Data Science Project Presentation

By Charlie Chu

Dataset: College Football Statistics for the 2021-2022 Season

Reasons why I chose this specific dataset:

- Big fan of college football (especially for the Michigan Wolverines)

- Recently won the National Championship

- Have gone on a 3-game winning streak against rivals Ohio State

-Winning streak started in the 2021-2022 college football season

	Team	Games	Win-Loss	Off Rank	Off Plays	Off Yards	Off Yards/Play	Off TDs	Off Yards per Game	Def Rank	Def Plays	Yards Allowed	Yards/Play Allowed	Off TDs Allowed	Total TDs Allowed
1	Air Force (Mountain West)	13	10-3	51	947	5,502	5.81	52	423.2	4	737	3,855	5.23	33	34
2	Akron (MAC)	12	2-10	106	783	4,089	5.22	29	340.8	122	783	5,628	7.19	58	62
3	Alabama (SEC)	15	13-2	7	1,119	7,323	6.54	69	488.2	7	956	4,561	4.77	34	36
4	App State (Sun Belt)	14	10-4	30	980	6,178	6.30	55	441.3	33	923	4,868	5.27	36	37
5	Arizona (Pac-12)	12	1-11	101	870	4,271	4.91	18	355.9	57	756	4,452	5.89	42	48
6	Arizona St. (Pac-12)	13	8-5	75	807	5,018	6.22	46	386.0	13	853	4,238	4.97	31	31
7	Arkansas (SEC)	13	9-4	27	902	5,742	6.37	48	441.7	50	884	4,780	5.41	33	35
8	Arkansas St. (Sun Belt)	12	2-10	69	874	4,756	5.44	31	396.3	129	840	6,069	7.22	58	62
9	Army West Point (FBS Independent)	13	9-4	83	868	4,881	5.62	57	375.5	15	739	4,269	5.78	34	37
10	Auburn (SEC)	13	6-7	67	899	5,219	5.81	39	401.5	61	919	4,860	5.29	31	32
11	Ball St. (MAC)	13	6-7	110	884	4,398	4.98	35	338.3	96	925	5,452	5.89	39	44
12	Baylor (Big 12)	14	12-2	53	937	5,918	6.32	53	422.7	31	940	4,843	5.15	29	29
13	Boise St. (Mountain West)	12	7-5	78	870	4,572	5.26	35	381.0	45	801	4,373	5.46	26	26
14	Boston College (ACC)	12	6-6	103	783	4,201	5.37	34	350.1	28	760	4,131	5.44	29	32
15	Bowling Green (MAC)	12	4-8	117	777	3,802	4.89	27	316.8	62	848	4,487	5.29	45	47
16	Buffalo (MAC)	12	4-8	62	905	4,905	5.42	42	408.8	93	788	4,997	6.34	47	47
17	BYU (FBS Independent)	13	10-3	17	868	5,884	6.78	56	452.6	74	881	5,054	5.74	38	38
18	California (Pac-12)	12	5-7	75	784	4,632	5.91	33	386.0	49	829	4,400	5.31	33	33
19	Central Mich. (MAC)	13	9-4	24	968	5,764	5.95	49	443.4	67	889	4,963	5.58	40	42
20	Charlotte (C-USA)	12	5-7	65	824	4,839	5.87	40	403.2	120	781	5,580	7.14	54	55
21	Cincinnati (AAC)	14	13-1	58	862	5,803	6.73	67	414.5	10	996	4,457	4.47	27	28
22	Clemson (ACC)	13	10-3	100	903	4,668	5.17	38	359.1	8	894	3,971	4.44	18	22
23	Coastal Carolina (Sun Belt)	13	11-2	5	831	6,431	7.74	69	494.7	25	836	4,412	5.28	36	36
24	Colorado (Pac-12)	12	4-8	129	707	3,089	4.37	22	257.4	98	834	5,051	6.06	39	40
25	Colorado St. (Mountain West)	12	3-9	57	911	4,989	5.48	29	415.8	64	829	4,528	5.46	41	43
26	Duke (ACC)	12	3-9	54	934	5,011	5.37	31	417.6	130	873	6,215	7.12	58	60
27	East Carolina (AAC)	12	7-5	39	899	5,200	5.78	41	433.3	80	789	4,733	6.00	37	41
28	Eastern Mich. (MAC)	13	7-6	79	910	4,950	5.44	45	380.8	107	901	5,661	6.28	46	50
29	FIU (C-USA)	12	1-11	94	750	4,400	5.87	31	366.7	128	882	5,896	6.68	58	62
30	Fla. Atlantic (C-USA)	12	5-7	68	835	4,800	5.75	38	400.0	90	843	4,911	5.83	36	39

Website on Kaggle for dataset: https://www.kaggle.com/code/marklandon79/python-data-prep-for-ml-college-football

More about the Data Set and Expectations

One of the key players that played a significant role in Michigan's 42 – 27 win over Ohio State back in 2021 was defensive end Aidan Hutchinson. He now plays for the Detroit Lions, leading them to the NFC Championship game since 1991. Finding inspiration from this, I wanted to find what defensive factors (variables) would have the strongest relationship with wins.

Out of the 151 columns in the data set, I would need to explore the elements would help me solve the question I wanted to find out: What specific defensive actions have a major influence in the result of a college football game. I would compare the data across all 130 teams out of the existing 133 currently in college football.

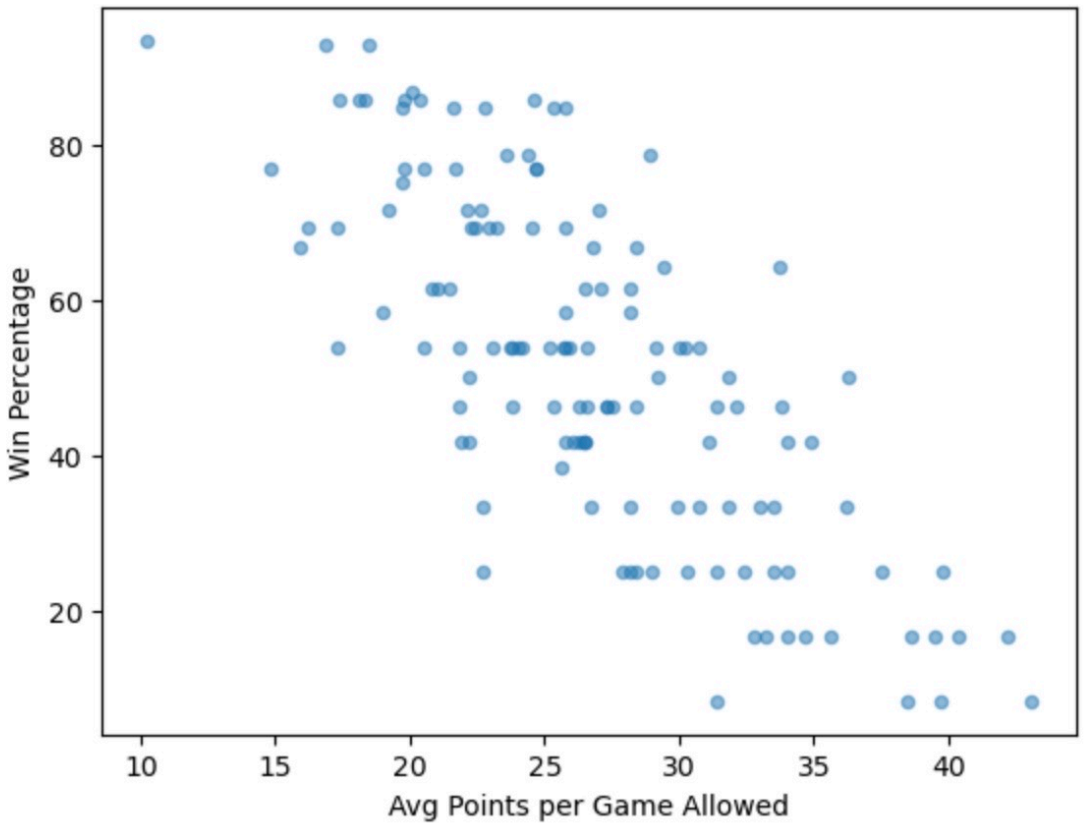
Analyzing & Visualizing the Data

Coding Language: Python

Libraries used:

- Pandas

- Matplotlib (Experimented but couldn’t successfully apply to my code)



Identifying data set

- To see how important defensive plays are in a football game, there needs to be a column that shows the number of wins teams have in a season

- Unfortunately, this data set does not have a column for that, and only has a win-loss section

Games	Win-Loss
13	10-3
12	2-10
15	13-2
14	10-4
12	1-11
13	8-5

Creating New Columns

- Since the original column was named “Win-Loss”, I split the string and transformed them into 2 separate columns with type int

- By creating a separated “Wins” column and a “Loss” column, I was able to create another column called “Win Percentage” by dividing the number of games won by the number of games played by the teams

- At first, the win percentage type float numbers would have many decimals after, so I rounded it to 2 numbers after the decimal for cleaner data

```
df[['Wins', 'Losses']] = df['Win-Loss'].str.split('-', expand=True).astype(int)
#df.head()
df['Win Percentage'] = ((df['Win'] / df['Games']) * 100).round(2)
df.drop('Loss', axis = 1)
df.drop('Win', axis = 1)
df.head()
```

Win Percentage	Wins	Losses
76.92	10	3
16.67	2	10
86.67	13	2
71.43	10	4
8.33	1	11

Trying to Create a New Column with Values of Different Types

```
In [6]: df["Average Number of Defensive Plays Per Game"] = df["Def Plays"] // df["Games"]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
File ~/anaconda3/lib/python3.11/site-packages/pandas/core/ops/array_ops.py:171, in _na_arithmetic_op
p, is_cmp)
    170 try:
--> 171     result = func(left, right)
    172 except TypeError:

File ~/anaconda3/lib/python3.11/site-packages/pandas/core/computation/expressions.py:240, in evaluat
_numexpr)
    239         return _evaluate(op, op_str, a, b)  # type: ignore[misc]
--> 240 return _evaluate_standard(op, op_str, a, b)

File ~/anaconda3/lib/python3.11/site-packages/pandas/core/computation/expressions.py:70, in _evaluat
op_str, a, b)
     69     _store_test_result(False)
---> 70 return op(a, b)

TypeError: unsupported operand type(s) for //: 'str' and 'int'

During handling of the above exception, another exception occurred:

TypeError                                 Traceback (most recent call last)
Cell In[6], line 1
----> 1 df["Average Number of Defensive Plays Per Game"] = df["Def Plays"] // df["Games"]

File ~/anaconda3/lib/python3.11/site-packages/pandas/core/ops/common.py:81, in _unpack_zerodim_and_d
ew_method(self, other)
     77             return NotImplemented
```

I tried to make a new column called “Average Number of Defensive Plays Per Game” by dividing the total number of defensive plays teams had over the season with the number of games teams played

```
In [131]: df.info(verbose = True)
```

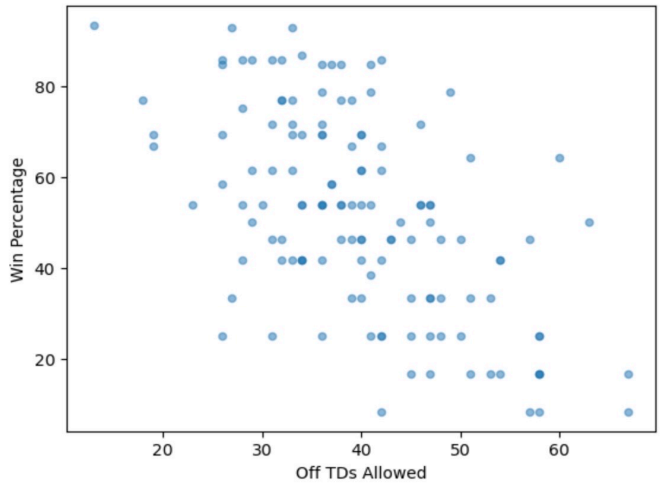
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 130 entries, 0 to 129
Data columns (total 152 columns):
 #    Column                                Dtype
---   ------                                -----
 0    Unnamed: 0                            int64
 1    Team                                  object
 2    Games                                 int64
 3    Win-Loss                              object
 4    Off Rank                              int64
 5    Off Plays                             object
 6    Off Yards                             object
 7    Off Yards/Play                        float64
 8    Off TDs                               int64
 9    Off Yards per Game                    float64
 10   Def Rank                              int64
 11   Def Plays                             object
 12   Yards Allowed                         object
 13   Yards/Play Allowed                    float64
```

However, the data type values for the number of defensive plays were strings, and not integers. Unfortunately, I have not acquired the skills to convert the data type values in a specific column of a data set. As a result, I was not able to create many of the new columns that I had wanted to create

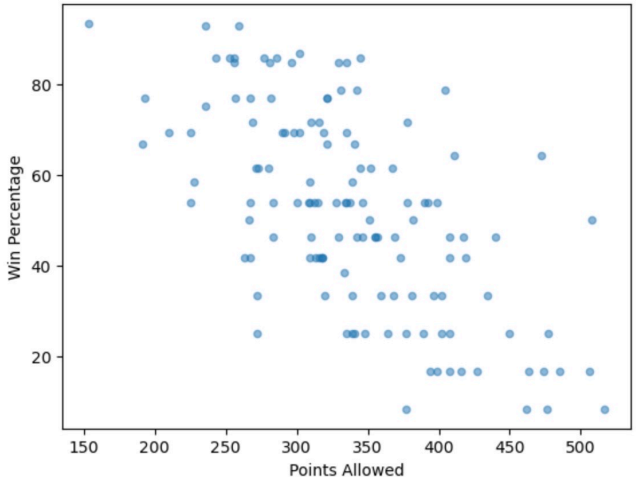
Visualizations

With the Pandas library, I made scatterplots with a constant y-axis (Win Percentage) and the x-axis consisting of factors that may play a role in the number of games won. I organized them into 3 categories: Strong Correlation, Medium/Weak Correlation, and No Correlation

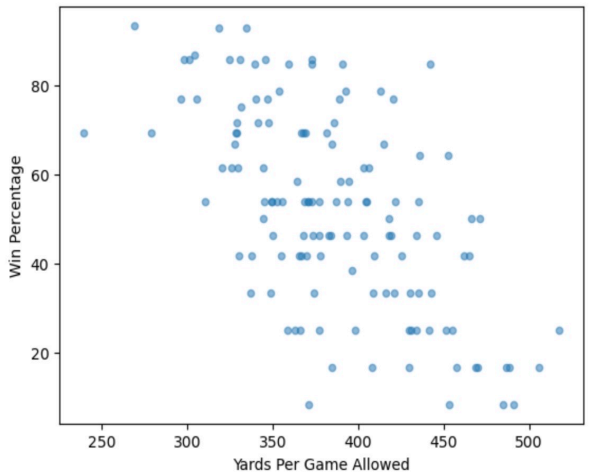
Strong Correlation



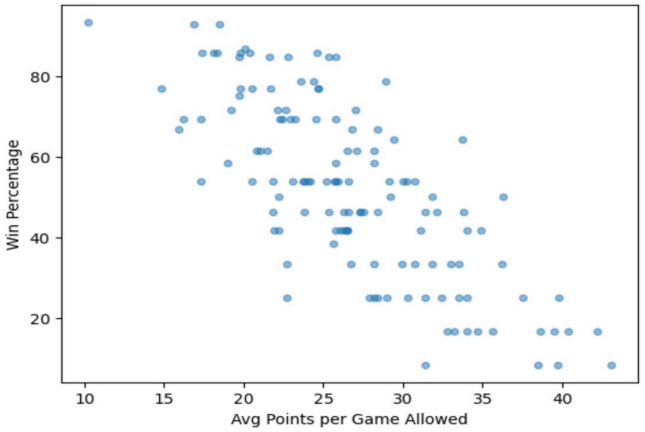

```
In [171]: df.plot.scatter(x='Off TDs Allowed', y = 'Win Percentage', alpha = 0.5)
Out[171]: <Axes: xlabel='Off TDs Allowed', ylabel='Win Percentage'>
```



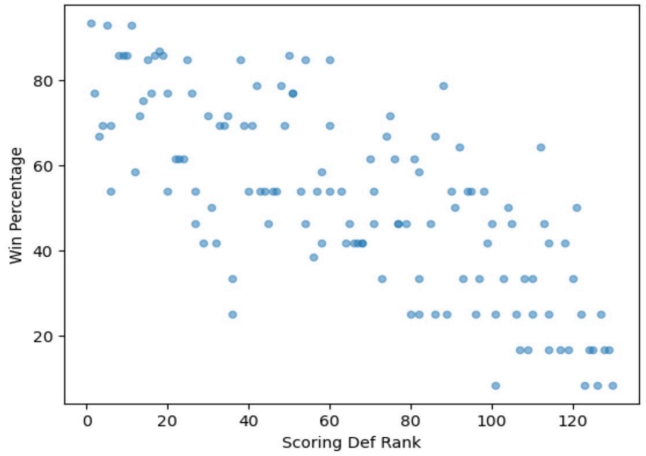

```
In [154]: df.plot.scatter(x='Redzone Points Allowed', y = 'Win Percentage', alpha = 0.5)
Out[154]: <Axes: xlabel='Redzone Points Allowed', ylabel='Win Percentage'>
```



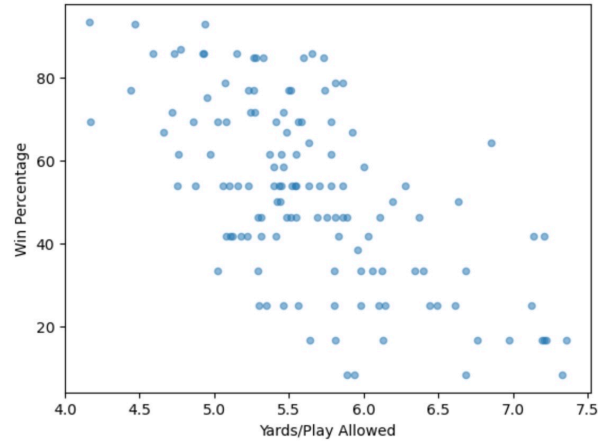

```
In [156]: df.plot.scatter(x='Pass Yards Per Game Allowed', y = 'Win Percentage', alpha = 0.5)
Out[156]: <Axes: xlabel='Pass Yards Per Game Allowed', ylabel='Win Percentage'>
```




```
In [187]: df.plot.scatter(x='Avg Points per Game Allowed', y = 'Win Percentage', alpha = 0.5)
Out[187]: <Axes: xlabel='Avg Points per Game Allowed', ylabel='Win Percentage'>
```

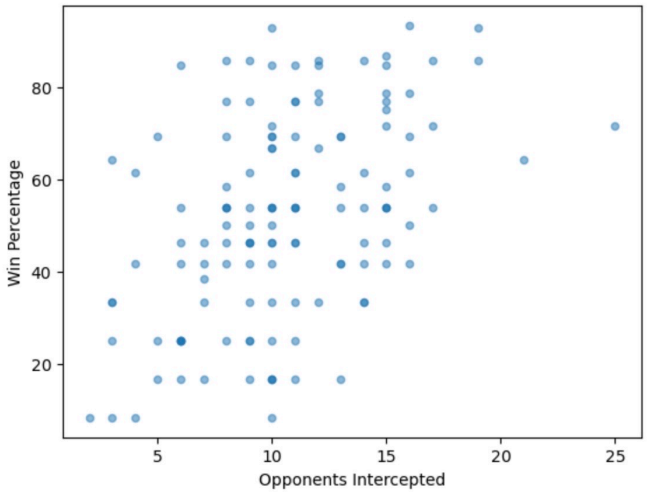



```
In [189]: df.plot.scatter(x='Scoring Def Rank', y = 'Win Percentage', alpha = 0.5)
Out[189]: <Axes: xlabel='Scoring Def Rank', ylabel='Win Percentage'>
```

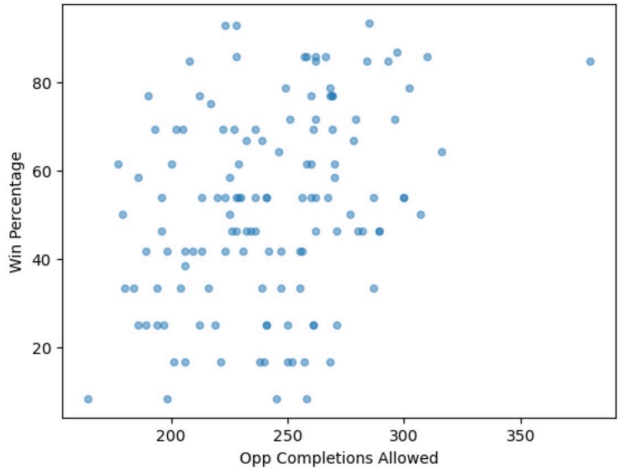



```
In [172]: df.plot.scatter(x='Yards/Play Allowed', y = 'Win Percentage', alpha = 0.5)
Out[172]: <Axes: xlabel='Yards/Play Allowed', ylabel='Win Percentage'>
```

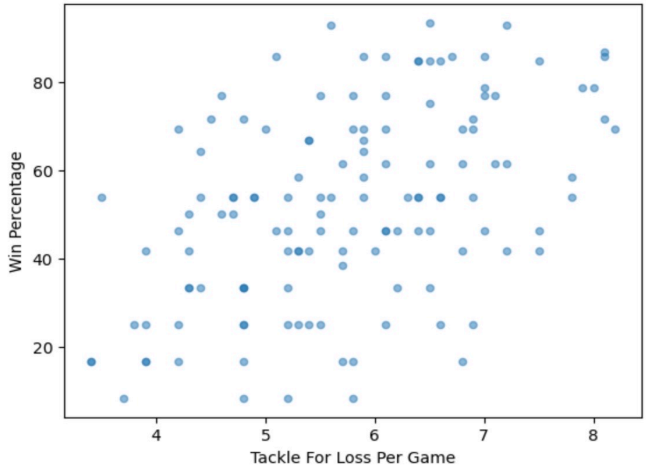
Weak – Medium Correlation



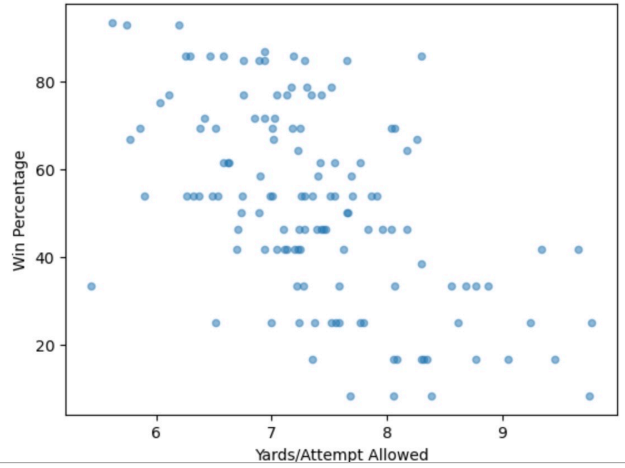

```
In [175]: df.plot.scatter(x='Opponents Intercepted', y = 'Win Percentage', alpha = 0.5)
Out[175]: <Axes: xlabel='Opponents Intercepted', ylabel='Win Percentage'>
```




```
In [163]: df.plot.scatter(x='Opp Completions Allowed', y = 'Win Percentage', alpha = 0.5)
Out[163]: <Axes: xlabel='Opp Completions Allowed', ylabel='Win Percentage'>
```

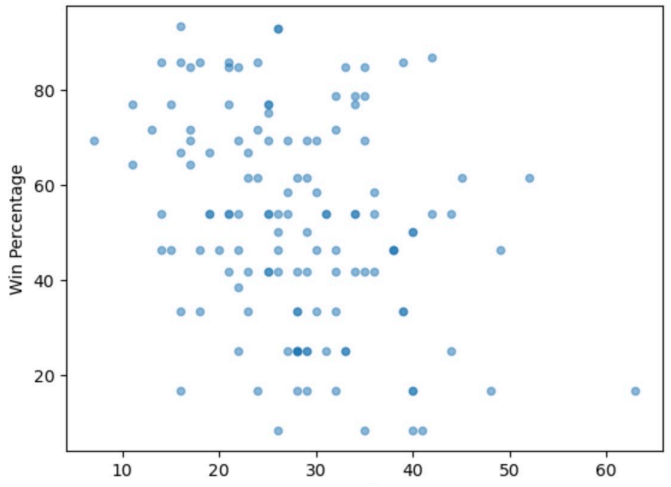



```
In [178]: df.plot.scatter(x='Tackle For Loss Per Game', y = 'Win Percentage', alpha = 0.5)
Out[178]: <Axes: xlabel='Tackle For Loss Per Game', ylabel='Win Percentage'>
```

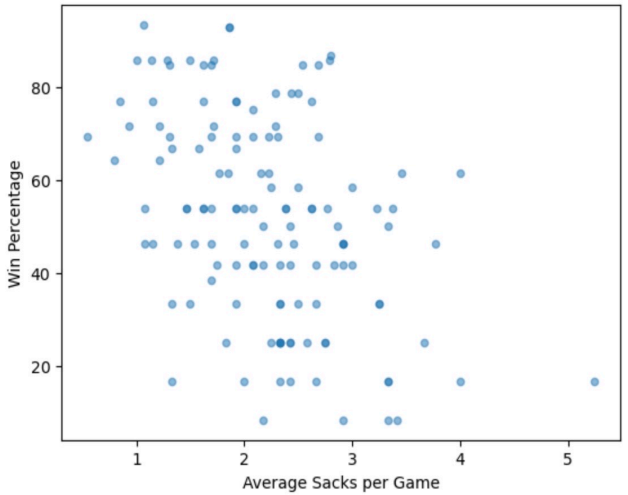



```
In [157]: df.plot.scatter(x='Yards/Completion Allowed', y = 'Win Percentage', alpha = 0.5)
Out[157]: <Axes: xlabel='Yards/Completion Allowed', ylabel='Win Percentage'>
```

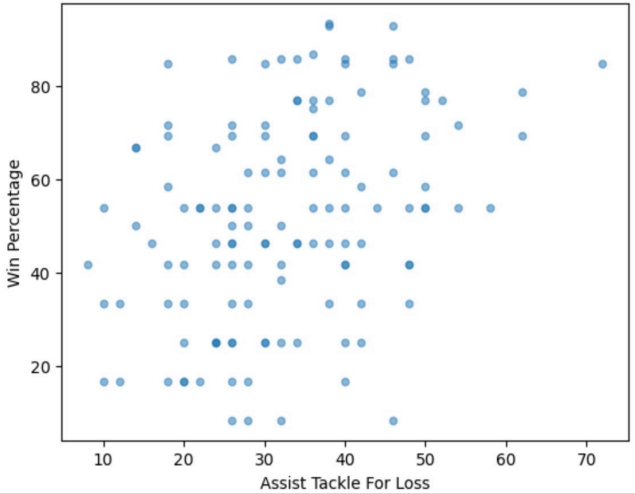
Weak – Medium Correlation Cont.




```
In [150]: df.plot.scatter(x='Sacks', y = 'Win Percentage', alpha = 0.5)
Out[150]: <Axes: xlabel='Sacks', ylabel='Win Percentage'>
```

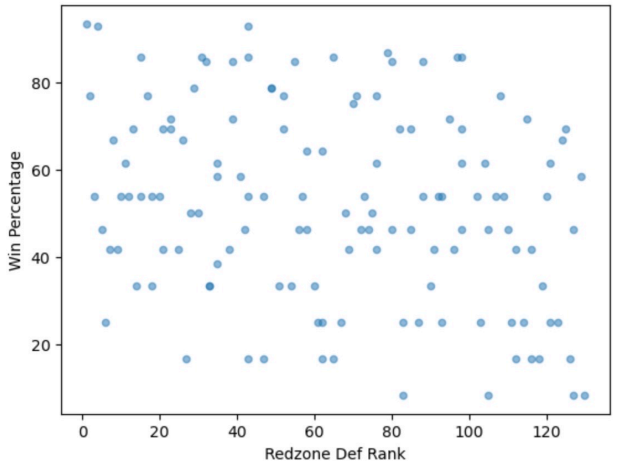



```
In [190]: df.plot.scatter(x='Average Sacks per Game', y = 'Win Percentage', alpha = 0.5)
Out[190]: <Axes: xlabel='Average Sacks per Game', ylabel='Win Percentage'>
```

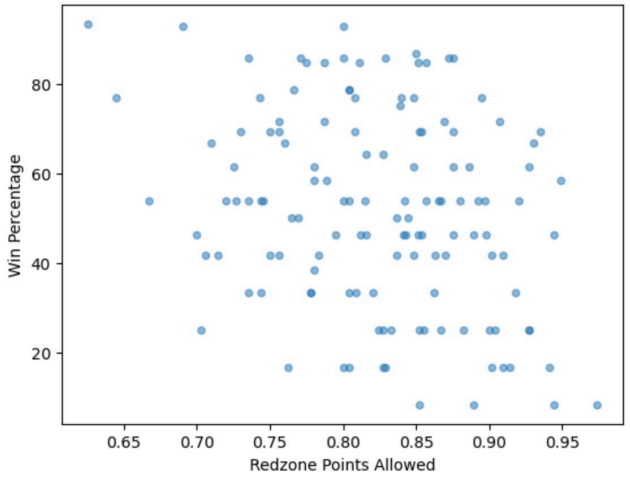



```
In [184]: df.plot.scatter(x='Assist Tackle For Loss', y = 'Win Percentage', alpha = 0.5)
Out[184]: <Axes: xlabel='Assist Tackle For Loss', ylabel='Win Percentage'>
```

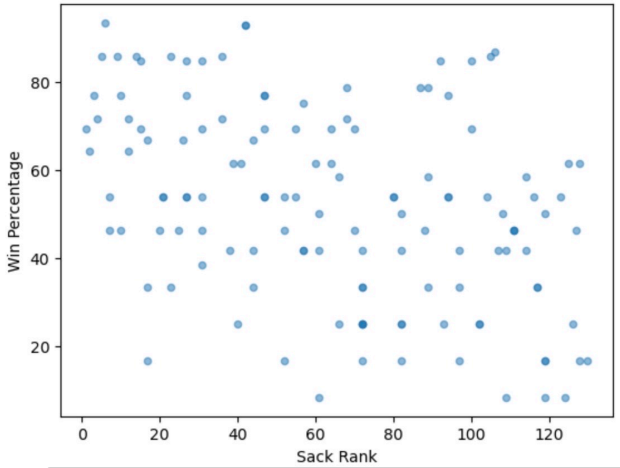
No Correlation



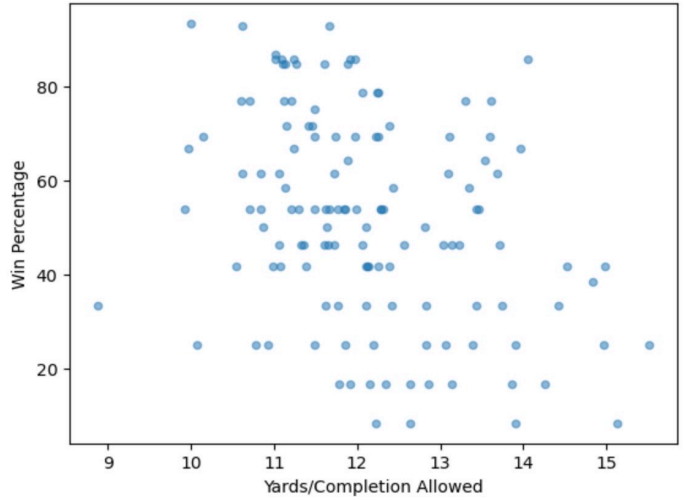

```
In [155]: df.plot.scatter(x='Redzone Def Rank', y = 'Win Percentage', alpha = 0.5)
Out[155]: <Axes: xlabel='Redzone Def Rank', ylabel='Win Percentage'>
```




```
In [154]: df.plot.scatter(x='Redzone Points Allowed', y = 'Win Percentage', alpha = 0.5)
Out[154]: <Axes: xlabel='Redzone Points Allowed', ylabel='Win Percentage'>
```

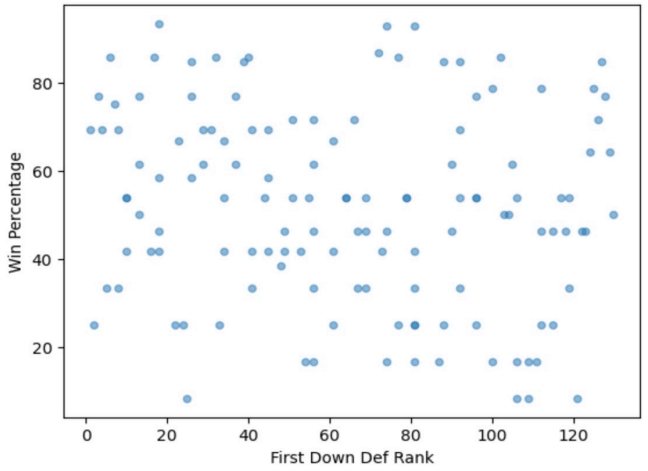



```
In [152]: df.plot.scatter(x='Sack Rank', y = 'Win Percentage', alpha = 0.5)
Out[152]: <Axes: xlabel='Sack Rank', ylabel='Win Percentage'>
```

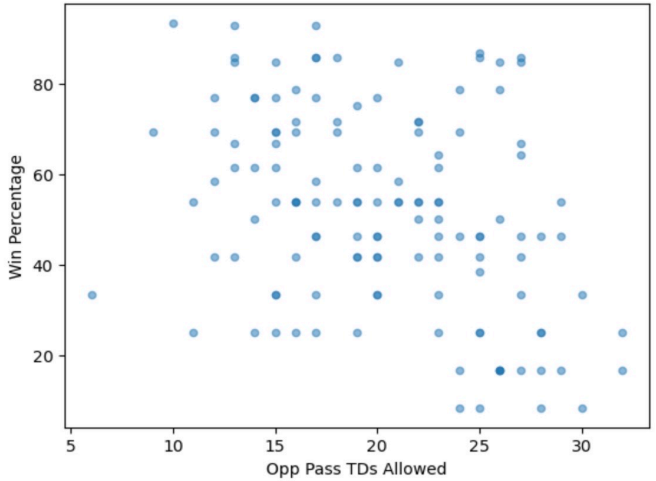



```
In [157]: df.plot.scatter(x='Yards/Completion Allowed', y = 'Win Percentage', alpha = 0.5)
Out[157]: <Axes: xlabel='Yards/Completion Allowed', ylabel='Win Percentage'>
```

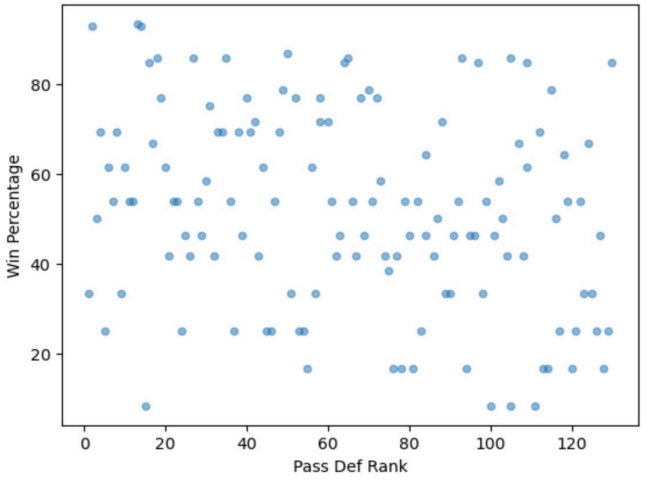
No Correlation Cont.



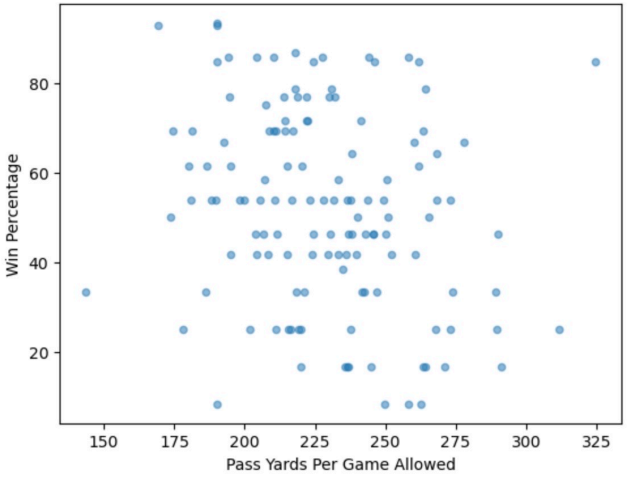

```
In [169]: df.plot.scatter(x='First Down Def Rank', y = 'Win Percentage', alpha = 0.5)
Out[169]: <Axes: xlabel='First Down Def Rank', ylabel='Win Percentage'>
```




```
In [159]: df.plot.scatter(x='Opp Pass TDs Allowed', y = 'Win Percentage', alpha = 0.5)
Out[159]: <Axes: xlabel='Opp Pass TDs Allowed', ylabel='Win Percentage'>
```

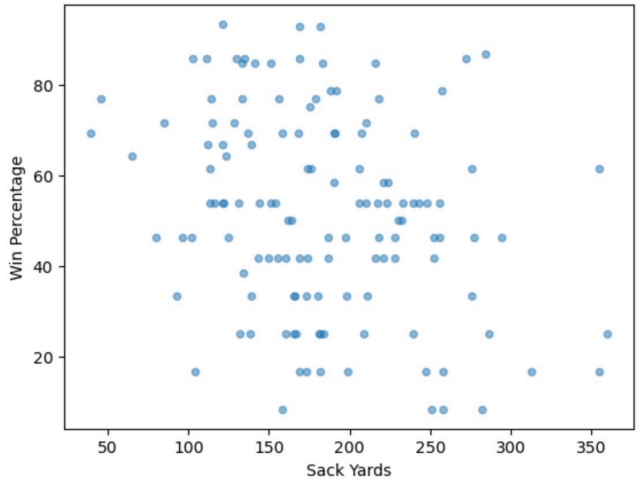



```
In [164]: df.plot.scatter(x='Pass Def Rank', y = 'Win Percentage', alpha = 0.5)
Out[164]: <Axes: xlabel='Pass Def Rank', ylabel='Win Percentage'>
```

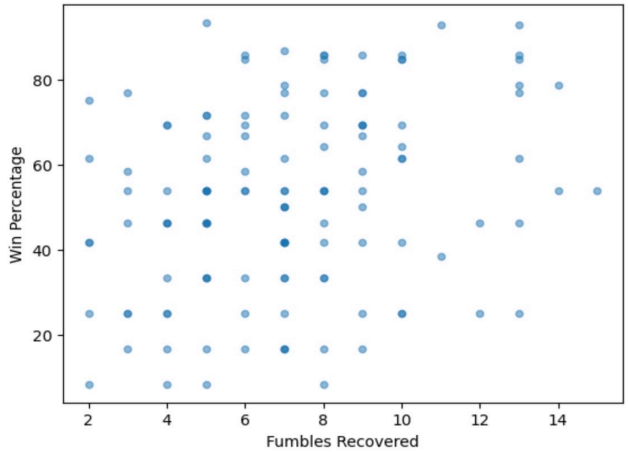



```
In [156]: df.plot.scatter(x='Pass Yards Per Game Allowed', y = 'Win Percentage', alpha = 0.5)
Out[156]: <Axes: xlabel='Pass Yards Per Game Allowed', ylabel='Win Percentage'>
```

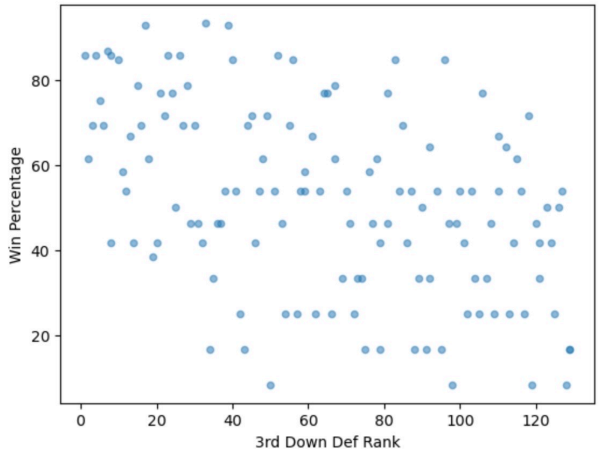
No Correlation Cont.



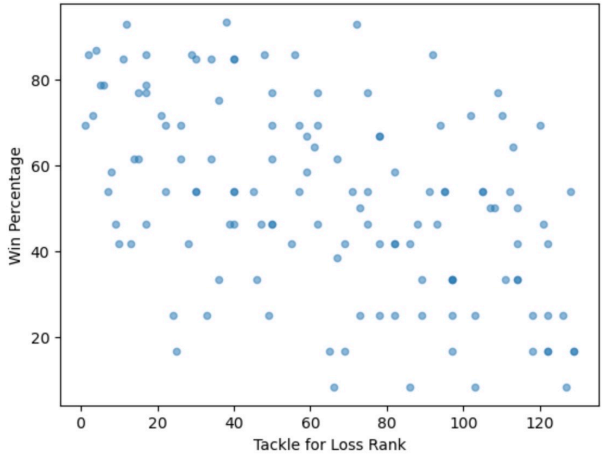

```
In [191]: df.plot.scatter(x='Sack Yards', y = 'Win Percentage', alpha = 0.5)
Out[191]: <Axes: xlabel='Sack Yards', ylabel='Win Percentage'>
```




```
In [176]: df.plot.scatter(x='Fumbles Recovered', y = 'Win Percentage', alpha = 0.5)
Out[176]: <Axes: xlabel='Fumbles Recovered', ylabel='Win Percentage'>
```




```
In [177]: df.plot.scatter(x='3rd Down Def Rank', y = 'Win Percentage', alpha = 0.5)
Out[177]: <Axes: xlabel='3rd Down Def Rank', ylabel='Win Percentage'>
```




```
In [186]: df.plot.scatter(x='Tackle for Loss Rank', y = 'Win Percentage', alpha = 0.5)
Out[186]: <Axes: xlabel='Tackle for Loss Rank', ylabel='Win Percentage'>
```

Key Findings

- The slope of the visualizations are not all in the same direction: this makes sense in the context of the variables chosen. For example, variables such as the number of opponents intercepted and tackles for loss per game prefer the number to be higher as it is a good result. However, variables such as the number of offensive touchdowns allowed and the number of passing yards per game would prefer the number to be lower.

- Dividing up the visualizations into three categories, it is evident that the following elements play a large role in a team's win percentage
  - Number of offensive touchdowns allowed
  - Number of redzone points allowed
  - Number of pass yards per game allowed
  - Average number of points per game allowed
  - Scoring defensive rank
  - Number of yards per play allowed

Surprises and Unexpected Results

For some of the graphs, I was expecting more correlation between the number of games won and some variables. This included the number of sack yards, fumbles recovered, and redzone points allowed. As a football enjoyer, I value these types of factors into the game and heavily believed that these statistics would deeply contribute to a team's win percentage. In contrast, there was little to no correlation between these variables and win percentage, causing me to truly rethink of how other defensive aspects may sway the result of a game.