In [121…

```
#import modules that will be used for analysis
%matplotlib inline
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style="darkgrid")
from sklearn.linear_model import LinearRegression
```

In [122…

```
#load data
nfl2020 = pd.read_csv('C:/Users/devis/OneDrive/Desktop/nfl_dst_raw_data2020.csv')
```

In [123…

```
nfl2020
```

Out[123]:

	game_id	team	def_int	def_int_td	sacks	fumbles_rec	fumbles_rec_td	blocked_kick	safet
0	202009100kan	HOU	0	0	1	0	0	0	
1	202009100kan	KAN	1	0	4	0	0	0	
2	202009130atl	ATL	0	0	3	0	0	0	
3	202009130atl	SEA	1	0	2	1	0	0	
4	202009130buf	BUF	1	0	3	1	0	0	
...	...	...	...	...	...	...	...	...	
533	202101240gnb	TAM	1	0	5	1	0	0	
534	202101240kan	BUF	0	0	1	1	0	0	
535	202101240kan	KAN	1	0	4	0	0	0	
536	202102070tam	KAN	0	0	1	0	0	0	
537	202102070tam	TAM	2	0	3	0	0	0	

538 rows × 37 columns

When the data is first loaded in, the defensive stats for EACH TEAM in every game is shown. We want to combine the stats of each team per game by combining each group of 2 rows together, so we can get total stats for the game. We do this using the code below.

In [124…

```
nfl_2020 = nfl2020.iloc[::2, :].reset_index(drop=True)
for i in range(1, len(nfl2020), 2):
    nfl_2020.loc[i//2, :] = nfl2020.iloc[i-1, :] + nfl2020.iloc[i, :]
```

In [125…

```
nfl_2020
```

Out[125]:

	game_id	team	def_int	def_int_td	sacks	fumbles_rec	fumbles_rec_td	bl
0	202009100kan202009100kan	HOUKAN	1	0	5	0	0	
1	202009130atl202009130atl	ATLSEA	1	0	5	1	0	
2	202009130buf202009130buf	BUFNYJ	1	0	6	3	0	
3	202009130car202009130car	CARLVR	0	0	1	0	0	
4	202009130cin202009130cin	CINLAC	1	0	5	1	0	
...	...	...	...	...	...	...	...	
264	202101170kan202101170kan	CLEKAN	2	0	2	1	0	
265	202101170nor202101170nor	NORTAM	3	0	1	1	0	
266	202101240gnb202101240gnb	GNBTAM	4	0	6	1	0	
267	202101240kan202101240kan	BUFKAN	1	0	5	1	0	
268	202102070tam202102070tam	KANTAM	2	0	4	0	0	

269 rows × 37 columns

After performing the previous operation, columns containing character vectors have the same word duplicated twice. We want to slice these strings in half, and we use this code to do so.

In [126…
```
nfl_2020['vis_team'] = nfl_2020['vis_team'].apply(lambda x: x[:len(x)//2])
nfl_2020['home_team'] = nfl_2020['home_team'].apply(lambda x: x[:len(x)//2])
nfl_2020['Roof'] = nfl_2020['Roof'].apply(lambda x: x[:len(x)//2])
nfl_2020['Surface'] = nfl_2020['Surface'].apply(lambda x: x[:len(x)//2])
nfl_2020['Vegas_Favorite'] = nfl_2020['Vegas_Favorite'].apply(lambda x: x[:len(x)//2])
nfl_2020['game_date'] = nfl_2020['game_date'].apply(lambda x: x[:len(x)//2])
```

In [127…
```
nfl_2020
```

Out[127]:

	game_id	team	def_int	def_int_td	sacks	fumbles_rec	fumbles_rec_td	bl
0	202009100kan202009100kan	HOUKAN	1	0	5	0	0	
1	202009130atl202009130atl	ATLSEA	1	0	5	1	0	
2	202009130buf202009130buf	BUFNYJ	1	0	6	3	0	
3	202009130car202009130car	CARLVR	0	0	1	0	0	
4	202009130cin202009130cin	CINLAC	1	0	5	1	0	
...	...	...	...	...	...	...	...	
264	202101170kan202101170kan	CLEKAN	2	0	2	1	0	
265	202101170nor202101170nor	NORTAM	3	0	1	1	0	
266	202101240gnb202101240gnb	GNBTAM	4	0	6	1	0	
267	202101240kan202101240kan	BUFKAN	1	0	5	1	0	
268	202102070tam202102070tam	KANTAM	2	0	4	0	0	

269 rows × 37 columns

For the numerical columns that are the same for both teams no matter what, we want to divide them all in two, since adding the two rows together doubled them. We do this using the code below.

In [128…

```
nfl_2020['Temperature'] = nfl_2020['Temperature'] / 2
nfl_2020['Humidity'] = nfl_2020['Humidity'] / 2
nfl_2020['Wind_Speed'] = nfl_2020['Wind_Speed'] / 2
nfl_2020['Vegas_Line'] = nfl_2020['Vegas_Line'] / 2
nfl_2020['Over_Under'] = nfl_2020['Over_Under'] / 2
```

In [129…

```
nfl_2020
```

Out[129]:

	game_id	team	def_int	def_int_td	sacks	fumbles_rec	fumbles_rec_td	bl
0	202009100kan202009100kan	HOUKAN	1	0	5	0	0	
1	202009130atl202009130atl	ATLSEA	1	0	5	1	0	
2	202009130buf202009130buf	BUFNYJ	1	0	6	3	0	
3	202009130car202009130car	CARLVR	0	0	1	0	0	
4	202009130cin202009130cin	CINLAC	1	0	5	1	0	
...	...	...	...	...	...	...	...	
264	202101170kan202101170kan	CLEKAN	2	0	2	1	0	
265	202101170nor202101170nor	NORTAM	3	0	1	1	0	
266	202101240gnb202101240gnb	GNBTAM	4	0	6	1	0	
267	202101240kan202101240kan	BUFKAN	1	0	5	1	0	
268	202102070tam202102070tam	KANTAM	2	0	4	0	0	

269 rows × 37 columns

Using code below, we drop columns that we do not need for the analysis.

In [131…

```
nfl_2020 = nfl_2020.drop(columns=['game_id', 'Team_abbrev', 'Opponent_abbrev', 'points
                                  'points_allowed_7_13', 'points_allowed_14_20', 'poir
                                  'points_allowed_28_34', 'points_allowed_35', 'Total_
```

In [132…

```
nfl_2020.head()
```

Out[132]:

	team	def_int	def_int_td	sacks	fumbles_rec	fumbles_rec_td	blocked_kick	safety	def_two_poi
0	HOUKAN	1	0	5	0	0	0	0	
1	ATLSEA	1	0	5	1	0	0	0	
2	BUFNYJ	1	0	6	3	0	0	0	
3	CARLVR	0	0	1	0	0	0	0	
4	CINLAC	1	0	5	1	0	0	0	

5 rows × 25 columns

The team column, which was combined every two rows, will be renamed as 'game' since this represents which two teams compete against another. Opponent score, which was added every other row, is now the total score of the game and will be renamed 'total_score'.

In [133…
```
nfl_2020 = nfl_2020.rename(columns={'team': 'game'})
nfl_2020 = nfl_2020.rename(columns={'Opponent_score': 'total_score'})
nfl_2020.head()
```

Out[133]:

	game	def_int	def_int_td	sacks	fumbles_rec	fumbles_rec_td	blocked_kick	safety	def_two_poi
0	HOUKAN	1	0	5	0	0	0	0	
1	ATLSEA	1	0	5	1	0	0	0	
2	BUFNYJ	1	0	6	3	0	0	0	
3	CARLVR	0	0	1	0	0	0	0	
4	CINLAC	1	0	5	1	0	0	0	

5 rows × 25 columns

We create a new column called 'score_overunder_diff' which is the difference of the points scored in the game and the over under, which is what Vegas predicted the total amount of points scored would be.

In [134…
```
nfl_2020['score_overunder_diff'] = nfl_2020['total_score'] - nfl_2020['Over_Under']
nfl_2020
```

Out[134]:

	game	def_int	def_int_td	sacks	fumbles_rec	fumbles_rec_td	blocked_kick	safety	def_two_
0	HOUKAN	1	0	5	0	0	0	0	
1	ATLSEA	1	0	5	1	0	0	0	
2	BUFNYJ	1	0	6	3	0	0	0	
3	CARLVR	0	0	1	0	0	0	0	
4	CINLAC	1	0	5	1	0	0	0	
...	...	...	...	...	...	...	...	...	
264	CLEKAN	2	0	2	1	0	0	0	
265	NORTAM	3	0	1	1	0	0	0	
266	GNBTAM	4	0	6	1	0	0	0	
267	BUFKAN	1	0	5	1	0	0	0	
268	KANTAM	2	0	4	0	0	0	0	

269 rows × 26 columns

We calculate the max and min differences of total score and over under and find the index of each.

In [135…

```
max_diff = nfl_2020['score_overunder_diff'].max()
max_diff_row = nfl_2020['score_overunder_diff'].idxmax()

min_diff = nfl_2020['score_overunder_diff'].min()
min_diff_row = nfl_2020['score_overunder_diff'].idxmin()
```

We then figure out which game had these differences and what the difference was.

In [136…

```
max_diff_game = nfl_2020.loc[max_diff_row, 'game']
print(max_diff_game, "with a difference of", max_diff)
```

```
BALCLE with a difference of 43.0
```

In [137…

```
min_diff_game = nfl_2020.loc[min_diff_row, 'game']
print(min_diff_game, "with a difference of", min_diff)
```

```
CLEHOU with a difference of -29.5
```

Tediously, we perform linear regressions and find the r-squared of 11 numerical independent variables, that could have an effect on the total score.

In [138…

```
x = nfl_2020['def_int'].values.reshape(-1, 1)
y = nfl_2020['total_score'].values.reshape(-1, 1)
linear_regressor = LinearRegression()
linear_regressor.fit(x, y)

reg = linear_regressor.fit(x, y)
reg.score(x, y)
```

Out[138]:

```
0.002819558949738399
```

In [139…

```
x = nfl_2020['def_int_td'].values.reshape(-1, 1)
y = nfl_2020['total_score'].values.reshape(-1, 1)
linear_regressor = LinearRegression()
linear_regressor.fit(x, y)

reg = linear_regressor.fit(x, y)
reg.score(x, y)
```

Out[139]:

```
0.0026685246600902657
```

In [140…

```
x = nfl_2020['sacks'].values.reshape(-1, 1)
y = nfl_2020['total_score'].values.reshape(-1, 1)
linear_regressor = LinearRegression()
linear_regressor.fit(x, y)

reg = linear_regressor.fit(x, y)
reg.score(x, y)
```

Out[140]:

```
0.05306996430028377
```

In [141…

```
x = nfl_2020['fumbles_rec'].values.reshape(-1, 1)
y = nfl_2020['total_score'].values.reshape(-1, 1)
```

```
linear_regressor = LinearRegression()
linear_regressor.fit(x, y)

reg = linear_regressor.fit(x, y)
reg.score(x, y)
```

```
Out[141]: 0.00014669627238639293
```

```
In [142…
x = nfl_2020['fumbles_rec_td'].values.reshape(-1, 1)
y = nfl_2020['total_score'].values.reshape(-1, 1)
linear_regressor = LinearRegression()
linear_regressor.fit(x, y)

reg = linear_regressor.fit(x, y)
reg.score(x, y)
```

```
Out[142]: 0.0003616118589075956
```

```
In [143…
x = nfl_2020['blocked_kick'].values.reshape(-1, 1)
y = nfl_2020['total_score'].values.reshape(-1, 1)
linear_regressor = LinearRegression()
linear_regressor.fit(x, y)

reg = linear_regressor.fit(x, y)
reg.score(x, y)
```

```
Out[143]: 0.001967414623703201
```

```
In [144…
x = nfl_2020['safety'].values.reshape(-1, 1)
y = nfl_2020['total_score'].values.reshape(-1, 1)
linear_regressor = LinearRegression()
linear_regressor.fit(x, y)

reg = linear_regressor.fit(x, y)
reg.score(x, y)
```

```
Out[144]: 0.00010363599908702614
```

```
In [145…
x = nfl_2020['def_two_point_conv'].values.reshape(-1, 1)
y = nfl_2020['total_score'].values.reshape(-1, 1)
linear_regressor = LinearRegression()
linear_regressor.fit(x, y)

reg = linear_regressor.fit(x, y)
reg.score(x, y)
```

```
Out[145]: 0.0
```

```
In [146…
x = nfl_2020['Temperature'].values.reshape(-1, 1)
y = nfl_2020['total_score'].values.reshape(-1, 1)
linear_regressor = LinearRegression()
linear_regressor.fit(x, y)

reg = linear_regressor.fit(x, y)
reg.score(x, y)
```

```
Out[146]: 0.014750153065433813
```

In [147…
```
x = nfl_2020['Humidity'].values.reshape(-1, 1)
y = nfl_2020['total_score'].values.reshape(-1, 1)
linear_regressor = LinearRegression()
linear_regressor.fit(x, y)

reg = linear_regressor.fit(x, y)
reg.score(x, y)
```

Out[147]:
```
0.007503547335413807
```

In [148…
```
x = nfl_2020['Wind_Speed'].values.reshape(-1, 1)
y = nfl_2020['total_score'].values.reshape(-1, 1)
linear_regressor = LinearRegression()
linear_regressor.fit(x, y)

reg = linear_regressor.fit(x, y)
reg.score(x, y)
```

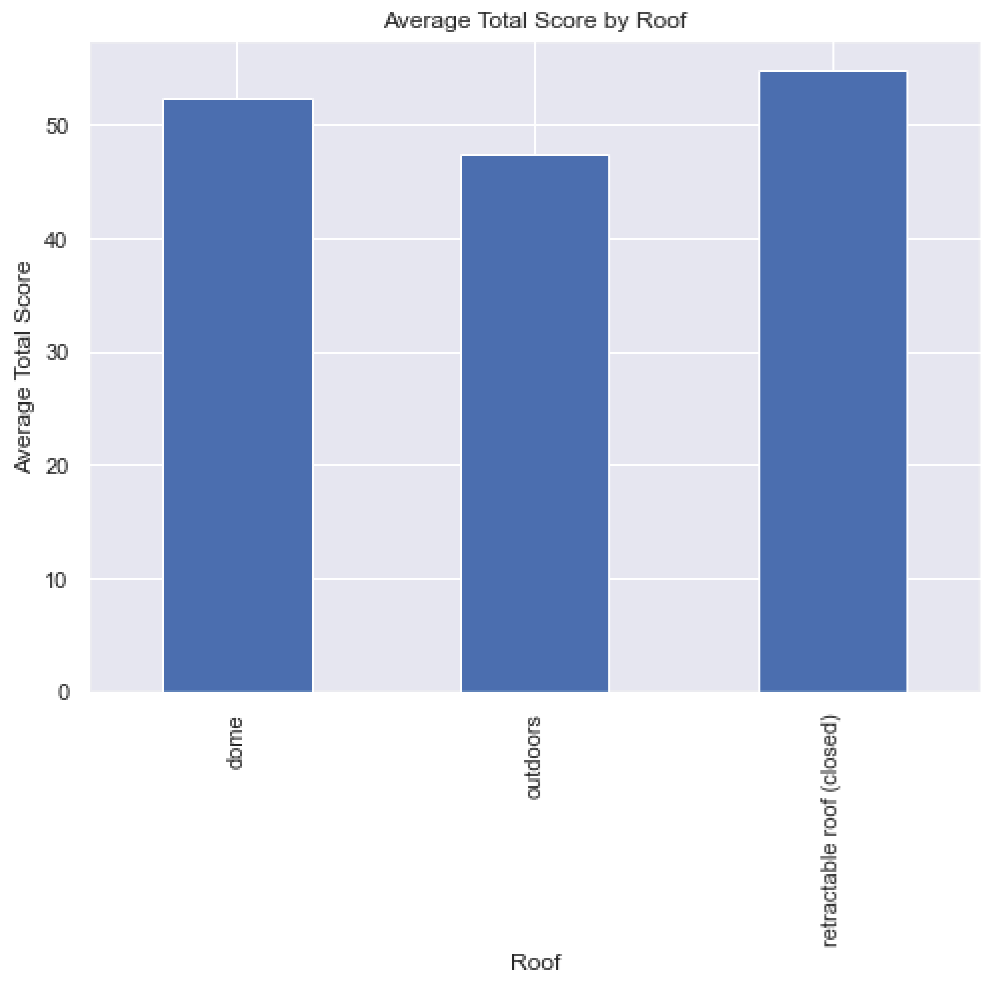
Out[148]:
```
0.028886801001449802
```

After the regressions, we can see that sacks were the most correlated with the total score with r squared about 0.05, while defensive two point conversions were the least corrleated with total score with a with r squared of 0.

We group the total score by the type of roof and type of surface the game was played at and graph the average total score for each roof type and surface type.

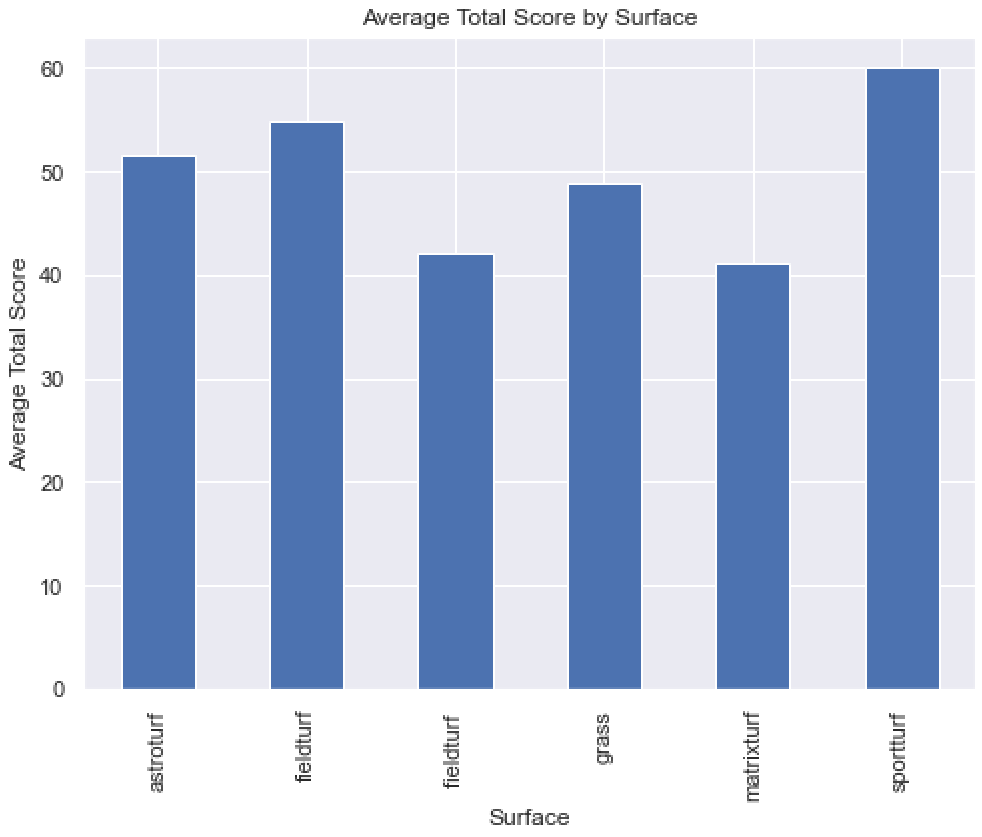
In [149…
```
Roof_mean_scores = nfl_2020.groupby("Roof")["total_score"].mean()
Surface_mean_scores = nfl_2020.groupby("Surface")["total_score"].mean()
```

In [150…
```
fig, ax = plt.subplots(figsize=(8, 6))
Roof_mean_scores.plot(kind='bar', ax=ax)

ax.set_title('Average Total Score by Roof')
ax.set_xlabel('Roof')
ax.set_ylabel('Average Total Score')

plt.show()
```



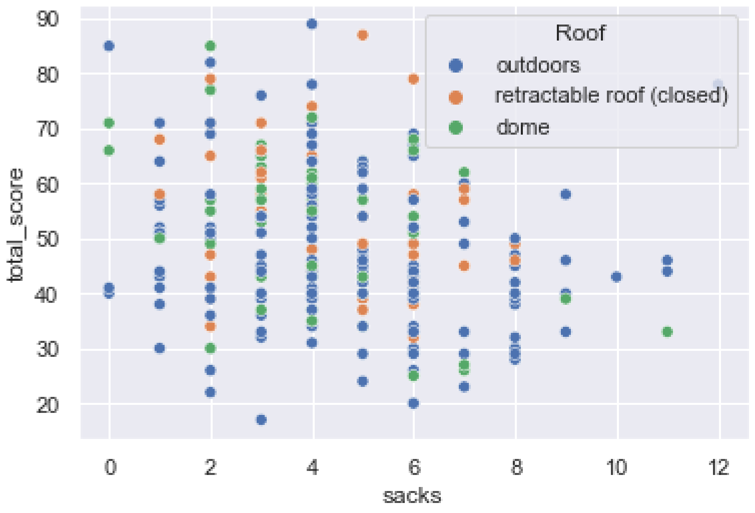
In [151…

```
fig, ax = plt.subplots(figsize=(8, 6))
Surface_mean_scores.plot(kind='bar', ax=ax)

ax.set_title('Average Total Score by Surface')
ax.set_xlabel('Surface')
ax.set_ylabel('Average Total Score')

plt.show()
```

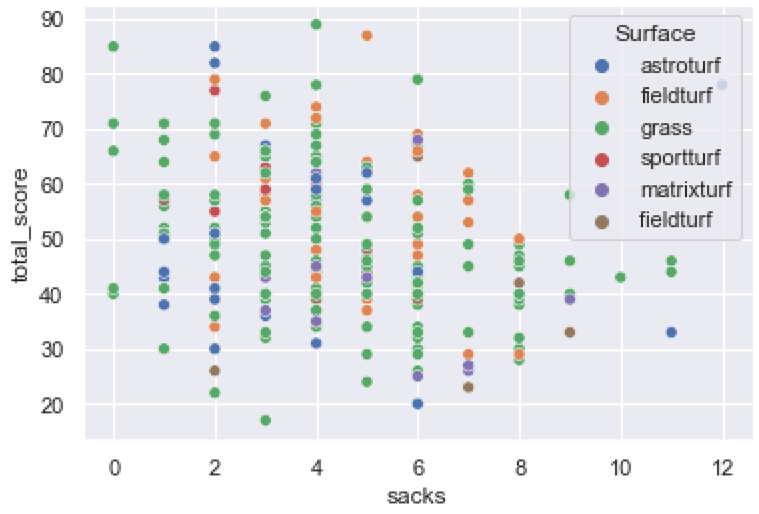


We plot the relationship between the number of sacks, which is the most correlated to the total score, and the total score itself, coloring the points by either Roof or Surface.

In [152…
```
sns.scatterplot(x='sacks', y='total_score', hue='Roof', data=nfl_2020)
plt.show()
```



In [153…
```
sns.scatterplot(x='sacks', y='total_score', hue='Surface', data=nfl_2020)
plt.show()
```



To calculate which games are the most and least defensive, I created my own formula for this. The game defensive score is a number that will be used. The lower the game defensive score is, the more defensive a game is.

The formula for this is game defensive score = (total score - defensive rating)

Defensive rating formula:

Sacks: 1 point

Interceptions, fumbles, blocked kicks, safeties, def two pt conv: 2 points

Def int td, fumble rec td: 6 points

We will be calculating the top 10 most defensive and least defensive games of the 2020 NFL Season.

In [154…
```
nfl_2020['def_rating'] = nfl_2020['sacks'] + 2*(nfl_2020['def_int']+nfl_2020['fumbles_
                                                nfl_2020['safety']+nfl_2020['def_two_p
                                                nfl_2020['def_int_td']+nfl_2020['fumbl
```

In [155…
```
nfl_2020
```

Out[155]:

	game	def_int	def_int_td	sacks	fumbles_rec	fumbles_rec_td	blocked_kick	safety	def_two_
0	HOUKAN	1	0	5	0	0	0	0	
1	ATLSEA	1	0	5	1	0	0	0	
2	BUFNYJ	1	0	6	3	0	0	0	
3	CARLVR	0	0	1	0	0	0	0	
4	CINLAC	1	0	5	1	0	0	0	
...	...	...	...	...	...	...	...	...	
264	CLEKAN	2	0	2	1	0	0	0	
265	NORTAM	3	0	1	1	0	0	0	
266	GNBTAM	4	0	6	1	0	0	0	
267	BUFKAN	1	0	5	1	0	0	0	
268	KANTAM	2	0	4	0	0	0	0	

269 rows × 27 columns

In [156…

```
nfl_2020['game_def_score'] = nfl_2020['total_score'] - nfl_2020['def_rating']
nfl_2020
```

Out[156]:

	game	def_int	def_int_td	sacks	fumbles_rec	fumbles_rec_td	blocked_kick	safety	def_two_
0	HOUKAN	1	0	5	0	0	0	0	
1	ATLSEA	1	0	5	1	0	0	0	
2	BUFNYJ	1	0	6	3	0	0	0	
3	CARLVR	0	0	1	0	0	0	0	
4	CINLAC	1	0	5	1	0	0	0	
...	...	...	...	...	...	...	...	...	
264	CLEKAN	2	0	2	1	0	0	0	
265	NORTAM	3	0	1	1	0	0	0	
266	GNBTAM	4	0	6	1	0	0	0	
267	BUFKAN	1	0	5	1	0	0	0	
268	KANTAM	2	0	4	0	0	0	0	

269 rows × 28 columns

After calculating the game defensive score, we create a new dataframe, with the columns that are of use.

In [157…

```
nfl_2020_def = nfl_2020.loc[:, ['game', 'vis_team', 'home_team', 'Roof', 'Surface', 'g
nfl_2020_def
```

Out[157]:

	game	vis_team	home_team	Roof	Surface	game_def_score	game_date
0	HOUKAN	HOU	KAN	outdoors	astroturf	47	9/10/2020
1	ATLSEA	SEA	ATL	retractable roof (closed)	fieldturf	54	9/13/2020
2	BUFNYJ	NYJ	BUF	outdoors	astroturf	30	9/13/2020
3	CARLVR	LVR	CAR	outdoors	grass	63	9/13/2020
4	CINLAC	LAC	CIN	outdoors	grass	20	9/13/2020
...	...	...	...	...	...	...	...
264	CLEKAN	CLE	KAN	outdoors	astroturf	31	1/17/2021
265	NORTAM	TAM	NOR	dome	astroturf	41	1/17/2021
266	GNBTAM	TAM	GNB	outdoors	grass	41	1/24/2021
267	BUFKAN	BUF	KAN	outdoors	astroturf	53	1/24/2021
268	KANTAM	TAM	KAN	outdoors	grass	32	2/7/2021

269 rows × 7 columns

In [158…

```
nfl_2020_def = nfl_2020_def.sort_values('game_def_score', ascending=True)
nfl_2020_def
```

Out[158]:

	game	vis_team	home_team	Roof	Surface	game_def_score	game_date
253	ARILAR	ARI	LAR	dome	matrixturf	5	1/3/2021
262	BALBUF	BAL	BUF	outdoors	astroturf	6	1/16/2021
115	DALPHI	DAL	PHI	outdoors	grass	6	11/1/2020
71	BALCIN	CIN	BAL	outdoors	grass	8	10/11/2020
238	CARWAS	CAR	WAS	outdoors	grass	8	12/27/2020
...	...	...	...	...	...	...	...
20	ATLDAL	ATL	DAL	retractable roof (closed)	fieldturf	71	9/20/2020
7	GNBMIN	GNB	MIN	dome	sportturf	73	9/13/2020
52	CLEDAL	CLE	DAL	retractable roof (closed)	fieldturf	76	10/4/2020
224	MINNOR	MIN	NOR	dome	astroturf	79	12/25/2020
207	BALCLE	BAL	CLE	outdoors	grass	81	12/14/2020

269 rows × 7 columns

In [159…
```
most_defensive = nfl_2020_def.head(10)
most_defensive
```

Out[159]:

	game	vis_team	home_team	Roof	Surface	game_def_score	game_date
253	ARILAR	ARI	LAR	dome	matrixturf	5	1/3/2021
262	BALBUF	BAL	BUF	outdoors	astroturf	6	1/16/2021
115	DALPHI	DAL	PHI	outdoors	grass	6	11/1/2020
71	BALCIN	CIN	BAL	outdoors	grass	8	10/11/2020
238	CARWAS	CAR	WAS	outdoors	grass	8	12/27/2020
172	MIANYJ	MIA	NYJ	outdoors	fieldturf	8	11/29/2020
148	CARDET	DET	CAR	outdoors	grass	10	11/22/2020
192	LARNWE	NWE	LAR	dome	matrixturf	10	12/10/2020
188	NYGSEA	NYG	SEA	outdoors	fieldturf	12	12/6/2020
205	SFOWAS	WAS	SFO	retractable roof (closed)	grass	12	12/13/2020

The Baltimore Ravens, Los Angeles Rams, Carolina Panthers, and Washington Football Team were each involved in two of these games, the most of any team. The Los Angeles Rams, who play in a domed stadium with matrixturf are the only team to host multiple of the top 10 most defensive games of 2020.

In [160…
```
least_defensive = nfl_2020_def.tail(10)
least_defensive = least_defensive.sort_values('game_def_score', ascending=False)
least_defensive
```

Out[160]:

	game	vis_team	home_team	Roof	Surface	game_def_score	game_date
207	BALCLE	BAL	CLE	outdoors	grass	81	12/14/2020
224	MINNOR	MIN	NOR	dome	astroturf	79	12/25/2020
52	CLEDAL	CLE	DAL	retractable roof (closed)	fieldturf	76	10/4/2020
7	GNBMIN	GNB	MIN	dome	sportturf	73	9/13/2020
261	CLEPIT	CLE	PIT	outdoors	grass	71	1/10/2021
20	ATLDAL	ATL	DAL	retractable roof (closed)	fieldturf	71	9/20/2020
248	HOUTEN	TEN	HOU	retractable roof (closed)	grass	69	1/3/2021
84	HOUTEN	HOU	TEN	outdoors	grass	68	10/18/2020
166	INDTEN	TEN	IND	retractable roof (closed)	fieldturf	66	11/29/2020
247	DETMIN	MIN	DET	dome	fieldturf	66	1/3/2021

The Minnesota Vikings, Cleveland Browns, and Tennesee Titans were each in three of these games, the most of any team. The Dallas Cowboys, who play in a retractable roof stadium with fieldturf surface were the only team to host multiple of these games.

```
In [ ]:
```