In [37]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [38]:
```
df = pd.read_csv('Full_scrape.csv')
```

In [45]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3638 entries, 0 to 3637
Data columns (total 50 columns):
Player    3638 non-null object
Pos       3638 non-null object
Age       3638 non-null int64
Tm        3638 non-null object
G         3638 non-null int64
GS        3638 non-null int64
MP        3638 non-null int64
FG        3638 non-null int64
FGA       3638 non-null int64
FG%       3638 non-null float64
3P        3638 non-null int64
3PA       3638 non-null int64
3P%       3638 non-null float64
2P        3638 non-null int64
2PA       3638 non-null int64
2P%       3638 non-null float64
eFG%      3638 non-null float64
FT        3638 non-null int64
FTA       3638 non-null int64
FT%       3638 non-null float64
ORB       3638 non-null int64
DRB       3638 non-null int64
TRB       3638 non-null int64
AST       3638 non-null int64
STL       3638 non-null int64
BLK       3638 non-null int64
TOV       3638 non-null int64
PF        3638 non-null int64
PER       3638 non-null float64
TS%       3638 non-null float64
3PAr      3638 non-null float64
FTr       3638 non-null float64
ORB%      3638 non-null float64
DRB%      3638 non-null float64
TRB%      3638 non-null float64
AST%      3638 non-null float64
STL%      3638 non-null float64
BLK%      3638 non-null float64
TOV%      3638 non-null float64
USG%      3638 non-null float64
OWS       3638 non-null float64
DWS       3638 non-null float64
WS        3638 non-null float64
WS/48     3638 non-null float64
OBPM      3638 non-null float64
DBPM      3638 non-null float64
BPM       3638 non-null float64
VORP      3638 non-null float64
Year      3638 non-null int64
Salary    3638 non-null float64
dtypes: float64(26), int64(21), object(3)
memory usage: 1.4+ MB
```

In [39]:
```
df.head(10)
```

Out[39]:

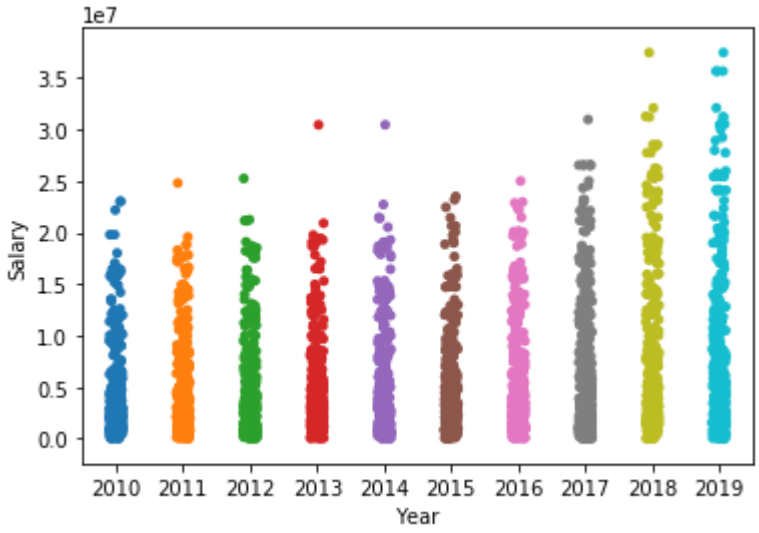
	Player	Pos	Age	Tm	G	GS	MP	FG	FGA	FG%	...	OWS	DWS	WS	WS/48	OBI
0	Arron Afflalo	SG	24	DEN	82	75	2221	272	585	0.465	...	2.8	1.4	4.3	0.092	-(
1	LaMarcus Aldridge	PF	24	POR	78	78	2922	579	1169	0.495	...	5.5	3.3	8.8	0.145	
2	Joe Alexander	SF	23	CHI	8	0	29	1	6	0.167	...	0.0	0.0	0.0	0.030	-9
3	Malik Allen	PF	31	DEN	51	3	456	46	116	0.397	...	-0.3	0.3	0.1	0.009	-4
4	Tony Allen	SG	28	BOS	54	8	889	129	253	0.510	...	0.2	1.6	1.9	0.100	-
5	Rafer Alston	PG	33	TOT	52	38	1421	155	446	0.348	...	-1.6	1.2	-0.4	-0.013	-3
6	Lou Amundson	PF	27	PHO	79	0	1168	152	276	0.551	...	1.5	1.3	2.9	0.118	-
7	Chris Andersen	PF	31	DEN	76	0	1695	142	251	0.566	...	3.2	2.7	5.9	0.166	-(
8	David Andersen	C	29	HOU	63	0	891	147	340	0.432	...	0.4	0.9	1.3	0.070	-2
9	Ryan Anderson	PF	21	ORL	63	6	911	169	388	0.436	...	1.7	1.4	3.1	0.161	2

10 rows × 50 columns

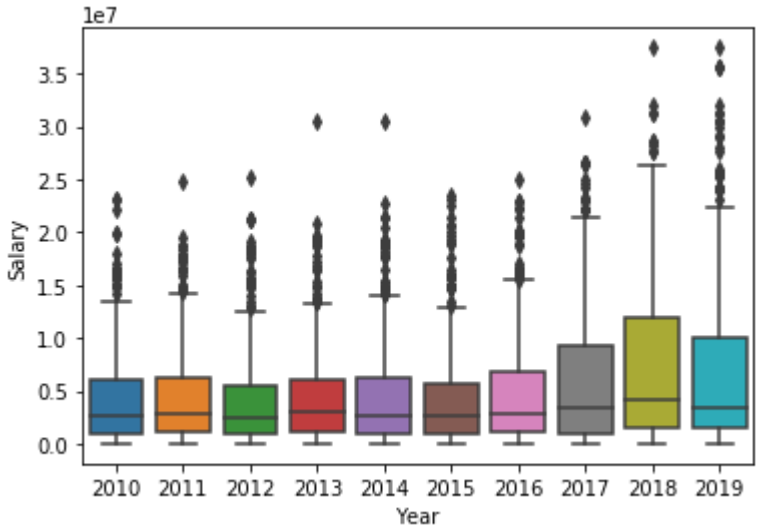
In [52]:
```
# Get columns
columns = df.columns
columns
```

Out[52]:
```
Index(['Player', 'Pos', 'Age', 'Tm', 'G', 'GS', 'MP', 'FG', 'FGA', 'FG%', '3P',
       '3PA', '3P%', '2P', '2PA', '2P%', 'eFG%', 'FT', 'FTA', 'FT%', 'ORB',
       'DRB', 'TRB', 'AST', 'STL', 'BLK', 'TOV', 'PF', 'PER', 'TS%', '3PAr',
       'FTr', 'ORB%', 'DRB%', 'TRB%', 'AST%', 'STL%', 'BLK%', 'TOV%', 'USG%',
       'OWS', 'DWS', 'WS', 'WS/48', 'OBPM', 'DBPM', 'BPM', 'VORP', 'Year',
       'Salary'],
      dtype='object')
```

In [75]:
```
# Salary x Year
sns.stripplot(x=df.Year, y=df.Salary);
```



In [76]:
```
# Salary x Year
sns.boxplot(x=df.Year, y=df.Salary);
```

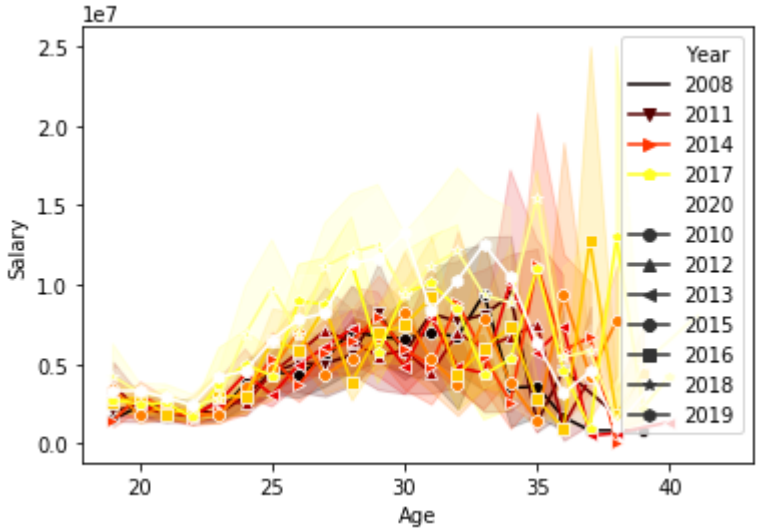


In [59]:
```
#plot data
df_grouped = df.groupby([df.Year])
#print(df_grouped.groups)
g2010 = df_grouped.get_group(2010)
g2011 = df_grouped.get_group(2011)
g2012 = df_grouped.get_group(2012)
g2013 = df_grouped.get_group(2013)
g2014 = df_grouped.get_group(2014)
g2015 = df_grouped.get_group(2015)
g2016 = df_grouped.get_group(2016)
g2017 = df_grouped.get_group(2017)
g2018 = df_grouped.get_group(2018)
g2019 = df_grouped.get_group(2019)
```

In [62]:
```
yearData = [g2010, g2011, g2012, g2013, g2014, g2015, g2016, g2017, g2018, g2019]
for eachYear in yearData:
    print(eachYear['Salary'].agg(np.mean))
```

```
4495217.947521866
4532404.076246334
4315241.95
4512389.885793872
4549454.985465116
4442079.382978723
4992980.581151833
6149016.659949622
7596788.384615385
7067359.482412061
```

In [86]:
```
#sns.lineplot(x = df.Age, y = df.Salary)
markers = ('o', 'v', '^', '<', '>', '8', 's', 'p', '*', 'h')
sns.lineplot(data = df, x = "Age", y = "Salary", hue = "Year", style = "Year", pa
plt.show()
```



In [42]:
```
# Subset by year
y2010 = df.loc[(df.Year == 2010)]
y2011 = df.loc[(df.Year == 2011)]
y2012 = df.loc[(df.Year == 2012)]
y2013 = df.loc[(df.Year == 2013)]
y2014 = df.loc[(df.Year == 2014)]
y2015 = df.loc[(df.Year == 2015)]
y2016 = df.loc[(df.Year == 2016)]
y2017 = df.loc[(df.Year == 2017)]
y2018 = df.loc[(df.Year == 2018)]
y2019 = df.loc[(df.Year == 2019)]

# Find average per year
avg2010 = y2010.Year.mean()
avg2011 = y2011.Year.mean()
avg2012 = y2012.Year.mean()
avg2013 = y2013.Year.mean()
avg2014 = y2014.Year.mean()
avg2015 = y2015.Year.mean()
avg2016 = y2016.Year.mean()
avg2017 = y2017.Year.mean()
avg2018 = y2018.Year.mean()
avg2019 = y2019.Year.mean()
```

In [ ]:
```
ax = plt.gca()

df.plot(kind='line',x='df.Age',y='df.Salary',ax=ax)
```