Subcategory Analysis

Doris Pira

Objective

- Determine how subcategory spend was trending over a period of 16 days (3/2/19-3/17/19)

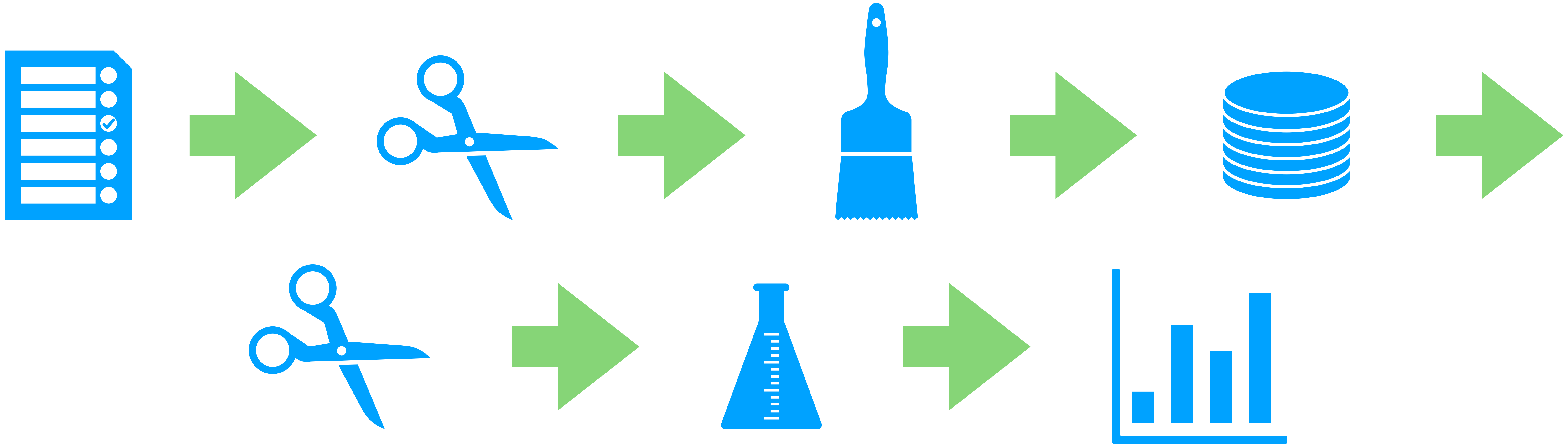
- High level view of each subcategory spend by day

- Look at a granular view of subcategory spend numbers by day

- High level view of spend vs. conversion rates

- Dig deeper to compare conversion rates by Subcategory

Process



Tools

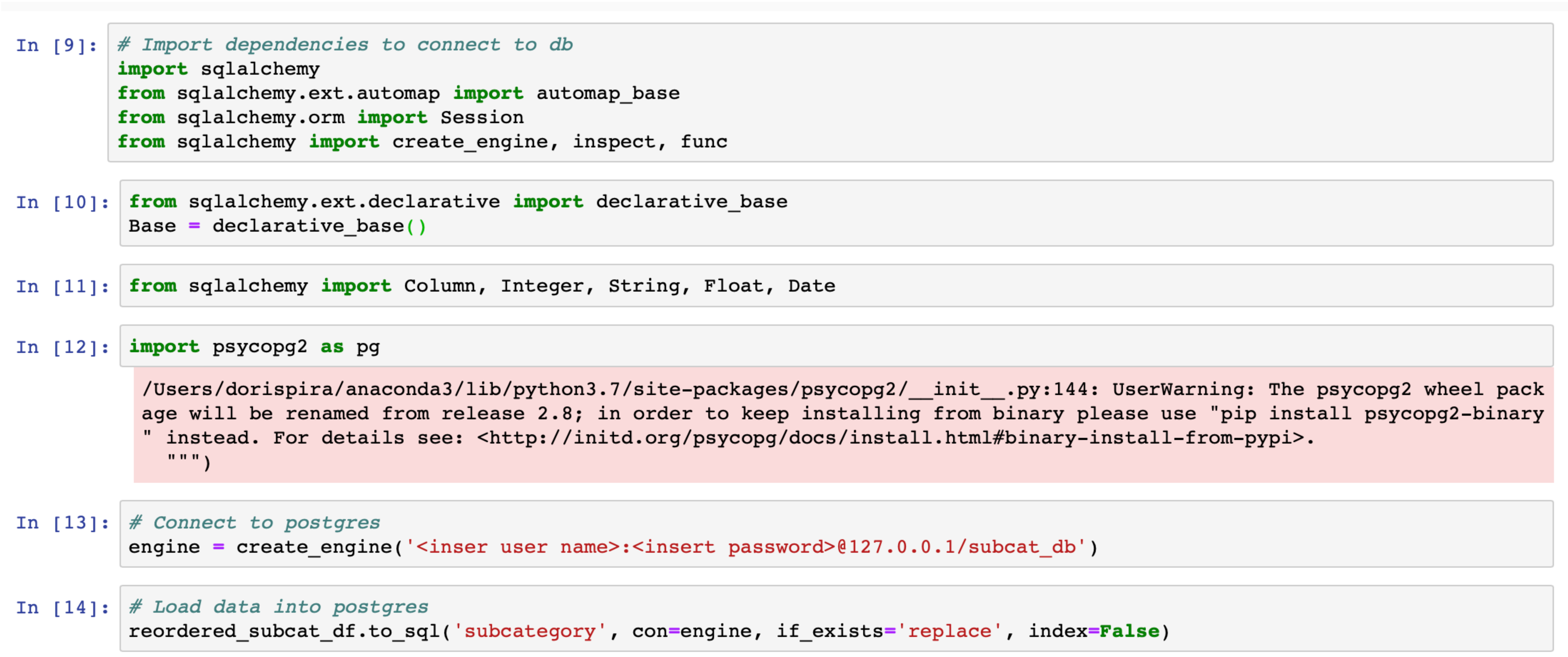
- Data: Dummified CSV
- Libraries: Pandas, NumPy, Matplotlib, SQLAlchemy, CSV
- Database: Postgres
- Tableau was utilized to create additional visuals

Roadblocks

- Loading data from Jupyter Notebook to Postgres
- Casting the date correctly as DATE type during the loading and extracting process
- Grouping the data vs Pivoting

Snippets

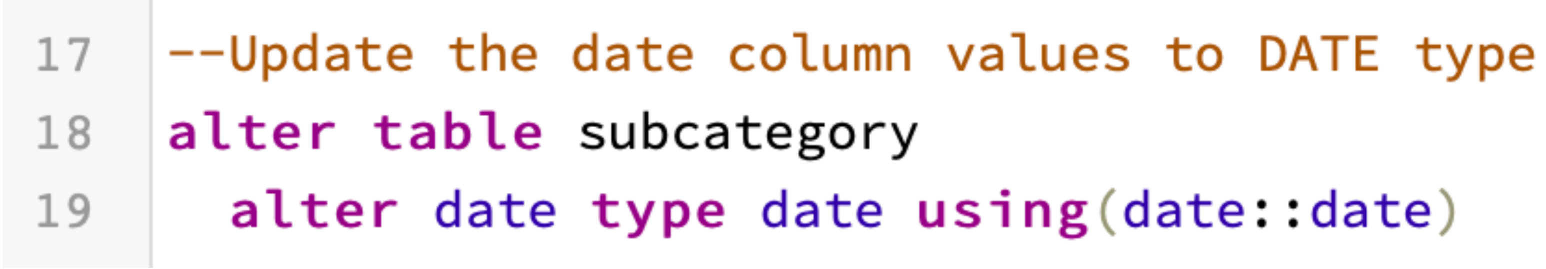
At first I tried to bulk insert the data as CSV using StringIO. Then I was able to successfully load it directly to Postgres using SQLAlchemy.



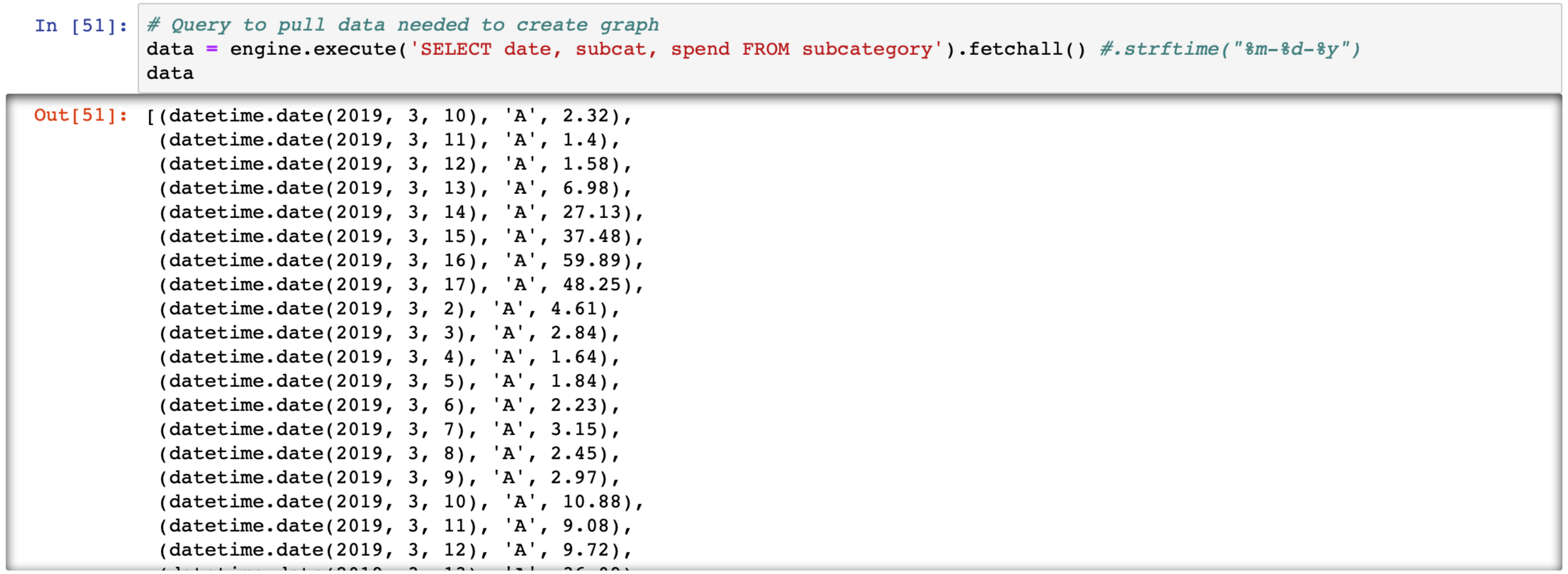
```
In [9]:  # Import dependencies to connect to db
         import sqlalchemy
         from sqlalchemy.ext.automap import automap_base
         from sqlalchemy.orm import Session
         from sqlalchemy import create_engine, inspect, func

In [10]: from sqlalchemy.ext.declarative import declarative_base
         Base = declarative_base()

In [11]: from sqlalchemy import Column, Integer, String, Float, Date

In [12]: import psycopg2 as pg

          /Users/dorispira/anaconda3/lib/python3.7/site-packages/psycopg2/__init__.py:144: UserWarning: The psycopg2 wheel pack
          age will be renamed from release 2.8; in order to keep installing from binary please use "pip install psycopg2-binary
          " instead. For details see: <http://initd.org/psycopg/docs/install.html#binary-install-from-pypi>.
            """)

In [13]: # Connect to postgres
         engine = create_engine('<inser user name>:<insert password>@127.0.0.1/subcat_db')

In [14]: # Load data into postgres
         reordered_subcat_df.to_sql('subcategory', con=engine, if_exists='replace', index=False)
```

Snippets

After loading my data into Postgres, my date values became text in the db. I had to convert the date values back to DATE type within Postgres, then I was able to query the data in the correct format.



```sql
--Update the date column values to DATE type
alter table subcategory
  alter date type date using(date::date)
```

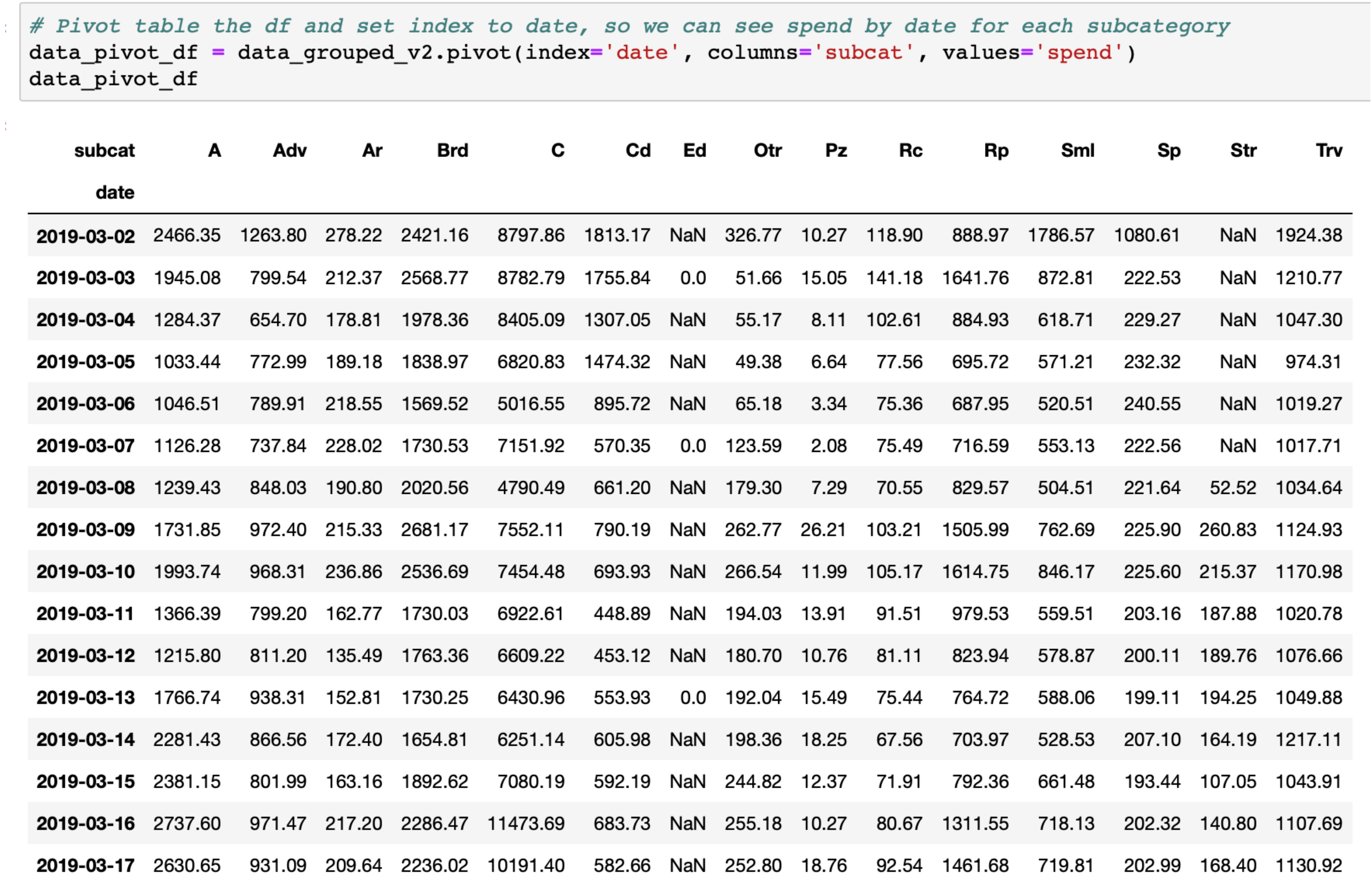


```
In [51]: # Query to pull data needed to create graph
         data = engine.execute('SELECT date, subcat, spend FROM subcategory').fetchall() #.strftime("%m-%d-%y")
         data
```

```
Out[51]: [(datetime.date(2019, 3, 10), 'A', 2.32),
          (datetime.date(2019, 3, 11), 'A', 1.4),
          (datetime.date(2019, 3, 12), 'A', 1.58),
          (datetime.date(2019, 3, 13), 'A', 6.98),
          (datetime.date(2019, 3, 14), 'A', 27.13),
          (datetime.date(2019, 3, 15), 'A', 37.48),
          (datetime.date(2019, 3, 16), 'A', 59.89),
          (datetime.date(2019, 3, 17), 'A', 48.25),
          (datetime.date(2019, 3, 2), 'A', 4.61),
          (datetime.date(2019, 3, 3), 'A', 2.84),
          (datetime.date(2019, 3, 4), 'A', 1.64),
          (datetime.date(2019, 3, 5), 'A', 1.84),
          (datetime.date(2019, 3, 6), 'A', 2.23),
          (datetime.date(2019, 3, 7), 'A', 3.15),
          (datetime.date(2019, 3, 8), 'A', 2.45),
          (datetime.date(2019, 3, 9), 'A', 2.97),
          (datetime.date(2019, 3, 10), 'A', 10.88),
          (datetime.date(2019, 3, 11), 'A', 9.08),
          (datetime.date(2019, 3, 12), 'A', 9.72),
```

Snippets

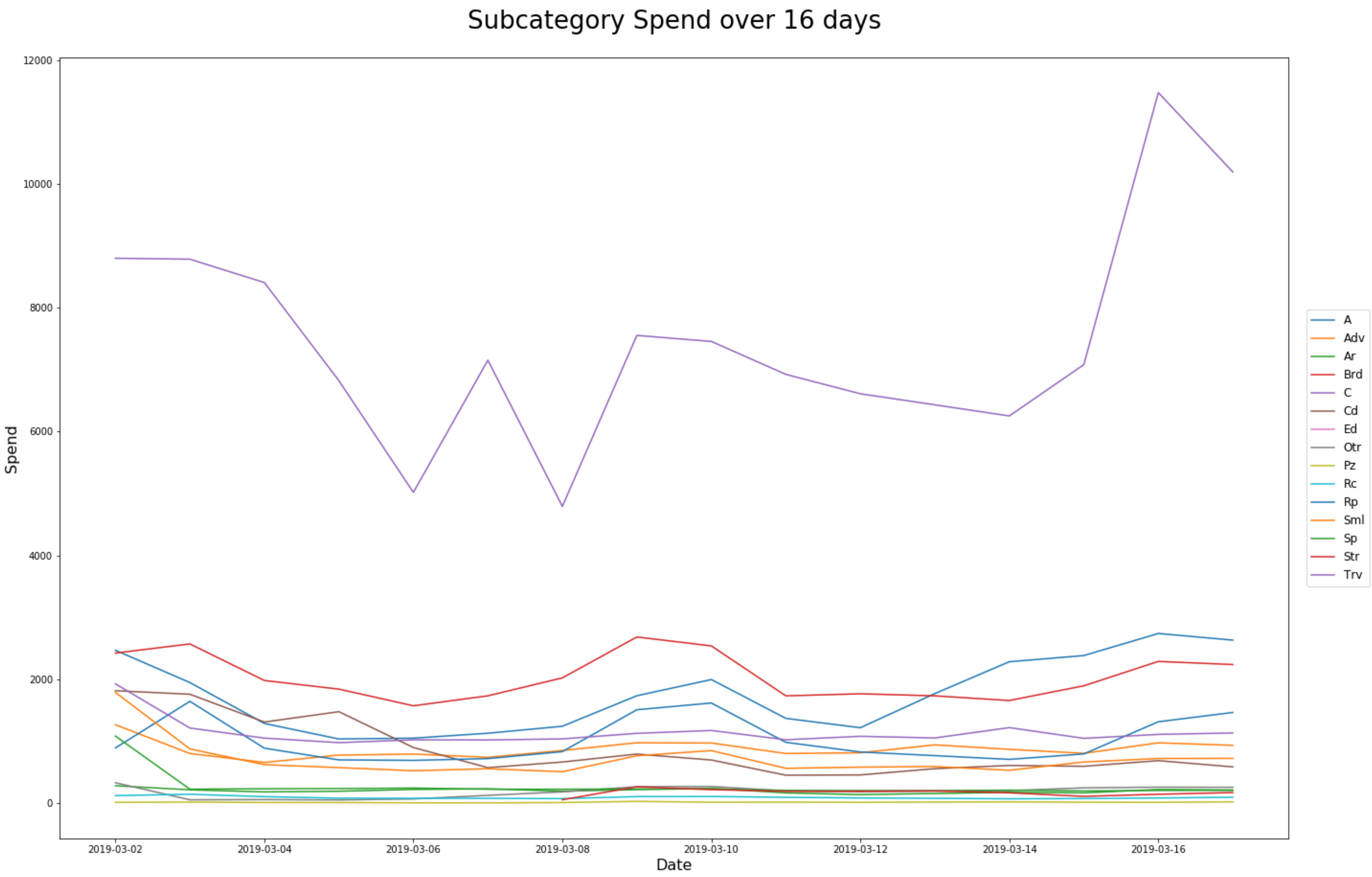
Initially tried to use the groupby function to get this result, but that created unwanted indices. This pivot function got the results in the format I needed to begin graphing.



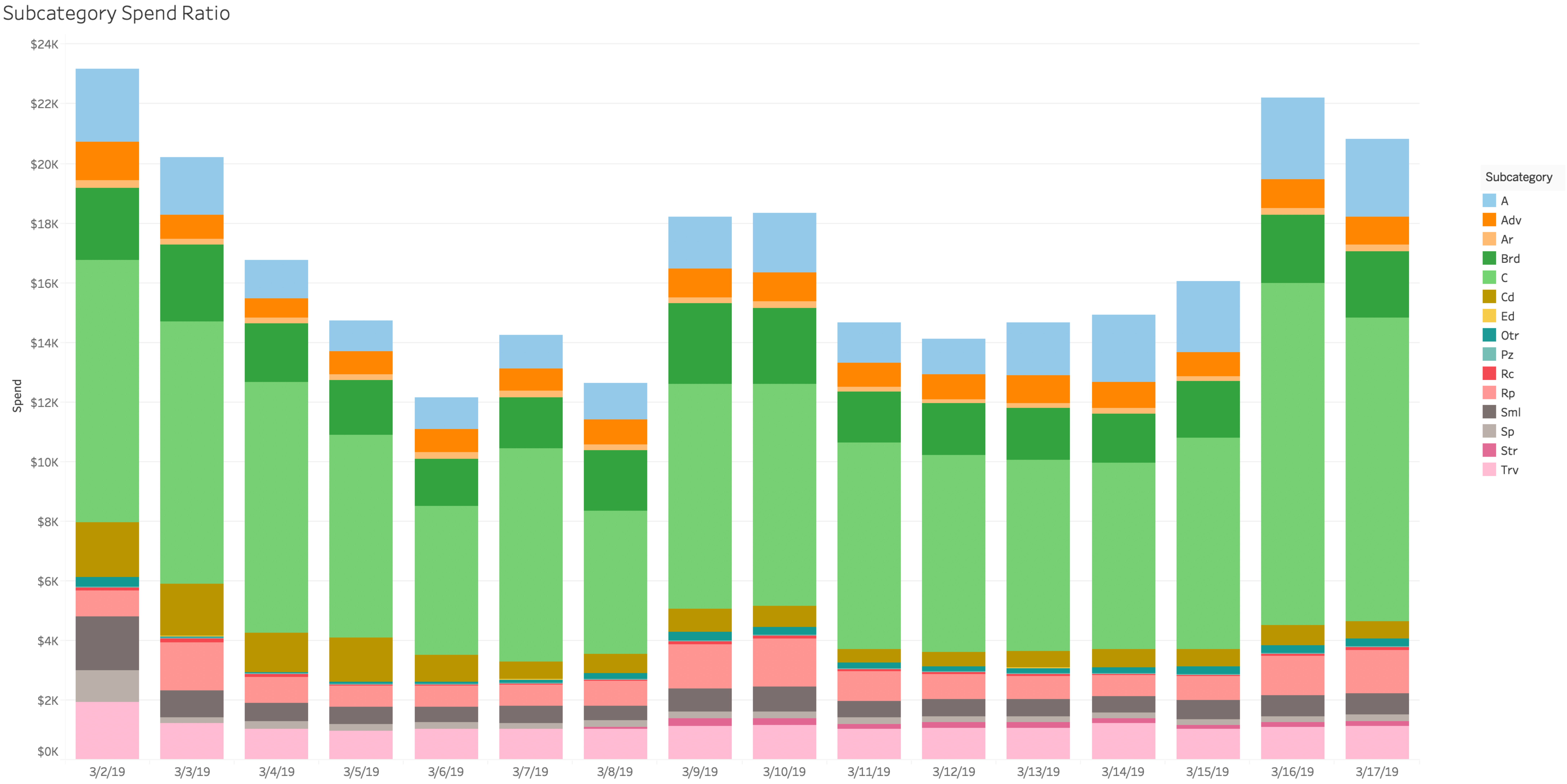
```
# Pivot table the df and set index to date, so we can see spend by date for each subcategory
data_pivot_df = data_grouped_v2.pivot(index='date', columns='subcat', values='spend')
data_pivot_df
```

subcat	A	Adv	Ar	Brd	C	Cd	Ed	Otr	Pz	Rc	Rp	Sml	Sp	Str	Trv
date															
2019-03-02	2466.35	1263.80	278.22	2421.16	8797.86	1813.17	NaN	326.77	10.27	118.90	888.97	1786.57	1080.61	NaN	1924.38
2019-03-03	1945.08	799.54	212.37	2568.77	8782.79	1755.84	0.0	51.66	15.05	141.18	1641.76	872.81	222.53	NaN	1210.77
2019-03-04	1284.37	654.70	178.81	1978.36	8405.09	1307.05	NaN	55.17	8.11	102.61	884.93	618.71	229.27	NaN	1047.30
2019-03-05	1033.44	772.99	189.18	1838.97	6820.83	1474.32	NaN	49.38	6.64	77.56	695.72	571.21	232.32	NaN	974.31
2019-03-06	1046.51	789.91	218.55	1569.52	5016.55	895.72	NaN	65.18	3.34	75.36	687.95	520.51	240.55	NaN	1019.27
2019-03-07	1126.28	737.84	228.02	1730.53	7151.92	570.35	0.0	123.59	2.08	75.49	716.59	553.13	222.56	NaN	1017.71
2019-03-08	1239.43	848.03	190.80	2020.56	4790.49	661.20	NaN	179.30	7.29	70.55	829.57	504.51	221.64	52.52	1034.64
2019-03-09	1731.85	972.40	215.33	2681.17	7552.11	790.19	NaN	262.77	26.21	103.21	1505.99	762.69	225.90	260.83	1124.93
2019-03-10	1993.74	968.31	236.86	2536.69	7454.48	693.93	NaN	266.54	11.99	105.17	1614.75	846.17	225.60	215.37	1170.98
2019-03-11	1366.39	799.20	162.77	1730.03	6922.61	448.89	NaN	194.03	13.91	91.51	979.53	559.51	203.16	187.88	1020.78
2019-03-12	1215.80	811.20	135.49	1763.36	6609.22	453.12	NaN	180.70	10.76	81.11	823.94	578.87	200.11	189.76	1076.66
2019-03-13	1766.74	938.31	152.81	1730.25	6430.96	553.93	0.0	192.04	15.49	75.44	764.72	588.06	199.11	194.25	1049.88
2019-03-14	2281.43	866.56	172.40	1654.81	6251.14	605.98	NaN	198.36	18.25	67.56	703.97	528.53	207.10	164.19	1217.11
2019-03-15	2381.15	801.99	163.16	1892.62	7080.19	592.19	NaN	244.82	12.37	71.91	792.36	661.48	193.44	107.05	1043.91
2019-03-16	2737.60	971.47	217.20	2286.47	11473.69	683.73	NaN	255.18	10.27	80.67	1311.55	718.13	202.32	140.80	1107.69
2019-03-17	2630.65	931.09	209.64	2236.02	10191.40	582.66	NaN	252.80	18.76	92.54	1461.68	719.81	202.99	168.40	1130.92

Graphs

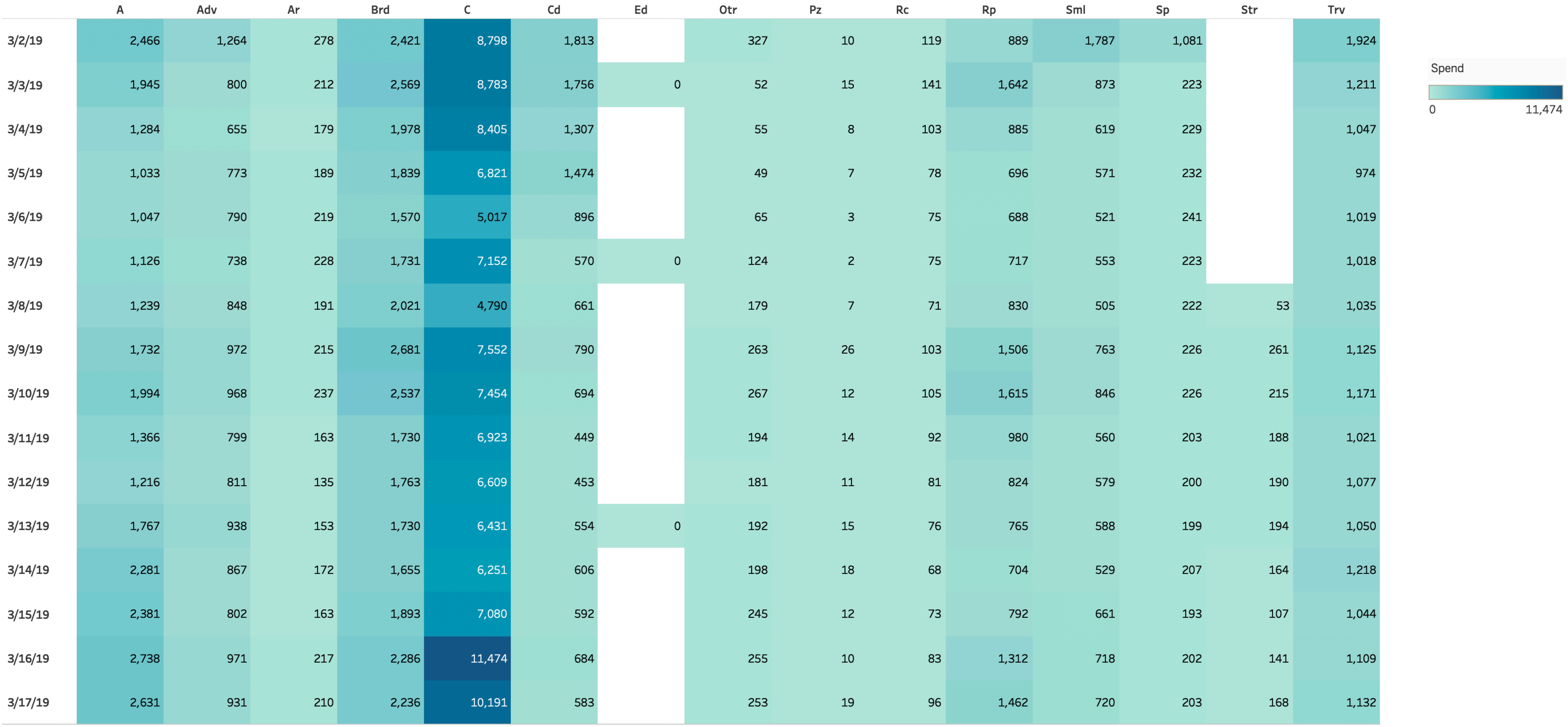

Subcategory Spend over 16 days
Spend
12000
10000
8000
6000
4000
2000
0
Date
2019-03-02
2019-03-04
2019-03-06
2019-03-08
2019-03-10
2019-03-12
2019-03-14
2019-03-16
A
Adv
Ar
Brd
C
Cd
Ed
Otr
Pz
Rc
Rp
Sml
Sp
Str
Trv

Graphs


Subcategory Spend Ratio
Spend
$24K
$22K
$20K
$18K
$16K
$14K
$12K
$10K
$8K
$6K
$4K
$2K
$0K
3/2/19
3/3/19
3/4/19
3/5/19
3/6/19
3/7/19
3/8/19
3/9/19
3/10/19
3/11/19
3/12/19
3/13/19
3/14/19
3/15/19
3/16/19
3/17/19
Subcategory
A
Adv
Ar
Brd
C
Cd
Ed
Otr
Pz
Rc
Rp
Sml
Sp
Str
Trv

Graphs

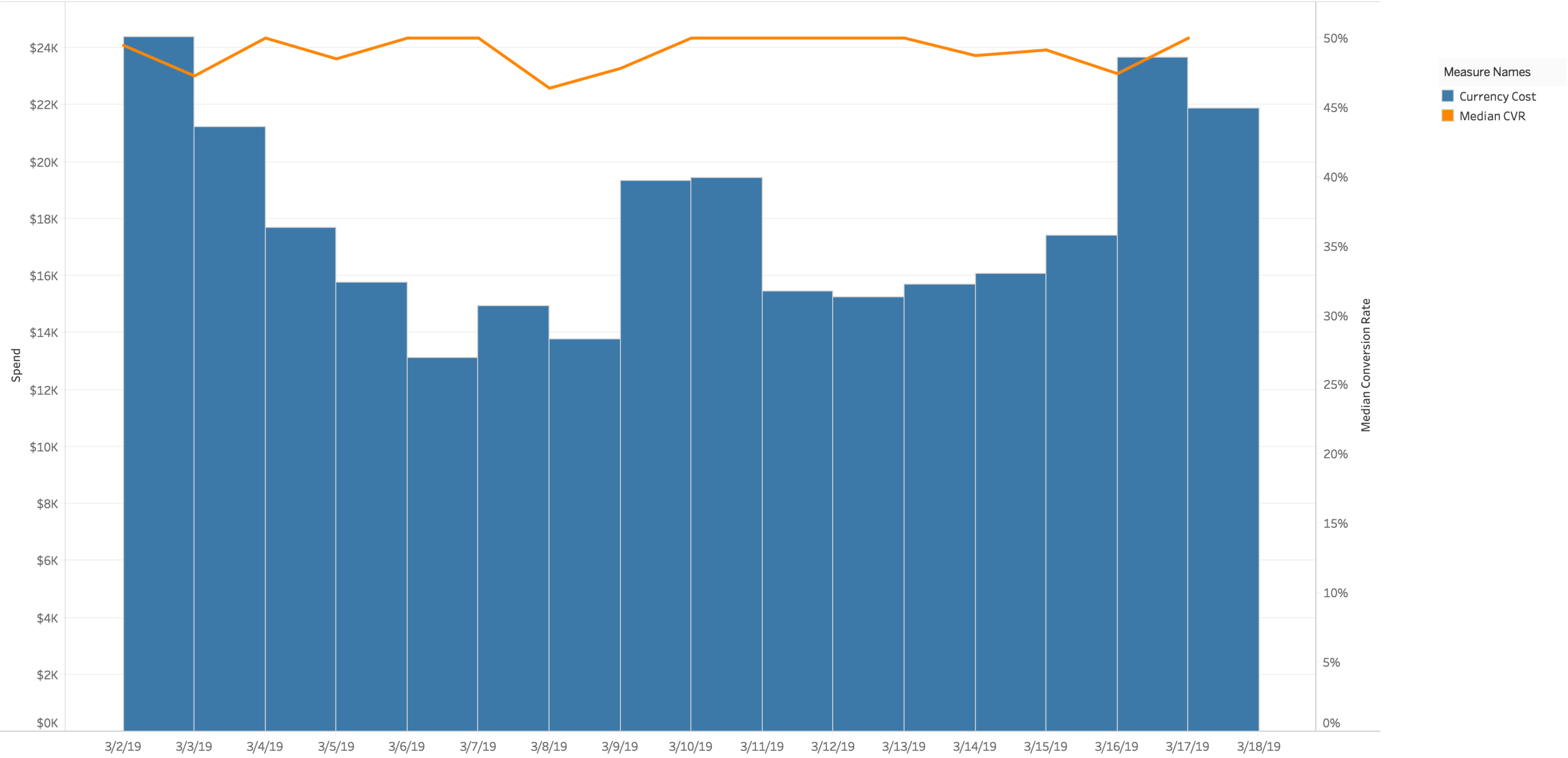
Subcategory Spend Comparison



	A	Adv	Ar	Brd	C	Cd	Ed	Otr	Pz	Rc	Rp	Sml	Sp	Str	Trv
3/2/19	2,466	1,264	278	2,421	8,798	1,813		327	10	119	889	1,787	1,081		1,924
3/3/19	1,945	800	212	2,569	8,783	1,756	0	52	15	141	1,642	873	223		1,211
3/4/19	1,284	655	179	1,978	8,405	1,307		55	8	103	885	619	229		1,047
3/5/19	1,033	773	189	1,839	6,821	1,474		49	7	78	696	571	232		974
3/6/19	1,047	790	219	1,570	5,017	896		65	3	75	688	521	241		1,019
3/7/19	1,126	738	228	1,731	7,152	570	0	124	2	75	717	553	223		1,018
3/8/19	1,239	848	191	2,021	4,790	661		179	7	71	830	505	222	53	1,035
3/9/19	1,732	972	215	2,681	7,552	790		263	26	103	1,506	763	226	261	1,125
3/10/19	1,994	968	237	2,537	7,454	694		267	12	105	1,615	846	226	215	1,171
3/11/19	1,366	799	163	1,730	6,923	449		194	14	92	980	560	203	188	1,021
3/12/19	1,216	811	135	1,763	6,609	453		181	11	81	824	579	200	190	1,077
3/13/19	1,767	938	153	1,730	6,431	554	0	192	15	76	765	588	199	194	1,050
3/14/19	2,281	867	172	1,655	6,251	606		198	18	68	704	529	207	164	1,218
3/15/19	2,381	802	163	1,893	7,080	592		245	12	73	792	661	193	107	1,044
3/16/19	2,738	971	217	2,286	11,474	684		255	10	83	1,312	718	202	141	1,109
3/17/19	2,631	931	210	2,236	10,191	583		253	19	96	1,462	720	203	168	1,132

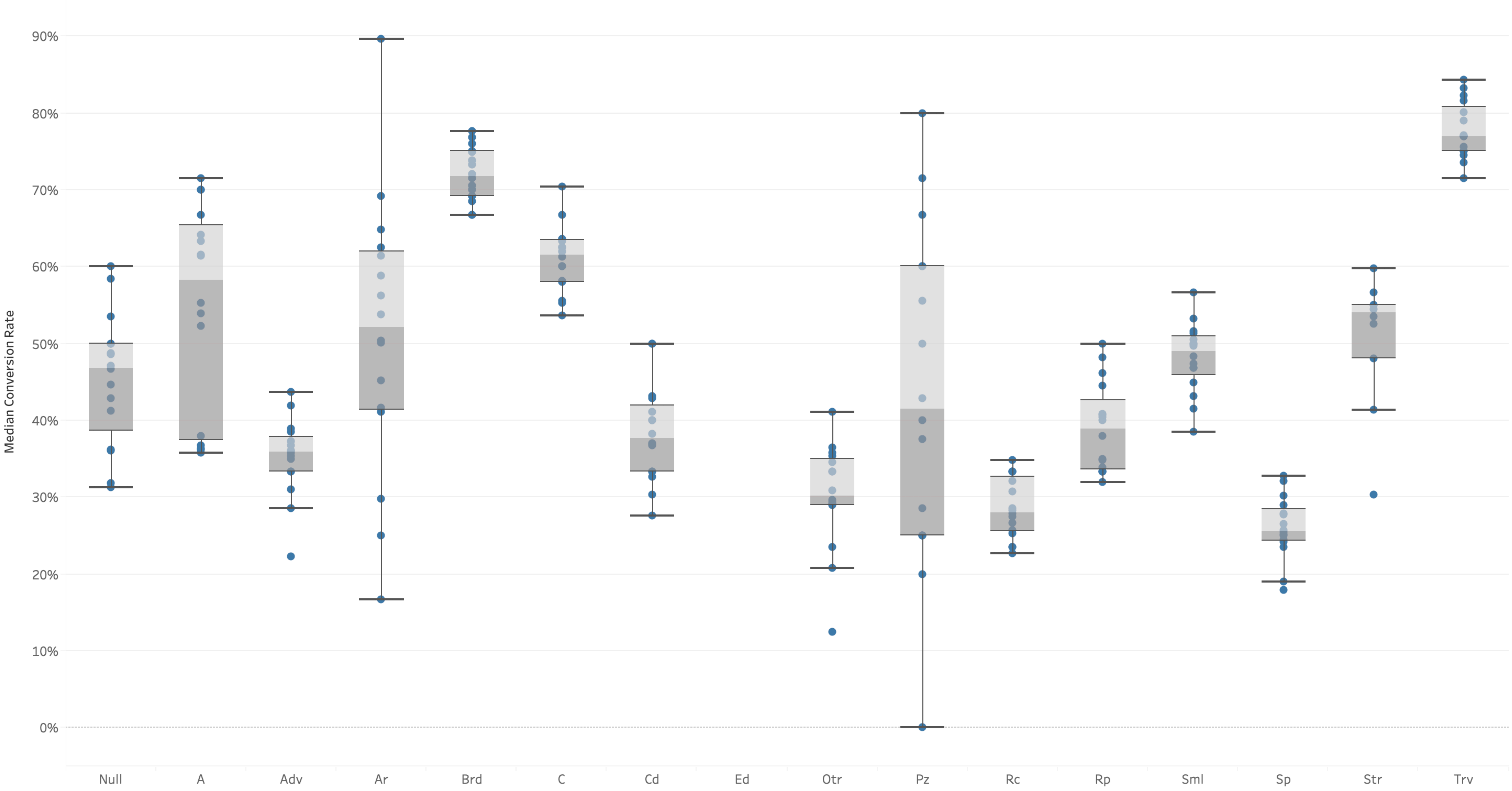
Graphs

Spend vs Conversion Rate



Graphs

Conversion Rate by Subcategory



Conclusion

- Subcategory C consistently had the highest spend as compared to other subcategories
- Total spend across all subcategories are always higher on weekends
- Conversion rates stay constant regardless of spend
- Conversion rates are most consistently high for Brd and Trv subcategories — a good next step would be to focus on driving more scale there
- Another good step would be to focus on improving CVR on subcategory C, given it already has such high spend

fin