Final Proposal: Pro Bowlers and their ages, teams, and contracts

This project was completed by El-Khalil Osman here in partial fulfilment of ECON-UB.0232, Data Bootcamp, Spring 2019. I certify that the NYU Stern Honor Code applies to this project. In particular, I have: Clearly acknowledged the work and efforts of others when submitting written work as our own. The incorporation of the work of others–including but not limited to their ideas, data, creative expression, and direct quotations (which should be designated with quotation marks), or paraphrasing thereof– has been fully and appropriately referenced using notations both in the text and the bibliography. And I understand that: Submitting the same or substantially similar work in multiple courses, either in the same semester or in a different semester, without the express approval of all instructors is strictly forbidden. I acknowledge that a failure to abide by NYU Stern Honor Code will result in a failing grade for the project and course.

Principle Investigator: Khalil Osman

Email: eko250@nyu.edu

Background:

This project will measure the top NFL players (Pro Bowlers) and look at how they are actually paid compared to non-Pro Bowlers. Over the years there have been many complaints regarding NFL teams not paying their top players accordingly and taking advantage of the system. For example, Leveon Bell who was one of the top NFL running backs in 2017 completely sat out of the 2018 season because he refused to be franchise tagged. The franchise tag is basically a ploy created by the NFL that allows them to keep their top player for a year if he is turning to free agency. Many individuals simply assume because Pro Bowlers are the best players that they are getting paid the most but this is not always the case.

The majority of my information will come from Pro Football Reference (https://www.pro-football-reference.com/ (https://www.pro-football-reference.com/)) which gives me access to the different salaries and NFL Pro Bowlers for 2019. Additionally, I have access to the teams the players play for and their ages.

The sections within the project will be:

Basic statistics and data such as the different players, names, age and team.

I will then split up the 2013 NFL salaries I have listed into different groups of players and represent this as a visualization. This will be based off of income so the highest paid players will be the first group, the second highest paid players will be the second group, and so on.

Finally, I will compare the different groups on a chart and see which one of them contained the most pro bowlers in 2019 using the data. I will then narrow down that data more to see where the bracket stands in terms of income and whether the best players are truly being paid the most.

Data Report

Overview:

As stated earlier, the data behind my entire project comes from Pro Football Reference (https://www.pro-football-reference.com/ (https://www.pro-football-reference.com/)) and the site provides access to the pro bowlers and their salaries and some other variables.

I will download the files and transport them as excel worksheets. Once I have them placed, I will get rid of the data I do not need and narrow down the focus on the charts.

I will add the average age of NFL Pro Bowlers through a visualization and which team had the most pro bowlers.

Finally, I will compare the salaries of Pro Bowlers and see if they are getting paid the highest.

Requisite Packages:

In [2]:
```
import pandas as pd # Import pandas as pd adds the pandas package to the noteb
ook as "pd" and it allows for the functionality of the pandas package
import matplotlib.pyplot as plt # Import matplotlib.pyplot as plt adds the mat
plotlib.pyplot package to the notebook as "plt" and it allows for the function
ality of the matplotlib.pyplot package
import numpy as np # For performing numerical analysis
import seaborn as sns
```

Bringing in the data: Below I export in the data as a csv file from my saved documents that I got from Pro Football Reference. The first chart is the 2019 Pro Bowl roster and the second is the 2018-2019 base pay for the all NFL players

In [3]:
```
df = pd.read_csv("ProBowlers.csv")
df
```

Out[3]:

	Pos	Player	Conf	Tm	Age	Yrs	G	GS	Cmp	Att	...	Att.1	Yds.1	TD.1	Rec
0	QB	Patrick Mahomes	AFC	KAN	23	1	16	16	383	580	...	60	272	2	0
1	QB	Drew Brees%	NFC	NOR	39	17	15	15	364	489	...	31	22	4	1
2	QB	Mitchell Trubisky+	NFC	CHI	24	1	14	14	289	434	...	68	421	3	0
3	QB	Andrew Luck+	AFC	IND	29	6	16	16	430	639	...	46	148	0	1
4	QB	Dak Prescott+	NFC	DAL	25	2	16	16	356	526	...	75	305	6	0
5	QB	Aaron Rodgers%	NFC	GNB	35	13	16	16	372	597	...	43	269	2	0
6	QB	Deshaun Watson+	AFC	HOU	23	1	16	16	345	505	...	99	551	5	0
7	QB	Tom Brady%	AFC	NWE	41	18	16	16	375	570	...	23	35	2	1
8	QB	Jared Goff%	NFC	LAR	24	2	16	16	364	561	...	43	108	2	0
9	QB	Philip Rivers%	AFC	LAC	37	14	16	16	347	508	...	18	7	0	0
10	QB	Russell Wilson+	NFC	SEA	30	6	16	16	280	427	...	67	376	0	1
11	RB	Todd Gurley%	NFC	LAR	24	3	14	14	0	0	...	256	1251	17	59
12	RB	Saquon Barkley	NFC	NYG	21	Rook	16	16	0	0	...	261	1307	11	91

	Pos	Player	Conf	Tm	Age	Yrs	G	GS	Cmp	Att	...	Att.1	Yds.1	TD.1	Rec
13	RB	Ezekiel Elliott	NFC	DAL	23	2	15	15	0	0	...	304	1434	6	77
14	RB	Alvin Kamara+	NFC	NOR	23	1	15	13	0	0	...	194	883	14	81
15	RB	Phillip Lindsay%	AFC	DEN	24	Rook	15	8	0	0	...	192	1037	9	35
16	RB	Lamar Miller+	AFC	HOU	27	6	14	14	0	0	...	210	973	5	25
17	RB	James Conner	AFC	PIT	23	1	13	12	0	0	...	215	973	12	55
18	RB	Melvin Gordon	AFC	LAC	25	3	12	12	0	0	...	175	885	10	50
19	FB	Anthony Sherman	AFC	KAN	30	7	16	1	0	0	...	1	2	0	8
20	FB	Kyle Juszczyk	NFC	SFO	27	5	16	14	0	0	...	8	30	0	30
21	WR	DeAndre Hopkins%	AFC	HOU	26	5	16	16	0	1	...	1	-7	0	115
22	WR	Michael Thomas%	NFC	NOR	25	2	16	16	0	0	...	0	0	0	125
23	WR	Tyreek Hill	AFC	KAN	24	2	16	16	0	0	...	22	151	1	87

	Pos	Player	Conf	Tm	Age	Yrs	G	GS	Cmp	Att	...	Att.1	Yds.1	TD.1	Rec
24	WR	Julio Jones%	NFC	ATL	29	7	16	16	0	0	...	2	12	0	113
25	WR	Keenan Allen	AFC	LAC	26	5	16	14	0	0	...	9	75	0	97
26	WR	Amari Cooper+	NFC	2TM	24	3	15	15	0	0	...	2	20	0	75
27	WR	Jarvis Landry+	AFC	CLE	26	4	16	14	1	2	...	3	60	1	81
28	WR	Davante Adams	NFC	GNB	26	4	15	15	0	0	...	0	0	0	111
29	WR	Adam Thielen	NFC	MIN	28	5	16	16	0	0	...	5	30	0	113
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
89	ILB	Leighton Vander Esch+	NFC	DAL	22	Rook	16	11	0	0	...	0	0	0	0
90	ILB	C.J. Mosley	AFC	BAL	26	4	15	15	0	0	...	0	0	0	0
91	ILB	Benardrick McKinney	AFC	HOU	26	3	16	16	0	0	...	0	0	0	0
92	CB	Kyle Fuller	NFC	CHI	26	4	16	16	0	0	...	0	0	0	0
93	CB	Stephon Gilmore%	AFC	NWE	28	6	16	16	0	0	...	0	0	0	0

	Pos	Player	Conf	Tm	Age	Yrs	G	GS	Cmp	Att	...	Att.1	Yds.1	TD.1	Rec
94	CB	Patrick Peterson	NFC	ARI	28	7	16	16	0	0	...	0	0	0	0
95	CB	Byron Jones	NFC	DAL	26	3	16	16	0	0	...	0	0	0	0
96	CB	Xavien Howard	AFC	MIA	25	2	12	12	0	0	...	0	0	0	0
97	CB	Denzel Ward	AFC	CLE	21	Rook	13	12	0	0	...	0	0	0	0
98	CB	Chris Harris+	AFC	DEN	29	7	12	12	0	0	...	0	0	0	0
99	CB	Darius Slay	NFC	DET	27	5	15	15	0	0	...	0	0	0	0
100	CB	Jalen Ramsey	AFC	JAX	24	2	16	16	0	0	...	0	0	0	0
101	SS	Jamal Adams	AFC	NYJ	23	1	16	16	0	0	...	0	0	0	0
102	SS	Landon Collins%	NFC	NYG	24	3	12	12	0	0	...	0	0	0	0
103	SS	Malcolm Jenkins+	NFC	PHI	31	9	16	16	0	0	...	0	0	0	0
104	FS	Eddie Jackson	NFC	CHI	26	1	14	14	0	0	...	0	0	0	0

	Pos	Player	Conf	Tm	Age	Yrs	G	GS	Cmp	Att	...	Att.1	Yds.1	TD.1	Rec
105	FS	Derwin James	AFC	LAC	22	Rook	16	16	0	0	...	0	0	0	0
106	FS	Harrison Smith	NFC	MIN	29	6	16	16	0	0	...	0	0	0	0
107	FS	Eric Weddle	AFC	BAL	33	11	16	16	0	0	...	0	0	0	0
108	LS	Casey Kreiter	AFC	DEN	28	2	16	0	0	0	...	0	0	0	0
109	LS	Don Muhlbach	NFC	DET	37	14	16	0	0	0	...	0	0	0	0
110	K	Aldrick Rosas	NFC	NYG	24	1	16	0	0	0	...	0	0	0	0
111	K	Jason Myers	AFC	NYJ	27	3	16	0	0	0	...	0	0	0	0
112	P	Michael Dickson	NFC	SEA	22	Rook	16	0	0	0	...	1	9	0	0
113	P	Brett Kern	AFC	TEN	32	10	16	0	0	0	...	0	0	0	0
114	PR	Andre Roberts	AFC	NYJ	30	8	16	1	0	0	...	2	20	0	10

	Pos	Player	Conf	Tm	Age	Yrs	G	GS	Cmp	Att	...	Att.1	Yds.1	TD.1	Rec
115	PR	Tarik Cohen	NFC	CHI	23	1	16	7	1	1	...	99	444	3	71
116	ST	Adrian Phillips	AFC	LAC	26	4	16	7	0	0	...	0	0	0	0
117	ST	Cory Littleton%	NFC	LAR	25	2	16	16	0	0	...	0	0	0	0
118	ST	Michael Thomas+	NFC	NYG	28	5	16	6	0	0	...	0	0	0	0

119 rows × 23 columns

In [4]:
```
df2 = pd.read_csv("AllPlayers.csv")
df2
```

Out[4]:

	Rk	Player	Pos	Tm	Salary
0	1.0	Kirk Cousins\CousKi00	QB	MIN	$22,500,000
1	2.0	Demarcus Lawrence\LawrDe00	DE	DAL	$17,143,000
2	NaN	Ezekiel Ansah\AnsaEz00	DE	DET	$17,143,000
3	4.0	Russell Wilson\WilsRu00	QB	SEA	$15,500,000
4	5.0	Calais Campbell\CampCa99	DE	JAX	$15,000,000
5	6.0	Justin Houston\HousJu00	LB	KAN	$14,750,000
6	7.0	Cam Newton\NewtCa00	QB	CAR	$14,500,000
7	8.0	Ndamukong Suh\SuhxNd99	DT	LAR	$14,000,000
8	9.0	Andy Dalton\DaltAn00	QB	CIN	$13,700,000
9	10.0	Malik Jackson\JackMa02	DT	JAX	$13,500,000
10	11.0	Josh Norman\NormJo01	CB	WAS	$13,300,000
11	12.0	A.J. Bouye\BouyA.00	CB	JAX	$13,000,000
12	NaN	Alex Smith\SmitAl03	QB	WAS	$13,000,000
13	14.0	Olivier Vernon\VernOl00	DE	NYG	$12,750,000
14	15.0	Chandler Jones\JoneCh03	DE	ARI	$12,500,000
15	NaN	Kawann Short\ShorKa00	DT	CAR	$12,500,000
16	NaN	DeAndre Hopkins\HopkDe00	WR	HOU	$12,500,000
17	18.0	Jadeveon Clowney\ClowJa00	DE	HOU	$12,306,000
18	NaN	Anthony Barr\BarrAn00	OLB	MIN	$12,306,000
19	20.0	Gerald McCoy\McCoGe99	DT	TAM	$12,250,000
20	21.0	Andrew Luck\LuckAn00	QB	IND	$12,000,000
21	NaN	Russell Okung\OkunRu20	T	LAC	$12,000,000
22	NaN	Ben Roethlisberger\RoetBe00	QB	PIT	$12,000,000
23	NaN	Joe Flacco\FlacJo00	QB	BAL	$12,000,000
24	25.0	Fletcher Cox\CoxxFl00	DT	PHI	$11,500,000
25	26.0	Lamarcus Joyner\JoynLa00	CB	LAR	$11,287,000
26	27.0	Jason Pierre-Paul\PierJa99	DE	TAM	$11,250,000
27	NaN	Melvin Ingram\IngrMe00	DE	LAC	$11,250,000
28	29.0	Eric Fisher\FishEr00	OT	KAN	$11,150,000
29	30.0	Patrick Peterson\PetePa00	CB	ARI	$11,000,000
...	...	...	...	...	...
1983	NaN	Kyle Wilson\WilsKy00	LB	LAC	$84,705
1984	NaN	Andrew East\EastAn01	LS	WAS	$84,705
1985	1986.0	Craig Mager\MageCr00	CB	LAC	$82,941
1986	1987.0	Jeremy Langford\LangJe00	RB	ATL	$74,118

	Rk	Player	Pos	Tm	Salary
1987	NaN	Ufomba Kamalu\KamaUf00	DE	NWE	$74,118
1988	1989.0	Steven Mitchell\MitcSt02	WR	HOU	$68,400
1989	1990.0	Breon Borders\BordBr00	CB	JAX	$65,294
1990	NaN	Harlan Miller\MillHa00	CB	WAS	$65,294
1991	NaN	Victor Bolden\BoldVi00	WR	2TM	$65,294
1992	NaN	Darius Jackson\JackDa01	RB	2TM	$65,294
1993	NaN	Nigel Harris\HarrNi00	OLB	TEN	$65,294
1994	NaN	Jeremiah Ledbetter\LedbJe00	DT	TAM	$65,294
1995	1996.0	Lavon Coleman\ColeLa02	RB	GNB	$56,470
1996	NaN	Donnie Ernsberger\ErnsDo00	TE	TAM	$56,470
1997	NaN	Elijah Nkansah\NkanEl00	T	SEA	$56,470
1998	NaN	John Atkins\AtkiJo01	DT	DET	$56,470
1999	NaN	Rico Gafford\GaffRi00	WR	OAK	$56,470
2000	NaN	Kyle Allen\AlleKy00	QB	CAR	$56,470
2001	NaN	Allen Lazard\LazaAl00	WR	GNB	$56,470
2002	NaN	Jeremy Reaves\ReavJe00	FS	WAS	$56,470
2003	2004.0	Clive Walford\WalfCl00	TE	NYJ	$41,471
2004	2005.0	Adolphus Washington\WashAd00	DT	2TM	$37,059
2005	NaN	Destiny Vaeao\VaeaDe01	DT	PHI	$37,059
2006	NaN	Aaron Wallace\WallAa00	OLB	TEN	$37,059
2007	2008.0	Max McCaffrey\McCaMa01	WR	SFO	$32,647
2008	2009.0	Mitchell Loewen\LoewMi00	DE	NOR	$28,235
2009	NaN	JJ Jones\JoneJJ00	WR	2TM	$28,235
2010	NaN	Malachi Dupre\DuprMa01	WR	ARI	$28,235
2011	NaN	Alex Carter\CartAl01	CB	WAS	$28,235
2012	NaN	Matthew McCrane\McCrMa00	K	3TM	$28,235

2013 rows × 5 columns

In [5]:
```
allplayers = df2#.filter(["Player", "Pos", "Tm", "Rk", "Salary"])  #Taking onl
y the data needed
```

In [6]:
```
proplayers = df#.filter(["Pos", "Player", "Tm", "Age", "Yrs", "Yds", "Cmp"])
 #Taking only the data needed
```

I now have the data to work with and can begin going more in depth.

Removing extra characters from proplayers and changing Yrs "Rook" to 0:

In [7]:
```
for index,player in proplayers.iterrows():
    name = player["Player"]
    if (name[-1]=="+" or name[-1]=="%"):
        name = name[:-1]
        proplayers.at[index,"Player"] = name

proplayers.loc[proplayers["Yrs"]=="Rook", "Yrs"] = 0
```

Removing extra charaters from all players:

In [8]:
```
for index,player in allplayers.iterrows():
    name = player["Player"]
    indexofslash = name.find('\\')
    name = name[:indexofslash]
    allplayers.at[index,"Player"] = name
```

Visualizing pro bowl players vs regular player salaries

turn salary into number:

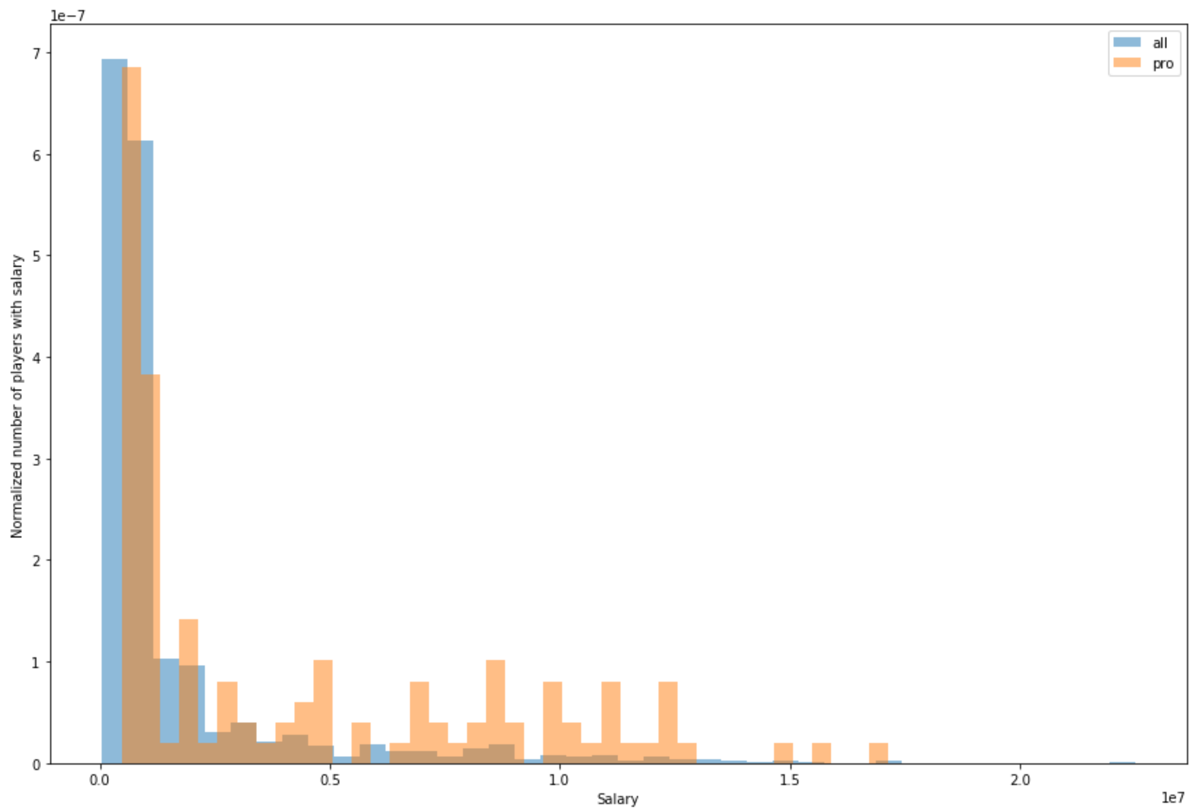
In [9]:
```
for index,player in allplayers.iterrows():
    salary = player["Salary"]
    nocomma = salary.replace(",","")
    nodollar = nocomma[1:]
    salarynumber = int(nodollar)
    allplayers.at[index,"Salary"] = salarynumber
```

Getting salary and ranks for pro players:

In [10]:
```
proplayers["Salary"] = 0
proplayers["Rk"] = np.NaN

for index, player in proplayers.iterrows():
    pplayer = allplayers.loc[allplayers["Player"]==player["Player"]]
    proplayers.at[index, "Salary"] = pplayer["Salary"].values[0]
    prank = pplayer["Rk"].values[0]

    if not np.isnan(prank):
        proplayers.at[index, "Rk"] = pplayer["Rk"].values[0]
```

In [15]:
```
plt.figure(figsize=(15,10))

plt.hist(allplayers["Salary"].tolist(),density=True, bins=40,alpha=0.5, label=
"all")
plt.hist(proplayers["Salary"].tolist(),density=True, bins=40, alpha=0.5, label
="pro")
plt.xlabel("Salary")
plt.ylabel("Normalized number of players with salary")
plt.legend()
#plt.savefig("graph1.png", bbox_inches = "tight", dpi = 1200)
```

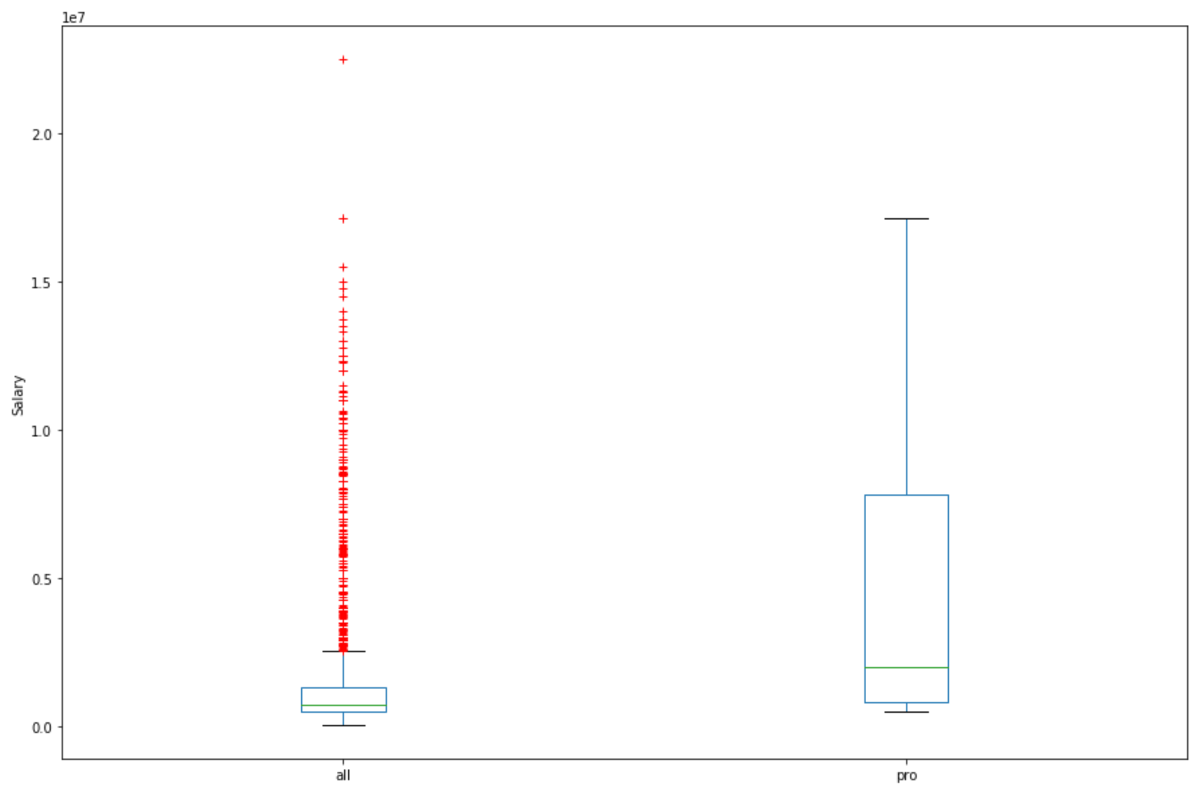
Out[15]: <matplotlib.legend.Legend at 0x1f9002f76a0>



Here are salaries plotted as a normalized histogram for both, as can be deduced from the graph, the average salary for a pro bowl player is slightly higher. Pro bowl players are also more likely to have a higher salary based on the histogram.

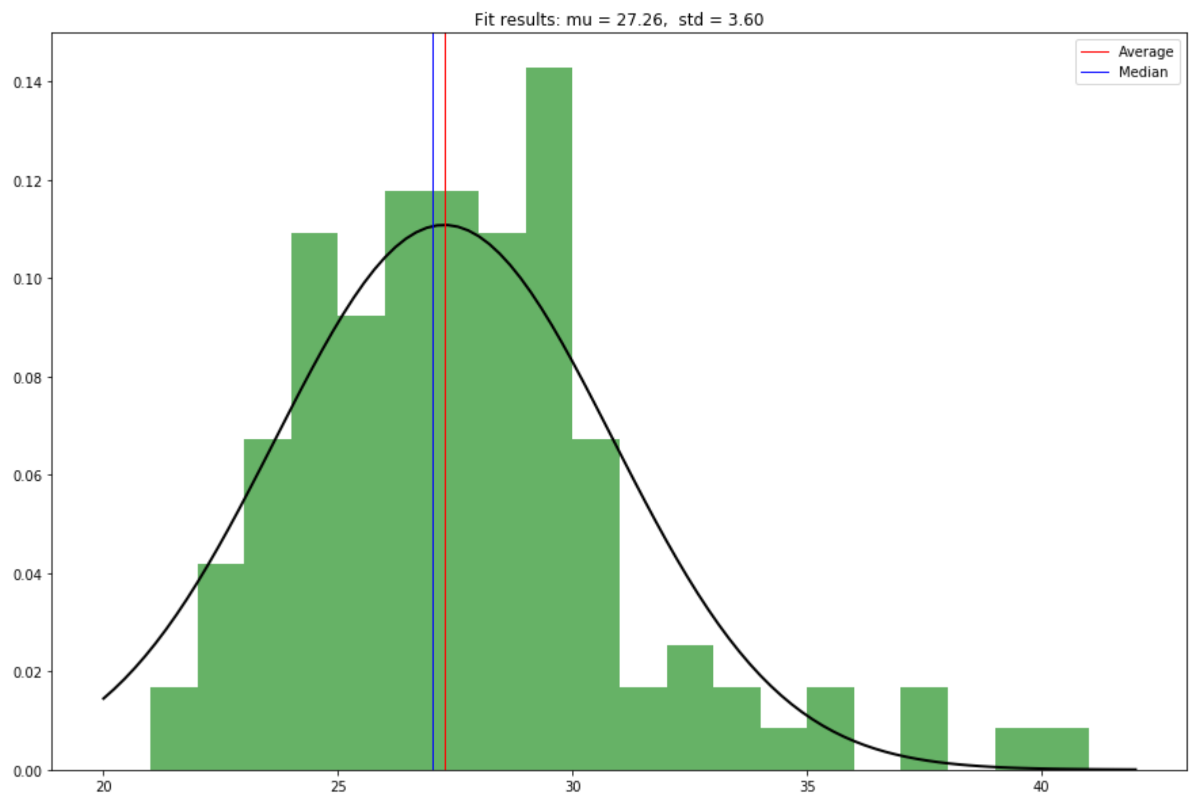
In [19]:
```
#allplayers["Salary"].describe().plot(kind='box')
#proplayers["Salary"].describe().plot(kind='box')

pd.DataFrame.from_dict({"all":allplayers["Salary"], "pro":proplayers["Salary"
]}).plot.box(figsize=(15,10),sym='r+')
plt.ylabel("Salary")
#plt.savefig("graph2.png", bbox_inches = "tight", dpi = 1200)
```

Out[19]:
```
Text(0, 0.5, 'Salary')
```



As can be seen from the plot for all players salary, there are many outliers (the different markers) This indicates that 97.5% of all players have salaries below ~$5 million$. However, the pro bowl players have a higher average salary, and generally don't have outliers in terms of salary (a.k.a all salaries are within 2 IQR from the mean).

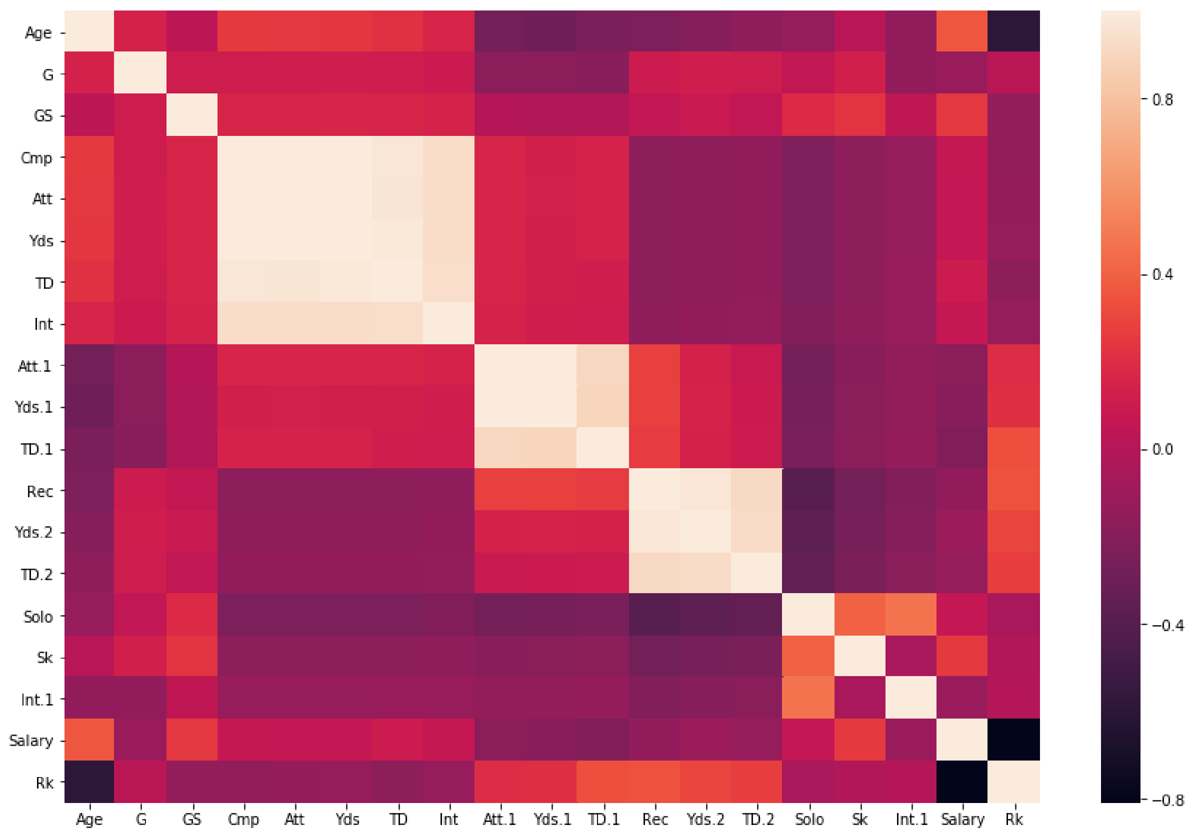
In [27]:
```
from scipy.stats import norm

mu, std = norm.fit(proplayers["Age"].values.tolist())

plt.figure(figsize=(15,10))

plt.hist(proplayers["Age"].tolist(), bins=20, density=True, alpha=0.6, color=
'g')

xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
p = norm.pdf(x, mu, std)
plt.plot(x, p, 'k', linewidth=2)
title = "Fit results: mu = %.2f,  std = %.2f" % (mu, std)
plt.title(title)

plt.axvline(x=proplayers["Age"].mean(),
          color="r", label="Average",linewidth=1)

plt.axvline(x=proplayers["Age"].median(),
          color="b", label="Median",linewidth=1)

plt.legend()

plt.show()
#plt.savefig("graph5.png", bbox_inches = "tight", dpi = 1200)
```



Here is a probability distribution of pro bowl players ages assuming age is a random gaussian variable. The median age is 27, and the average is 27.26

In [31]:
```
#proplayers[["Age","Salary","Rk"]].describe().astype('int')
plt.figure(figsize=(15,10))
corr = proplayers.corr()
sns.heatmap(corr,
            xticklabels=corr.columns.values,
            yticklabels=corr.columns.values)
#plt.savefig("graph6.png", bbox_inches = "tight", dpi = 1200)
```

Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0x1f9005c28d0>



Heatmap for the correlation matrix between all columns for pro bowl players. Yards and completions are very heavily correlated. Surprising to see that ranking is negatively correlated with most other data fields, and pretty uncorrelated to any field. It seems that it correlates a bit to Touchdowns and receptions throughout the season (depending by position of course), but not as much as I would have initially thought.

Key

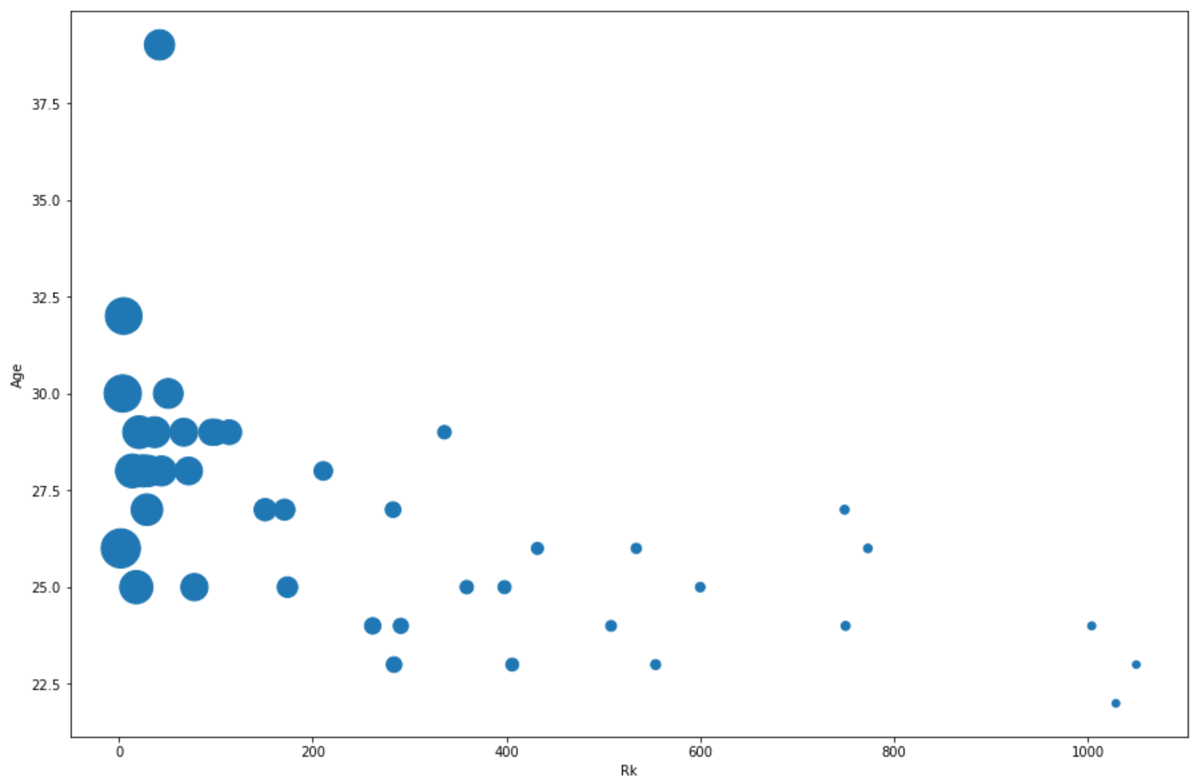
=================

+1 heavily correlated

-1 negatively correlated

0 not correlated

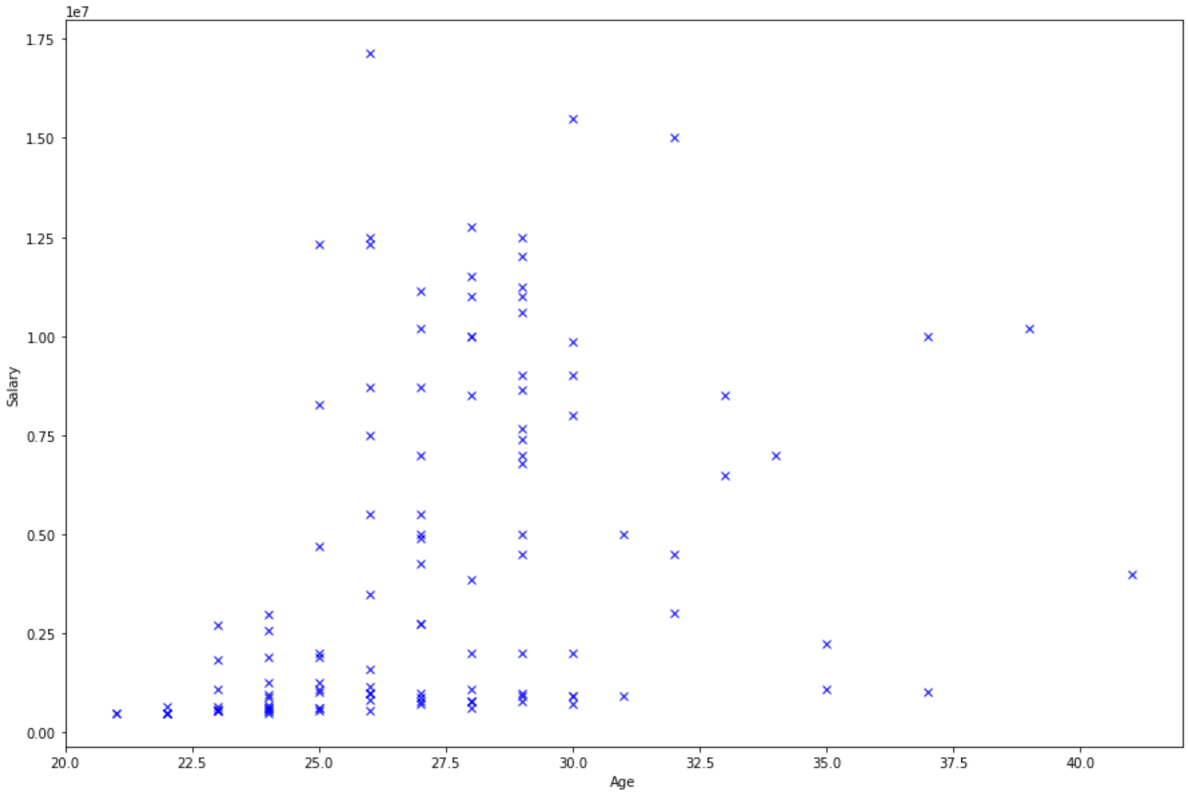
In [15]:
```
proplayers.plot.scatter(x="Rk", y="Age", s=proplayers["Salary"] //20000, figsize = (15,10));
```



Bubble plot with bubble size representing Salary. Higher rank has a clear correlation with salary, while age seems to not play a big role in salary.

In [16]:
```
plt.figure(figsize=(15,10))
plt.plot(proplayers["Age"], proplayers["Salary"], 'bx')
plt.xlabel("Age")
plt.ylabel("Salary")
```

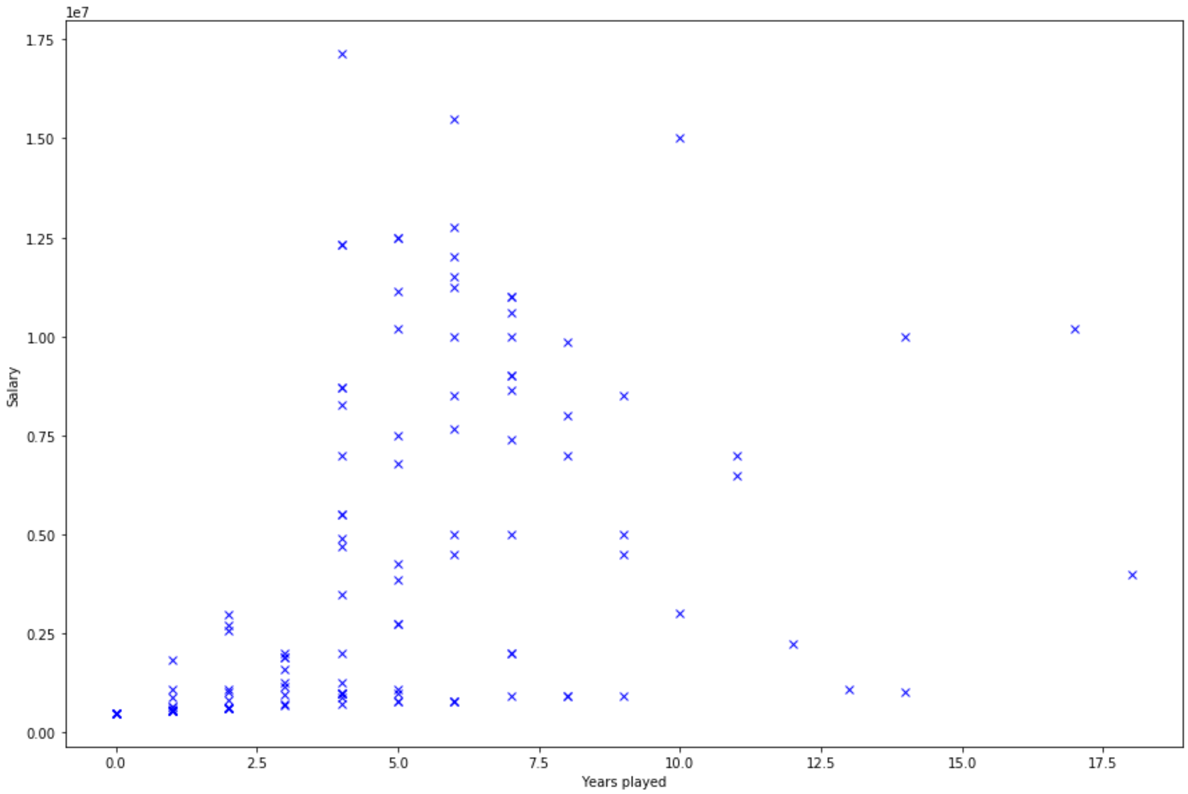
Out[16]: `Text(0, 0.5, 'Salary')`



Correlation between Age and salary. After 25 years, there is no significant correlation, however, younger players are generally paid less. This is mostly due to rookie contracts.

In [17]:
```
plt.figure(figsize=(15,10))
plt.xlabel("Years played")
plt.ylabel("Salary")

plt.plot(proplayers["Yrs"].astype('int'), proplayers["Salary"], 'bx')
```

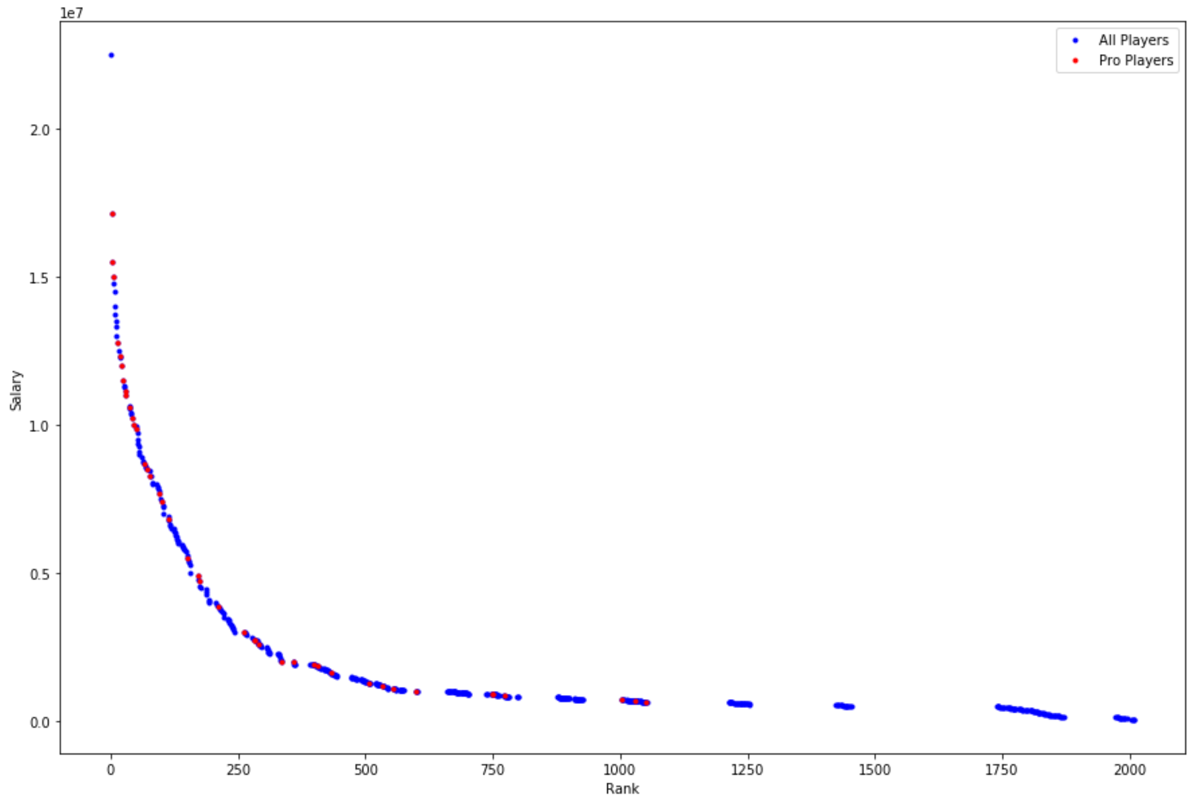
Out[17]: [<matplotlib.lines.Line2D at 0x20e15d0bc18>]



Correlation between years played and salary. Rookies (0 years) Are all mostly paid the same, meanwhile as more years are played, talent seems to separate the players into different salary amounts. Similar to age, the players who play longer usually end up getting paid more.

In [18]:
```
plt.figure(figsize=(15,10))
plt.plot(allplayers["Rk"],allplayers["Salary"], 'b.', label = "All Players")
plt.plot(proplayers["Rk"],proplayers["Salary"], 'r.', label = "Pro Players")
plt.xlabel("Rank")
plt.ylabel("Salary")
plt.legend()
```

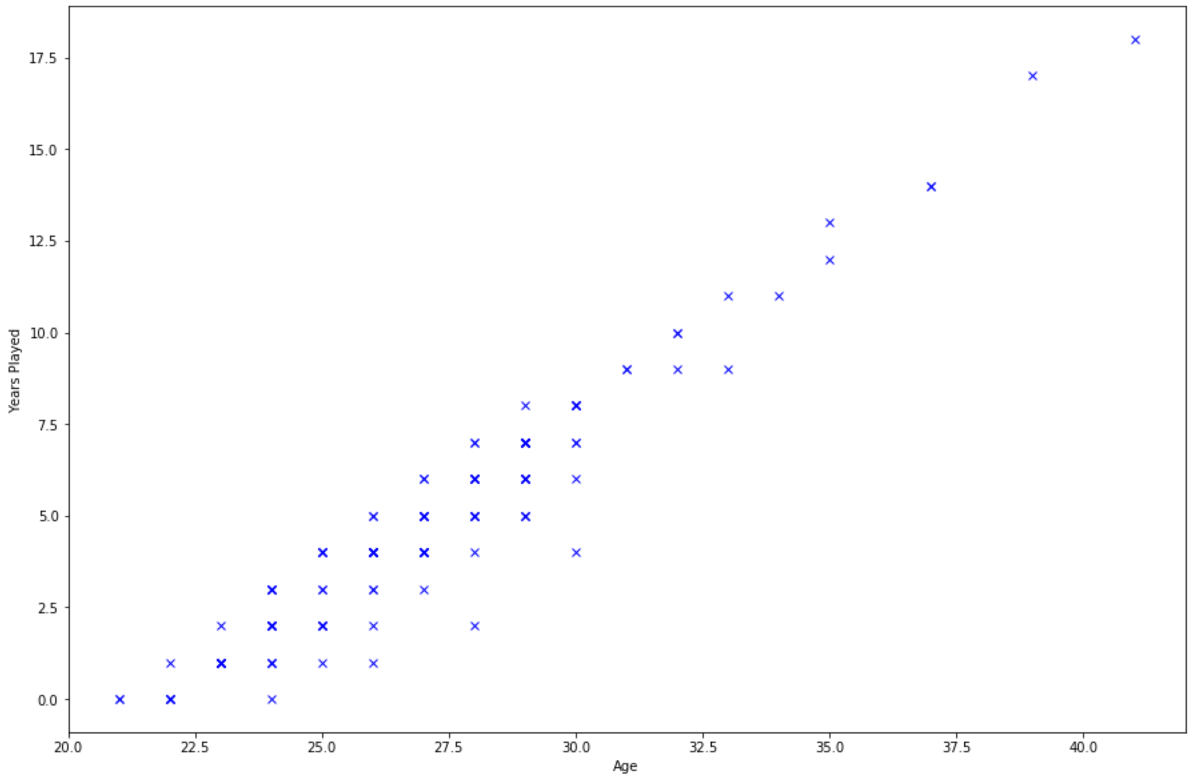
Out[18]: <matplotlib.legend.Legend at 0x20e15d7b6d8>



Clear correlation between ranking and salary. As can be seen from the graph, no pro bowl players are ranked >1100, but not all high ranking players are pro bowl players. Interesting to look at this graph and see that many players are getting paid like pro bowlers.

In [19]:
```
plt.figure(figsize=(15,10))
plt.plot(proplayers["Age"], proplayers["Yrs"].astype('int'), 'bx')
plt.xlabel("Age")
plt.ylabel("Years Played")
```

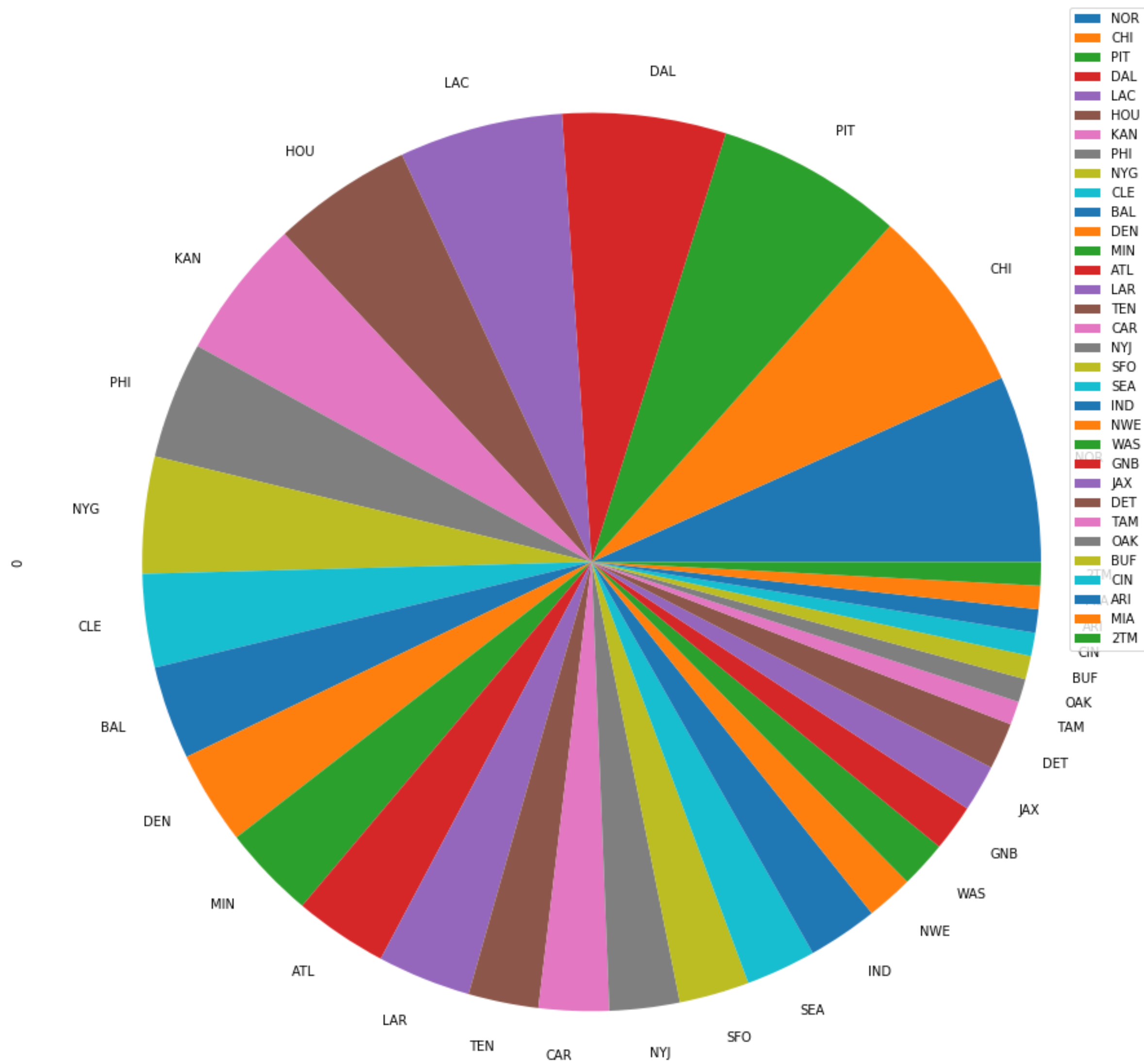
Out[19]: Text(0, 0.5, 'Years Played')

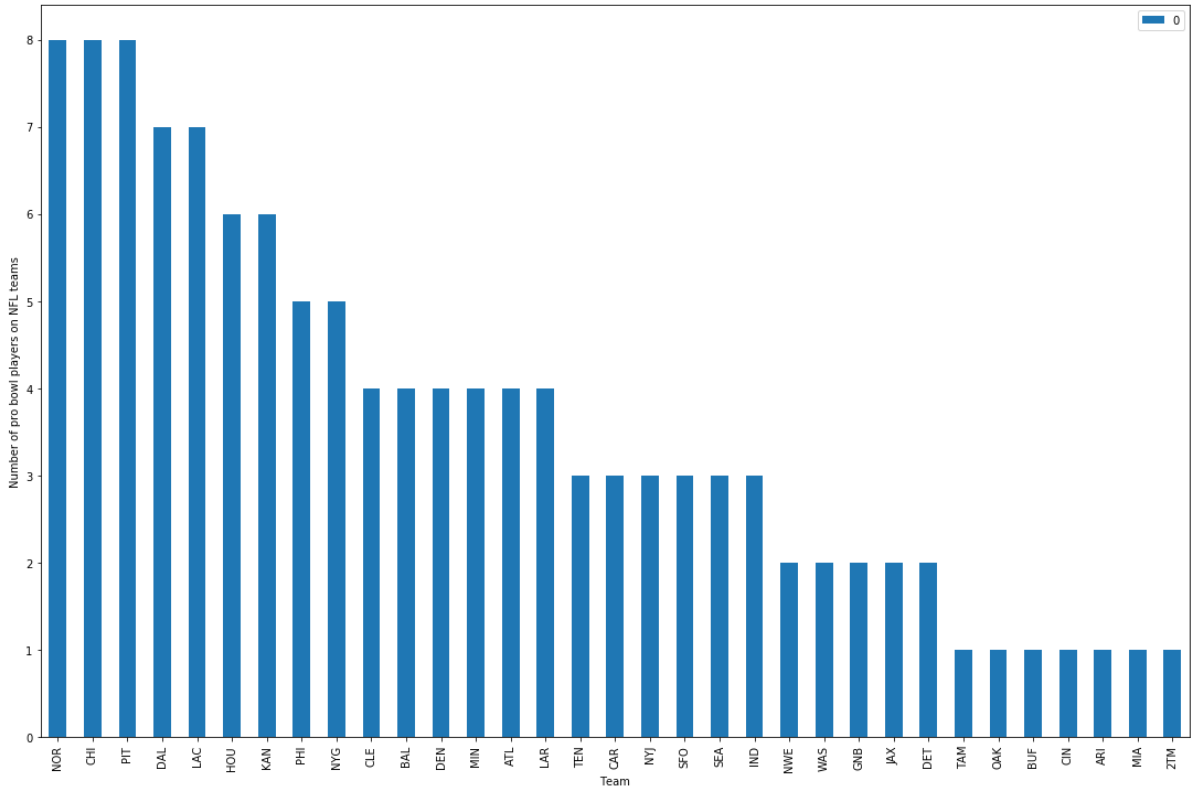


As expected, the correlation between age and years played is mostly linear.

In [33]:
```
from collections import Counter
teamcounts = Counter(proplayers["Tm"])
df = pd.DataFrame.from_dict(teamcounts, orient='index').sort_values(0,0,False)
df.plot.pie(figsize=(16, 16), subplots=True)

df.plot(kind='bar',figsize=(15,10))
plt.tight_layout()
plt.xlabel("Team")
plt.ylabel("Number of pro bowl players on NFL teams")
#plt.savefig("graph6.png", bbox_inches = "tight", dpi = 1200) ..... only using
bar chart.
```

Out[33]: Text(117.625, 0.5, 'Number of pro bowl players on NFL teams')

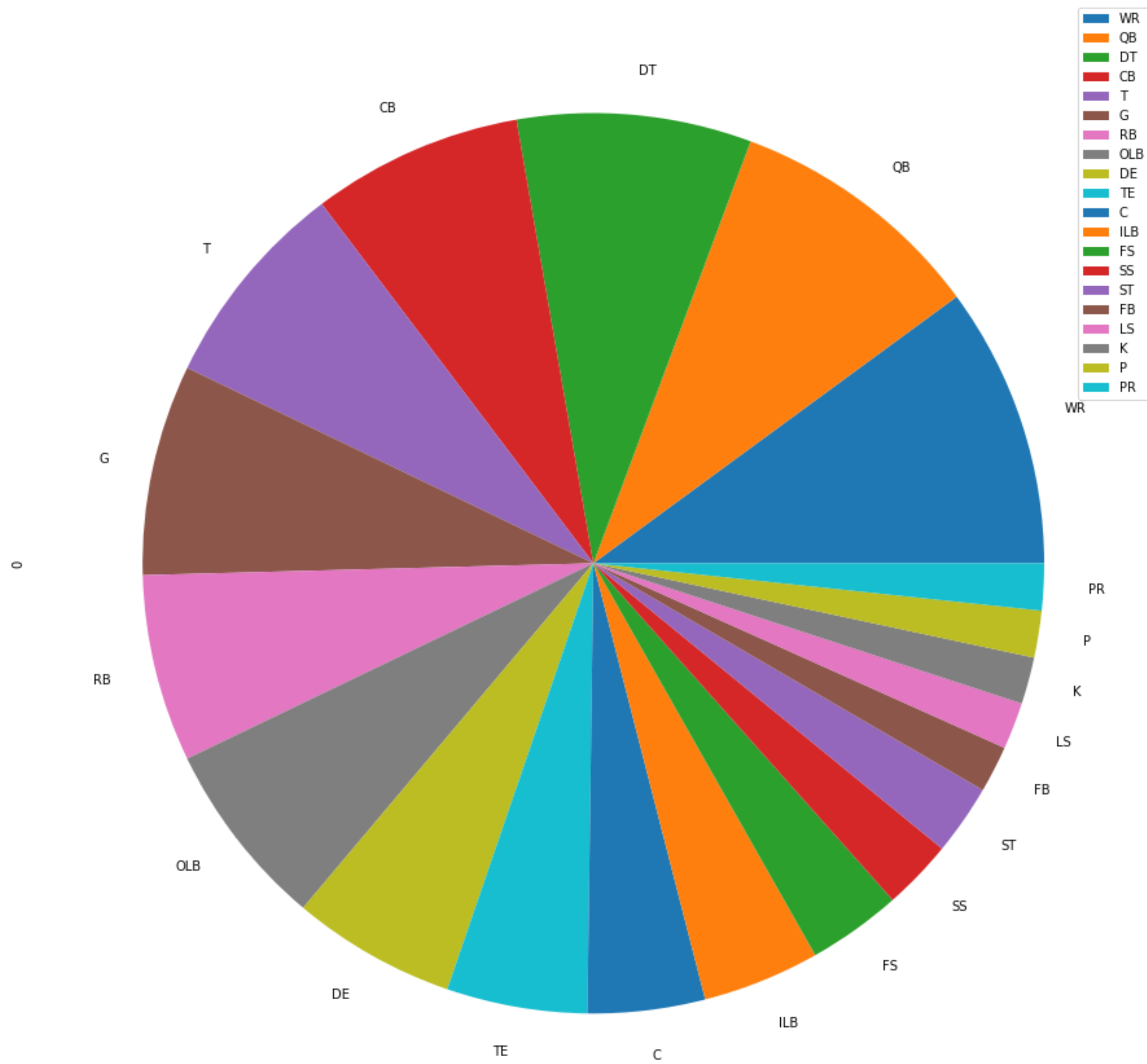


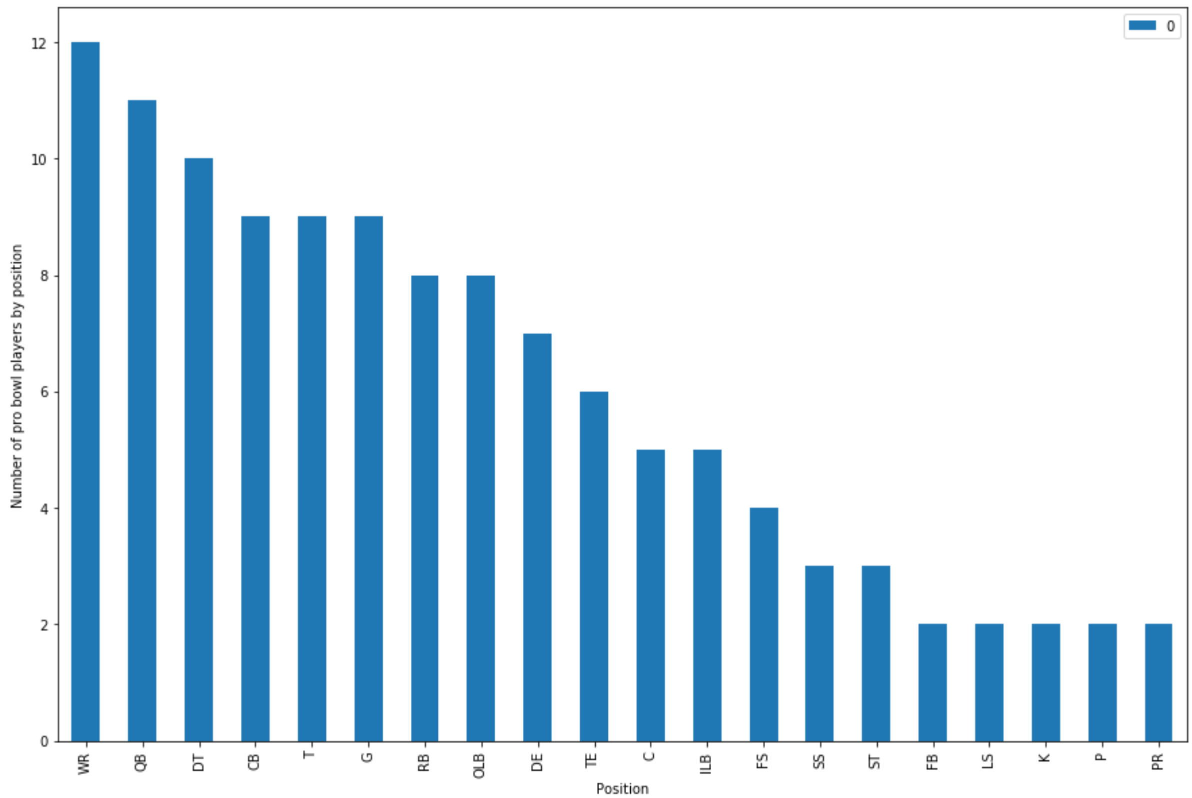


Both charts representing how many pro bowl players there were by team in 2019.

In [21]:
```
poscounts = Counter(proplayers["Pos"])
df = pd.DataFrame.from_dict(poscounts, orient='index').sort_values(0,0,False)

df.plot.pie(figsize=(16,16),subplots=True)
df.plot(kind='bar',figsize=(15,10))
plt.xlabel("Position")
plt.ylabel("Number of pro bowl players by position")
```

Out[21]: Text(0, 0.5, 'Number of pro bowl players by position')

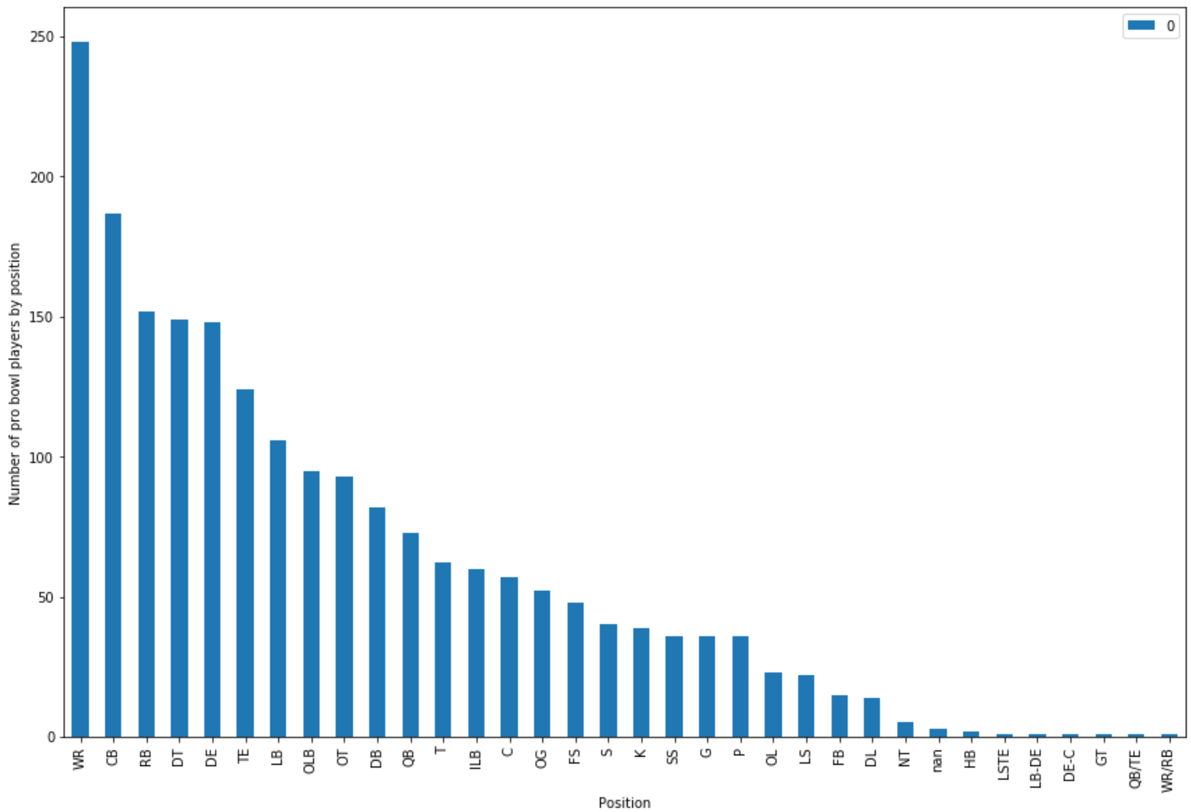




Both charts representing how many pro bowl players there were by position in 2019.

In [22]:
```
poscounts = Counter(allplayers["Pos"])
df = pd.DataFrame.from_dict(poscounts, orient='index').sort_values(0,0,False)

df.plot(kind='bar',figsize=(15,10))
plt.xlabel("Position")
plt.ylabel("Number of pro bowl players by position")
```

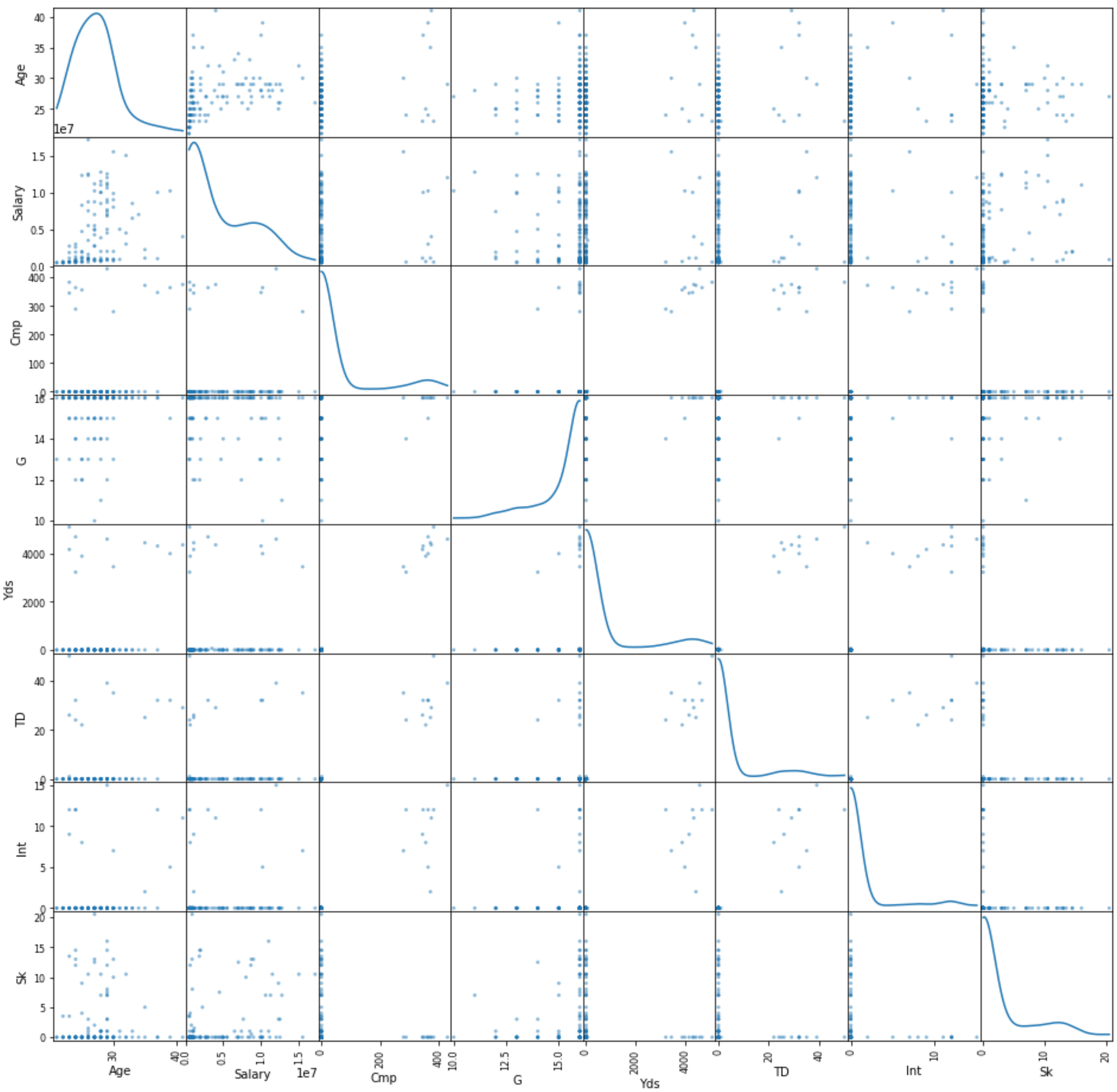
Out[22]:
```
Text(0, 0.5, 'Number of pro bowl players by position')
```



In [23]:
```
pd.plotting.scatter_matrix(proplayers[["Age","Salary","Cmp","G","Yds","TD","Int","Sk"]], figsize=(16, 16), diagonal='kde')
```

Out[23]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000020E1581B1D0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E1737A160>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E173A16A0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E173C7BE0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E173FA198>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17421710>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E1744AC88>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E1747C278
>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x0000020E1747C2B0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E174CBD30>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E174FD2E8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17522860>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E1754BDD8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E178AC390>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E178D3908>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E178FCE80
>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x0000020E1792D438>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E179559B0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E1797EF28>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E179AF4E0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E179D5A58>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E179FEFD0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17A31588>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17A58B00
>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17A890B8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17AB2630>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17AD9BA8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17B09160>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17B2F6D8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17B59C50>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17B8B208>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17BB3780
>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17BDCCF8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17C0C2B0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17C34828>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17C5EDA0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17C8D358>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17CB38D0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17CDCE48>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17D0D400
>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17D36978>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17D5DEF0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E165FF0F0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E162B2A58>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E1629A978>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E155CB320>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17D9B898>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E15C3DE10
>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x0000020E15DFE3C8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E165EC940>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E16502EB8>,
```

```
         <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E162FA470>,
         <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E162DF9E8>,
         <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17582F60>,
         <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E175B1518>,
         <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E175D9A90
>],
        [<matplotlib.axes._subplots.AxesSubplot object at 0x0000020E1760B048>,
         <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E176325C0>,
         <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17659B38>,
         <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E1768C0F0>,
         <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E176B4668>,
         <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E176DBBE0>,
         <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E1770F198>,
         <matplotlib.axes._subplots.AxesSubplot object at 0x0000020E17736710
>]],
       dtype=object)
```



A Scatter-matrix plot between subsets of data. This graph gives us a rough idea about the correlations between different data. For example, high values for Yards usually correlate with high values for Touchdowns. Interesting to put all the data I have on one chart.

Conclusion:

My final conclusion based off of all the data I was able to record and put into different visualizations includes multiple thoughts. First off, it was pretty clear that pro bowl players are actually paid well relative to the rest of the NFL. Regardless, there are many players that still have a similar salary and the difference between the salaries of pro bowlers and regular NFL players is not that crazy (with some exceptions of course). Secondly, the average age of pro bowlers is 27.6. This shows that the majority of pro bowlers are actually older considering that many studies have the average length of an NFL career being about 3-4 years out of college. This fact took me by surprise because I honestly thought pro bowlers would be younger and more in the 24-25 year old range. Thirdly, the NFL team with the most Pro Bowlers in 2019 was the New Orleans Saints. Ironically enough, the super bowl winner patriots only had two selections for the 2019 Pro Bowl but they were not able to participate because of the play-offs. Finally, there is no real correlation between age and salary although it is clear that as players get older, the ones that still play usually get paid more simply because they are probably talented and more skilled at their position to even be playing at an older age as compared to their teammates/rivals.

Thank you!

In [ ]:

```

```