Rafael Hernandez
George Washington Consulting Report
June 11, 2021

FINAL PROJECT REPORT

Link to colab notebook:

https://colab.research.google.com/drive/1zf0SfKo7ulbXR8BnasS2iyFGah17qT46?usp=sharing

Part I: Describe the data and it's sources(s)

The data comes from https://www.kaggle.com/andrewsundberg/xollege-basketball-dataset The data set consists of 10 csv files starting from 2013 to 2021. The 2020 file is incomplete because the 2020 college basketball season was cut short due to the Corona Virus pandemic. The 2020 file will not be part of the examination. Below is the index for the variables in the files.

Variables

Team: Division 1 college school

Conf: The conference the team represents

G: number of games played

W: Number of games won

ADJOE: Adjusted Offensive Efficiency (An estimate of the offensive efficiency (points scored per 100 possessions) a team would have against the average Division 1 defense)

ADJDE: Adjusted Defensive Efficiency (An estimate of the defensive efficiency (points allowed per 100 possessions) a team would have against the average Division 1 team)

EFG_0: Effective Field Goal Percentage shot

EFG_D: Effective field goal percentage allowed

TOR: Turnover Percentage Allowed (Turnover Rate)

TORD: Turnover Percentage Committed (Steal Rate)

ORB: Offensive Rebound Rate

DRB: Offensive Rebound Rate Allowed

FTR: Free Throw Rate (How often the given team shoots Free Throws)

FTRD: Free Throw Rate Allowed

2P_0: Two-Point Shooting Percentage

2P_D: Two-Point Shooting Percentage Allowed

3P_0: Three-Point Shooting Percentage

3P_D: Three-Point Shooting Percentage Allowed

ADJ_T: Adjusted Temp(An estimate of the tempo (possession per 40 minutes) a team would have against the team that wants to play an average Division 1 tempo)

WAB: Wins Above Bubble (The bubble refers to the cut off between making the NCAA March Madness Tournament and not making it)

POSTSEASON: Round where the given team was eliminated or where their season ended

SEED: Seed in the NCAA March Madness Tournament

Rafael Hernandez
George Washington Consulting Report
June 11, 2021

YEAR: Season

PART II: Describe your methods of analysis, including the questions that will be answered, in what field the data will be used, and what resulting outputs will be

The method of analysis centers on a client ranked 90th in the 2020 college basketball dataset. The client (George Washington University) did not make the tournament and wants to examine the data set to see the average baseline for lower seeded teams in the tournament and compare those averages with the averages for top seeded teams in the NCAA tournament. The focus on lowest tournament seed and highest tournament seeds is a reflection of short-term goal(make the NCAA tournament in 2022) and long-term goal(elevate the program to elite status and consistently be a high seed in the tournament. To accomplish this analysis the following questions will be answered.

1. What are the average defensive and offensive stats for 16 seeds from 2013 to 2021?
2. What is the average defensive and offensive stats for 1 seed from 2013 to 2021?
3. Out of all the variables is there a variable(s) that can most predict who is a 16 seed and who is a 1 seed?
4. What areas does George Washington University need to improve on to make the ncaa tournament in 2022?

The dataset has a total of 23 variables. TEAM and SEED will be used to identify teams and whether they made the tournament. Conference, G, BARTHAG, ADJ_T and WAB will not be used. The analysis will be done on W =Wins and stats associated with defense and offense.

The output files will have offensive and defensive averages for all high seed and 16 seeds from 2013 to 2020. The files will compare the averages with George Washington statistics in 2021. The report will also include a few visualizations to try and discover what statistic has a correlation with making the tournament and being a top seed in the NCAA tournament.

PART III: Overall description of the program

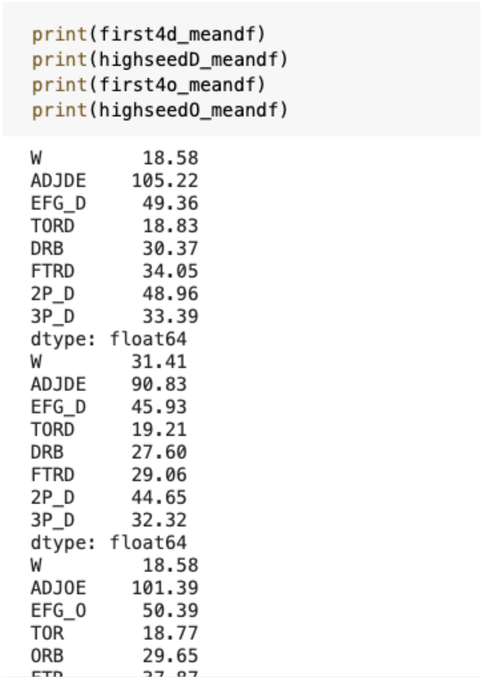
The program imports the required datasets. The focus will be on teams that made the tournament thus the 2020 dataset will not be used because not tournament was held in 2020 and the selection process was cancelled.

Cleaning: dropna() was used to get rid of any teams that did not make the tournament. NaN only existed in the SEED variable for teams that were not seeded. If teams were not seeded then they did not make the tournament and they were dropped from the data set using dropna.

Rafael Hernandez
George Washington Consulting Report
June 11, 2021

Creation and Merging of Datasets: The program merges all of the csv files into a master dataframe using pd.concat(). A data frame is also created for George Washington University(gwclientdf) because George Washington will be dropped from the 2021 data set due to not making the tournament. The program rounds the data to two decimals using .round(decimals=2).

The master data frame is broken up into sub data frames using groupby. It is broken up into a data frame for all the 16 seeds from 2013 to 2021 and for all the one seeds from 2013 to 2021. These data frames are then broken up into offensive and defensive data frames for 16 seeds and 1 seeds. Offensive and Defensive columns extracted to the corresponding data frame and the mean is calculated by using .mean(). Data frames are created for the mean outcomes. An example screenshot is provided below.



```
print(first4d_meandf)
print(highseedD_meandf)
print(first4o_meandf)
print(highseedO_meandf)
```

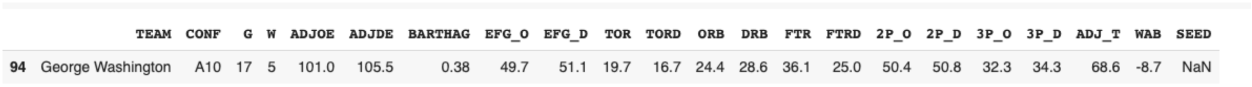
```
W          18.58
ADJDE     105.22
EFG_D      49.36
TORD       18.83
DRB        30.37
FTRD       34.05
2P_D       48.96
3P_D       33.39
dtype: float64
W         31.41
ADJDE     90.83
EFG_D     45.93
TORD      19.21
DRB       27.60
FTRD      29.06
2P_D      44.65
3P_D      32.32
dtype: float64
W          18.58
ADJOE     101.39
EFG_O      50.39
TOR        18.77
ORB        29.65
FTR        37.87
```

The program creates four output files for the offensive and defensive mean data frames. Indexing is used to give values to the means in order to write output files that will compare George Washington 2021 stats with the offensive and defensive averages for 16 seed teams from 2013 to 2021.

The last part of the program tinkers with visualization to try and find a statistic that has a direct correlation to teams making the tournament and being a high seed. Plot, scatter matrix and density visualizations are used to explore the statistics. Matplotlib, pandas.plotting and plotly libraries are used to create the visualizations.

Rafael Hernandez
George Washington Consulting Report
June 11, 2021

PART IV: Conclusions

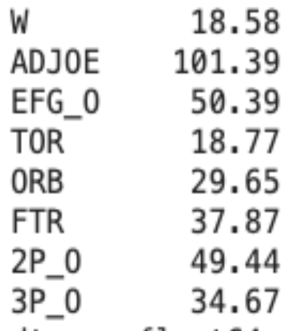


	TEAM	CONF	G	W	ADJOE	ADJDE	BARTHAG	EFG_O	EFG_D	TOR	TORD	ORB	DRB	FTR	FTRD	2P_O	2P_D	3P_O	3P_D	ADJ_T	WAB	SEED
94	George Washington	A10	17	5	101.0	105.5	0.38	49.7	51.1	19.7	16.7	24.4	28.6	36.1	25.0	50.4	50.8	32.3	34.3	68.6	-8.7	NaN

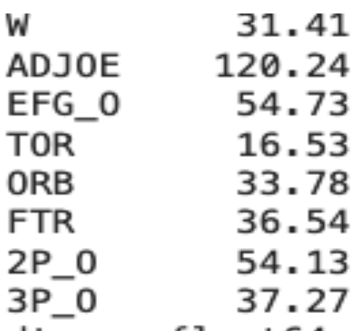
George Washington University had a lackluster 2021 season. The season consisted of 17 games. George Washington won five games out of those 17 games. George Washington wants to make the tournament in 2022.

Questions 1: What are the average defensive and offensive stats for the lowest seeded teams from 2013 to 2021?
Questions 2: What is the average defensive and offensive stats for the highest seeded teams from 2013 to 2021

Offensive Means: **Low Seeds.** Offensive Means: **High Seeds**



```
W          18.58
ADJOE     101.39
EFG_O      50.39
TOR        18.77
ORB        29.65
FTR        37.87
2P_O       49.44
3P_O       34.67
```



```
W          31.41
ADJOE     120.24
EFG_O      54.73
TOR        16.53
ORB        33.78
FTR        36.54
2P_O       54.13
3P_O       37.27
```

Above are the mean offensive stats for all 16 seeds(left) and one seeds(right) from 2013 to 2021. One seeds outperform 16 seeds in all categories except FTR(Free throw rate). 16 seeds average one free throw more than one seeds. One seeds have better shooting percentages than 16 seeds from the three point line and from inside the three point line.

Before we continue let's elaborate on what Adjusted offensive efficiency(ADJOE) and Adjusted defensive efficiency(ADJDE). ADJOE and ADJDE calculate point per possessions and how a team defends points per possession. Offensively, anything above 1 point per possession is considered a good offense. Defensively, anything below 1 point per possession is considered a good defense. 16 seeds average 1.01 point per possession compared to 1.20 for one seeds. Ken Pomeroy multiplies points per possession times 100 to give us the average points per possession per 100 possessions. A more elaborate explanation can be found at https://www.sportsrec.com/calculate-teams-offensive-defensive-efficiencies-7775395.html and it is quoted below:

Rafael Hernandez
George Washington Consulting Report
June 11, 2021

Offensive Efficiency

Calculate the number of total number of possessions for your team using the formula: field goals attempted - offensive rebounds + turnovers + (0.4 x free throws attempted) = total number of possessions for the season. This works because a possession can end only in one of three ways: an attempted field goal, a turnover or a free throw, with an offensive rebound negating additional field goal attempts.

Divide the team's total points scored for the season by the possessions you calculated in Step 1. For example, 938 total points scored divided by 998 total possessions gives your team 0.94 points scored per possession. Numbers above 1.0 are generally considered good.

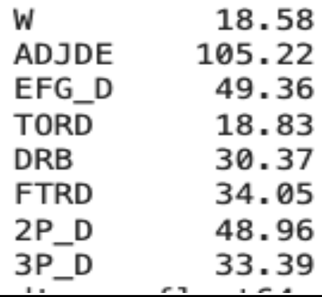
Convert the offensive PPP number to an efficiency rating by simply multipling by 100. So 0.94 points scored per possession becomes an offensive efficiency rating of 94.

Defensive Efficiency

Use the formula field goals attempted - offensive rebounds + turnovers + (0.4 x free throws attempted) = total number of possessions for the season to calculate total team possessions.

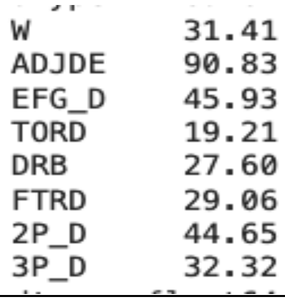
Divide the total number of points allowed by your team by the possession total you calculated in Step 1. For example, 1009 total points allowed divided by 998 total possessions gives your team 1.01 points allowed per possession. The opposite is true for defensive PPP: Above 1.0 is bad; below 1.0 is considered good.

Defensive Averages: **Low Seeds**



```
W          18.58
ADJDE     105.22
EFG_D      49.36
TORD       18.83
DRB        30.37
FTRD       34.05
2P_D       48.96
3P_D       33.39
```

Defensive Averages: **High Seeds**

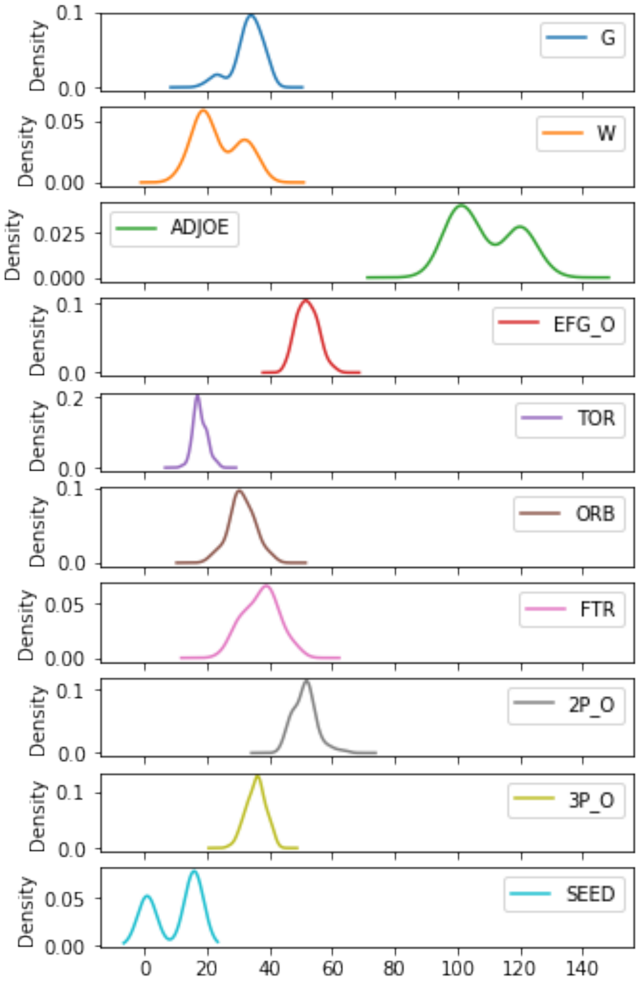


```
W          31.41
ADJDE      90.83
EFG_D      45.93
TORD       19.21
DRB        27.60
FTRD       29.06
2P_D       44.65
3P_D       32.32
```

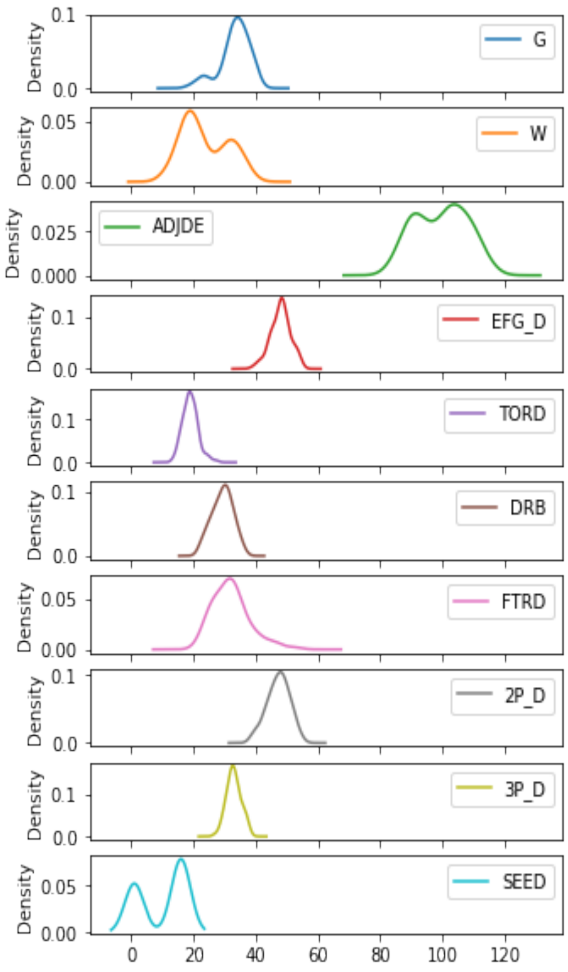
The means point to high seeds having better defensive statistics. One seeds cause more turnovers, force teams to shoot lower percentage from behind and inside the three point line. When we looked at ADJOE in the offensive stats, both high and low seeds had good offenses but on the defensive side 16 seeds average over 1.0 point per possession indicating that 16 seeds on average have bad defenses.

Question 3: Out of all the variables is there a variable(s) that can predict who is a 16 seed and who is a 1 seed?

Defensive Density

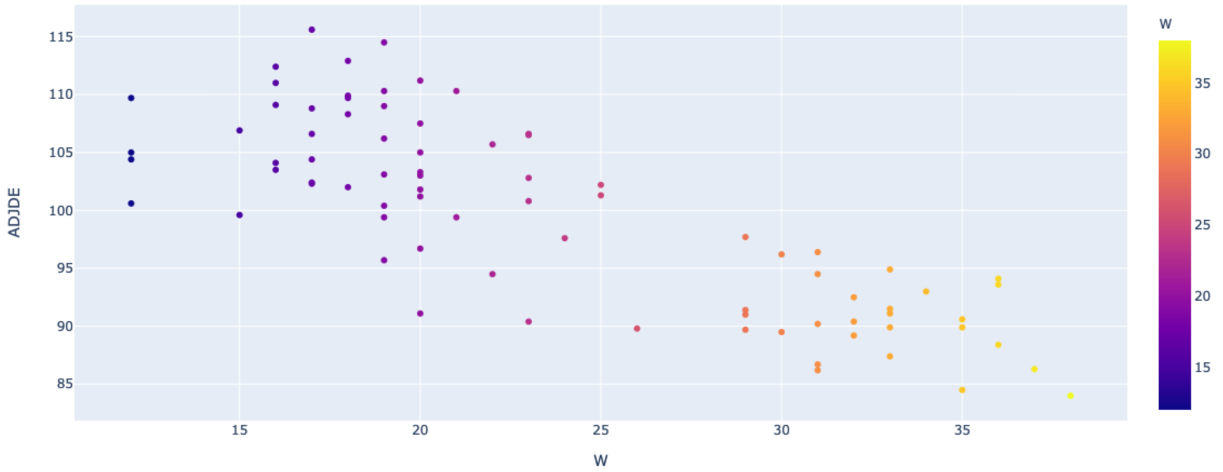


Offensive Density

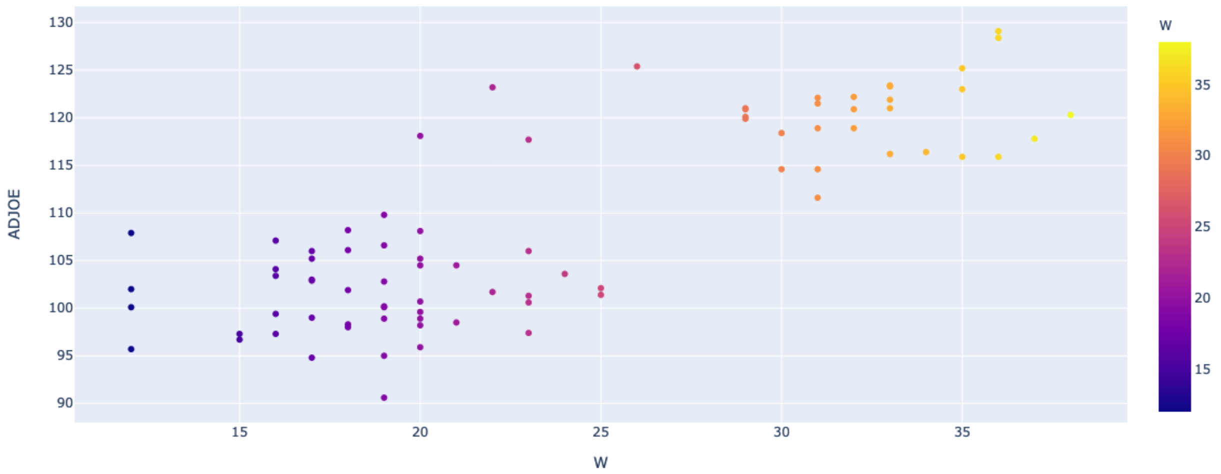


Rafael Hernandez
George Washington Consulting Report
June 11, 2021

The data frame was concatenated from the high and low seeds data frames and a density visualization was created using matplotlib out of the offensive and defensive stats to find if there is a statistical value that can be used to predict who is a 16 seed and a one seed. “Predict” is used in a general sense and not to be construed as a machine learning outcome because this is not a machine learning predictor. The visualizations indicate that there is less density in the ADJOE, ADJDE, Wins, FTR, FTRD and SEED values. Out of all of these categories, I focused on ADJOE, ADJDE and Wins because there is more variance and there can be a clearer correlation. The other values are too close together and the lack of variance will make it difficult to determine the direct point where there is separation between one seeds and 16 seeds.



In the visualization above we see the variance and correlation between Adjusted Defensive efficiency and Wins. The visualization is colored by wins. There is a large cluster of teams ranging from 15 to 20 wins and 100 to 115 ADJDE. At the other end of the spectrum, we see that the cluster takes place at 30 wins and beyond and an ADJDE of 95 or below.

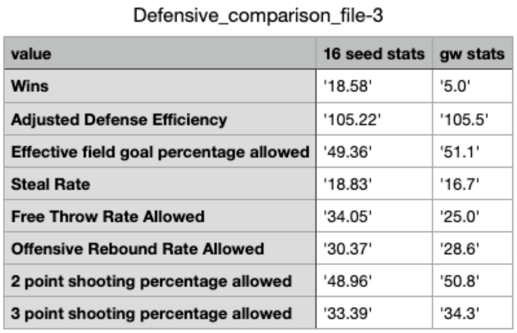


This visualization is similar to the previous visualization except it compares Wins(x-axis) with Adjusted Offensive Efficiency(y-axis). The visualization uses Wins to color the plots. The gap between one seeds and high seeds appears to be wider in this chart. We know from the tables that 16 seeds averaged above 1.0 point per possession, but the placement of all the offensive efficiency statistics paints a picture of truly elite offenses for top seeds and their correlation to wins.

The answer to question 3 is that ADJOE and ADJDE are the variables that should be used to determine wins which will in turn lead to a team having a high seed. From the data and the visuals, we can say that a team with an ADJOE of 120 and ADJDE of 95 will have around 30 wins. At the other end of the spectrum, we will not want to wager if a team with 105 ADJDE and 100 ADJOE will make it into the tournament. Whether that team gets into the tournament becomes subjective and is in the hands of the selective committee. A team will have to have over 20 wins and be closer to 25 wins to get into the tournament. The committee will take into consideration other factors such as strength of schedule and number of quality wins which are not part of this dataset, however we can determine who the elite teams are and begin to establish a baseline of what a team needs to have in order to get into the tournament. Out of all the variables ADJOE, ADJDE and Wins paint a clearer picture of which teams will make the tournament than any other variable.

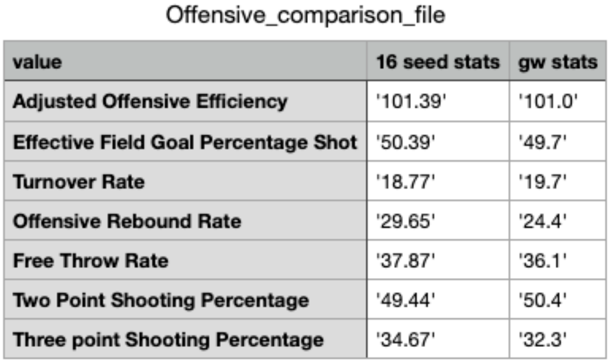
QUESTION 4: What areas does George Washington University need to improve on to make the NCAA tournament in 2022?

Rafael Hernandez
George Washington Consulting Report
June 11, 2021



Defensive_comparison_file-3

value	16 seed stats	gw stats
Wins	'18.58'	'5.0'
Adjusted Defense Efficiency	'105.22'	'105.5'
Effective field goal percentage allowed	'49.36'	'51.1'
Steal Rate	'18.83'	'16.7'
Free Throw Rate Allowed	'34.05'	'25.0'
Offensive Rebound Rate Allowed	'30.37'	'28.6'
2 point shooting percentage allowed	'48.96'	'50.8'
3 point shooting percentage allowed	'33.39'	'34.3'



Offensive_comparison_file

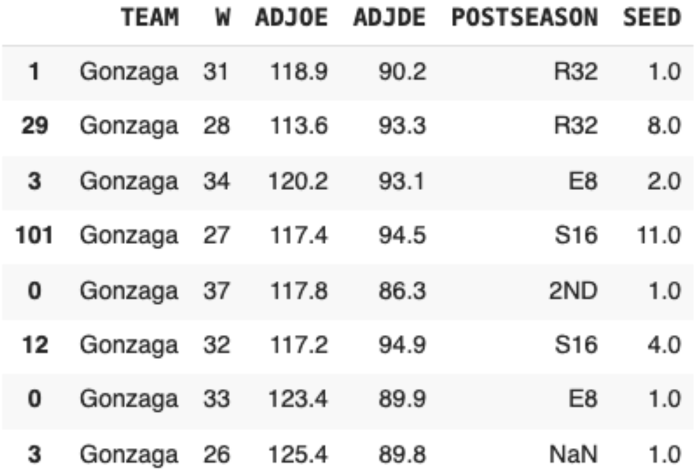
value	16 seed stats	gw stats
Adjusted Offensive Efficiency	'101.39'	'101.0'
Effective Field Goal Percentage Shot	'50.39'	'49.7'
Turnover Rate	'18.77'	'19.7'
Offensive Rebound Rate	'29.65'	'24.4'
Free Throw Rate	'37.87'	'36.1'
Two Point Shooting Percentage	'49.44'	'50.4'
Three point Shooting Percentage	'34.67'	'32.3'

The two charts above compare the defensive and offensive means of 16 seeds with George Washington's 2021 stats. George Washington's problem is not the offense. They have good offensive efficiency even though it's slightly below the mean of 16 seeds. They can decrease their turnover rate which will give the more offensive possessions that would lead to more points. George Washington shoots the ball well from two-point range and can slightly improve their three-point percentage but as it stands, George Washington runs a good offense.

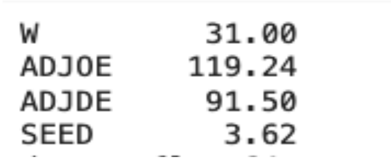
If George Washington wants to make the tournament in 2022, they have to improve on the defensive end. 16 seed teams on average have bad defense and George Washington's defense is worse than the 16 seed mean. George Washington's adjusted defensive efficiency is 105.5 compared to 105.22. George Washington's goal should not be to be at the mean for 16 seed defenses. George Washington has to be better, and they need to get closer to 100 than 105 or be slightly below 100. George Washington is bad at creating turnovers and their opponents have a high shooting percentage from two-point range and are above the mean on three-point shots. If you are not generating turnover and allowing teams to have high shooting percentages then you are not playing defense. Whatever George Washington's defensive philosophy is, it's not working.

The Story of Gonzaga:

Rafael Hernandez
George Washington Consulting Report
June 11, 2021



	TEAM	W	ADJOE	ADJDE	POSTSEASON	SEED
1	Gonzaga	31	118.9	90.2	R32	1.0
29	Gonzaga	28	113.6	93.3	R32	8.0
3	Gonzaga	34	120.2	93.1	E8	2.0
101	Gonzaga	27	117.4	94.5	S16	11.0
0	Gonzaga	37	117.8	86.3	2ND	1.0
12	Gonzaga	32	117.2	94.9	S16	4.0
0	Gonzaga	33	123.4	89.9	E8	1.0
3	Gonzaga	26	125.4	89.8	NaN	1.0



```
W          31.00
ADJOE     119.24
ADJDE      91.50
SEED        3.62
```

Once upon a time Gonzaga was a George Washington. Gonzaga was a mid-major team playing in a sub-par conference in the west coast. At one point the school was considering closing it's doors and shutting down. Gonzaga's basketball team found some sustainability and started making the NCAA tournament. The program has gradually grown and since 2013 the team has made it to the championship game twice (Lost to Baylor in 2021). Gonzaga's reputation is that of having elite offenses. The charts above support that perception with a mean ADJOE of 119 and an elite ADJOE of 125.4 in 2021. Gonzaga has become one of the top programs in the country because they have good to very good defenses(ADJDE mean of 91.50). Gonzaga is not just an elite offensive team but there is also a commitment to defense. Since 2013 Gonzaga has made it to the NCAA championship game and in both of those seasons, they had elite defenses with ADJDE of 86.3 and 89.8 in 2021.

George Washington aspires to make the NCAA tournament. George Washington should not get caught up in the number of wins they had in 2021. George Washington should look at its adjusted defensive efficiency and seek to improve in that category. George Washington has a bad defense, and it should follow Gonzaga's model and focus on defense. Gonzaga's defensive commitment is making Gonzaga a blue blood in college basketball. George Washington has a good offense and is committed to offense but the area where George Washington has to focus is defense and it's adjusted defensive efficiency.

LESSONS LEARNED

Rafael Hernandez
George Washington Consulting Report
June 11, 2021

I have coached basketball since I was 18. Prior to this program and working with this data set, my perception of basketball statistics or my prism of viewing statistics was a sole indicator of performance with no correlation to each other but solely to wins and losses. After games as a staff, we would look at the stats sheet and compare the other stats with ours. We would conclude that the reason we lost was because we had too many turnovers or the other team out rebounded us. What I learned from this project is that offensive and defensive statistics are what make up offensive and defensive efficiency. That is, offensive and defensive efficiency are the cake and statistics like offensive and defense rebound, turnovers and shooting percentage are the ingredients needed to make the cake.

I always taught basketball from a defensive point of view and in my 15 years of coaching, I have never used defensive efficiency as a matrix for defensive performance. I have heard of points per possessions, but it was always a point of discussion for offense not defense. I have been fortunate to have really good mentors and I'm a really good coach. My teams have won sectional and regional championships. We have been state runner-up's in the highest division in California and I wish I had come across this data set and acquired these skills early in my career.

Analytics has impacted basketball and the way that it is played today but by looking at the dataset it is amazing to see that what separates the elite teams from the rest is defensive efficiency. In this project George Washington's statistics are at the mean across the board in offense and defense to all of the 16 seeds that have made the tournament. It is clear as day that if you want to win you have to focus more on defense and yet the data indicates that teams spend more time on offense which leads to a good offensive efficiency and bad defensive efficiency. If everyone is doing the same thing, it only makes sense that to differentiate yourself and to separate from the pack you should focus on what everyone else is not focusing on and that's defense.

CODE:

```
# -*- coding: utf-8 -*-
"""Rafael_Hernandez Final Project.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1zf0SfKo7ulbXR8BnasS2iyFGah17qT46
"""
```

Rafael Hernandez
George Washington Consulting Report
June 11, 2021

```
from google.colab import drive
drive.mount("/content/gdrive")

import pandas as pd
#importing csv files for seasons 13 to 15 and giving them values according to season

cbb13df = pd.read_csv('/content/gdrive/My Drive/cbb13.csv')
cbb14df = pd.read_csv('/content/gdrive/My Drive/cbb14.csv')
cbb15df = pd.read_csv('/content/gdrive/My Drive/cbb15.csv')

#viewing season 13
cbb13df

#Dropping rows with NaN because we only want data from teams that made ncaa tournament.
#Teams that did not make tournamen will have NaN in poste season and seed columns.
#Data frame cut down to 67 teams.
cbb13df = cbb13df.dropna()

cbb13df

#view info on 2013 season
cbb13df.info()

#dropping NaN's
cbb14df = cbb14df.dropna()

cbb14df

#dropping NaN
cbb15df = cbb15df.dropna()

cbb15df

#importing data frame for ncaa seasons 2016-2018
cbb16df = pd.read_csv('/content/gdrive/My Drive/cbb16.csv')
cbb17df = pd.read_csv('/content/gdrive/My Drive/cbb17.csv')
cbb18df = pd.read_csv('/content/gdrive/My Drive/cbb18.csv')

#Dropping NaN for 16, 17 and 18df
cbb16df = cbb16df.dropna()
cbb17df = cbb17df.dropna()
```

Rafael Hernandez
George Washington Consulting Report
June 11, 2021

```
cbb18df = cbb18df.dropna()

cbb16df

#view cbb17df
cbb17df

#view cbb18df
cbb18df

#importing season 2019 - 2021 and cbb.csv
cbb19df = pd.read_csv('/content/gdrive/My Drive/cbb19.csv')
cbb20df = pd.read_csv('/content/gdrive/My Drive/cbb20.csv')
cbb21df = pd.read_csv('/content/gdrive/My Drive/cbb21.csv')
cbbdf = pd.read_csv('/content/gdrive/My Drive/cbb.csv')

#Identifying the stats for client George Washington
#ranked 95 in 2021 and did not make the tournamnet
gwclientdf = (cbb21df.iloc[[94]])

#view gwclientdf
gwclientdf

#rounding stats
gwclientdf = gwclientdf.round(decimals=2)
gwclientdf

#dropping NaN for 2019 and 2021.
#Can't erase NaN for 2020 because there was no postseason due to covid-19.
#Don't know which teams would have made the tournament.
cbb19df = cbb19df.dropna()
cbb21df = cbb21df.dropna()

cbb19df

#Not cleaning up cbb20df due to no postseason play because of covid pandemic.
cbb20df

#view cbb21df.  #File has not updated end of season results.
#Final four teams were UCLA, Gonzaga, Baylor and Houston.
#Baylor won the National Championship.
cbb21df
```

Rafael Hernandez
George Washington Consulting Report
June 11, 2021

```
#not using cbbdf
cbbdf

#concactenating dataframes into one from all other seasons
frames = [cbb13df, cbb14df, cbb15df, cbb16df, cbb17df, cbb18df, cbb19df, cbb21df]

result = pd.concat(frames)
display(result)

#view result
result.info()

#create new dataframe of all 16 seeds
first4 = result.groupby(result.SEED)
first4df = first4.get_group(16)
first4df

#view first4df
first4df.info()

#first4df has 48 entries
#there are NaN in the postseason column for teams in the 2021 dataframe because the
dataframe was produced without final tournament outcomes.

#creating dataframe for all the 1 seeds
highseed = result.groupby(result.SEED)
highseeddf = highseed.get_group(1)

highseeddf

#create df for offense and defense to do statistical analysis for 16 seeds and 1 seeds.
first4defensedf = first4df[['TEAM', 'CONF', 'G', 'W', 'ADJDE', 'EFG_D', 'TORD', 'DRB', 'FTRD',
'2P_D', '3P_D', 'POSTSEASON', 'SEED']]
first4defensedf

#defense statistics for high seeds
highseedDefensedf = highseeddf[['TEAM', 'CONF', 'G', 'W', 'ADJDE', 'EFG_D', 'TORD', 'DRB',
'FTRD', '2P_D', '3P_D', 'POSTSEASON', 'SEED']]
highseedDefensedf

#Offense statistics for 16 seeds
```

Rafael Hernandez
George Washington Consulting Report
June 11, 2021

```
first4offensedf = first4df[['TEAM', 'CONF', 'G', 'W', 'ADJOE', 'EFG_O', 'TOR', 'ORB', 'FTR', '2P_O',
'3P_O', 'POSTSEASON', 'SEED']]
first4offensedf

#offense statistics for high seeds
highseedOffensedf = highseeddf[['TEAM', 'CONF', 'G', 'W', 'ADJOE', 'EFG_O', 'TOR', 'ORB', 'FTR',
'2P_O', '3P_O', 'POSTSEASON', 'SEED']]
highseedOffensedf

#defensive means
first4d_meandf = first4defensedf[['W', 'ADJDE', 'EFG_D', 'TORD', 'DRB', 'FTRD', '2P_D',
'3P_D',]].mean()
highseedD_meandf = highseedDefensedf[['W', 'ADJDE', 'EFG_D', 'TORD', 'DRB', 'FTRD', '2P_D',
'3P_D',]].mean()

print(round(first4d_meandf,2))
print(round(highseedD_meandf,2))

#offensive means
first4o_meandf = first4offensedf[['W', 'ADJOE', 'EFG_O', 'TOR', 'ORB', 'FTR', '2P_O',
'3P_O']].mean()
highseedO_meandf = highseedOffensedf[['W', 'ADJOE', 'EFG_O', 'TOR', 'ORB', 'FTR', '2P_O',
'3P_O']].mean()

print(round(first4o_meandf,2))
print(round(highseedO_meandf,2))

#testing index outcome
display(first4d_meandf.iloc[1])

#rounding decimal places on dataframes
first4d_meandf = first4d_meandf.round(decimals=2)
highseedD_meandf = highseedD_meandf.round(decimals=2)
first4o_meandf = first4o_meandf.round(decimals=2)
highseedO_meandf = highseedO_meandf.round(decimals=2)

print(first4d_meandf)
print(highseedD_meandf)
print(first4o_meandf)
print(highseedO_meandf)

print(first4d_meandf)
print(first4o_meandf)

#merging defensive stats for both high seeds and 16 seeds
dframes = [first4d_meandf, highseedD_meandf]

defensivedf = pd.concat(dframes)

defensivedf

#merging offensive stats for both high seeds and 16 seeds
oframes = [first4o_meandf, highseedO_meandf]

offensivedf = pd.concat(oframes)

offensivedf

#creating csv files comparing offensive and defensive means
defensivedf.to_csv('Comparing_defensive_means.csv')
offensivedf.to_csv('Comparing_offensive_means.csv')

#creating csv files for mean dataframesf
first4d_meandf.to_csv('Average_defensive_stats_for_16_seeds.csv')
first4o_meandf.to_csv('Average_offensive_stats_for_16_seeds.csv')
highseedD_meandf.to_csv('Average_defensive_stats_for_1_seeds.csv')
highseedO_meandf.to_csv('Average_offensive_stats_for_1_seeds.csv')

#creating values through indexing for George Washington stats
gwwins = gwclientdf["W"].mean()
gwadjoe = gwclientdf["ADJOE"].mean()
gwadjde = gwclientdf["ADJDE"].mean()
gwefgo = gwclientdf["EFG_O"].mean()
gwefgd = gwclientdf["EFG_D"].mean()
gwtor = gwclientdf["TOR"].mean()
gwtord = gwclientdf["TORD"].mean()
gworb = gwclientdf["ORB"].mean()
gwdrb = gwclientdf["DRB"].mean()
gwftr = gwclientdf["FTR"].mean()
gwftrd = gwclientdf["FTRD"].mean()
gw2po = gwclientdf["2P_O"].mean()
gw2pd = gwclientdf["2P_D"].mean()
gw3po = gwclientdf["3P_O"].mean()
gw3pd = gwclientdf["3P_D"].mean()
```

Rafael Hernandez
George Washington Consulting Report
June 11, 2021

```
#testing value outcome
gwwins

#creating defensive values through indexing for first 4 defensive means
first4wins = first4d_meandf.iloc[0]
first4adjde = first4d_meandf.iloc[1]
first4efgd = first4d_meandf.iloc[2]
first4tord = first4d_meandf.iloc[3]
first4drb = first4d_meandf.iloc[4]
first4ftrd = first4d_meandf.iloc[5]
first42pd = first4d_meandf.iloc[6]
first43pd = first4d_meandf.iloc[7]

#creating offensive values through indexing for first 4 offensive means.  Will exclude wins
first4adjoe = first4o_meandf.iloc[1]
first4efgo = first4o_meandf.iloc[2]
first4tor = first4o_meandf.iloc[3]
first4orb = first4o_meandf.iloc[4]
first4ftr = first4o_meandf.iloc[5]
first42po = first4o_meandf.iloc[6]
first43po = first4o_meandf.iloc[7]

first43po

#write out report csv file comparing George Washington defensive stats with 16 seed mean
stats
import csv
#label report file
with open('Defensive_comparison_file.csv', mode='w')as csv_file:
  #define fieldnames
  fieldnames=['value','16 seed stats', 'gw stats']
  writer=csv.DictWriter(csv_file, fieldnames=fieldnames)

  #create header
  writer.writeheader()
  #create rows for comparison
  writer.writerow({'value':'Wins', '16 seed stats':'\'' + str(first4wins) + '\'', 'gw stats':'\'' +
str(round(gwwins,2)) + '\''})
  writer.writerow({'value':'Adjusted Defense Efficiency', '16 seed stats':'\'' +str(first4adjde) + '\'',
'gw stats':'\'' +str(round(gwadjde,2)) + '\''})
```

```
  writer.writerow({'value':'Effective field goal percentage allowed', '16 seed stats':'\'' +
str(first4efgd) + '\'', 'gw stats':'\'' + str(round(gwefgd,2)) + '\''})
  writer.writerow({'value':'Steal Rate', '16 seed stats':'\'' + str(first4tord) + '\'', 'gw stats':'\''
+str(round(gwtord,2)) + '\''})
  writer.writerow({'value':'Free Throw Rate Allowed', '16 seed stats':'\'' +str(first4ftrd) + '\'', 'gw
stats':'\'' + str(round(gwftrd,2)) + '\''})
  writer.writerow({'value':'Offensive Rebound Rate Allowed', '16 seed stats':'\'' +str(first4drb) +
'\'', 'gw stats':'\'' +str(round(gwdrb,2)) + '\''})
  writer.writerow({'value':'2 point shooting percentage allowed', '16 seed stats':'\''
+str(first42pd) + '\'', 'gw stats':'\'' + str(round(gw2pd,2)) + '\''})
  writer.writerow({'value':'3 point shooting percentage allowed', '16 seed stats':'\''
+str(first43pd) + '\'', 'gw stats':'\'' + str(round(gw3pd,2)) + '\''})

#write out report csv file comparing George Washing offensive stats with 16 seed mean stats
import csv#label report file
with open('Offensive_comparison_file.csv', mode='w')as csv_file:
  #define field names
  fieldnames1=['value', '16 seed stats', 'gw stats']
  writer=csv.DictWriter(csv_file, fieldnames=fieldnames1)

  #create header
  writer.writeheader()
  #create rows for comparison
  writer.writerow({'value':'Adjusted Offensive Efficiency', '16 seed stats':'\'' +str(first4adjoe) +
'\'', 'gw stats':'\'' + str(round(gwadjoe,2)) + '\''})
  writer.writerow({'value':'Effective Field Goal Percentage Shot', '16 seed stats':'\'' +
str(first4efgo) + '\'', 'gw stats':'\'' +str(round(gwefgo,2)) + '\''})
  writer.writerow({'value':'Turnover Rate','16 seed stats':'\'' + str(first4tor) + '\'', 'gw stats':'\'' +
str(round(gwtor,2)) + '\''})
  writer.writerow({'value':'Offensive Rebound Rate', '16 seed stats':'\'' + str(first4orb) +'\'', 'gw
stats':'\'' + str(round(gworb,2)) + '\''})
  writer.writerow({'value':'Free Throw Rate', '16 seed stats':'\'' + str(first4ftr) + '\'', 'gw stats':'\''
+ str(round(gwftr,2)) + '\''})
  writer.writerow({'value':'Two Point Shooting Percentage', '16 seed stats':'\'' + str(first42po) +
'\'', 'gw stats':'\'' + str(round(gw2po,2)) + '\''})
  writer.writerow({'value':'Three point Shooting Percentage', '16 seed stats':'\'' + str(first43po) +
'\'', 'gw stats':'\'' + str(round(gw3po,2)) + '\''})

#visualizations

import matplotlib
import matplotlib_inline
```

Rafael Hernandez
George Washington Consulting Report
June 11, 2021

```
#merging 1 seed and 16 seed dfs by offensive efficiency and defensive efficiency
frames2 = [first4defensedf, highseedDefensedf]

highlowd = pd.concat(frames2)
display(highlowd)

#merging offensive stats
frames3 = [first4offensedf, highseedOffensedf]

highlowo = pd.concat(frames3)
display(highlowo)

#plotting defense by defensive efficiency
#interesting but x-axis is sloppy
highlowd.plot(x="TEAM", y=["ADJDE","EFG_D","TORD","DRB"])

#comparing ADJDE with 2P and 3p shooting percentage allowed.
highlowd.plot(x="TEAM", y=["ADJDE", "2P_D", "3P_D"])

#testing outcome
#x-axis is sloppy.
highlowo.plot(x="TEAM", y=["ADJOE"])

from pandas.plotting import scatter_matrix

#scatter matrix to compare variables and see if there are correlations
#Kind of pretty but difficult to read.
scatter_matrix(highlowo, figsize=(20,20))

#visualiation of statistical density for defense
highlowd.plot.kde(subplots=True, figsize=(5,9))

#visuazliation for stistical density for offense
highlowo.plot.kde(subplots=True, figsize=(5,9))

import plotly
import plotly.express as px

#visualization testing correlation between wins and ADJDE
fig = px.scatter(highlowd, x="W", y="ADJDE", color='W')
fig.show()
```

Rafael Hernandez
George Washington Consulting Report
June 11, 2021

```
#visualization testing correlation between wins and 2 point percentage defense
fig1 = px.scatter(highlowd, x="W", y="2P_D", color='W')
fig1.show()

#visualization comparing correlation between wins and adjoe
fig3 = px.scatter(highlowo, x="W", y="ADJOE", color='W')
fig3.show()

#Extracting gonzaga from result dataframe.
gonzaga = result.groupby(result.TEAM)
gonzagadf = gonzaga.get_group("Gonzaga")
gonzagadf = gonzagadf[["TEAM", "W", "ADJOE", "ADJDE", "POSTSEASON", "SEED"]]

gonzagadf

#using mean function and round function to create a meandf for gonzaga
gonzaga_meandf = gonzagadf.mean()
round(gonzaga_meandf,2)
```