

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as py
import seaborn as sb

In [2]: # read in the csv file
data = pd.read_csv("/users/baylife/desktop/csv files/nfl_pass_rush_receive_raw
type(data)

Out[2]: pandas.core.frame.DataFrame

In [3]: # display the columns of the data
data.columns

Out[3]: Index(['game_id', 'player_id', 'pos', 'player', 'team', 'pass_cmp', 'pass_at
t',
            'pass_yds', 'pass_td', 'pass_int', 'pass_sacked', 'pass_sacked_yds',
            'pass_long', 'pass_rating', 'rush_att', 'rush_yds', 'rush_td',
            'rush_long', 'targets', 'rec', 'rec_yds', 'rec_td', 'rec_long',
            'fumbles_lost', 'rush_scrambles', 'designed_rush_att',
            'comb_pass_rush_play', 'comb_pass_play', 'comb_rush_play',
            'Team_abbrev', 'Opponent_abbrev', 'two_point_conv', 'total_ret_td',
            'offensive_fumble_recovery_td', 'pass_yds_bonus', 'rush_yds_bonus',
            'rec_yds_bonus', 'Total_DKP', 'Off_DKP', 'Total_FDP', 'Off_FDP',
            'Total_SDP', 'Off_SDP', 'pass_target_yds', 'pass_poor_throws',
            'pass_blitzed', 'pass_hurried', 'rush_yds_before_contact', 'rush_yac',
            'rush_broken_tackles', 'rec_air_yds', 'rec_yac', 'rec_drops', 'offens
e',
            'off_pct', 'vis_team', 'home_team', 'vis_score', 'home_score', 'OT',
            'Roof', 'Surface', 'Temperature', 'Humidity', 'Wind_Speed',
            'Vegas_Line', 'Vegas_Favorite', 'Over_Under', 'game_date'],
            dtype='object')

In [4]: # rename the a few of the columns, then display the number of rows in the file
data.rename(columns={'pos':'position','game_date':'date'},inplace=True)
print("number of row in the data set:", len(data))

number of row in the data set: 18481

In [5]: # sort the dataframe by position, then by name and lastly the date
data.sort_values(by=['position','player','date'],inplace = True)

In [6]: #print data
data
```

Out[6]:

	game_id	player_id	position	player	team	pass_cmp	pass_att	pass_yds	pass_td
10939	202012060oti	BrewAa01	C	Aaron Brewer	TEN	0	0	0	0
4776	201912080min	BradGa00	C	Garrett Bradbury	MIN	0	0	0	0
6209	202001110sfo	BradGa00	C	Garrett Bradbury	MIN	0	0	0	0
11721	202012200nyg	GateNi00	C	Nick Gates	NYG	0	0	0	0
9355	202011080clt	SkurMa01	C/G	Matt Skura	BAL	0	0	0	0
...
9216	202011010rav	SneaWi00	WR/R	Willie Snead	BAL	0	0	0	0
9352	202011080clt	SneaWi00	WR/R	Willie Snead	BAL	0	0	0	0
9822	202011150nwe	SneaWi00	WR/R	Willie Snead	BAL	0	0	0	0
10223	202011220rav	SneaWi00	WR/R	Willie Snead	BAL	0	0	0	0
12799	202101160buf	SneaWi00	WR/R	Willie Snead	BAL	0	0	0	0

18481 rows x 69 columns

In [7]:

```
#check if the column names were changed
data.columns
```

Out[7]:

```
Index(['game_id', 'player_id', 'position', 'player', 'team', 'pass_cmp',
      'pass_att', 'pass_yds', 'pass_td', 'pass_int', 'pass_sacked',
      'pass_sacked_yds', 'pass_long', 'pass_rating', 'rush_att', 'rush_yds',
      'rush_td', 'rush_long', 'targets', 'rec', 'rec_yds', 'rec_td',
      'rec_long', 'fumbles_lost', 'rush_scrambles', 'designed_rush_att',
      'comb_pass_rush_play', 'comb_pass_play', 'comb_rush_play',
      'Team_abbrev', 'Opponent_abbrev', 'two_point_conv', 'total_ret_td',
      'offensive_fumble_recovery_td', 'pass_yds_bonus', 'rush_yds_bonus',
      'rec_yds_bonus', 'Total_DKP', 'Off_DKP', 'Total_FDP', 'Off_FDP',
      'Total_SDP', 'Off_SDP', 'pass_target_yds', 'pass_poor_throws',
      'pass_blitzed', 'pass_hurried', 'rush_yds_before_contact', 'rush_yac',
      'rush_broken_tackles', 'rec_air_yds', 'rec_yac', 'rec_drops', 'offens
e',
      'off_pct', 'vis_team', 'home_team', 'vis_score', 'home_score', 'OT',
      'Roof', 'Surface', 'Temperature', 'Humidity', 'Wind_Speed',
      'Vegas_Line', 'Vegas_Favorite', 'Over_Under', 'date'],
      dtype='object')
```

In [8]:

```
# get the list of players who are QBs
df = data[data["position"]=="QB"]
```

In [9]:

```
# sort the QB's by name
df.sort_values(by=['player'])
```

Out[9]:

	game_id	player_id	position	player	team	pass_cmp	pass_att	pass_yds	pas
3724	201911170rav	McCaA.00	QB	A.J. McCarron	HOU	0	1	0	
5864	201912290htx	McCaA.00	QB	A.J. McCarron	HOU	21	36	225	
11160	202012130chi	McCaA.00	QB	A.J. McCarron	HOU	0	0	0	
12474	202101030htx	McCaA.00	QB	A.J. McCarron	HOU	1	1	20	
10045	2020111220clt	RodgAa00	QB	Aaron Rodgers	GNB	27	38	311	
...	
13825	202109260den	WilsZa00	QB	Zach Wilson	NYJ	19	35	160	
13032	202109120car	WilsZa00	QB	Zach Wilson	NYJ	20	37	258	
17958	202112120nyj	WilsZa00	QB	Zach Wilson	NYJ	19	42	202	
14418	202110030nyj	WilsZa00	QB	Zach Wilson	NYJ	21	34	297	
18248	202112190mia	WilsZa00	QB	Zach Wilson	NYJ	13	23	170	

1849 rows x 69 columns

In [10]:

```
#sort by the number of touchdowns
touch_downs = df.sort_values(by=['pass_td'])
```

In [11]:

```
# save the columns of interest in an array, create new dataframe for the quate
interest = ['position', 'player', 'team', 'pass_cmp',
            'pass_att', 'pass_yds', 'pass_td', 'pass_int','pass_rating','Surface',
QB_data = df[interest]
```

In [12]:

```
# get the QB who played in the most games: the frequence of the name that appe
QB_data.loc[:, "player"].describe()
```

Out[12]:

```
count      1849
unique       114
top      Tom Brady
freq         51
Name: player, dtype: object
```

In [13]:

```
# get the collection of data for the QB who played in the most amount of games
most_prominent = QB_data.loc[:, "player"].describe()
brady = QB_data[QB_data['player'] == most_prominent[2]]
brady
```

Out[13]:

	position	player	team	pass_cmp	pass_att	pass_yds	pass_td	pass_int	pass_rating
210	QB	Tom Brady	NWE	24	36	341	3	0	124.9
555	QB	Tom Brady	NWE	20	28	264	2	0	124.7
977	QB	Tom Brady	NWE	28	42	306	2	0	103.9
1202	QB	Tom Brady	NWE	18	39	150	0	1	45.9
1787	QB	Tom Brady	NWE	28	42	348	3	1	106.1
1846	QB	Tom Brady	NWE	31	41	334	0	1	88.9
2485	QB	Tom Brady	NWE	31	45	249	1	1	80.7
2763	QB	Tom Brady	NWE	20	36	259	2	0	96.9
3100	QB	Tom Brady	NWE	30	46	285	1	1	80.4
3669	QB	Tom Brady	NWE	26	47	216	0	0	67.3
4017	QB	Tom Brady	NWE	17	37	190	1	0	70.8
4370	QB	Tom Brady	NWE	24	47	326	3	1	85.9
4809	QB	Tom Brady	NWE	19	36	169	1	1	63.3
4987	QB	Tom Brady	NWE	15	29	128	2	0	86.6
5324	QB	Tom Brady	NWE	26	33	271	1	0	111.0
5952	QB	Tom Brady	NWE	16	29	221	2	1	88.4
6115	QB	Tom Brady	NWE	20	37	209	0	1	59.4
6547	QB	Tom Brady	TAM	23	36	239	2	2	78.4
7087	QB	Tom Brady	TAM	23	35	217	1	1	80.3
7275	QB	Tom Brady	TAM	25	38	297	3	0	115.8
7809	QB	Tom Brady	TAM	30	46	369	5	1	117.0
7894	QB	Tom Brady	TAM	25	41	253	1	0	86.7
8520	QB	Tom Brady	TAM	17	27	166	2	0	104.9
8830	QB	Tom Brady	TAM	33	45	369	4	0	127.0
9249	QB	Tom Brady	TAM	28	40	279	2	0	106.1
9561	QB	Tom Brady	TAM	22	38	209	0	3	40.4
9657	QB	Tom Brady	TAM	28	39	341	3	0	124.0
10297	QB	Tom Brady	TAM	26	48	216	2	2	62.5
10632	QB	Tom Brady	TAM	27	41	345	3	2	96.1

	position	player	team	pass_cmp	pass_att	pass_yds	pass_td	pass_int	pass_rating
11436	QB	Tom Brady	TAM	15	23	196	2	0	120.9
11545	QB	Tom Brady	TAM	31	45	390	2	0	110.4
11926	QB	Tom Brady	TAM	22	27	348	4	0	158.3
12638	QB	Tom Brady	TAM	26	41	399	4	1	117.8
12699	QB	Tom Brady	TAM	22	40	381	2	0	104.3
12883	QB	Tom Brady	TAM	18	33	199	2	0	92.9
12887	QB	Tom Brady	TAM	20	36	280	3	3	73.8
12953	QB	Tom Brady	TAM	21	29	201	3	0	125.8
12969	QB	Tom Brady	TAM	32	50	379	4	2	97.0
13710	QB	Tom Brady	TAM	24	36	276	5	0	129.2
14088	QB	Tom Brady	TAM	41	55	432	1	0	103.0
14390	QB	Tom Brady	TAM	22	43	269	0	0	70.8
14858	QB	Tom Brady	TAM	30	41	411	5	0	144.4
14931	QB	Tom Brady	TAM	34	42	297	2	1	102.1
15559	QB	Tom Brady	TAM	20	36	211	4	0	109.8
15847	QB	Tom Brady	TAM	28	40	375	4	2	112.0
16626	QB	Tom Brady	TAM	23	34	220	2	2	80.5
17027	QB	Tom Brady	TAM	30	46	307	2	1	89.7
17150	QB	Tom Brady	TAM	25	34	226	1	1	88.6
17440	QB	Tom Brady	TAM	38	51	368	4	1	112.2
18034	QB	Tom Brady	TAM	31	46	363	2	0	105.6
18369	QB	Tom Brady	TAM	26	48	214	0	1	57.1

In [14]:

```
# narrow down the columns of interest for Tom Brady
observation= ['player', 'team', 'pass_cmp',
              'pass_att', 'pass_yds', 'pass_td', 'pass_int', 'pass_rating']

# get the dates of each event for the Tom Brady
time_laps = brady.loc[:, 'date']
cluster = brady.loc[:, observation[2]:observation[7]]
cluster
```

Out[14]:

	pass_cmp	pass_att	pass_yds	pass_td	pass_int	pass_rating
210	24	36	341	3	0	124.9
555	20	28	264	2	0	124.7
977	28	42	306	2	0	103.9
1202	18	39	150	0	1	45.9
1787	28	42	348	3	1	106.1
1846	31	41	334	0	1	88.9
2485	31	45	249	1	1	80.7
2763	20	36	259	2	0	96.9
3100	30	46	285	1	1	80.4
3669	26	47	216	0	0	67.3
4017	17	37	190	1	0	70.8
4370	24	47	326	3	1	85.9
4809	19	36	169	1	1	63.3
4987	15	29	128	2	0	86.6
5324	26	33	271	1	0	111.0
5952	16	29	221	2	1	88.4
6115	20	37	209	0	1	59.4
6547	23	36	239	2	2	78.4
7087	23	35	217	1	1	80.3
7275	25	38	297	3	0	115.8
7809	30	46	369	5	1	117.0
7894	25	41	253	1	0	86.7
8520	17	27	166	2	0	104.9
8830	33	45	369	4	0	127.0
9249	28	40	279	2	0	106.1
9561	22	38	209	0	3	40.4
9657	28	39	341	3	0	124.0
10297	26	48	216	2	2	62.5
10632	27	41	345	3	2	96.1
11436	15	23	196	2	0	120.9
11545	31	45	390	2	0	110.4
11926	22	27	348	4	0	158.3
12638	26	41	399	4	1	117.8
12699	22	40	381	2	0	104.3
12883	18	33	199	2	0	92.9
12887	20	36	280	3	3	73.8
12953	21	29	201	3	0	125.8
12969	32	50	379	4	2	97.0
13710	24	36	276	5	0	129.2
14088	41	55	432	1	0	103.0
14390	22	43	269	0	0	70.8
14858	30	41	411	5	0	144.4
14931	34	42	297	2	1	102.1
15559	20	36	211	4	0	109.8
15847	28	40	375	4	2	112.0

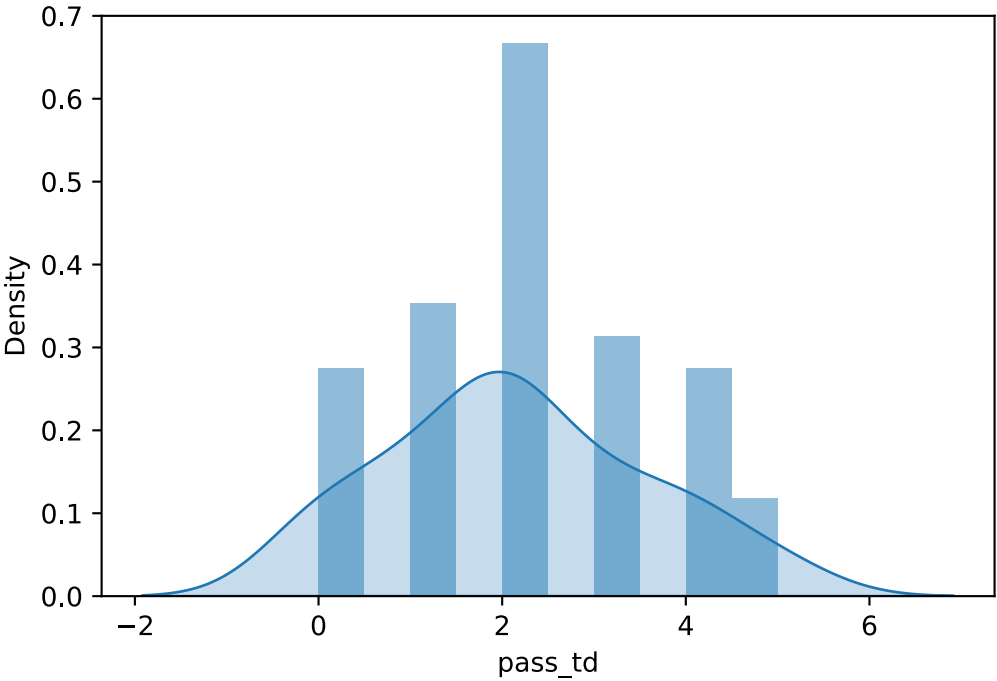
	pass_cmp	pass_att	pass_yds	pass_td	pass_int	pass_rating
16626	23	34	220	2	2	80.5
17027	30	46	307	2	1	89.7
17150	25	34	226	1	1	88.6
17440	38	51	368	4	1	112.2
18034	31	46	363	2	0	105.6
18369	26	48	214	0	1	57.1

In [15]:

```
# histogram of Tom Brady pass touchdowns, Kdeplot for touchdowns
#for col in temp2['pass_cmp']:
    #density unitizes each histogram such that the area under the curver is 1
    #alpha essential allows one to visibly see through each graph
    #kdeplot kernal density plot
#sb.kdeplot(cluster['pass_td'],cluster['pass_yds'],shade=True)
sb.kdeplot(cluster['pass_td'],shade=True)
py.hist(cluster["pass_td"],density = True, alpha = 0.5)
#py.hist(cluster["pass_yds"],density = True, alpha = 0.5)
```

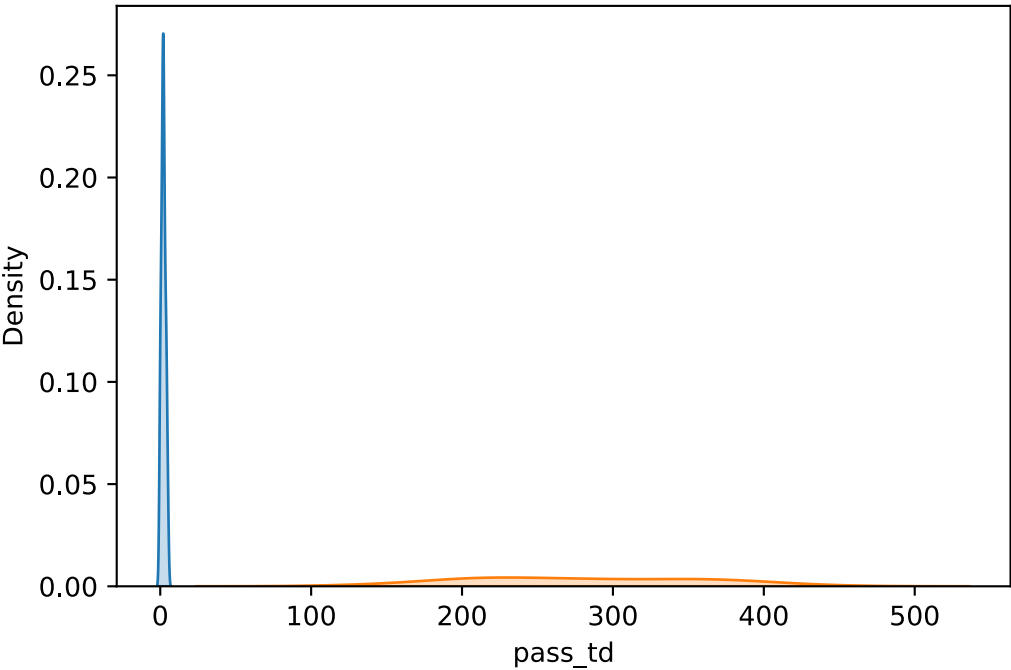
Out[15]:

```
(array([0.2745098 , 0.          , 0.35294118, 0.          , 0.66666667,
        0.          , 0.31372549, 0.          , 0.2745098 , 0.11764706]),
 array([0. , 0.5, 1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5, 5. ]),
 <BarContainer object of 10 artists>)
```



In [16]:

```
# get the kedplot for the pass_td and the pass_yds
l = ['pass_td','pass_yds']
for col in l:
    sb.kdeplot(cluster[col],shade=True)
```



```
In [17]: # get the sum of all the rows for each column
print(cluster.sum())
```

```
pass_cmp      1279.0
pass_att      2000.0
pass_yds     14308.0
pass_td        110.0
pass_int        35.0
pass_rating   4930.5
dtype: float64
```

```
In [18]: # get a general description of the data cluster from Tom Brady
cluster.describe()
```

Out[18]:

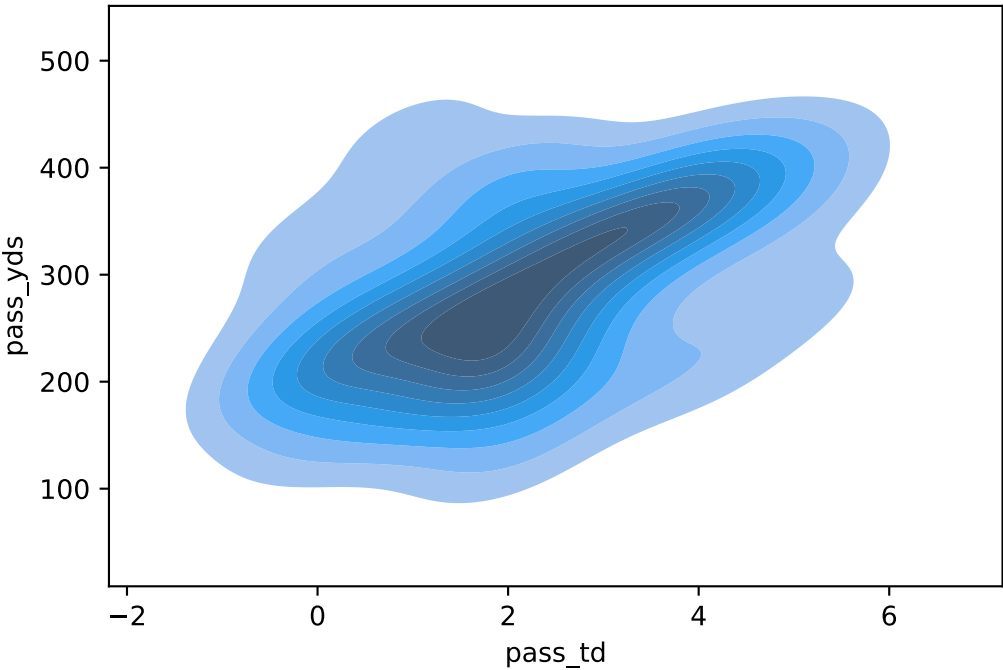
	pass_cmp	pass_att	pass_yds	pass_td	pass_int	pass_rating
count	51.000000	51.000000	51.000000	51.000000	51.000000	51.000000
mean	25.078431	39.215686	280.549020	2.156863	0.686275	96.676471
std	5.747497	6.862401	76.536348	1.405312	0.836426	24.598484
min	15.000000	23.000000	128.000000	0.000000	0.000000	40.400000
25%	20.500000	36.000000	216.000000	1.000000	0.000000	80.450000
50%	25.000000	40.000000	276.000000	2.000000	0.000000	97.000000
75%	29.000000	45.000000	346.500000	3.000000	1.000000	112.100000
max	41.000000	55.000000	432.000000	5.000000	3.000000	158.300000

```
In [19]: # the cluster shows the corrolation between the passing td and passing yds
sb.kdeplot(cluster[l[0]],cluster[l[1]],shade=True)
```

/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variable as a key word arg: y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

```
Out[19]: <AxesSubplot:xlabel='pass_td', ylabel='pass_yds'>
```

```
In [20]: # We will determine the corrolation between each column for Tom Brady to deter
analyze = cluster.corr()
```

```
In [21]: # after multiplying the cluster by the Transpose of the cluster we have a symm
analyze
```

Out[21]:

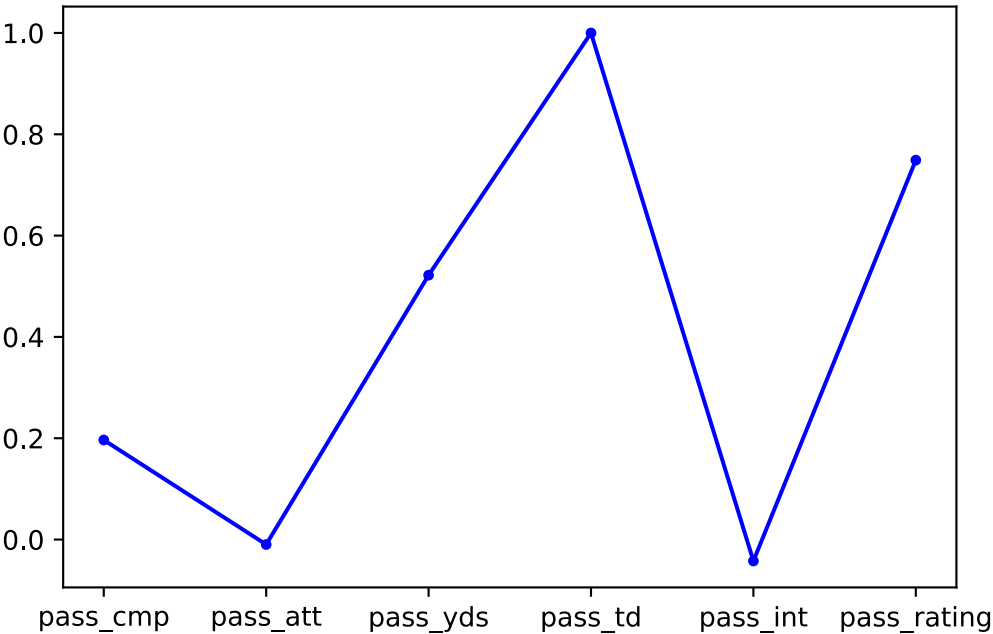
	pass_cmp	pass_att	pass_yds	pass_td	pass_int	pass_rating
pass_cmp	1.000000	0.797197	0.741627	0.196539	0.063465	0.237020
pass_att	0.797197	1.000000	0.537903	-0.009800	0.182760	-0.206267
pass_yds	0.741627	0.537903	1.000000	0.521881	-0.036308	0.566525
pass_td	0.196539	-0.009800	0.521881	1.000000	-0.042370	0.748998
pass_int	0.063465	0.182760	-0.036308	-0.042370	1.000000	-0.509630
pass_rating	0.237020	-0.206267	0.566525	0.748998	-0.509630	1.000000

```
In [22]: analyze.columns
```

Out[22]: Index(['pass_cmp', 'pass_att', 'pass_yds', 'pass_td', 'pass_int', 'pass_rating'], dtype='object')

```
In [23]: # we find that the graph is consistant with the corralations, so we can surmis
#zone = brady.loc[:, 'date']
py.plot(analyze["pass_td"],marker="." , color = "blue")
```

Out[23]: [



```
In [24]: # simple group by
insert = ['player','pass_cmp',
          'pass_att', 'pass_yds', 'pass_td', 'pass_int','pass_rating']

test = brady.groupby(by=insert[0])[[insert[1],insert[2],insert[3],insert[4],insert[5]]]
```

```
In [25]: #cumulative stats for Tom Brady
test.head()
```

Out[25]:

	pass_cmp	pass_att	pass_yds	pass_td	pass_int	pass_rating
player						
Tom Brady	1279	2000	14308	110	35	4930.5

```
In [ ]:
```