Storytelling Data Visualization on UFC Fighters 2024

Introduction:

- Mixed Martial Arts (MMA) is a popular combat sport, and the Ultimate Fighting Championship (UFC) is the premier organization in the MMA world.
- This dataset provides detailed statistics of UFC fighters, including information on their wins, losses, draws, physical attributes.
- In this notebook we will find out a little more about the world of mma using a dataframe containing data on UFC fighters.

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df = pd.read_csv("ufc-fighters-statistics.csv")
```

In [3]:
```
df
```

Out[3]:

	name	nickname	wins	losses	draws	height_cm	weight_in_kg	reach_in_cm	stance	date_of_birth
0	Robert Drysdale	NaN	7	0	0	190.50	92.99	NaN	Orthodox	1981-10-05
1	Daniel McWilliams	The Animal	15	37	0	185.42	83.91	NaN	NaN	NaN
2	Dan Molina	NaN	13	9	0	177.80	97.98	NaN	NaN	NaN
3	Paul Ruiz	NaN	7	4	0	167.64	61.23	NaN	NaN	NaN
4	Collin Huckbody	All In	8	2	0	190.50	83.91	193.04	Orthodox	1994-09-29
...	...	...	...	...	...	...	...	...	...	...
4106	John Campetella	NaN	0	1	0	175.26	106.59	NaN	Orthodox	NaN
4107	Andre Pederneiras	NaN	1	1	2	172.72	70.31	NaN	Orthodox	1967-03-22
4108	Bryson Kamaka	NaN	12	20	1	180.34	77.11	NaN	Orthodox	NaN
4109	Matej Penaz	Money	6	1	0	190.50	83.91	210.82	Southpaw	1996-10-14
4110	Pauline Macias	PITA	4	1	0	162.56	52.16	162.56	Southpaw	1988-06-27

4111 rows × 18 columns

In [4]:
```
df.shape
```

Out[4]:
```
(4111, 18)
```

In [5]:
```
df.head()
```

Out[5]:

	name	nickname	wins	losses	draws	height_cm	weight_in_kg	reach_in_cm	stance	date_of_birth	sig
0	Robert Drysdale	NaN	7	0	0	190.50	92.99	NaN	Orthodox	1981-10-05	
1	Daniel McWilliams	The Animal	15	37	0	185.42	83.91	NaN	NaN	NaN	
2	Dan Molina	NaN	13	9	0	177.80	97.98	NaN	NaN	NaN	
3	Paul Ruiz	NaN	7	4	0	167.64	61.23	NaN	NaN	NaN	
4	Collin Huckbody	All In	8	2	0	190.50	83.91	193.04	Orthodox	1994-09-29	

In [6]:
```
df["date_of_birth"] = pd.to_datetime(df["date_of_birth"])
```

In [7]:
```
df.isna().sum()
```

Out[7]:
```
name                                          0
nickname                                   1854
wins                                          0
losses                                        0
draws                                         0
height_cm                                   298
weight_in_kg                                 87
reach_in_cm                                1927
stance                                      823
date_of_birth                              1135
significant_strikes_landed_per_minute         0
significant_striking_accuracy                 0
significant_strikes_absorbed_per_minute       0
significant_strike_defence                    0
average_takedowns_landed_per_15_minutes       0
takedown_accuracy                             0
takedown_defense                              0
average_submissions_attempted_per_15_minutes  0
dtype: int64
```

In [8]:
```
df.drop(columns = ["nickname"],inplace = True)
df.drop(columns = ["reach_in_cm"],inplace = True)
df.replace({'DOB:': pd.NA},inplace = True)
df.fillna(pd.NA,inplace=True)
```

In [9]:
```
df.head()
```

Out[9]:

	name	wins	losses	draws	height_cm	weight_in_kg	stance	date_of_birth	significant_strikes_landed_pe
0	Robert Drysdale	7	0	0	190.50	92.99	Orthodox	1981-10-05	
1	Daniel McWilliams	15	37	0	185.42	83.91	<NA>	NaT	
2	Dan Molina	13	9	0	177.80	97.98	<NA>	NaT	
3	Paul Ruiz	7	4	0	167.64	61.23	<NA>	NaT	
4	Collin Huckbody	8	2	0	190.50	83.91	Orthodox	1994-09-29	

In [10]:
```
df["matches_played"] = df["wins"]+ df["losses"] + df["draws"]
df["win_percentage"] = round((df["wins"]/df["matches_played"])*100,2)
df["lose_percentage"] = round((df["losses"]/df["matches_played"])*100,2)
df["draw_percentage"] = round((df["draws"]/df["matches_played"])*100,2)
```

In [11]:
```
df
```

Out[11]:

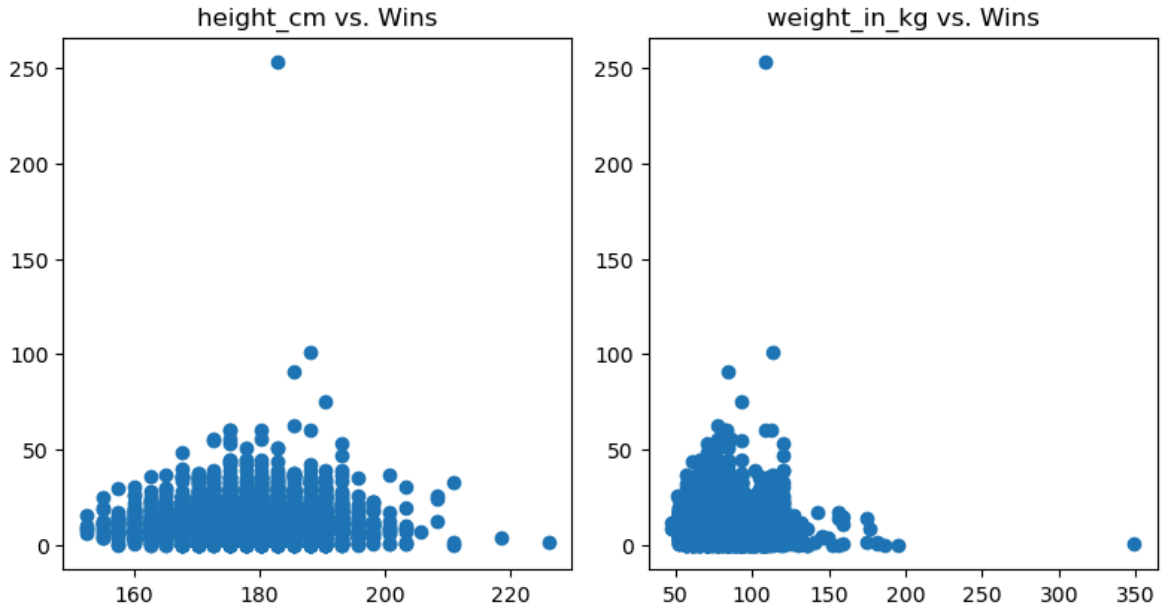
	name	wins	losses	draws	height_cm	weight_in_kg	stance	date_of_birth	significant_strikes_lande
0	Robert Drysdale	7	0	0	190.50	92.99	Orthodox	1981-10-05	
1	Daniel McWilliams	15	37	0	185.42	83.91	<NA>	NaT	
2	Dan Molina	13	9	0	177.80	97.98	<NA>	NaT	
3	Paul Ruiz	7	4	0	167.64	61.23	<NA>	NaT	
4	Collin Huckbody	8	2	0	190.50	83.91	Orthodox	1994-09-29	
...	...	...	...	...	...	...	...	...	
4106	John Campetella	0	1	0	175.26	106.59	Orthodox	NaT	
4107	Andre Pederneiras	1	1	2	172.72	70.31	Orthodox	1967-03-22	
4108	Bryson Kamaka	12	20	1	180.34	77.11	Orthodox	NaT	
4109	Matej Penaz	6	1	0	190.50	83.91	Southpaw	1996-10-14	
4110	Pauline Macias	4	1	0	162.56	52.16	Southpaw	1988-06-27	

4111 rows × 20 columns

In [12]:
```
physical_attributes = ["height_cm","weight_in_kg"]
performance_metrices = ["wins","lossses","draws"]
```

Scatter Plots

In [13]:
```
plt.figure(figsize=(8, 8))

for i, attribute in enumerate(physical_attributes):
    plt.subplot(2,2,i+1)
    plt.scatter(df[attribute],df['wins'])
    plt.title(f'{attribute} vs. Wins')
plt.tight_layout()
plt.show()
```



In [14]:
```
grouped_height = df.groupby('height_cm')['wins'].mean().reset_index()

grouped_height
```

Out[14]:

	height_cm	wins
0	152.40	10.000000
1	154.94	11.400000
2	157.48	9.814815
3	160.02	10.557377
4	162.56	10.705263
5	165.10	11.627737
6	167.64	13.000000
7	170.18	12.814465
8	172.72	13.360000
9	175.26	13.794554
10	177.80	13.515000
11	180.34	13.347107
12	182.88	13.066362
13	185.42	13.366242
14	187.96	13.003831
15	190.50	12.672897
16	193.04	12.691589
17	195.58	9.880952
18	198.12	12.350000
19	200.66	10.636364
20	203.20	10.000000
21	205.74	7.000000
22	208.28	21.000000
23	210.82	11.666667
24	218.44	4.000000
25	226.06	2.000000

In [15]:
```
grouped_weight = df.groupby('weight_in_kg')['wins'].mean().reset_index()
grouped_weight
```

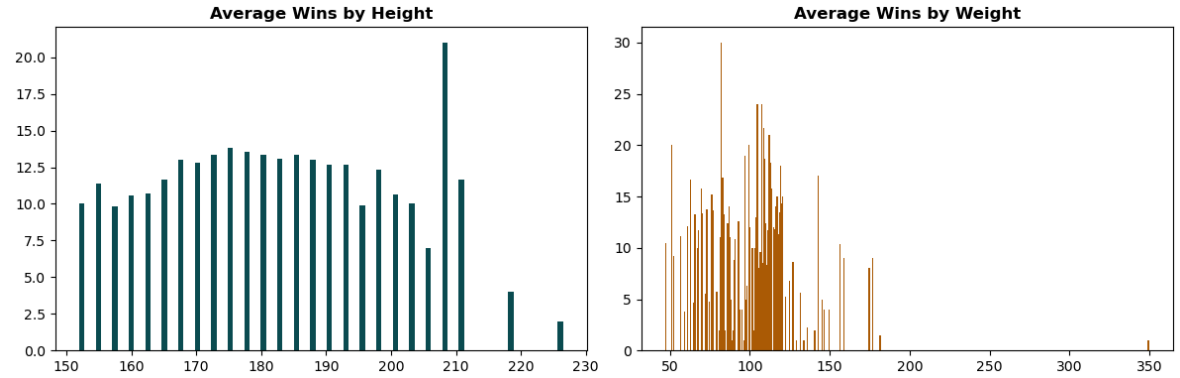
Out[15]:

	weight_in_kg	wins
0	47.63	10.500000
1	51.26	20.000000
2	52.16	9.223140
3	56.70	11.159722
4	58.97	3.750000
...	...	...
107	176.90	9.000000
108	181.44	1.500000
109	185.97	0.000000
110	195.04	0.000000
111	349.27	1.000000

112 rows × 2 columns

Height vs Weight

In [16]:
```
plt.figure(figsize=(12,4))

plt.subplot(1,2,1)
plt.bar(grouped_height['height_cm'], grouped_height['wins'],color = '#0d4c54')
plt.title('Average Wins by Height',weight= 'bold')

plt.subplot(1,2,2)
plt.bar(grouped_weight["weight_in_kg"],grouped_weight['wins'],color = '#ad5a07')
plt.title("Average Wins by Weight",weight = 'bold')

plt.tight_layout()
plt.show()
```



In [17]:
```
stance_performance = df.groupby('stance')['wins'].mean().reset_index()
stance_performance
```

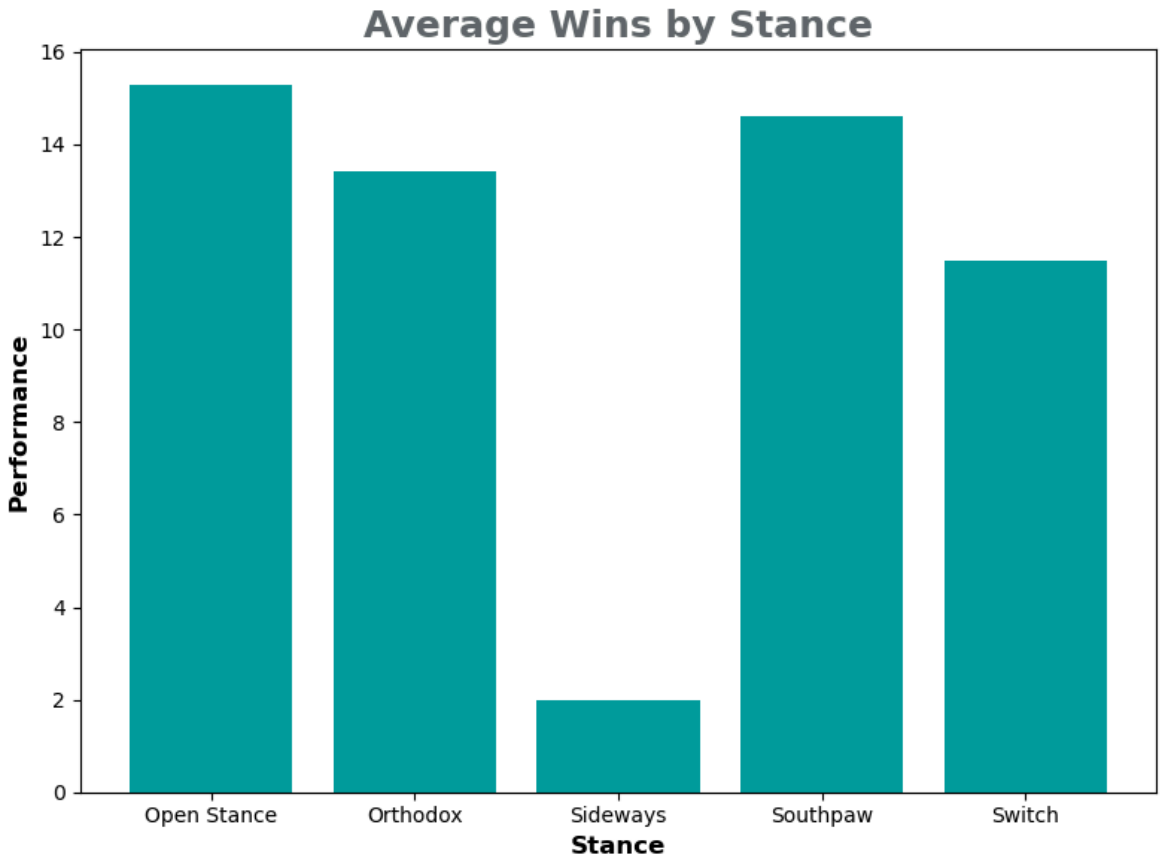
Out[17]:

	stance	wins
0	Open Stance	15.285714
1	Orthodox	13.426762
2	Sideways	2.000000
3	Southpaw	14.626786
4	Switch	11.484375

Determining which stances are more effective and analyze the impact of striking accuracy on fight outcomes

In [18]:

```
plt.figure(figsize=(8,6))

plt.bar(stance_performance["stance"],stance_performance['wins'],color = '#039c9e')
plt.title("Average Wins by Stance",color = '#606869',weight = 'bold',size = 18)
plt.xlabel("Stance",size = 12, weight = 'bold')
plt.ylabel("Performance",size = 12, weight = 'bold')
plt.tight_layout()
plt.show()
```



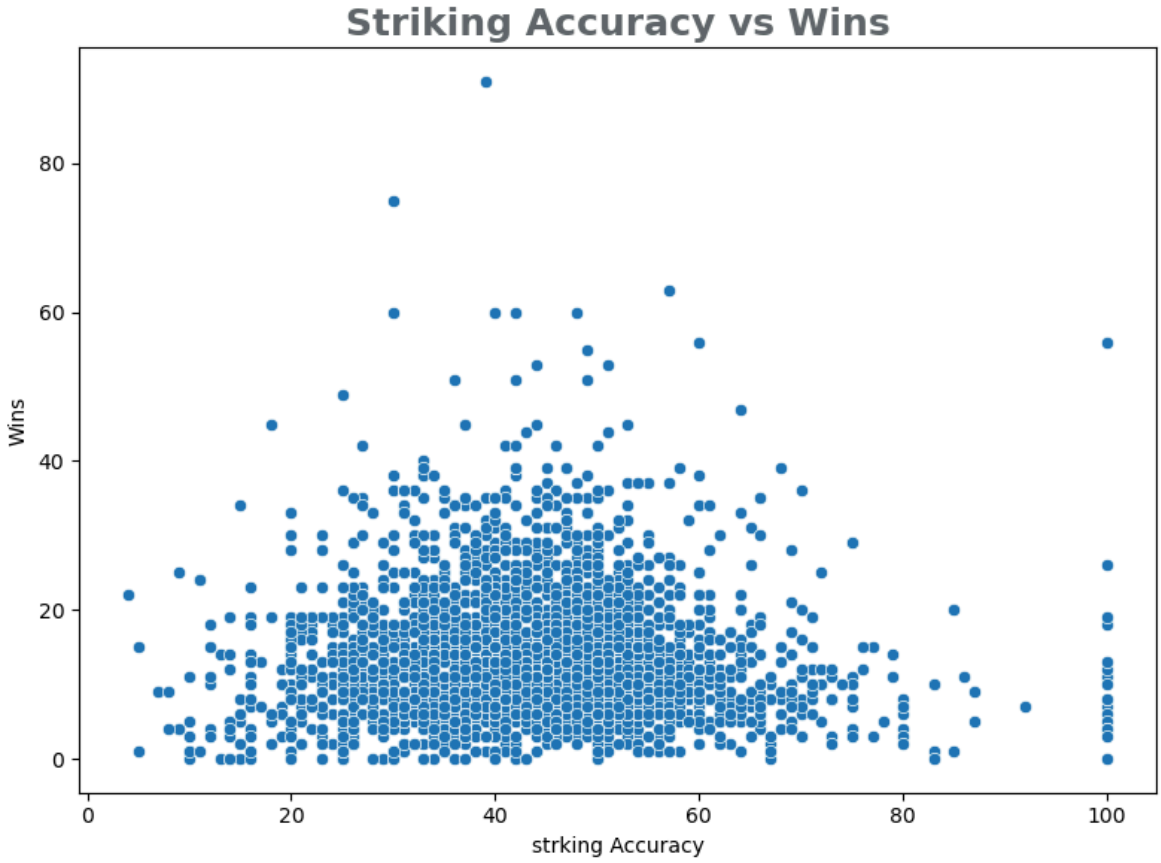
In [19]:
```
df_filtered = df[df['significant_striking_accuracy']>0]
df_filtered
```

Out[19]:

	name	wins	losses	draws	height_cm	weight_in_kg	stance	date_of_birth	significant_strikes_lande
1	Daniel McWilliams	15	37	0	185.42	83.91	<NA>	NaT	
3	Paul Ruiz	7	4	0	167.64	61.23	<NA>	NaT	
4	Collin Huckbody	8	2	0	190.50	83.91	Orthodox	1994-09-29	
6	Isaiah Hill	5	7	1	177.80	70.31	<NA>	NaT	
7	Kenneth Seegrist	4	7	0	182.88	83.91	Orthodox	NaT	
...	...	...	...	...	...	...	...	...	
4103	Abner Lloveras	20	9	1	180.34	70.31	<NA>	NaT	
4104	Brian Melancon	7	3	0	172.72	77.11	Orthodox	1982-05-28	
4108	Bryson Kamaka	12	20	1	180.34	77.11	Orthodox	NaT	
4109	Matej Penaz	6	1	0	190.50	83.91	Southpaw	1996-10-14	
4110	Pauline Macias	4	1	0	162.56	52.16	Southpaw	1988-06-27	

3339 rows × 20 columns

In [20]:
```
plt.figure(figsize = (8,6))
sns.scatterplot(x = 'significant_striking_accuracy', y = 'wins', data = df_filtered)

plt.title("Striking Accuracy vs Wins",weight = 'bold',size = 18,color = '#606869')
plt.xlabel("strking Accuracy")
plt.ylabel("Wins")

plt.tight_layout()
plt.show()
```



In [21]:
```
correlation_coefficient = df_filtered['significant_striking_accuracy'].corr(df_filtered['win
print(f"Correlation Coefficient: {correlation_coefficient}")
```

```
Correlation Coefficient: -0.007914999601794752
```

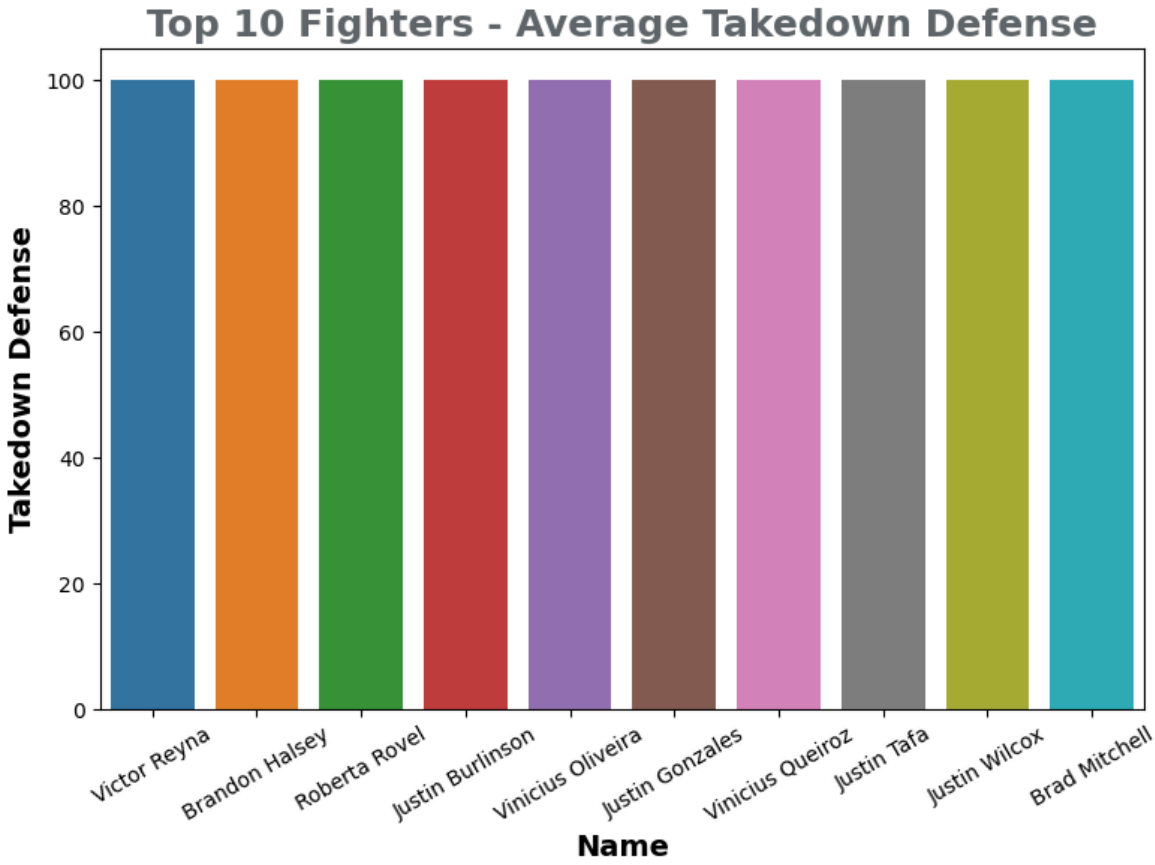
Comparing the takedown defense of different fighters and identify trends in takedown accuracy

In [22]:
```
fighter_takedown_defense = df.groupby('name')['takedown_defense'].mean().reset_index()
top_10_fighters = fighter_takedown_defense.sort_values(by = 'takedown_defense',ascending = F
top_10_fighters
```

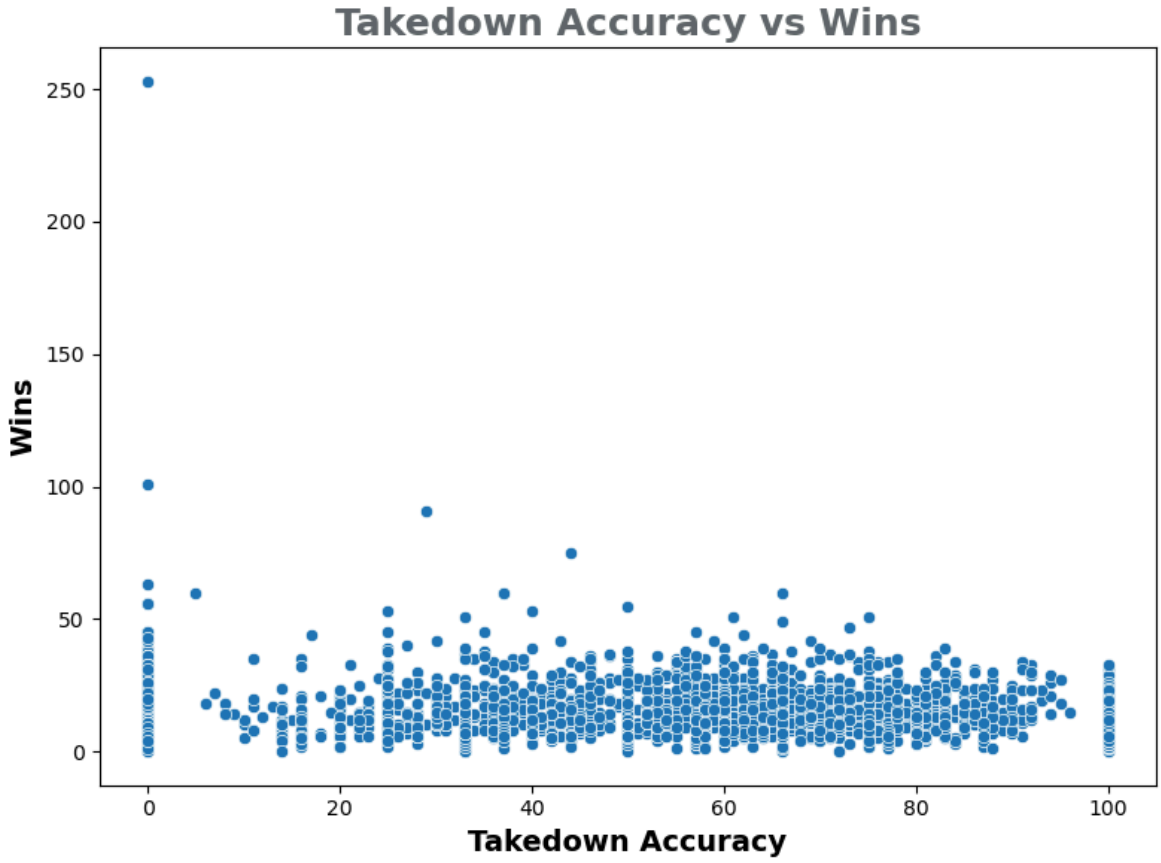
Out[22]:

	name	takedown_defense
3918	Victor Reyna	100.0
455	Brandon Halsey	100.0
3307	Roberta Rovel	100.0
2160	Justin Burlinson	100.0
3934	Vinicius Oliveira	100.0
2167	Justin Gonzales	100.0
3935	Vinicius Queiroz	100.0
2188	Justin Tafa	100.0
2189	Justin Wilcox	100.0
439	Brad Mitchell	100.0

In [23]:
```
plt.figure(figsize=(8,6))

sns.barplot(x= 'name',y='takedown_defense',data = top_10_fighters)
plt.title("Top 10 Fighters - Average Takedown Defense",weight = 'bold',size = 18,color = '#6
plt.xticks(rotation = 30 )
plt.xlabel("Name",size = 14,weight = 'bold')
plt.ylabel("Takedown Defense",size = 14,weight = 'bold')
plt.tight_layout()
plt.show()
```



In [24]:
```
plt.figure(figsize=(8,6))

sns.scatterplot(x= 'takedown_defense',y='wins',data = df)
plt.title("Takedown Accuracy vs Wins",weight = 'bold',size = 18,color = '#606869')

plt.xlabel("Takedown Accuracy",size = 14,weight = 'bold')
plt.ylabel("Wins",size = 14,weight = 'bold')
plt.tight_layout()
plt.show()
```



In [25]:
```
correlation_coefficient = df['takedown_accuracy'].corr(df['wins'])
print(f"Correlation Coefficient: {correlation_coefficient}")
```

```
Correlation Coefficient: 0.19231052370019144
```

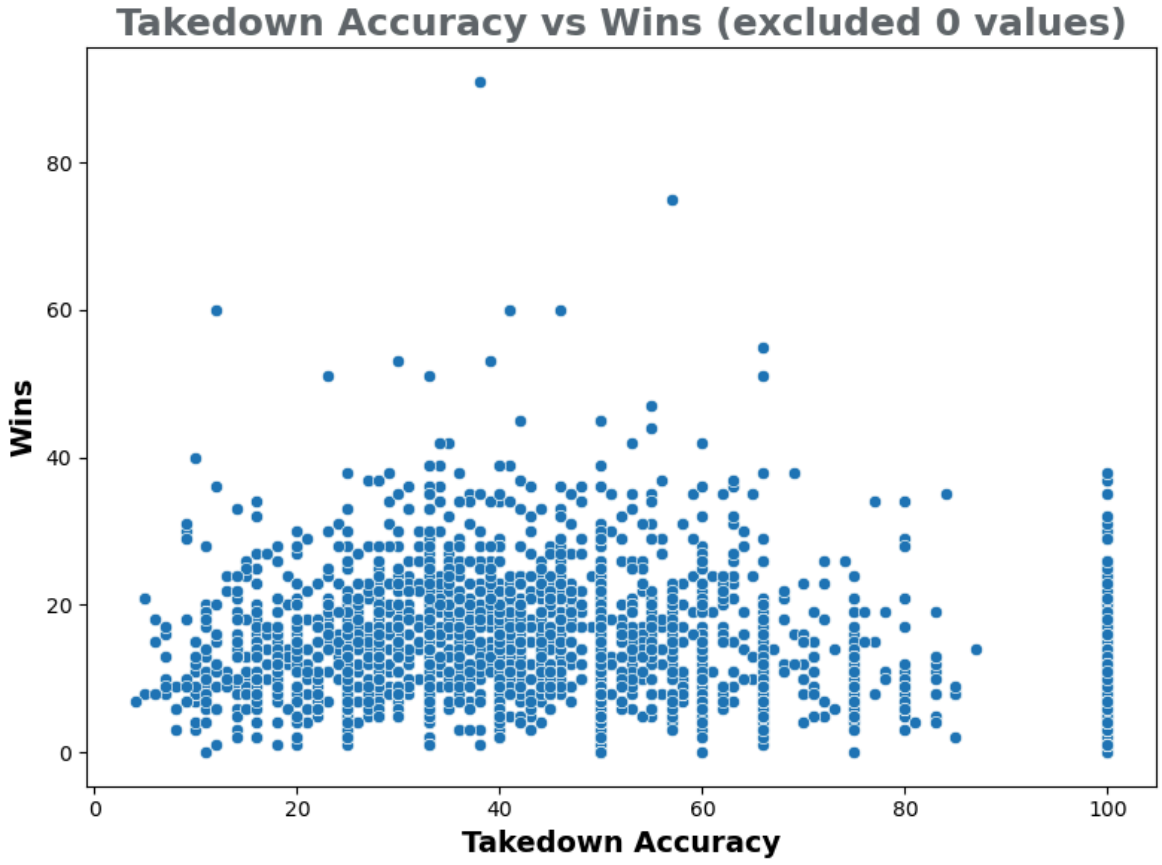
In [26]:
```
df_filtered_takedown = df[df['takedown_accuracy']>0]
df_filtered_takedown
```

Out[26]:

	name	wins	losses	draws	height_cm	weight_in_kg	stance	date_of_birth	significant_strikes_landed
0	Robert Drysdale	7	0	0	190.50	92.99	Orthodox	1981-10-05	
4	Collin Huckbody	8	2	0	190.50	83.91	Orthodox	1994-09-29	
11	Bo Nickal	5	0	0	185.42	83.91	Southpaw	1996-01-14	
15	Kenny Ento	14	15	0	187.96	87.09	<NA>	NaT	
16	AJ McKee	8	0	0	177.80	65.77	<NA>	NaT	
...	...	...	...	...	...	...	...	...	
4096	Eperaim Ginting	6	1	0	162.56	61.23	Orthodox	1995-11-30	
4098	Jack May	9	3	0	203.20	115.67	Switch	1981-04-14	
4103	Abner Lloveras	20	9	1	180.34	70.31	<NA>	NaT	
4104	Brian Melancon	7	3	0	172.72	77.11	Orthodox	1982-05-28	
4110	Pauline Macias	4	1	0	162.56	52.16	Southpaw	1988-06-27	

2373 rows × 20 columns

In [27]:
```
plt.figure(figsize=(8,6))
sns.scatterplot(x= 'takedown_accuracy',y = 'wins', data = df_filtered_takedown)
plt.title('Takedown Accuracy vs Wins (excluded 0 values)',weight = 'bold',size = 18,color =
plt.xlabel("Takedown Accuracy",size = 14,weight = 'bold')
plt.ylabel("Wins",size = 14,weight = 'bold')
plt.tight_layout()
plt.show()
```



In [28]:
```
correlation_coefficient = df_filtered_takedown['takedown_accuracy'].corr(df_filtered_takedow
print(f"Correlation Coefficient: {correlation_coefficient}")
```

```
Correlation Coefficient: -0.09334276792217182
```

Analyzing striking accuracy and defense to identify fighters with the best stand-up game

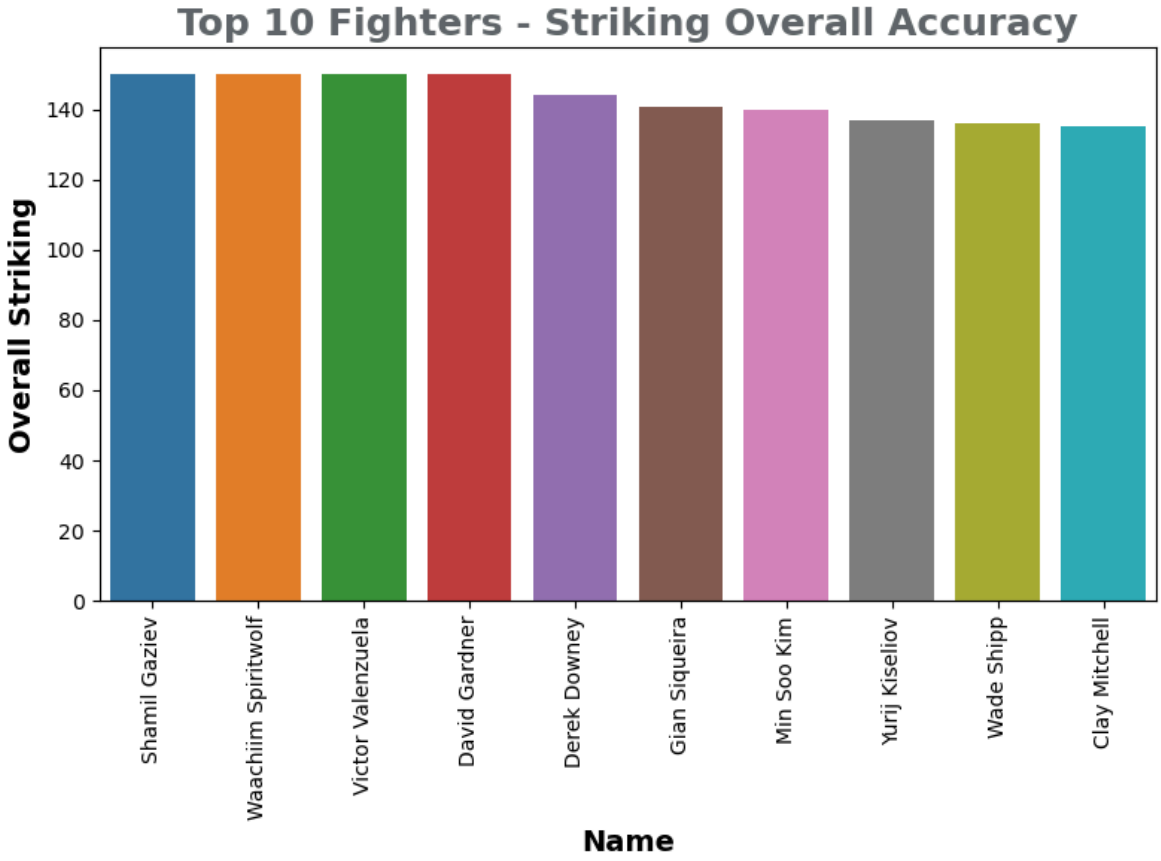
In [29]:
```
df["overall_striking"] = (df['significant_striking_accuracy'] + (100 - df['significant_strik
df["overall_striking"]
```

Out[29]:
```
0        50.0
1       127.0
2        20.0
3        45.5
4        89.0
        ...
4106     50.0
4107     50.0
4108    110.0
4109     66.5
4110     58.0
Name: overall_striking, Length: 4111, dtype: float64
```

In [30]:
```
top_fighters = df[['name', 'overall_striking']].sort_values(by='overall_striking', ascending
print(top_fighters)
```

```
                    name  overall_striking
56          Shamil Gaziev             150.0
3236  Waachiim Spiritwolf             150.0
3607    Victor Valenzuela             150.0
2540        David Gardner             150.0
2927         Derek Downey             144.0
3856        Gian Siqueira             140.5
4024          Min Soo Kim             140.0
3469       Yurij Kiseliov             137.0
3674           Wade Shipp             136.0
100         Clay Mitchell             135.0
```

In [34]:
```
plt.figure(figsize=(8,6))
sns.barplot(x = 'name', y = 'overall_striking',data = top_fighters)
plt.title('Top 10 Fighters - Striking Overall Accuracy',weight = 'bold',size = 18,color = '#

plt.xticks(rotation = 90)
plt.xlabel("Name",size = 14,weight = 'bold')
plt.ylabel("Overall Striking",size = 14,weight = 'bold')
plt.tight_layout()
plt.show()
```



Conclusion:

By analyzing various aspects of UFC fighters' statistics, including physical attributes, fighting techniques, and performance metrics, you can gain insights into the factors that contribute to success in mixed martial arts. These insights can be valuable for fighters, coaches, and fans alike, helping to inform training strategies, game plans, and predictions for future fights.

In [ ]: