Importing Libraries

In [92]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import chart_studio.plotly as py
from plotly import __version__
import cufflinks as cf
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
init_notebook_mode(connected = True)
cf.go_offline()
%matplotlib inline
MVPsteph = pd.read_csv("MVPseason")
preSteph = pd.read_csv("players_stats(08-09).csv")
current = pd.read_csv("players_stats(18-19).csv") # Loading Base Data and Libraries
```

2008 metrics

In [93]:
```
Attempts2008 = preSteph["3PA"].sum()
made2008 = preSteph["3P"].sum()
percent2008 = made2008 / Attempts2008
```

2018 Metrics

In [94]:
```
Attempts2018 = current["3PA"].sum()
made2018 = current["3P"].sum()
percent2018 = made2018 / Attempts2018
```

Increase Percentage over 10-year span

In [95]:
```
increase = ((Attempts2018-Attempts2008) / (Attempts2008))*100
```

Average threes attempted by centers before and after the decade

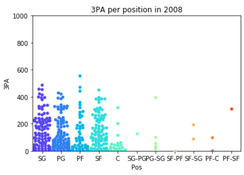
In [96]:
```
preSteph[preSteph["Pos"] == "C"]["3PA"].mean()
current[current["Pos"] == "C"]["3PA"].mean()
```

Out[96]:
```
46.09166666666667
```

Plotting threes attempted per position, 2008 and 2018

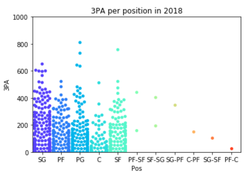
In [97]:
```
sns.swarmplot(x="Pos", y="3PA", data=preSteph,palette='rainbow')
plt.ylim(0,1000)
plt.title("3PA per position in 2008") # 2008 positions swarmplot
```

Out[97]:
```
Text(0.5, 1.0, '3PA per position in 2008')
```



In [98]:
```
sns.swarmplot(x="Pos", y="3PA", data=current,palette='rainbow')
plt.ylim(0,1000)
plt.title("3PA per position in 2018") # 2018 positions swarmplot
```

Out[98]:
```
Text(0.5, 1.0, '3PA per position in 2018')
```



Point guard scoring and rebounding increases over 10-year span

In [99]:
```
(current[current["Pos"] == "PG"]["TRB"].sum() - preSteph[preSteph["Pos"] == "PG"]["TRB"].sum()) / preSteph[preSteph[
"Pos"] == "PG"]["TRB"].sum() * 100
# Increase in PG TRB %
```

Out[99]:
```
32.93163891323401
```

In [100]:
```
(current[current["Pos"] == "PG"]["PTS"].sum() - preSteph[preSteph["Pos"] == "PG"]["PTS"].sum()) / preSteph[preSteph[
"Pos"] == "PG"]["PTS"].sum() * 100
# Increase in PG PTS %
```

Out[100]:
```
23.62299039883214
```

Cumulative Statistics since 1980

In [101]:
```
totalStats = pd.read_csv("totalLeague-80to19")
```

In [102]:
```
totalStats
```

Out[102]:

	Rk	Season	Lg	Age	Ht	Wt	G	MP	FG	FGA	...	PTS	FG%	3P%	FT%	Pace	eFG%	TOV%	ORB%	FT/FGA	ORtg
0	1	2019-20	NBA	26.0	6-6	216	971	469380	79292	172463	...	216429	0.460	0.357	0.771	100.2	0.528	12.8	22.6	0.199	110.4
1	2	2018-19	NBA	26.3	6-6	217	1230	594450	101062	219458	...	273573	0.461	0.355	0.766	100.0	0.524	12.4	22.9	0.198	110.4
2	3	2017-18	NBA	26.4	6-7	219	1230	593850	97435	211709	...	261580	0.460	0.362	0.767	97.3	0.521	13.0	22.3	0.193	108.6
3	4	2016-17	NBA	26.6	6-7	220	1230	594400	96061	210115	...	259753	0.457	0.358	0.772	96.4	0.514	12.7	23.3	0.209	108.8
4	5	2015-16	NBA	26.7	6-7	221	1230	594850	94065	208049	...	252572	0.452	0.354	0.757	95.8	0.502	13.2	23.8	0.209	106.4
5	6	2014-15	NBA	26.7	6-7	222	1230	595200	92287	205570	...	246035	0.449	0.350	0.750	93.9	0.496	13.3	25.1	0.205	105.6
6	7	2013-14	NBA	26.5	6-7	223	1230	595200	92779	204172	...	248482	0.454	0.360	0.756	93.9	0.501	13.6	25.5	0.215	106.6
7	8	2012-13	NBA	26.7	6-7	223	1229	594470	91282	201609	...	241223	0.453	0.359	0.753	92.0	0.496	13.7	26.5	0.204	105.8
8	9	2011-12	NBA	26.6	6-7	223	990	479050	72218	161225	...	190594	0.448	0.349	0.752	91.3	0.487	13.8	27.0	0.208	104.6
9	10	2010-11	NBA	26.6	6-7	223	1230	595050	91624	199790	...	244894	0.459	0.358	0.763	92.1	0.498	13.4	26.4	0.229	107.3
10	11	2009-10	NBA	26.6	6-7	222	1230	594500	92730	200989	...	247100	0.461	0.355	0.759	92.7	0.501	13.3	26.3	0.228	107.6
11	12	2008-09	NBA	26.6	6-7	221	1230	594650	91310	199054	...	245879	0.459	0.367	0.771	91.7	0.500	13.3	26.7	0.236	108.3
12	13	2007-08	NBA	26.8	6-7	220	1230	594100	91669	200501	...	245811	0.457	0.362	0.755	92.4	0.497	13.2	26.7	0.231	107.5
13	14	2006-07	NBA	26.6	6-7	219	1230	595750	89860	196073	...	242899	0.458	0.358	0.752	91.9	0.496	14.2	27.1	0.246	106.5
14	15	2005-06	NBA	26.5	6-7	220	1230	595550	88166	194315	...	238641	0.454	0.358	0.745	90.5	0.490	13.7	27.3	0.248	106.2
15	16	2004-05	NBA	26.9	6-7	220	1230	595000	88435	197626	...	239109	0.447	0.356	0.756	90.9	0.482	13.6	28.7	0.245	106.1
16	17	2003-04	NBA	27.0	6-7	219	1189	574770	83254	189802	...	222097	0.439	0.347	0.752	90.1	0.471	14.2	28.6	0.228	102.9
17	18	2002-03	NBA	27.2	6-7	219	1189	575420	84937	192109	...	226102	0.442	0.349	0.758	91.0	0.474	14.0	28.5	0.229	103.6
18	19	2001-02	NBA	27.4	6-7	218	1189	574670	86011	193263	...	227043	0.445	0.354	0.752	90.7	0.477	13.6	28.9	0.221	104.5
19	20	2000-01	NBA	27.7	6-7	216	1189	575520	84865	191664	...	225459	0.443	0.354	0.748	91.3	0.473	14.1	28.2	0.231	103.0
20	21	1999-00	NBA	27.8	6-7	216	1189	574270	87591	195220	...	231789	0.449	0.353	0.750	93.1	0.478	14.2	28.9	0.231	104.1
21	22	1998-99	NBA	27.9	6-7	215	725	350650	49549	113379	...	132792	0.437	0.339	0.728	88.9	0.466	14.6	30.2	0.240	102.2
22	23	1997-98	NBA	27.7	6-7	214	1189	575120	85383	189537	...	227269	0.450	0.346	0.737	90.3	0.478	14.5	31.4	0.243	105.0
23	24	1996-97	NBA	27.7	6-7	213	1189	575170	85777	188594	...	230433	0.455	0.360	0.738	90.1	0.493	14.8	30.8	0.236	106.7
24	25	1995-96	NBA	27.5	6-7	213	1189	574570	88096	190675	...	236619	0.462	0.367	0.740	91.8	0.499	14.7	30.6	0.243	107.6
25	26	1994-95	NBA	27.2	6-7	212	1107	535610	84105	180423	...	224518	0.466	0.359	0.737	92.9	0.500	14.6	31.4	0.245	108.3
26	27	1993-94	NBA	27.2	6-7	211	1107	533810	87064	186951	...	224737	0.466	0.333	0.734	95.1	0.485	14.3	32.2	0.232	106.3
27	28	1992-93	NBA	27.1	6-7	210	1107	535160	90056	190295	...	233073	0.473	0.336	0.754	96.8	0.491	14.0	32.0	0.243	108.0
28	29	1991-92	NBA	27.2	6-7	208	1107	535310	91371	193391	...	233155	0.472	0.331	0.759	96.6	0.487	13.6	32.9	0.232	108.2
29	30	1990-91	NBA	27.2	6-7	209	1107	535260	91551	193059	...	235370	0.474	0.320	0.765	97.8	0.487	13.9	32.3	0.245	107.9
30	31	1989-90	NBA	27.1	6-7	208	1107	534760	91914	192951	...	236882	0.476	0.331	0.764	98.3	0.489	13.9	32.1	0.250	108.1
31	32	1988-89	NBA	26.9	6-7	208	1025	495150	87060	182385	...	223797	0.477	0.323	0.768	100.6	0.489	14.5	33.0	0.249	107.8
32	33	1987-88	NBA	26.9	6-7	209	943	455040	79473	165441	...	203993	0.480	0.316	0.766	99.6	0.489	14.3	32.8	0.254	108.0
33	34	1986-87	NBA	26.6	6-7	209	943	455590	80422	167455	...	207338	0.480	0.301	0.763	100.8	0.488	14.3	33.4	0.262	108.3
34	35	1985-86	NBA	26.8	6-7	208	943	455640	81465	167166	...	207875	0.487	0.282	0.756	102.1	0.493	14.9	32.4	0.258	107.2
35	36	1984-85	NBA	26.4	6-7	207	943	455340	82532	168048	...	209041	0.491	0.282	0.764	102.1	0.496	14.9	32.9	0.252	107.9
36	37	1983-84	NBA	26.4	6-7	206	943	456490	82007	166638	...	207668	0.492	0.250	0.760	101.4	0.495	15.0	33.0	0.255	107.6
37	38	1982-83	NBA	26.1	6-7	205	943	455090	82087	169098	...	204658	0.485	0.238	0.740	103.1	0.488	15.8	33.4	0.233	104.7
38	39	1981-82	NBA	26.2	6-7	205	943	455690	81732	166418	...	204776	0.491	0.262	0.746	100.9	0.495	15.0	33.0	0.241	106.9
39	40	1980-81	NBA	26.2	6-7	205	943	455340	81025	166769	...	203886	0.486	0.245	0.751	101.8	0.489	15.6	33.5	0.245	105.5

40 rows × 32 columns

In [103]:
```
totalStats = totalStats.reindex(index=totalStats.index[::-1]) # Reversing row order for better visuals
```

Linear regression prediction model (training, testing, fit, predictions)

In [104]:
```
stephYearly = pd.read_csv("StephYearlyStats")
stephAvgPredict = pd.read_csv("stephAvgPredict") # Loading more data sets
```

In [105]:
```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
lm = LinearRegression()
# Libraries and creating model
```

In [106]:
```
X_train = stephYearly[['GS', 'MP', 'FG', 'FGA', 'FG%',
       '3P', '3PA', '3P%', '2P', '2PA', '2P%', 'eFG%', 'FT', 'FTA', 'FT%',
       'ORB', 'DRB', 'STL', 'BLK', 'TOV', 'PF']]
y_train = stephYearly[["TRB", "AST", "PTS"]] # Training data is his 5-year compiledd statistics

X_test =  stephAvgPredict # Test data is an average set, used for prediction
```

In [107]:
```
X_test = stephAvgPredict[['GS', 'MP', 'FG', 'FGA', 'FG%',
       '3P', '3PA', '3P%', '2P', '2PA', '2P%', 'eFG%', 'FT', 'FTA', 'FT%',
       'ORB', 'DRB', 'STL', 'BLK', 'TOV', 'PF']] # Test set, used to predict major statistical categories
```

In [108]:
```
lm.fit(X_train,y_train) # Fitting the model to the data set
```

Out[108]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [109]:
```
prediction = lm.predict(X_test) # Predicting based on adjusted coefficients
```

In [110]:
```
for i in range(0, 3):
    prediction[0][i] = (prediction[0][i])/(71.6) # Standardizing to per-game statistics
```

Final Prediction

In [111]:
```
prediction
```

Out[111]:
```
array([[ 5.07857979, 10.7077409 , 23.29896939]])
```