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_data.csv")
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_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', 'offense',
       '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
```

	game_id	player_id	position	player	team	pass_cmp	pass_att	pass_yds	pass_td	pass_int	...	OT	Roof	Surface	Temperature	Humidity	Wind_Speed	Vegas_l
10939	202012060oti	BrewAa01	C	Aaron Brewer	TEN	0	0	0	0	0	...	False	outdoors	grass	52	68	1	
4776	201912080min	BradGa00	C	Garrett Bradbury	MIN	0	0	0	0	0	...	False	dome	sportturf	72	45	0	-
6209	202001110sfo	BradGa00	C	Garrett Bradbury	MIN	0	0	0	0	0	...	False	outdoors	grass	56	67	11	
11721	202012200nyg	GateNi00	C	Nick Gates	NYG	0	0	0	0	0	...	False	outdoors	fieldturf	36	82	0	
9355	202011080clt	SkurMa01	C/G	Matt Skura	BAL	0	0	0	0	0	...	False	retractable roof (closed)	fieldturf	72	45	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
9216	202011010rav	SneaWi00	WR/R	Willie Snead	BAL	0	0	0	0	0	...	False	outdoors	grass	52	97	7	
9352	202011080clt	SneaWi00	WR/R	Willie Snead	BAL	0	0	0	0	0	...	False	retractable roof (closed)	fieldturf	72	45	0	
9822	202011150nwe	SneaWi00	WR/R	Willie Snead	BAL	0	0	0	0	0	...	False	outdoors	grass	52	81	13	
10223	202011220rav	SneaWi00	WR/R	Willie Snead	BAL	0	0	0	0	0	...	True	outdoors	grass	53	67	6	
12799	202101160buf	SneaWi00	WR/R	Willie Snead	BAL	0	0	0	0	0	...	False	outdoors	astroturf	34	81	13	

18481 rows × 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', 'offense',
       '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  pass_td  pass_int  ...  OT     Roof                       Surface     Temperature  Humidity  Wind_Spee
3724   201911170rav  McCaA.00  QB        A.J. McCarron  HOU          0         1         0        0         0  ...  False  outdoors                   grass                44        56          1
5864   201912290htx  McCaA.00  QB        A.J. McCarron  HOU         21        36       225        0         1  ...  False  retractable roof (closed)  grass                72        45
11160  202012130chi  McCaA.00  QB        A.J. McCarron  HOU          0         0         0        0         0  ...  False  outdoors                   grass                34        70
12474  202101030htx  McCaA.00  QB        A.J. McCarron  HOU          1         1        20        0         0  ...  False  retractable roof (closed)  grass                72        45
10045  202011220clt  RodgAa00  QB        Aaron Rodgers  GNB         27        38       311        3         1  ...  True   retractable roof (closed)  fieldturf            72        45
...             ...       ...  ...                 ...  ...        ...       ...       ...      ...       ...  ...    ...                        ...        ...          ...       ...          .
13825  202109260den  WilsZa00  QB        Zach Wilson    NYJ         19        35       160        0         2  ...  False  outdoors                   grass                82        13
13032  202109120car  WilsZa00  QB        Zach Wilson    NYJ         20        37       258        2         1  ...  False  outdoors                   grass                81        51
17958  202112120nyj  WilsZa00  QB        Zach Wilson    NYJ         19        42       202        0         0  ...  False  outdoors                   fieldturf            46        30          1
14418  202110030nyj  WilsZa00  QB        Zach Wilson    NYJ         21        34       297        2         1  ...  True   outdoors                   fieldturf            78        50
18248  202112190mia  WilsZa00  QB        Zach Wilson    NYJ         13        23       170        0         0  ...  False  outdoors                   grass                82        70

1849 rows × 69 columns
```

In [10]:

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

In [11]:

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

In [12]:

```
# get the QB who played in the most games: the frequence of the name that appears the most is analogous to the number of games they played in
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	Surface	Temperature	Humidity	Wind_Speed	date
210	QB	Tom Brady	NWE	24	36	341	3	0	124.9	grass	66	62	2	2019-09-08
555	QB	Tom Brady	NWE	20	28	264	2	0	124.7	grass	87	67	4	2019-09-15
977	QB	Tom Brady	NWE	28	42	306	2	0	103.9	grass	81	49	10	2019-09-22
1202	QB	Tom Brady	NWE	18	39	150	0	1	45.9	astroturf	58	71	10	2019-09-29
1787	QB	Tom Brady	NWE	28	42	348	3	1	106.1	grass	70	66	8	2019-10-06
1846	QB	Tom Brady	NWE	31	41	334	0	1	88.9	grass	54	61	13	2019-10-10
2485	QB	Tom Brady	NWE	31	45	249	1	1	80.7	fieldturf	58	73	0	2019-10-21
2763	QB	Tom Brady	NWE	20	36	259	2	0	96.9	grass	49	91	11	2019-10-27
3100	QB	Tom Brady	NWE	30	46	285	1	1	80.4	grass	48	41	1	2019-11-03
3669	QB	Tom Brady	NWE	26	47	216	0	0	67.3	grass	42	57	13	2019-11-17
4017	QB	Tom Brady	NWE	17	37	190	1	0	70.8	grass	38	90	16	2019-11-24
4370	QB	Tom Brady	NWE	24	47	326	3	1	85.9	grass	72	45	0	2019-12-01
4809	QB	Tom Brady	NWE	19	36	169	1	1	63.3	grass	32	56	8	2019-12-08
4987	QB	Tom Brady	NWE	15	29	128	2	0	86.6	grass	36	58	6	2019-12-15
5324	QB	Tom Brady	NWE	26	33	271	1	0	111.0	grass	33	64	2	2019-12-21
5952	QB	Tom Brady	NWE	16	29	221	2	1	88.4	grass	43	41	2	2019-12-29
6115	QB	Tom Brady	NWE	20	37	209	0	1	59.4	grass	44	88	4	2020-01-04
6547	QB	Tom Brady	TAM	23	36	239	2	2	78.4	astroturf	72	45	0	2020-09-13
7087	QB	Tom Brady	TAM	23	35	217	1	1	80.3	grass	85	72	17	2020-09-20
7275	QB	Tom Brady	TAM	25	38	297	3	0	115.8	grass	55	39	7	2020-09-27
7809	QB	Tom Brady	TAM	30	46	369	5	1	117.0	grass	75	87	6	2020-10-04
7894	QB	Tom Brady	TAM	25	41	253	1	0	86.7	grass	57	74	7	2020-10-08
8520	QB	Tom Brady	TAM	17	27	166	2	0	104.9	grass	88	48	12	2020-10-18
8830	QB	Tom Brady	TAM	33	45	369	4	0	127.0	grass	72	45	0	2020-10-25
9249	QB	Tom Brady	TAM	28	40	279	2	0	106.1	fieldturf	39	46	8	2020-11-02
9561	QB	Tom Brady	TAM	22	38	209	0	3	40.4	grass	77	76	17	2020-11-08

	position	player	team	pass_cmp	pass_att	pass_yds	pass_td	pass_int	pass_rating	Surface	Temperature	Humidity	Wind_Speed	date
9657	QB	Tom Brady	TAM	28	39	341	3	0	124.0	grass	65	90	15	2020-11-15
10297	QB	Tom Brady	TAM	26	48	216	2	2	62.5	grass	69	65	5	2020-11-23
10632	QB	Tom Brady	TAM	27	41	345	3	2	96.1	grass	81	50	5	2020-11-29
11436	QB	Tom Brady	TAM	15	23	196	2	0	120.9	grass	78	63	5	2020-12-13
11545	QB	Tom Brady	TAM	31	45	390	2	0	110.4	fieldturf	72	45	0	2020-12-20
11926	QB	Tom Brady	TAM	22	27	348	4	0	158.3	fieldturf	72	45	0	2020-12-26
12638	QB	Tom Brady	TAM	26	41	399	4	1	117.8	grass	65	77	4	2021-01-03
12699	QB	Tom Brady	TAM	22	40	381	2	0	104.3	grass	40	47	10	2021-01-09
12883	QB	Tom Brady	TAM	18	33	199	2	0	92.9	astroturf	72	45	0	2021-01-17
12887	QB	Tom Brady	TAM	20	36	280	3	3	73.8	grass	29	75	10	2021-01-24
12953	QB	Tom Brady	TAM	21	29	201	3	0	125.8	grass	63	78	9	2021-02-07
12969	QB	Tom Brady	TAM	32	50	379	4	2	97.0	grass	82	85	9	2021-09-09
13710	QB	Tom Brady	TAM	24	36	276	5	0	129.2	grass	85	72	0	2021-09-19
14088	QB	Tom Brady	TAM	41	55	432	1	0	103.0	matrixturf	72	45	0	2021-09-26
14390	QB	Tom Brady	TAM	22	43	269	0	0	70.8	grass	62	84	4	2021-10-03
14858	QB	Tom Brady	TAM	30	41	411	5	0	144.4	grass	84	59	7	2021-10-10
14931	QB	Tom Brady	TAM	34	42	297	2	1	102.1	grass	74	56	8	2021-10-14
15559	QB	Tom Brady	TAM	20	36	211	4	0	109.8	grass	83	65	7	2021-10-24
15847	QB	Tom Brady	TAM	28	40	375	4	2	112.0	astroturf	72	45	0	2021-10-31
16626	QB	Tom Brady	TAM	23	34	220	2	2	80.5	grass	47	40	8	2021-11-14
17027	QB	Tom Brady	TAM	30	46	307	2	1	89.7	grass	68	61	6	2021-11-22
17150	QB	Tom Brady	TAM	25	34	226	1	1	88.6	fieldturf	72	45	0	2021-11-28
17440	QB	Tom Brady	TAM	38	51	368	4	1	112.2	fieldturf	72	45	0	2021-12-05
18034	QB	Tom Brady	TAM	31	46	363	2	0	105.6	grass	82	66	4	2021-12-12
18369	QB	Tom Brady	TAM	26	48	214	0	1	57.1	grass	75	99	0	2021-12-19

```
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

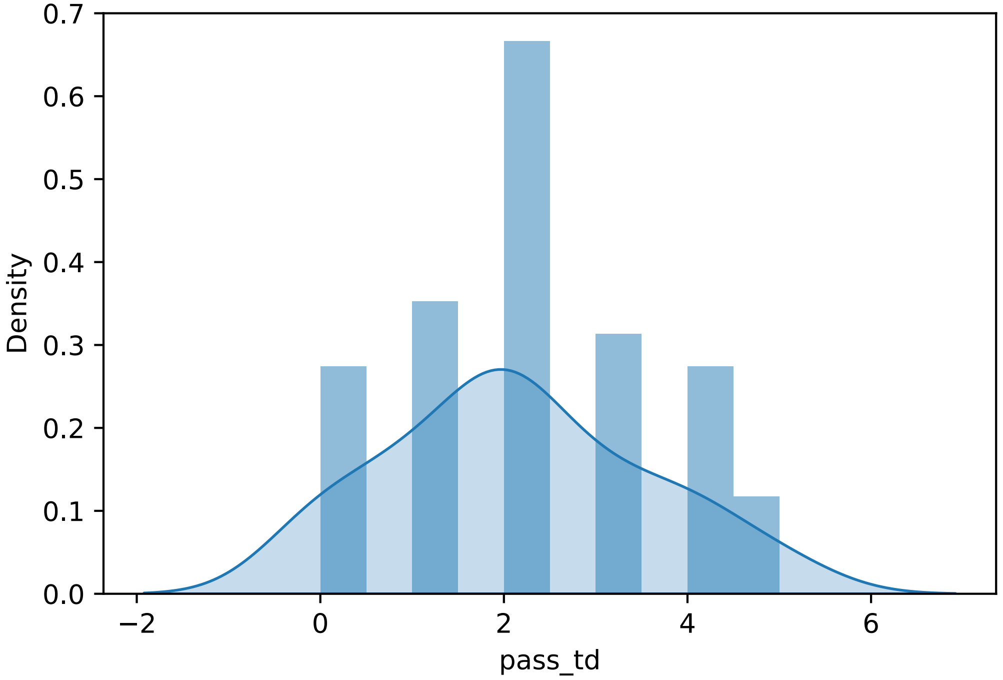
	pass_cmp	pass_att	pass_yds	pass_td	pass_int	pass_rating
9657	28	39	341	3	0	124.0
10297	26	48	216	2	2	62.5
10632	27	41	345	3	2	96.1
11436	15	23	196	2	0	120.9
11545	31	45	390	2	0	110.4
11926	22	27	348	4	0	158.3
12638	26	41	399	4	1	117.8
12699	22	40	381	2	0	104.3
12883	18	33	199	2	0	92.9
12887	20	36	280	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
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)
```

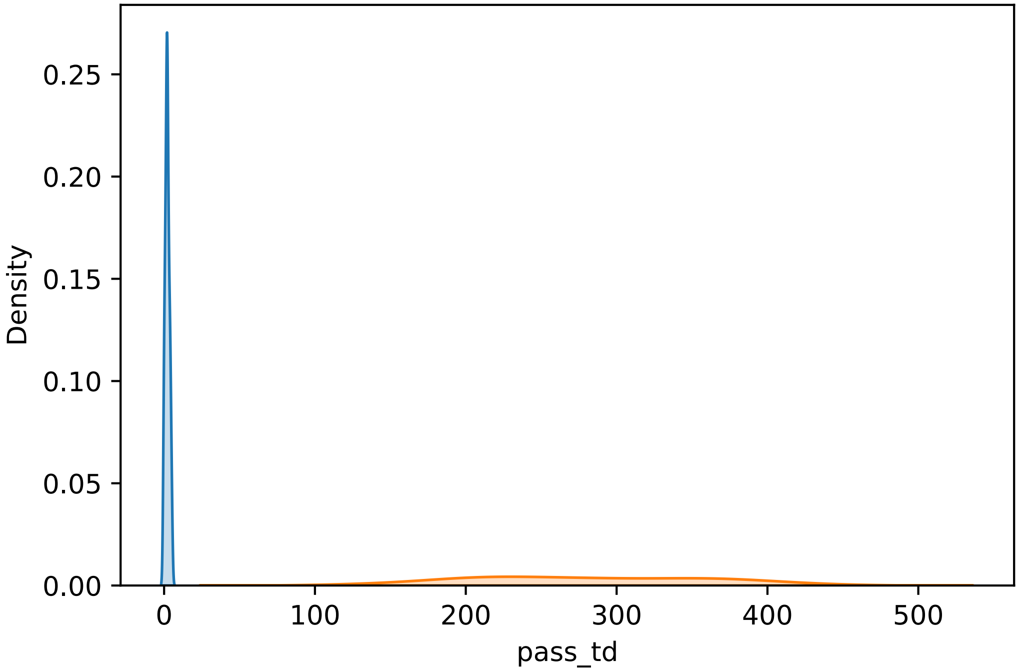
Out[15]:

```
(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

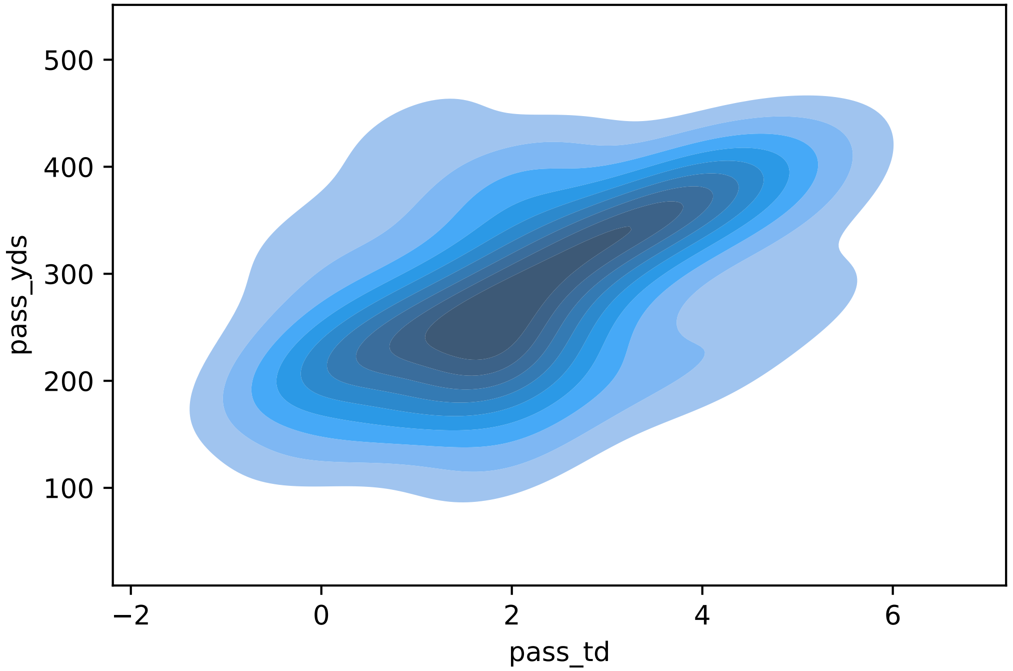
In [19]:

```python
# 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 variabl
e as a keyword arg: y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keywo
rd 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 determine how dependent each column is on another. by pass_cmp ----- pass
analyze = cluster.corr()
```

In [21]:

```
# after multiplying the cluster by the Transpose of the cluster we have a symmetric Matrix that will interm display the corrolations
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

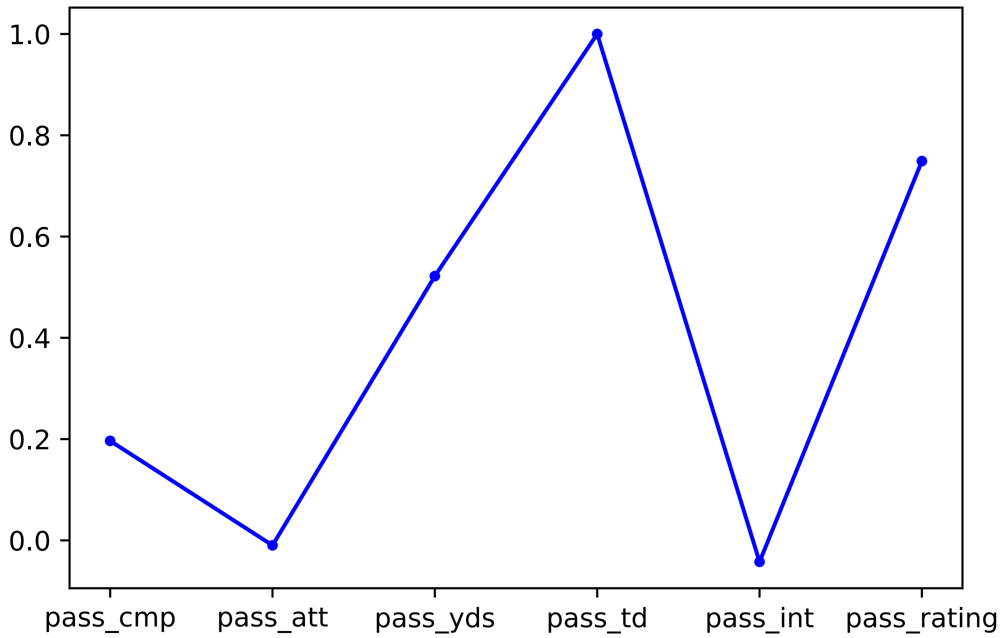
In [22]:

```
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 surmise that for Tom Brady the corralation between Brady and the amount of
#zone = brady.loc[:,'date']
py.plot(analyze["pass_td"],marker="." , color = "blue")
```

Out[23]:
```
[<matplotlib.lines.Line2D at 0x12f3854c0>]
```



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

test = brady.groupby(by=inser[0])[[inser[1],inser[2],inser[3],inser[4],inser[5],inser[6]]].sum()
```

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 [ ]: