Effectiveness of Tanking in the NBA: A 30-Year Study

Michael O’Donnell

City University of New York, School of Professional Studies

DATA621, Business Analytics and Data Mining

December 13, 2020

# Abstract

In 1966, the National Basketball Association (NBA) installed a reverse order draft; the order of draft picks was the reverse of teams’ regular season records. Inadvertently, this created an incentive for teams to lose (Becker and Huselid, 1992). In many cases, the incentive to lose was exposed by midseason teams without playoff potential. As an early example, the 1983-84 Houston Rockets decided to play more bench players after a disappointing 20-26 start to the season (Hallisey, 2016). They ended that season at 29-53 and drafted Hakeem Olajuwon 1st overall the next year.

Since 1966, the NBA draft was reformed but kept a structure that rewarded losing teams with higher draft picks. Thus, the term “tanking” was coined for teams that aimed for losing (Paxton, 2019). This paper acknowledges tanking exists, but it challenges its effectiveness. More specifically, it seeks to determine if tanking helps teams reach the NBA Finals.

# Problem Statement

Tanking in the NBA is prevalent. It is even sometimes transparent, like the 76ers “Trust the Process” years under General Manager Sam Hinkie (Choi, 2019). It is clear tanking guarantees better draft picks. But does it guarantee success?

To examine the relationship between tanking and success, this paper will analyze 30 consecutive NBA seasons from 1990-2020. In this analysis, success is defined as an NBA Finals appearance and tanking is defined as multiple losing seasons. The “length of a tanking” is defined as the number of consecutive losing seasons and “years since tanking” is defined as the number of seasons since a losing season. Overall, this paper will answer:

Does tanking help NBA Teams reach the Finals?

# Literature Review

## Incentive to Lose

Many different models are used in sports to determine success (Becker and Huselid, 1922). Some models reward only winning, while others reward winning and losing. In the NBA, both winning and losing are rewarded (Taylor and Trogdon, 2002); winning is rewarded with championship titles and losing is rewarded with better draft picks. Thus, NBA teams have both incentives to win or lose, but no incentive for a 50% winning percentage (Thomas, 2020).

## History of Tanking in the NBA

In 1966, the NBA installed a reverse order draft; the order of draft picks was the reverse of teams’ regular season records. Inadvertently, this created an incentive for teams to lose (Becker and Huselid, 1992). In many cases, the incentive to lose was exposed by midseason teams without playoff potential. As an early example, the 1983-84 Houston Rockets decided to play more bench players after a disappointing 20-26 start to the season (Hallisey, 2016). They ended that season at 29-53 and drafted Hakeem Olajuwon 1st overall the next year.

Since 1966, the NBA draft was reformed but kept a structure that rewarded losing teams with higher draft picks. Thus, the term “tanking” was coined for teams that aimed for losing (Paxton, 2019)

# Methodology

As stated above, this project’s research focused on answering the following question:

Does tanking help NBA Teams reach the Finals?

To answer this question, a five-step methodology was used:

1. **Data Acquisition**: The required data to answer this research question was NBA team regular season results and NBA team playoff results. Thus, all data was acquired from the NBA statistics website, basketball-reference.com. Since the data lived on multiple pages, a python web scraper was built to acquire the required 30 years of NBA data into one dataset.
2. **Data Preparation**: The data scraped from basketball-reference.com contained all NBA team regular season results and playoff results. But, the data did not contain information about tanking. To add this data, a python script was built to determine the “years since tanking” and “length of last tank” for each row in the dataset.
3. **Data Exploration**: Each variable in the complete NBA dataset was explored for data type, correlation, and distribution. Then, the three most important variables were closely explored: consecutive years in playoffs, years since tanking, and length of tank.
4. **Binary Logistic Regression Model**: After the data was prepared and explored, a Binary Logistic Regression model was created in R with a binary response variable, “NBA Finals Appearance”.
5. **Odds Ratio and Standardized Regression Coefficients**: After the Binary Logistic Regression Model was setup, the Odds Ratio and Standardized Regression Coefficient of each predictor variable were calculated in R to determine the most important predictors for an NBA Finals Appearance.

# Experimentation and Results

Question: Does tanking help NBA Teams reach the Finals?

## Data Acquisition

To acquire 30 seasons of NBA team data from basketball-reference.com, a python web scraper was built with the beautifulsoup library. The web scraper went to 30 web pages, which each contained 1 year of NBA Standings, and grabbed the NBA Teams’ regular season records. Then, all 30 years of standings were combined into one pandas dataframe and exported as a CSV. The details of the acquired data in the dataframe are below:

Table 1  
*NBA Seasons Collected by Team*

|  |  |  |
| --- | --- | --- |
| NBA Team | Seasons | Finals Appearances |
| Atlanta Hawks | 30 | 0 |
| Boston Celtics | 30 | 2 |
| Brooklyn Nets (NJ Nets) | 30 | 2 |
| Charlotte Bobcats | 10 | 0 |
| Charlotte Hornets | 20 | 0 |
| Chicago Bulls | 30 | 6 |
| Cleveland Cavaliers | 30 | 5 |
| Dallas Mavericks | 30 | 2 |
| Denver Nuggets | 30 | 0 |
| Detriot Pistons | 30 | 3 |
| Golden State Warriors | 30 | 5 |
| Houston Rockets | 30 | 2 |
| Indiana Pacers | 30 | 1 |
| Los Angeles Clippers | 30 | 0 |
| Los Angeles Lakers | 30 | 9 |
| Memphis Grizzlies | 19 | 0 |
| Miami Heat | 30 | 6 |
| Milwaukee Bucks | 30 | 0 |
| New Orleans Pelicans | 7 | 0 |
| New York Knicks | 30 | 2 |
| Oklahoma City Thunder (Seattle) | 30 | 2 |
| Orlando Magic | 30 | 2 |
| Philidelphia 76ers | 30 | 1 |
| Phoenix Suns | 30 | 1 |
| Portland Trail Blazers | 30 | 2 |
| Sacremento Kings | 30 | 0 |
| San Antonio Spurs | 30 | 6 |
| Toronto Raptors | 25 | 1 |
| Utah Jazz | 30 | 2 |
| Vancouver Grizzlies | 6 | 0 |
| Washington Bullets | 8 | 0 |
| Washington Wizards | 22 | 0 |

## Data Preparation

To isolate tanking as a predictor, two variables were created from the acquired data: “years since tanking” and “length of tanking”. With these variables, the effect of tanking on NBA Finals appearance could be measured. To create these variables, a python script went through the dataset with a loop and calculated the “years since tanking” and “length of tanking” for each row, which was 1 seasons for 1 team. An example of the results are shown below:

Table 2  
*Example of Dataset with Tanking Predictors*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Year | NBA Team | Finals Appearance | Consecutive Playoffs | | Years Since Tanking | Length of Tanking |
| 2020 | Atlanta Hawks | N | 0 | 0 | | 3 |
| 2020 | Boston Celtics | N | 6 | 6 | | 2 |
| 2020 | Brooklyn Nets | N | 2 | 0 | | 1 |
| 2020 | Charlotte Hornets | N | 0 | 0 | | 4 |
| 2020 | Chicago Bulls | N | 0 | 0 | | 3 |
| 2020 | Cleveland Cavaliers | N | 0 | 0 | | 2 |

## Data Exploration

Each variable in the complete NBA dataset was explored for data type, correlation, and distribution. Then, the three most important variables were closely explored: consecutive years in playoffs, years since tanking, and length of tank.

Figure 1  
*Histogram of Important Predictors*

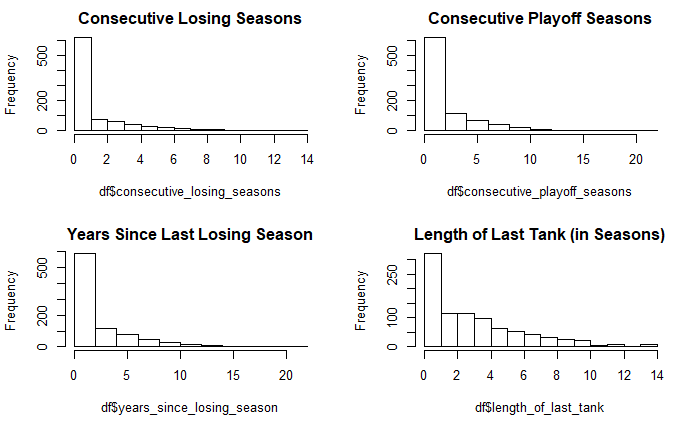
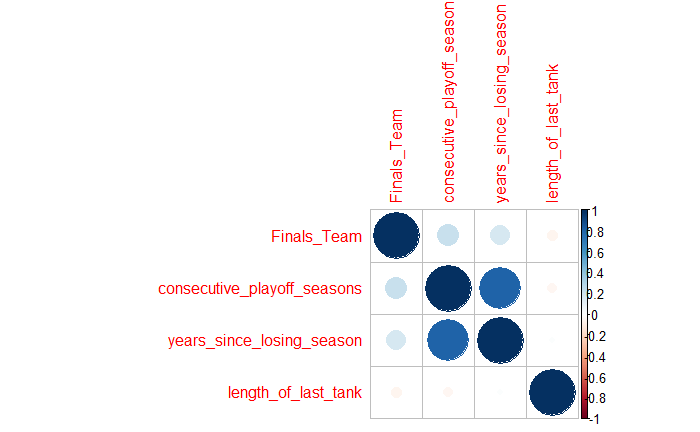


Figure 2  
*Correlation Plot of Important Predictors*



## Binary Logistic Regression Model

After Natural Language Processing was run on all 10,472 episode descriptions, the data was prepared to answer the research question: Which types of exercise are more popular in the US post-COVID-19?

To scientifically answer this question, a two-sample t-test was set up for each of the seven types of exercise included in the project: running, cycling, swimming, walking, weightlifting, Crossfit, and yoga.

## Odds Ratio and Standardized Regression Coefficients

After Natural Language Processing was run on all 10,472 episode descriptions, the data was prepared to answer the research question: Which types of exercise are more popular in the US post-COVID-19?

To scientifically answer this question, a two-sample t-test was set up for each of the seven types of exercise included in the project: running, cycling, swimming, walking, weightlifting, Crossfit, and yoga.

# Conclusions

This project began with a question: Which types of exercise are most popular in the US post-COVID-19? To measure exercise popularity, a connection was made between podcast episode content and relevancy to podcast listeners. Therefore, 11,447 podcast episode descriptions from 47 major US exercise-related podcasts were acquired.

During the data exploration phase, podcast episodes were divided between pre-COVID-19 and post-COVID-19 episodes. In total, there were 9,522 pre-COVID-19 episodes and 1,914 post-COVID-19 episodes.

During the data preparation phase, 975 podcast episodes were removed because their corresponding podcasts could not be compared between pre-COVID-19 and post-COVID-19.

After the data was explored and prepared, all podcast episode descriptions were run through a natural language processor in python to count the number of words related to each of the following seven types of exercise: running, cycling, swimming, walking, weightlifting, Crossfit, and yoga.

Using the natural language processing results, seven two-sample t-tests were run to compare the means of exercise-related terms used pre-COVID-19 and post-COVID-19 for each of the seven types of exercise. From the two-sample t-tests, it was clear that running and weightlifting became more popular post-COVID-19:

Table 7  
*Research Conclusions*

|  |  |
| --- | --- |
| Type of Exercise | Conclusion |
| Running | More popular |
| Cycling | Less popular |
| Swimming | Less popular |
| Walking | No change |
| Weightlifting | More popular |
| Crossfit | Less popular |
| Yoga | No change |

# References

Ahmad, T. (2020, April 5). Corona virus (COVID-19) pandemic and work from home: Challenges of cybercrimes and cybersecurity. [http://dx.doi.org/10.2139/ssrn.3568830](https://dx.doi.org/10.2139/ssrn.3568830)

Bird, S., Loper, E., & Klein, E. (2009). *Natural Language Processing with Python*. O’Reilly Media, Inc.

Bollen, J., & Mao, H. (2011). Twitter mood as a stock market predictor. *Computer,* *44*(10), 91–94. doi:https://doi.org/10.1016/j.jocs.2010.12.007

Chowell, G. (2020). The COVID-19 pandemic in the USA: What might we expect? *The Lancet,* *395*(10230), 1093–1094. doi:https://doi.org/10.1016/S0140-6736(20)30743-1

Chu, S., Chen, H., & Sung, Y. (2015). Following brands on Twitter: An extension of theory of planned behavior. *International Journal of Advertising,* *35*(3), 421­–437. doi:10.1080/02650487.2015.1037708

Czeisler, M. É., Tynan M. A., Howard, M.E. (2020). Public attitudes, behaviors, and beliefs related to COVID-19, stay-at-home orders, nonessential business closures, and public health guidance: United States, New York City, and Los Angeles. *Morbidity and Mortality Weekly Report* *2020*(69):751–758. doi: [http://dx.doi.org/10.15585/mmwr.mm6924e1external icon](http://dx.doi.org/10.15585/mmwr.mm6924e1)

Dias, M. (2020). Netflix: From Apollo 13 to the coronavirus pandemic. *Global Scientific Journal,* *8*(8): doi:10.11216/gsj.2020.08.42678

Durrani, M., Gotkin, K., & Laughlin, C. (2015). Serial, seriality, and the possibilities for the podcast format. *American Anthropologist*

Fletcher, G. F., Blair, S. N., Blumenthal, J., Caspersen, C., Chaitman, B., Epstein, S., . . . Pina, I. L. (1992). Statement on exercise. Benefits and recommendations for physical activity programs for all Americans. A statement for health professionals by the Committee on Exercise and Cardiac Rehabilitation of the Council on Clinical Cardiology, American Heart association. *Circulation,* *86*(1), 340–344. doi:10.1161/01.cir.86.1.340

Gentil, P., Ramirez-Campillo, R., & Souza, D. (2020). Resistance training in face of the coronavirus outbreak: Time to think outside the box. *Frontiers in Physiology*, *11*, 859. https://doi.org/10.3389/fphys.2020.00859

Gostin, L. O., & Wiley, L. F. (2020). Governmental public health powers during the COVID-19 pandemic. *Jama,* *323*(21), 2137–2138. doi:10.1001/jama.2020.5460Markman, K. M., & Sawyer, C. E. (2014). Why pod? Further explorations of the motivations for independent podcasting. *Journal of Radio & Audio Media,* *21*(1), 20–35. doi:10.1080/19376529.2014.891211

Mchugh, S. (2020). Wide angle: Podcasts: Radio reinvented. *The UNESCO Courier,* *2020*(1), 6–9. doi:10.18356/4c5699b4-en

Mcclung, S., & Johnson, K. (2010). Examining the motives of podcast users. *Journal of Radio & Audio Media,* *17*(1), 82–95. doi:10.1080/19376521003719391

Moreland, A., Herlihy, C., Tynan, M. A. 2020). Timing of state and territorial COVID-19 stay-at-home orders and changes in population movement: United States. *Morbidity and Mortality Weekly Report* *2020*(69):1198–1203. doi: [http://dx.doi.org/10.15585/mmwr.mm6935a2external icon](http://dx.doi.org/10.15585/mmwr.mm6935a2).

Ngoc, L. (2014). Behavior pattern of individual investors in stock market. *International Journal of Business and Management,* *9*(1). doi:10.5539/ijbm.v9n1p1

Roberts, J. D., & Tehrani, S. O. (2020). Environments, behaviors, and inequalities: Reflecting on the impacts of the influenza and coronavirus pandemics in the United States. *International Journal of Environmental Research and Public Health,* *17*(12). doi:https://doi.org/10.3390/ijerph17124484

Tobarra, L., Robles-Gómez, A., Ros, S., Hernández, R., & Caminero, A. C. (2014). Analyzing the students’ behavior and relevant topics in virtual learning communities. *Computers in Human Behavior,* *31*, 659–669. doi:10.1016/j.chb.2013.10.001

WHO. (2020, June 29). Listings of WHO's response to COVID-19. <https://www.who.int/news/item/29-06-2020-covidtimeline>

Zhang, H., Dantu, R., & Cangussu, J. W. (2011). Socioscope: Human relationship and behavior analysis in social networks. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans,* *41*(6), 1122–1143. doi:10.1109/tsmca.2011.2113335

# Appendix with Code

## Data Acquisition

1. # get podcast episodes from a show
2. **def** get\_all\_podcast\_episodes(self, showid, market = 'US'):
3. headers = self.get\_resource\_header()
5. podcast\_name = self.get\_podcast\_info\_by\_id(showid)['name']
6. num\_episodes = self.get\_podcast\_info\_by\_id(showid)['total\_episodes']
8. limit = 25
9. offset = 0
10. num\_runs = num\_episodes//limit
12. episode\_df = pd.DataFrame(columns = ['podcast', 'name','release\_date','duration\_min',
13. 'external\_urls','id', 'language',
14. 'release\_date\_precision', 'uri','description'])
16. **for** i **in** range(num\_runs):
18. endpoint = f"https://api.spotify.com/v1/shows/{showid}/episodes?offset={offset}&limit={limit}&market=US"
19. lookup\_url = f"{endpoint}"
21. r = requests.get(lookup\_url, headers = headers)
22. **if** r.status\_code **not** **in** range(200,299):
23. **return** "somethings wrong..."
25. raw\_json = r.json()


29. **for** i **in** range(limit):
30. # create a dict with the data
31. temp\_dict = {'podcast': podcast\_name,
32. 'name': raw\_json['items'][i]['name'],
33. 'release\_date': raw\_json['items'][i]['release\_date'],
34. 'duration\_min': round((raw\_json['items'][i]['duration\_ms'])/60000,2),
35. 'external\_urls': raw\_json['items'][i]['external\_urls'],
36. 'id': raw\_json['items'][i]['id'],
37. 'language': raw\_json['items'][i]['language'],
38. 'release\_date\_precision': raw\_json['items'][i]['release\_date\_precision'],
39. 'uri': raw\_json['items'][i]['uri'],
40. 'description': raw\_json['items'][i]['description']}
42. df = pd.DataFrame(temp\_dict, columns = ['podcast','name','release\_date','duration\_min',
43. 'external\_urls','id', 'language',
44. 'release\_date\_precision', 'uri','description'])
45. episode\_df = episode\_df.append(df)
47. offset = offset + limit
49. episode\_df = episode\_df.reset\_index(drop=True)
50. **return** episode\_df

## Data Exploration

1. # create histogram of episode duration
2. plt.figure(figsize=(8,4))
4. plt.hist(df['duration\_min'], bins = 20)
5. plt.xlabel("Duration (minutes)")
6. plt.ylabel("Number of Episodes")
7. plt.title("Distribution of Podcast Duration (minutes)")
8. plt.grid(True)
10. plt.show()
12. # create boxplot of episode duration by year
13. df[df['year\_number']>2014].boxplot(by='year\_number',
14. column=['duration\_min'],
15. grid=True,
16. figsize = (10,4))
17. plt.title("Duration of Episodes by Year")
18. plt.suptitle("")
19. plt.ylabel("Duration (minutes)")
20. plt.xlabel("Year")
21. plt.show()

## Data Preparation

1. # add year, month, and week numbers to dataframe
2. df['year\_number'] = pd.to\_datetime(df['release\_date']).dt.year
3. df['month\_number'] = pd.to\_datetime(df['release\_date']).dt.month
4. df['week\_number'] = pd.to\_datetime(df['release\_date']).dt.week
5. df['year\_week'] = df['year\_number'].map(str) + df['week\_number'].map(str)
6. df['year\_month'] = df['year\_number'].map(str) + df['month\_number'].map(str)
8. **for** i **in** df['podcast'].unique():
9. temp\_df = df[df['podcast']==i]
10. **print**(i, "|", round(len(temp\_df)/len(temp\_df['year\_month'].unique()),2))

## Natural Language Processing and Two-Sample T-Test

1. # function to setup hypothesis test for podcast descriptions pre vs post COVID
2. **def** podcast\_description\_hypothesis\_test(csv, list\_of\_words, alpha, title = "hypothesis test"):
4. # first, read in the csv
5. # 'data/final\_datasets/relevant\_episode\_data\_v2.csv'
6. episode\_desc\_df = read\_episode\_descriptions(csv)
7. preCOVID\_episode\_desc\_df = episode\_desc\_df[pd.to\_datetime(episode\_desc\_df['release\_date']) < '2020-03-13']
8. postCOVID\_episode\_desc\_df = episode\_desc\_df[pd.to\_datetime(episode\_desc\_df['release\_date']) >= '2020-03-13']
10. # format description column for tokenization
11. desc\_df = create\_episode\_df(episode\_desc\_df)
12. preCOVID\_df = create\_episode\_df(preCOVID\_episode\_desc\_df)
13. postCOVID\_df = create\_episode\_df(postCOVID\_episode\_desc\_df)
15. # tokenize all 3 dataframes
16. episode\_tokens = tokenize\_descriptions(desc\_df)
17. preCOVID\_tokens = tokenize\_descriptions(preCOVID\_df)
18. postCOVID\_tokens = tokenize\_descriptions(postCOVID\_df)
20. # create final dataframes for analyses
21. episode\_dataset = pd.DataFrame(episode\_tokens[1])
22. preCOVID\_dataset = pd.DataFrame(preCOVID\_tokens[1])
23. postCOVID\_dataset = pd.DataFrame(postCOVID\_tokens[1])
25. # add number of exercise terms to each row in above datasets
26. episodes\_terms = []
27. **for** i **in** range(len(episode\_dataset)):
28. episodes\_terms.append(count\_terms(episode\_dataset['description'][i], list\_of\_words))
29. episode\_dataset['number\_of\_words'] = episodes\_terms
31. preCOVID\_terms = []
32. **for** i **in** range(len(preCOVID\_dataset)):
33. preCOVID\_terms.append(count\_terms(preCOVID\_dataset['description'][i], list\_of\_words))
34. preCOVID\_dataset['number\_of\_words'] = preCOVID\_terms
36. postCOVID\_terms = []
37. **for** i **in** range(len(postCOVID\_dataset)):
38. postCOVID\_terms.append(count\_terms(postCOVID\_dataset['description'][i], list\_of\_words))
39. postCOVID\_dataset['number\_of\_words'] = postCOVID\_terms
41. # find mean, standard deviation, and count of pre-COVID-19 exercise terms
42. preCOVID\_mean = preCOVID\_dataset['number\_of\_words'].mean()
43. preCOVID\_sd = preCOVID\_dataset['number\_of\_words'].std()
44. preCOVID\_episodes = len(preCOVID\_dataset['number\_of\_words'])
46. # find mean, standard deviation, and count of post-COVID-19 exercise terms
47. postCOVID\_mean = postCOVID\_dataset['number\_of\_words'].mean()
48. postCOVID\_sd = postCOVID\_dataset['number\_of\_words'].std()
49. postCOVID\_episodes = len(postCOVID\_dataset['number\_of\_words'])
51. # use scipy to get test statistic and p-value
52. hyp\_test = stats.ttest\_ind(postCOVID\_dataset['number\_of\_words'],preCOVID\_dataset['number\_of\_words'], equal\_var=False)
53. t\_statistic = hyp\_test[0]
54. p\_value = hyp\_test[1]
56. # print findings
57. **print**("words tested:", list\_of\_words)
58. **print**("variance of postCOVID:", round(postCOVID\_dataset['number\_of\_words'].var(), 2))
59. **print**("variance of preCOVID:", round(preCOVID\_dataset['number\_of\_words'].var(), 2))
60. **print**("difference in means = ", round((postCOVID\_mean-preCOVID\_mean), 3))
61. **print**("t = ", round(t\_statistic, 3))
62. **print**("p = ", round(p\_value, 3))
63. **print**("alpha = ", str(alpha))
64. **if** stats.ttest\_ind(postCOVID\_dataset['number\_of\_words'],preCOVID\_dataset['number\_of\_words'], equal\_var=False)[1] < alpha:
65. **print**("Hypothesis result: REJECT the null hypothesis")
66. **else**:
67. **print**("Hypothesis result: cannot reject the null hypothesis")
69. # export dataset
70. postCOVID\_dataset['post\_COVID'] = 1
71. preCOVID\_dataset['post\_COVID'] = 0
72. df = preCOVID\_dataset
73. df = preCOVID\_dataset.append(postCOVID\_dataset)
74. df.to\_csv(f"{title}\_data.csv")