Name: Austin Snyder

Title: CS512 Final UFC Database extended

Date: 4/18/2024

Problem Statement: Make informed decisions when placing UFC bets. The UFC is very unpredictable, which can make betting exciting. With a wide variety of fighters, it is unlikely you will be familiar with everyone's record and fighting styles. It would be useful to have a way to compare matchups you know little about by easily pulling up all information without having to research and sort through info. When placing bets, it is likely you are in a social setting, you don't want to be that person in the corner glued to your phone when you should be having a good time, so I want to design something to streamline the process. I want to expand upon the work I did on the midterm, improving upon what I started.

1. Obtain: I pulled all data directly from UFC's website http://www.ufcstats.com/statistics/events/completed.
I used python to write my scraping script. The necessary packages include Selenium, Chrome Driver, and Pandas. The data I chose to scrape includes information on every completed UFC event card, Its title, date, and location. I pulled the results overview for every card as well, this includes each fight, the result, KD, STR, TD, SUB, Weight Class, Method, Round and Time. Finally for every fight I also attempted to pull details including the totals and significant strikes. Totals refers to KD, SIG. STR and its percentage, Total STR, TD and its percentage, SUB. ATT, REV and CTRL. Significant Strikes contains SIG STR and its percentage, Head, Body, Leg, Distance, Clinch and Ground. I pulled he same information from the midterm but improved upon the process by restructuring my functions, so they worked all together, and instead of writing the data frames to a SQLite database I stored the information in csv, as well as JSON dictionaries. I was able to successfully scrape the entirety of fight total and significant strikes information this time.

Estimated Complexity: Data Split across multiple files: 1pt
Data not in CSV, Json, in DB: 2pts
Data is split up across multiple files to begin with: 1pt
Data Contains Strings….: 1pt
Data is larger than 1GB: 1pt
Data comprised of more than one type of related data: 2pts
Data needs to be accessed…(scraped): 1 pt
total: 9pts

2. Scrub: I made some improvements upon the midterm when it came to cleaning my data during the pull. Initially I was hoping to be able to clean and reformat data after it was placers in a SQLite database but that proved to be quite the haste so I took some preventative measures to avoid that issue this time. I added in three cleaning functions to take care of the strings that I ran into on the midterm. Some of the stats contained of, for example, in significant strikes, a possible result could be 29 of 61. While the percentage, is also given the sting prevents any form of numerical analysis and contains important information like strikes attempted and strikes landed. My functions keep these columns but in addition split them and add new columns containing attempted and landed fields so analysis on these categories can be preformed. Below shows how the information is presented on UFC's statistic website where the data is scraped from

TOTALS

FIGHTER	KD	SIG. STR.	SIG. STR. %	TOTAL STR.	TD	TD %	SUB. ATT	REV.	CTRL
Kevin Holland	0	29 of 61	47%	64 of 99	2 of 5	40%	1	0	4:22
Michael Page	0	41 of 62	66%	59 of 81	0 of 3	0%	0	0	1:26

PER ROUND

SIGNIFICANT STRIKES

FIGHTER	SIG. STR	SIG. STR. %	HEAD	BODY	LEG	DISTANCE	CLINCH	GROUND
Kevin Holland	29 of 61	47%	11 of 37	4 of 6	14 of 18	18 of 47	4 of 5	7 of 9
Michael Page	41 of 62	66%	28 of 47	8 of 10	5 of 5	38 of 58	2 of 3	1 of 1

below is an example of the functions I added to scrub

```python
def clean_fight_details(df):
    df["KD"] = pd.to_numeric(df["KD"], errors="coerce").fillna(0).astype(int)
    df[["SIG_STR_landed", "SIG_STR_attempted"]] = df["SIG_STR"].apply(
        lambda x: pd.Series(convert_to_landed_attempted(x))
    )
    df[["TOTAL_STR_landed", "TOTAL_STR_attempted"]] = df["TOTAL_STR"].apply(
        lambda x: pd.Series(convert_to_landed_attempted(x))
    )
    df[["TD_landed", "TD_attempted"]] = df["TD"].apply(
        lambda x: pd.Series(convert_to_landed_attempted(x))
    )

    # Handle percentage fields
    df["TD_PCT"] = df["TD_PCT"].replace("---", "0%").str.rstrip("%").astype(float) / 100
    df["SIG_STR_PCT"] = (
        df["SIG_STR_PCT"].replace("---", "0%").str.rstrip("%").astype(float)
    )

    df["SUB_ATT"] = pd.to_numeric(df["SUB_ATT"], errors="coerce").fillna(0).astype(int)
    df["REV"] = pd.to_numeric(df["REV"], errors="coerce").fillna(0).astype(int)
    df["CTRL"] = df["CTRL"].fillna("0:00")
    df["CTRL"] = df["CTRL"].astype(str)

    return df


def clean_significant_strikes(df):
    for col in ["SIG_STR", "HEAD", "BODY", "LEG", "DISTANCE", "CLINCH", "GROUND"]:
        df[[f"{col}_landed", f"{col}_attempted"]] = df[col].apply(
            lambda x: pd.Series(convert_to_landed_attempted(x))
        )
    df["SIG_STR_PCT"] = df["SIG_STR_PCT"].str.rstrip("%").astype(float)
    df.fillna(0, inplace=True)
    return df.drop(
        ["SIG_STR", "HEAD", "BODY", "LEG", "DISTANCE", "CLINCH", "GROUND"], axis=1
    )
```

As a result I was able to write the data for each table into JSON in a very clean format, some examples of the scraped data in JSON are below.

Cards table:

cards.json > {} 670 > card_link

```json
[
    {
        "card_link": "http://www.ufcstats.com/event-details/a9df5ae20a97b090",
        "title": "UFC 299: O'Malley vs. Vera 2",
        "date": "2024-03-09",
        "location": "Miami, Florida, USA",
        "eventID": "0681"
    },
    {
        "card_link": "http://www.ufcstats.com/event-details/e4a9dbade7c7e1a7",
        "title": "UFC Fight Night: Rozenstruik vs. Gaziev",
        "date": "2024-03-02",
        "location": "Las Vegas, Nevada, USA",
        "eventID": "0680"
    },
```

Fights table:

fights.json > {} 0

```json
[
    {
        "fight_link": "http://www.ufcstats.com/fight-details/894c44c3d04aaf6f",
        "eventID": "0681"
    },
    {
        "fight_link": "http://www.ufcstats.com/fight-details/6e73ebbd089aff5e",
        "eventID": "0681"
    },
```

Fight details table:

fight_details.json > {} 1

```json
[
    {
        "FIGHTER": "Sean O'Malley",
        "KD": 0,
        "SIG_STR": "230 of 356",
        "SIG_STR_PCT": 64.0,
        "TOTAL_STR": "232 of 358",
        "TD": "0 of 0",
        "TD_PCT": 0.0,
        "SUB_ATT": 0,
        "REV": 0,
        "CTRL": "0:03",
        "EVENT_TITLE": "UFC 299: O'Malley vs. Vera 2",
        "SIG_STR_landed": 230,
        "SIG_STR_attempted": 356,
        "TOTAL_STR_landed": 232,
        "TOTAL_STR_attempted": 358,
        "TD_landed": 0,
        "TD_attempted": 0
    },
    {
```

Significant strikes table:


```
[
    {
        "FIGHTER": "Sean O'Malley",
        "SIG_STR": "230 of 356",
        "SIG_STR_PCT": "64%",
        "HEAD": "150 of 268",
        "BODY": "61 of 68",
        "LEG": "19 of 20",
        "DISTANCE": "227 of 352",
        "CLINCH": "3 of 4",
        "GROUND": "0 of 0",
        "EVENT_TITLE": "UFC 299: O'Malley vs. Vera 2",
        "SIG_STR_landed": 230,
        "SIG_STR_attempted": 356,
        "HEAD_landed": 150,
        "HEAD_attempted": 268,
        "BODY_landed": 61,
        "BODY_attempted": 68,
        "LEG_landed": 19,
        "LEG_attempted": 20,
        "DISTANCE_landed": 227,
        "DISTANCE_attempted": 352,
        "CLINCH_landed": 3,
        "CLINCH_attempted": 4,
        "GROUND_landed": 0,
        "GROUND_attempted": 0
    }
```


After I was happy with the state my data was in after writing to JSON, I uploaded my data to Google Big Query. I began by creating a new project titles UFC_Final, then created a new bucket to store my data in. I uploaded my JSON files to this bucket and proceeded to data prep. My data required very minimal cleaning in prep itself, some steps I needed to take were removing a percent symbol from significant strikes percentage column in significant strike that I missed within my cleaning function in python. I also removed some null values that occurred from specific events that took place in the early 90s where the format did not follow the typical statistic format used to display the fighter's metrics. These inconsistencies resulted in mismatched datatypes. When I was satisfied with how my data looked I ran each individual recipe job to Big Query. The results are below.

Fights

fight_link	eventID
http://www.ufcstats.com/fight-details/39be68189f2d5074	256
http://www.ufcstats.com/fight-details/a6e494c82d136172	256
http://www.ufcstats.com/fight-details/5e4f358e0bcc2f15	256
http://www.ufcstats.com/fight-details/27635cc4d9fcac98	256
http://www.ufcstats.com/fight-details/a060b6b0dfe275b7	256
http://www.ufcstats.com/fight-details/b1fd21231f0604e1	256
http://www.ufcstats.com/fight-details/2a17686c6b5f334e	256
http://www.ufcstats.com/fight-details/11706648d34ff3e8	256
http://www.ufcstats.com/fight-details/315968a74789600a	256
http://www.ufcstats.com/fight-details/f1a9c3187855155c	256

2 columns 7698 rows This is a preview of the current data in your destination. It might not reflect the

View on BigQuery View details

Cards

card_link	title	date	location	eventID
http://www.ufcstats.com/event-details/a8fa0c4e95512806	UFC 23: Ultimate Japan 2	1999-11-19T00:00:00	Chiba, Japan	26
http://www.ufcstats.com/event-details/770b9d4813c25902	UFC Fight Night: Bisping vs Le	2014-08-23T00:00:00	Macau, China	285
http://www.ufcstats.com/event-details/319fa1bd3176bded	UFC Macao: Franklin vs Le	2012-11-10T00:00:00	Macau, China	218
http://www.ufcstats.com/event-details/a26198ba5093147e	UFC Fight Night: Kim vs Hathaway	2014-03-01T00:00:00	Macau, China	263
http://www.ufcstats.com/event-details/08ae5cd9aef7ddd3	UFC 29: Defense of the Belts	2000-12-16T00:00:00	Tokyo, Japan	32
http://www.ufcstats.com/event-details/d2b1c1317a39f6c6	UFC 25: Ultimate Japan 3	2000-04-14T00:00:00	Tokyo, Japan	28
http://www.ufcstats.com/event-details/de25520d54eab12d	UFC Fight Night: Blaydes vs. Ngannou 2	2018-11-24T00:00:00	Beijing, China	457
http://www.ufcstats.com/event-details/d6b68eaf4b68b160	UFC Fight Night: Cerrone vs. Till	2017-10-21T00:00:00	Gdansk, Poland	413
http://www.ufcstats.com/event-details/5de61b03868035ff	UFC Fight Night: Gonzaga vs Cro Cop 2	2015-04-11T00:00:00	Krakow, Poland	313
http://www.ufcstats.com/event-details/a4dd5c9a75763295	UFC Fight Night: Barnett vs Nelson	2015-09-26T00:00:00	Saitama, Japan	333

5 columns 681 rows This is a preview of the current data in your destination. It might not reflect the output from this particular job run.

View on BigQuery View details

Fight details

FIGHTER	KD	SIG_STR	SIG_STR_PCT	TOTAL_STR	TD	TD_PCT	SUB_ATT
Je[illegible]rade	4	53 of 107	49	53 of 107	0 of 0	0	0
Viacheslav Borshchev	3	85 of 148	57	85 of 148	0 of 0	0	0
Zak Cummings	3	100 of 175	57	114 of 189	0 of 0	0	0
Steve Garcia	3	29 of 44	65	31 of 46	0 of 0	0	0
Marlon Vera	3	61 of 156	39	63 of 160	0 of 0	0	0
Gregory Rodrigues	3	49 of 72	68	50 of 74	0 of 0	0	0
Marlon Vera	3	159 of 283	56	167 of 291	0 of 0	0	0
Matt Frevola	4	60 of 101	59	71 of 114	0 of 0	0	0
Daniel Rodriguez	3	15 of 32	46	15 of 32	0 of 0	0	0

17 columns 14986 rows This is a preview of the current data in your destination. It might not reflect the output from this particular job run.

View on BigQuery View details

SIG_STR_landed	SIG_STR_attempted	TOTAL_STR_landed	TOTAL_STR_attempted	TD_landed	TD_attempted
53	107	53	107	0	0
85	148	85	148	0	0
100	175	114	189	0	0
29	44	31	46	0	0
61	156	63	160	0	0
49	72	50	74	0	0
159	283	167	291	0	0
60	101	71	114	0	0
15	32	15	32	0	0

17 columns 14986 rows This is a preview of the current data in your destination. It might not reflect the output from this particular job run.

Significant strikes

FIGHTER	SIG_STR	SIG_STR_PCT	HEAD	BODY	LEG	DISTANCE	CLINCH	
Andre Petroski	0 of 1	0	0 of 1	0 of 0	0 of 0	0 of 1	0 of 0	0 of
Yazmin Jauregui	0 of 1	0	0 of 1	0 of 0	0 of 0	0 of 1	0 of 0	0 of
Alexandr Romanov	0 of 1	0	0 of 1	0 of 0	0 of 0	0 of 1	0 of 0	0 of
Khamzat Chimaev	0 of 1	0	0 of 1	0 of 0	0 of 0	0 of 0	0 of 0	0 of
Makwan Amirkhani	0 of 1	0	0 of 1	0 of 0	0 of 0	0 of 1	0 of 0	0 of
Ramiz Brahimaj	0 of 1	0	0 of 1	0 of 0	0 of 0	0 of 1	0 of 0	0 of
Li Jingliang	0 of 1	0	0 of 1	0 of 0	0 of 0	0 of 1	0 of 0	0 of
Fabio Cherant	0 of 1	0	0 of 1	0 of 0	0 of 0	0 of 1	0 of 0	0 of
Gerald Meerschaert	0 of 1	0	0 of 1	0 of 0	0 of 0	0 of 1	0 of 0	0 of

24 columns 15259 rows This is a preview of the current data in your destination. It might not reflect the output from this particular job run.

View on BigQuery View details

DISTANCE_landed	DISTANCE_attempted	CLINCH_landed	CLINCH_attempted	GROUND_landed	GROUND_attempted
	1	0	0	0	0
	1	0	0	0	0
	1	0	0	0	0
	0	0	0	0	1
	1	0	0	0	0
	1	0	0	0	0
	1	0	0	0	0
	1	0	0	0	0
	1	0	0	0	0

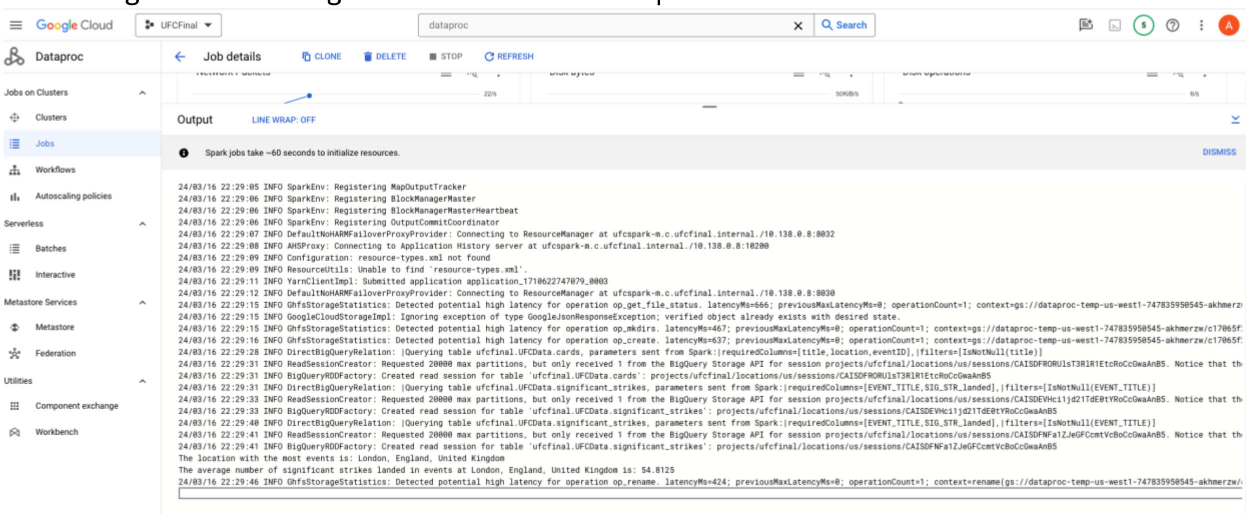
nections

24 columns 15259 rows This is a preview of the current data in your destination. It might not reflect the output from this particular job run.

View on BigQuery View details

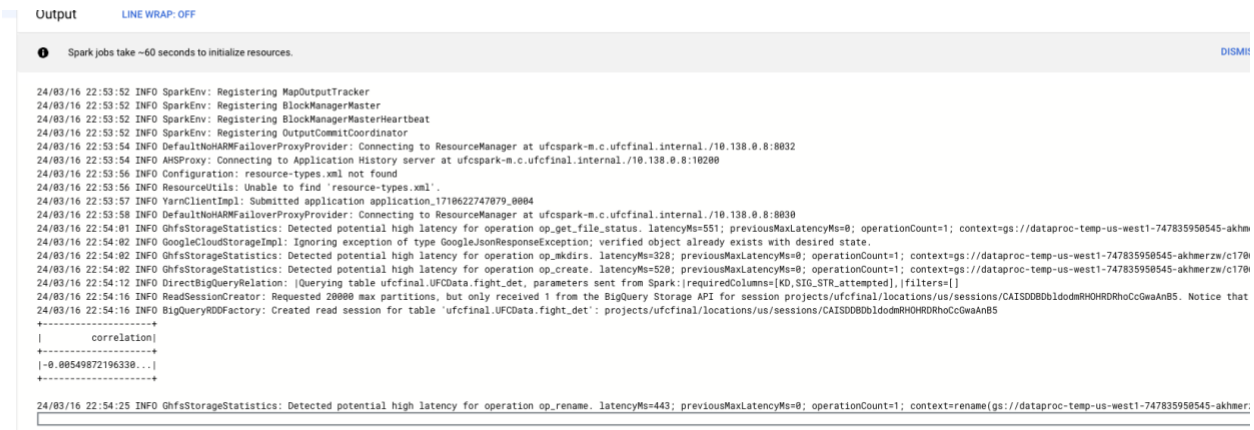
3. Explore: I ran all my questions in spark using dataproc. When a job is executed in Spark the execution engine breaks down the job into tasks that are distributed across the nodes in the cluster. Each node processes a subset of the data in parallel with other nodes. My jobs are job utilizing parallel processing.

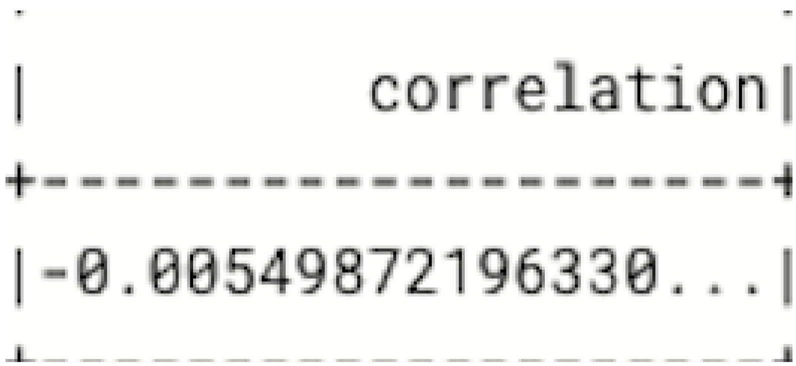
1. UFC Events are held in many different locations. I was curious to know which location has hosted the greatest number of events and what the average number of significant strikes landed was at this location. In order to answer this, I used spark. I needed to group cards by location, and count the number of events for each card resulting in the location that hosted the most events. A join would be performed with significant strikes, on event title to then calculate the average number of significant strikes for that specific location.



Data Proc output shows London UK to be the most common location with an average of 54 significant strikes landed.

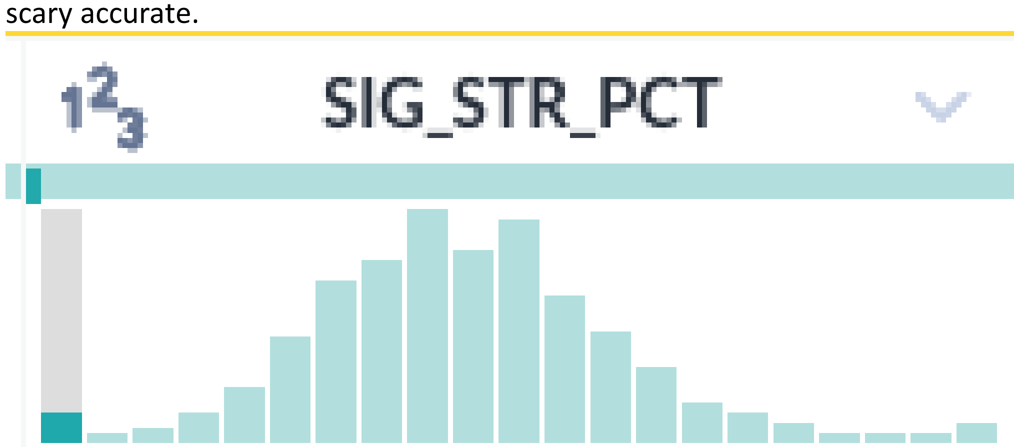
2. Knockdowns occur when a fighter touches the floor of the octagon with any body part aside from their feet following some sort of hit but is able to return to their feet and continue the fight. There are many different styles of fighting but I'd assume the more strikes that occur the more likely it is to see a larger quantity of KD's. To test this I wanted to see if there was a correlation b=between significant strikes attempted and KD. To get this result I used the fight_det table to extract KD and SIG_STR_attempted columns. Then calculate the correlation between these two variables to see if there's any statistical relationship between the frequency of knockdowns and the aggressiveness (as indicated by the number of attempts) of the fighters.





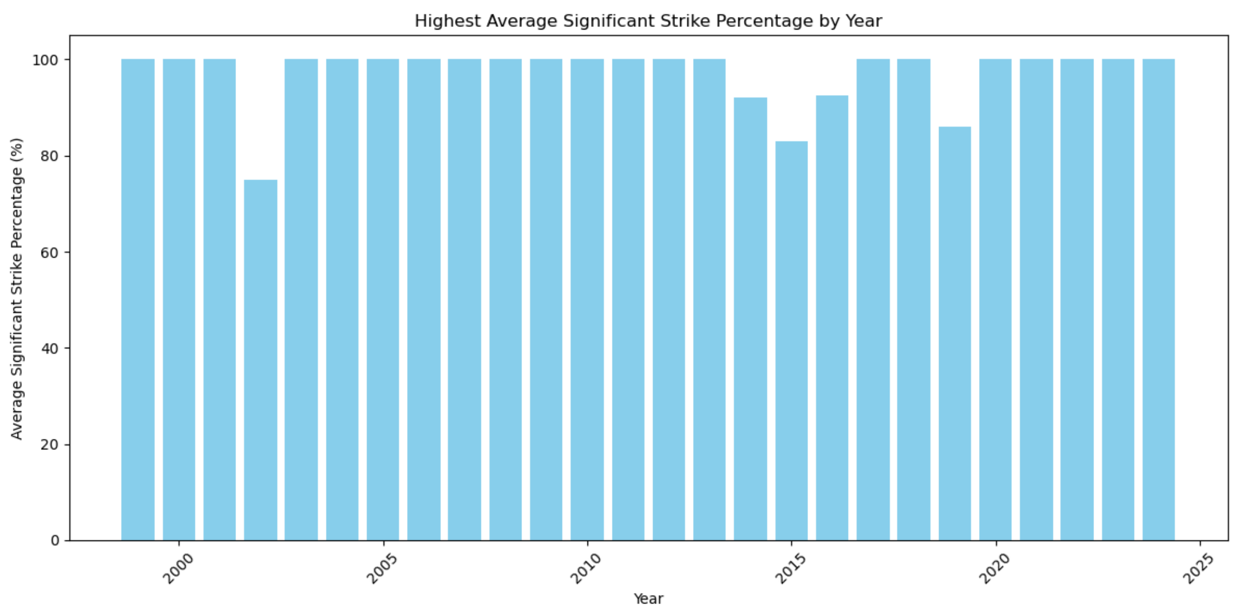
The output resulted in -.00549, which I found to be surprising. You would expect to see a strong correlation, but this may be because I'm looking at the total data as a whole. If I dug a little deeper, by weight class, and round I may find out more information. It is common that many strikes are thrown lightly and don't result in KDs.

3. A significant strike percentage of 100 is definitely an outlier but some fighters can be scary accurate.



You can see that sig strike percentage appears to be close to normally distributed so 100 percent is uncommon, I wanted to see the best preforming significant striker for each year. To do this I used the fight_det and cards table. I extracted the year from the date column within the cards data frame, joined fight details with cards on event_title. I then grouped the data by year and fighter to calculate average significant strikle percentage for each fighter per year. Finally I used the window function to find the fighter with the highest average significant strike percentage for each year.

```
24/03/16 23:44:49 INFO BigQueryRDDFactory: Created read session for table 'ufcfinal.UFCData.fight_det': projects/ufcfinal/locations/us/sessions/CAISDHNnTnU5eWxYYXc4dxoC
+----+--------------------+---------------+
|year|             FIGHTER|avg_SIG_STR_PCT|
+----+--------------------+---------------+
|1999|           Joe Slick|          100.0|
|2000|      Marcelo Aguiar|          100.0|
|2001|      Paul Rodriguez|          100.0|
|2002|Vladimir Matyushenko|           75.0|
|2003|        Keith Rockel|          100.0|
|2004|    Jonathan Wiezorek|          100.0|
|2005|          Bill Mahood|          100.0|
|2006|        Cory Walmsley|          100.0|
|2007|          Demian Maia|          100.0|
|2008|       Ivan Salaverry|          100.0|
|2009|      Jason MacDonald|          100.0|
|2010|        Ricardo Funch|          100.0|
|2011|          Mike Guymon|          100.0|
|2012|           Ryan Jimmo|          100.0|
|2013|           Mike Brown|          100.0|
|2014|      Chris Clements|           92.0|
|2015|     Anthony Hamilton|           83.0|
|2016|          Demian Maia|           92.5|
|2017|       Teemu Packalen|          100.0|
|2018|        Gunnar Nelson|          100.0|
+----+--------------------+---------------+
```



I then saved this data frame as a csv, then made a bar chart using matplotlib. There was only a few years where no fighter was able to hit a 100 percent average significant strike percentage.

4. Model: Modeling was not used for this part of the project but is something I am looking forward on implementing in the future. I would like to make a functional app in the future that allows you to search by fighter and a breakdown of summary statistics will be presented to you. You would also be able to view upcoming fights and compare two fighters' statistics side by side.

5. Interpret:

In the future I will need to add a few more statistics that are provided on the site but I did not capture.