KINGS	2

In [136]:
```
#Create a new df with only the counties for each registration
county1 = vehicle_reg[['County']]
county1[:5]
#rename so it is easier to join later
#county1.columns = ['County']
#county1[:5]
```

Out[136]:

	County
0	ERIE
1	ONONDAGA
2	NIAGARA
3	NASSAU
4	KINGS

In [137]:
```
#use groupby to organize the counties, then count how many times a couny
 appears in df
#Add a column for the number of times a county shows up in the df. This
 is the number of EV registered in the county
county['Number of Vehicles'] = ''
reg_county = county.groupby(['County'], as_index = False)
count_county = reg_county.agg({'Number of Vehicles':'count'}).sort_value
s(['Number of Vehicles','County'], ascending=[False, True])
count_county[:5]
```

Out[137]:

	County	Number of Vehicles
59	WESTCHESTER	1577
28	NASSAU	1486
51	SUFFOLK	1196
29	NEW YORK	875
40	QUEENS	456

In [138]:
```
#import Region data
from pandas import DataFrame
region = pd.read_excel(r'/Users/ceciliapershyn/CIS_512_Term_Project/coun
ty_to_region.xlsx')
region = DataFrame(region, columns=['Region_Code','Region','County'])
region[:5]
```

Out[138]:

	Region_Code	Region	County
0	1	Long Island	Nassau
1	1	Long Island	Suffolk
2	2	New York City	Kings
3	2	New York City	Bronx
4	2	New York City	New York

In [139]:
```
#convert County column to uppercase so it will match the County columns
 in other tables after join
region['County'] = region['County'].str.upper()
region.rename(index=str, columns={"County": "county"})
region[:5]
```

Out[139]:

	Region_Code	Region	County
0	1	Long Island	NASSAU
1	1	Long Island	SUFFOLK
2	2	New York City	KINGS
3	2	New York City	BRONX
4	2	New York City	NEW YORK

In [140]:
```
#remove Region_Code
region = region[['County','Region']]
region[:5]
```

Out[140]:

	County	Region
0	NASSAU	Long Island
1	SUFFOLK	Long Island
2	KINGS	New York City
3	BRONX	New York City
4	NEW YORK	New York City

In [141]:
```
#1st attempt to join
veh_count_reg = pd.merge_ordered(count_county, region, on='County', fill
_method=None, how='outer')
veh_count_reg

#Question1: How might I be able to join this dataframe and get rid of Na
N values?

#Edit: further down in the notebook I realize that it would be
#a better strategy to add the county to the region table before
#doing the group by.  This way, I can groupby the region
#and get the number of vehicles per region(see below for example)
```

Out[141]:

	County	Number of Vehicles	Region
0	ALBANY	NaN	Capital Region
1	ALBANY	226.0	NaN
2	ALLEGANY	NaN	Western New York
3	ALLEGANY	2.0	NaN
4	BRONX	NaN	New York City
5	BRONX	77.0	NaN
6	BROOME	NaN	Central New York
7	BROOME	45.0	NaN
8	CATTARAUGUS	NaN	Western New York
9	CATTARAUGUS	7.0	NaN
10	CAYUGA	NaN	Central New York
11	CAYUGA	13.0	NaN
12	CHAUTAUQUA	NaN	Western New York
13	CHAUTAUQUA	16.0	NaN
14	CHEMUNG	NaN	Western Finger Lakes
15	CHEMUNG	21.0	NaN
16	CHENANGO	NaN	Central New York
17	CHENANGO	7.0	NaN
18	CLINTON	NaN	Eastern Adirondacks
19	CLINTON	21.0	NaN
20	COLUMBIA	NaN	Capital Region
21	COLUMBIA	31.0	NaN
22	CORTLAND	NaN	Central New York
23	CORTLAND	7.0	NaN
24	DELAWARE	NaN	Capital Region
25	DELAWARE	9.0	NaN
26	DUTCHESS	NaN	Lower Hudson Valley
27	DUTCHESS	202.0	NaN
28	ERIE	NaN	Western New York
29	ERIE	283.0	NaN
...	...	...	...
94	SCHUYLER	NaN	Western Finger Lakes

	County	Number of Vehicles	Region
95	SCHUYLER	8.0	NaN
96	SENECA	NaN	Western Finger Lakes
97	SENECA	6.0	NaN
98	ST LAWRENCE	9.0	NaN
99	ST. LAWRENCE	NaN	Western Adirondacks
100	STEUBEN	NaN	Western Finger Lakes
101	STEUBEN	24.0	NaN
102	SUFFOLK	NaN	Long Island
103	SUFFOLK	1196.0	NaN
104	SULLIVAN	NaN	Lower Hudson Valley
105	SULLIVAN	21.0	NaN
106	TIOA	NaN	Central New York
107	TIOGA	11.0	NaN
108	TOMPKINS	NaN	Central New York
109	TOMPKINS	113.0	NaN
110	ULSTER	NaN	Lower Hudson Valley
111	ULSTER	156.0	NaN
112	WARREN	NaN	Eastern Adirondacks
113	WARREN	26.0	NaN
114	WASHINGTON	NaN	Eastern Adirondacks
115	WASHINGTON	15.0	NaN
116	WAYNE	NaN	Western Finger Lakes
117	WAYNE	15.0	NaN
118	WESTCHESTER	NaN	Lower Hudson Valley
119	WESTCHESTER	1577.0	NaN
120	WYOMING	NaN	Western New York
121	WYOMING	3.0	NaN
122	YATES	NaN	Western Finger Lakes
123	YATES	3.0	NaN

124 rows × 3 columns

In [142]:
```
#I'm creating a sample dataframe as an example of the output
#I am trying to achieve

sample = {'County':['Albany', 'Allegany', 'Columbia',  'Cattaragus', 'Cl
inton', 'Erie'],
          '# of Veh':[226, 2, 31, 7, 21, 283],
          'Region':['Capital', 'WNY', 'Capital', 'WNY', 'Eastern ADK',
'WNY' ]}
sample_veh_reg_count = DataFrame (sample, columns = ['County', '# of Ve
h', 'Region'])
sample_veh_reg_count
```

Out[142]:

	County	# of Veh	Region
0	Albany	226	Capital
1	Allegany	2	WNY
2	Columbia	31	Capital
3	Cattaragus	7	WNY
4	Clinton	21	Eastern ADK
5	Erie	283	WNY

In [143]:
```
#The next step is to add together all the vehicles in each region
sample_veh_reg = sample_veh_reg_count[['# of Veh', 'Region']]
sample_veh_reg
```

Out[143]:

	# of Veh	Region
0	226	Capital
1	2	WNY
2	31	Capital
3	7	WNY
4	21	Eastern ADK
5	283	WNY

In [144]:
```
sample_veh_reg = sample_veh_reg.groupby(['Region'], as_index = False)
sample_total_veh_region = sample_veh_reg.agg({'# of Veh':'count'}).sort_
values(['# of Veh','Region'], ascending=[False, True])
sample_total_veh_region[:5]

#So this is just showing the number of times the region shows
#up in the dataframe, which is not what I am looking for
```

Out[144]:

	Region	# of Veh
2	WNY	3
0	Capital	2
1	Eastern ADK	1

In [147]:
```
#Maybe I should try to add the county to the region table BEFORE
#doing the initial groupby (which produces the # of vehicles/county).
#This way, I can do the group by on regions instead of counties

veh_count_reg = pd.merge_ordered(county1, region, on='County', fill_meth
od=None, how='outer')
veh_count_reg

#The result is the region only shows up one time,
#the first time the county shows up in the dataframe
#How do i get it to match so that every time Albany appears in the
#dataframe, it is matched with "Capital Region"?
```

Out[147]:

	County	Region
0	ALBANY	Capital Region
1	ALBANY	NaN
2	ALBANY	NaN
3	ALBANY	NaN
4	ALBANY	NaN
5	ALBANY	NaN
6	ALBANY	NaN
7	ALBANY	NaN
8	ALBANY	NaN
9	ALBANY	NaN
10	ALBANY	NaN
11	ALBANY	NaN
12	ALBANY	NaN
13	ALBANY	NaN
14	ALBANY	NaN
15	ALBANY	NaN
16	ALBANY	NaN
17	ALBANY	NaN
18	ALBANY	NaN
19	ALBANY	NaN
20	ALBANY	NaN
21	ALBANY	NaN
22	ALBANY	NaN
23	ALBANY	NaN
24	ALBANY	NaN
25	ALBANY	NaN
26	ALBANY	NaN
27	ALBANY	NaN
28	ALBANY	NaN
29	ALBANY	NaN
...	...	...
9254	WESTCHESTER	NaN

	County	Region
9255	WESTCHESTER	NaN
9256	WESTCHESTER	NaN
9257	WESTCHESTER	NaN
9258	WESTCHESTER	NaN
9259	WESTCHESTER	NaN
9260	WESTCHESTER	NaN
9261	WESTCHESTER	NaN
9262	WESTCHESTER	NaN
9263	WESTCHESTER	NaN
9264	WESTCHESTER	NaN
9265	WESTCHESTER	NaN
9266	WESTCHESTER	NaN
9267	WESTCHESTER	NaN
9268	WESTCHESTER	NaN
9269	WESTCHESTER	NaN
9270	WESTCHESTER	NaN
9271	WESTCHESTER	NaN
9272	WESTCHESTER	NaN
9273	WESTCHESTER	NaN
9274	WESTCHESTER	NaN
9275	WESTCHESTER	NaN
9276	WYOMING	Western New York
9277	WYOMING	NaN
9278	WYOMING	NaN
9279	WYOMING	NaN
9280	YATES	Western Finger Lakes
9281	YATES	NaN
9282	YATES	NaN
9283	YATES	NaN

9284 rows × 2 columns

In [148]:
```
#This is another sample dataset to show my new desired outcome

sample2 = {'County':['Albany', 'Albany', 'Albany',  'Cattaragus', 'Erie'
, 'Clinton', 'Dutchess', 'Dutchess'],
          'Region':['Capital', 'Capital', 'Capital', 'WNY', 'WNY', 'East
ern ADK', 'LHV', 'LHV'  ]}
sample_county_reg = DataFrame (sample2, columns = ['County', 'Region'])
sample_county_reg
```

Out[148]:

	County	Region
0	Albany	Capital
1	Albany	Capital
2	Albany	Capital
3	Cattaragus	WNY
4	Erie	WNY
5	Clinton	Eastern ADK
6	Dutchess	LHV
7	Dutchess	LHV

In [149]:
```
reg = sample_county_reg[['Region']]
reg['# of Veh'] = ''

sample_county_reg = reg.groupby(['Region'], as_index = False)
sample_total_county_reg = sample_county_reg.agg({'# of Veh':'count'}).so
rt_values(['# of Veh','Region'], ascending=[False, True])
sample_total_county_reg

#The below table is my desired outcome for this portion of the project
```

Out[149]:

	Region	# of Veh
0	Capital	3
2	LHV	2
3	WNY	2
1	Eastern ADK	1

In [150]:
```
#The above method would work, if I could figure out a way to give
#each county its corresponding region in the data table above
#I am thinking a iterative loop might be the best option?
```

Charging Hubs

In [151]:
```
charge_hub = pd.read_csv('Electric_Vehicle_Charging_Stations_in_New_Yor
k.csv')
charge_hub.head()
```

Out[151]:

	Fuel Type Code	Station Name	Street Address	Intersection Directions	City	State	ZIP	Plus4	Station Phone	St C
0	ELEC	Hudson Valley Community College - TEC-SMART Bu...	345 Hermes Rd	NaN	Malta	NY	12020	NaN	518-629-7075	E
1	ELEC	EDISONPARKFAST	451 9th Ave	LOC #250 #2 LOT#250; ChargePoint America Program	New York	NY	10018	NaN	888-758-4389	E
2	ELEC	CARCHARGING	350 W 50th St	ICON MERCURY; Icon parking see attendant for a...	New York	NY	10019	NaN	888-758-4389	E
3	ELEC	CARCHARGING	310 W 39th St	ICON 310 W 39TH; Icon Parking systems see vale...	New York	NY	10018	NaN	888-758-4389	E
4	ELEC	EDISONPARKFAST	50 W 44th St	LOC #100 LEVEL3; Located in basement level of ...	New York	NY	10036	NaN	888-758-4389	E

5 rows × 32 columns

In [152]:
```
Zip = charge_hub[['ZIP']]
Zip[:4]
```

Out[152]:

	ZIP
0	12020
1	10018
2	10019
3	10018

In [153]:
```
#upload zip/county cross reference data from NYS open data

upload = pd.read_csv('zip_county_cross_reference.csv')
upload[:5]
```

Out[153]:

	County Name	State FIPS	County Code	County FIPS	ZIP Code	File Date
0	Albany	36	1	36001	12046	07/25/2007
1	Albany	36	1	36001	12083	07/25/2007
2	Albany	36	1	36001	12085	07/25/2007
3	Albany	36	1	36001	12201	07/25/2007
4	Albany	36	1	36001	12203	07/25/2007

In [154]:
```
zip_county = upload[['County Name', 'ZIP Code']]
zip_county[:5]
```

Out[154]:

	County Name	ZIP Code
0	Albany	12046
1	Albany	12083
2	Albany	12085
3	Albany	12201
4	Albany	12203

In [155]:
```
#join zip_county and Zip.  Output will be a datatable that will
#give county values to the zip codes where charge hubs are located

zip_co = Zip.merge(zip_county, how='inner', left_on='ZIP', right_on='ZIP
 Code', validate='many_to_many')
zip_co
```

Out[155]:

	ZIP	County Name	ZIP Code
0	12020	Saratoga	12020
1	10018	New York	10018
2	10018	New York	10018
3	10018	New York	10018
4	10018	New York	10018
5	10018	New York	10018
6	10019	New York	10019
7	10019	New York	10019
8	10019	New York	10019
9	10019	New York	10019
10	10019	New York	10019
11	10019	New York	10019
12	10019	New York	10019
13	10019	New York	10019
14	10019	New York	10019
15	10019	New York	10019
16	10019	New York	10019
17	10019	New York	10019
18	10019	New York	10019
19	10019	New York	10019
20	10019	New York	10019
21	10019	New York	10019
22	10019	New York	10019
23	10019	New York	10019
24	10019	New York	10019
25	10019	New York	10019
26	10019	New York	10019
27	10019	New York	10019
28	10019	New York	10019
29	10019	New York	10019
...	...	...	...
1114	13088	Onondaga	13088

	ZIP	County Name	ZIP Code
1115	13088	Onondaga	13088
1116	13088	Onondaga	13088
1117	12075	Columbia	12075
1118	12075	Columbia	12075
1119	12045	Albany	12045
1120	13669	St. Lawrence	13669
1121	11706	Suffolk	11706
1122	10523	Westchester	10523
1123	12043	Otsego	12043
1124	12043	Schoharie	12043
1125	12514	Dutchess	12514
1126	14870	Schuyler	14870
1127	14870	Steuben	14870
1128	11803	Nassau	11803
1129	11581	Nassau	11581
1130	12047	Albany	12047
1131	14760	Cattaraugus	14760
1132	11355	Queens	11355
1133	11978	Suffolk	11978
1134	14485	Livingston	14485
1135	14485	Ontario	14485
1136	11378	Queens	11378
1137	12498	Ulster	12498
1138	12498	Ulster	12498
1139	11768	Suffolk	11768
1140	13320	Otsego	13320
1141	13320	Schoharie	13320
1142	13320	Montgomery	13320
1143	14086	Erie	14086

1144 rows × 3 columns

In [156]:
```
county_hub = zip_co[['County Name']]
county_hub['County Name'] = county_hub['County Name'].str.upper()
county_hub[:5]
```

```
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:2: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-d
ocs/stable/indexing.html#indexing-view-versus-copy
```

Out[156]:

	County Name
0	SARATOGA
1	NEW YORK
2	NEW YORK
3	NEW YORK
4	NEW YORK

In [157]:
```
county_reg = county_hub.merge(region, how='inner', left_on='County Name'
, right_on='County', validate='many_to_many')
county_reg
```

Out[157]:

	County Name	County	Region
0	SARATOGA	SARATOGA	Eastern Adirondacks
1	SARATOGA	SARATOGA	Eastern Adirondacks
2	SARATOGA	SARATOGA	Eastern Adirondacks
3	SARATOGA	SARATOGA	Eastern Adirondacks
4	SARATOGA	SARATOGA	Eastern Adirondacks
5	SARATOGA	SARATOGA	Eastern Adirondacks
6	SARATOGA	SARATOGA	Eastern Adirondacks
7	SARATOGA	SARATOGA	Eastern Adirondacks
8	SARATOGA	SARATOGA	Eastern Adirondacks
9	SARATOGA	SARATOGA	Eastern Adirondacks
10	SARATOGA	SARATOGA	Eastern Adirondacks
11	SARATOGA	SARATOGA	Eastern Adirondacks
12	SARATOGA	SARATOGA	Eastern Adirondacks
13	SARATOGA	SARATOGA	Eastern Adirondacks
14	SARATOGA	SARATOGA	Eastern Adirondacks
15	SARATOGA	SARATOGA	Eastern Adirondacks
16	SARATOGA	SARATOGA	Eastern Adirondacks
17	SARATOGA	SARATOGA	Eastern Adirondacks
18	SARATOGA	SARATOGA	Eastern Adirondacks
19	SARATOGA	SARATOGA	Eastern Adirondacks
20	SARATOGA	SARATOGA	Eastern Adirondacks
21	SARATOGA	SARATOGA	Eastern Adirondacks
22	SARATOGA	SARATOGA	Eastern Adirondacks
23	SARATOGA	SARATOGA	Eastern Adirondacks
24	SARATOGA	SARATOGA	Eastern Adirondacks
25	SARATOGA	SARATOGA	Eastern Adirondacks
26	NEW YORK	NEW YORK	New York City
27	NEW YORK	NEW YORK	New York City
28	NEW YORK	NEW YORK	New York City
29	NEW YORK	NEW YORK	New York City
...	...	...	...
1112	CORTLAND	CORTLAND	Central New York

	County Name	County	Region
1113	PUTNAM	PUTNAM	Lower Hudson Valley
1114	PUTNAM	PUTNAM	Lower Hudson Valley
1115	GREENE	GREENE	Capital Region
1116	GREENE	GREENE	Capital Region
1117	GREENE	GREENE	Capital Region
1118	GREENE	GREENE	Capital Region
1119	GENESEE	GENESEE	Western Finger Lakes
1120	GENESEE	GENESEE	Western Finger Lakes
1121	GENESEE	GENESEE	Western Finger Lakes
1122	WYOMING	WYOMING	Western New York
1123	WYOMING	WYOMING	Western New York
1124	ORLEANS	ORLEANS	Western Finger Lakes
1125	ORLEANS	ORLEANS	Western Finger Lakes
1126	CATTARAUGUS	CATTARAUGUS	Western New York
1127	CATTARAUGUS	CATTARAUGUS	Western New York
1128	LIVINGSTON	LIVINGSTON	Western Finger Lakes
1129	LIVINGSTON	LIVINGSTON	Western Finger Lakes
1130	LIVINGSTON	LIVINGSTON	Western Finger Lakes
1131	LIVINGSTON	LIVINGSTON	Western Finger Lakes
1132	HERKIMER	HERKIMER	Western Adirondacks
1133	HERKIMER	HERKIMER	Western Adirondacks
1134	HERKIMER	HERKIMER	Western Adirondacks
1135	HERKIMER	HERKIMER	Western Adirondacks
1136	HERKIMER	HERKIMER	Western Adirondacks
1137	HERKIMER	HERKIMER	Western Adirondacks
1138	OSWEGO	OSWEGO	Central New York
1139	SCHOHARIE	SCHOHARIE	Capital Region
1140	SCHOHARIE	SCHOHARIE	Capital Region
1141	SCHOHARIE	SCHOHARIE	Capital Region

1142 rows × 3 columns

In [158]:
```
reg_count_hub1 = county_reg[['Region']]
reg_count_hub1[:5]
```

Out[158]:

	Region
0	Eastern Adirondacks
1	Eastern Adirondacks
2	Eastern Adirondacks
3	Eastern Adirondacks
4	Eastern Adirondacks

In [159]:
```
reg_count_hub1['Number of Hubs'] = ''
reg_count_hub2 = reg_count_hub1.groupby(['Region'], as_index = False)
reg_count_hub = reg_count_hub2.agg({'Number of Hubs':'count'}).sort_valu
es(['Number of Hubs','Region'], ascending=[False, True])
reg_count_hub
```

```
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:1: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-d
ocs/stable/indexing.html#indexing-view-versus-copy
  """Entry point for launching an IPython kernel.
```

Out[159]:

	Region	Number of Hubs
5	New York City	397
4	Lower Hudson Valley	161
0	Capital Region	127
3	Long Island	115
7	Western Finger Lakes	96
1	Central New York	84
2	Eastern Adirondacks	67
8	Western New York	62
6	Western Adirondacks	33

In [160]:
```
#wondering if I can use the same
#county_reg = county_hub.merge(region, how='inner', left_on='County Nam
e', right_on='County', validate='many_to_many')
#code to merge county and region for the vehicle data?
```

In [161]:
```
#Question3: Any ideas why this code would turn up an empty dataframe?
v_c_reg = county1.merge(region, how='inner', left_on='County', right_on=
'County', validate='many_to_many')
v_c_reg
```

Out[161]:

	County	Region