

# Project Proposal

Jack Kincannon

12/3/2020

## Blog Link

My blog is available at <https://jdkincan.netlify.app/>

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Warning: 1 parsing failure.
```

```
##      row col   expected      actual                                     file
## 898078 -- 21 columns 5 columns '/home/jdkincan/DASC1104_PROJ/jdk/data/blkjckhands.csv'
```

```
## Warning: 2 parsing failures.
```

```
##      row col   expected      actual                                     file
## 618445 date date like  2015-09    '/home/jdkincan/DASC1104_PROJ/jdk/data/all_stocks_5yr.csv'
## 618445 NA    7 columns  1 columns  '/home/jdkincan/DASC1104_PROJ/jdk/data/all_stocks_5yr.csv'
```

## Dataset Exploration

## Blackjack Analysis

```
glimpse(dat1)
```

```
## Observations: 898,078
```

```
## Variables: 21
```

## \$ X1	<dbl> 0, 1, 2, 3, 4, 5, 0, 1, 2, 3, 4, 5, 0, 1, 2, 3, 4, 5, 0, 1, 2, 3, 4, 5, 0, 1, 2,
## \$ PlayerNo	<chr> "Player1", "Player2", "Player3", "Player4", "Player5", "Player6", "Player1", "Pl
## \$ card1	<dbl> 7, 10, 9, 2, 10, 6, 10, 3, 10, 9, 10, 3, 10, 9, 5, 10, 2, 9, 10, 2, 3, 10, 10, 4
## \$ card2	<dbl> 10, 9, 8, 10, 2, 1, 4, 10, 2, 6, 4, 5, 2, 10, 1, 10, 9, 5, 4, 10, 10, 10, 10, 8,
## \$ card3	<dbl> 0, 0, 0, 0, 0, 8, 0, 0, 0, 0, 0, 2, 0, 0, 10, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
## \$ card4	<dbl> 0, 0, 0, 5, 5, 10, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 5, 5, 10, 0, 0, 10, 9, 0,
## \$ card5	<dbl> 0,
## \$ sumofcards	<dbl> 17, 19, 17, 17, 17, 25, 14, 13, 12, 15, 14, 14, 12, 19, 16, 20, 12, 14, 19, 17, 1
## \$ dealcard1	<dbl> 10, 10, 10, 10, 10, 10, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 10, 10, 10, 10, 10, 1
## \$ dealcard2	<dbl> 8, 8, 8, 8, 8, 8, 4, 4, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 8, 8, 8, 8, 8, 8, 7, 7, 7,
## \$ dealcard3	<dbl> 0, 0, 0, 0, 0, 0, 5, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 0, 0, 0, 0, 0, 0, 4, 4, 4,
## \$ dealcard4	<dbl> 0, 0, 0, 0, 0, 0, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 0, 0, 0, 0, 0,
## \$ dealcard5	<dbl> 0,
## \$ sumofdeal	<dbl> 18, 18, 18, 18, 18, 18, 23, 23, 23, 23, 23, 23, 26, 26, 26, 26, 26, 26, 18, 18,

```
## $ blkjck      <chr> "nowin", "nowin", "nowin", "nowin", "nowin", "nowin", "nowin", "nowin", "nowin",
## $ winloss     <chr> "Loss", "Win", "Loss", "Loss", "Loss", "Loss", "Win", "Win", "Win", "Win", "Win"
## $ plybustbeat <chr> "Beat", "Plwin", "Beat", "Beat", "Beat", "Bust", "DlBust", "DlBust", "DlBust", "B
## $ dl bustbeat <chr> "Dlwin", "Beat", "Dlwin", "Dlwin", "Dlwin", "PlBust", "Bust", "Bust", "Bust", "B
## $ plwinamt    <dbl> 0, 20, 0, 0, 0, 0, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 0, 0, 20,
## $ dlwinamt    <dbl> 10, 0, 10, 10, 10, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 10, 10, 0, 0, 10,
## $ ply2cardsum <dbl> 17, 19, 17, 12, 12, 7, 14, 13, 12, 15, 14, 8, 12, 19, 6, 20, 11, 14, 14, 12, 13,
```

For this project, I am examining a dataset that contains 90,000 blackjack hands with 6 players per hand. The data is stored in the blkjckhands.csv and it was downloaded from a repository on Kaggle that is linked below. The data consists of 898,078 observations of 21 variables. The variables X1 and PlayerNo describe the 6 players present in each hand, they are dbl numeric and a character string respectively. The variables: card1, card2, card3, card4, card5, dealcard1, dealcard2, dealcard3, dealcard4, and dealcard5 describe the numeric value of the card dealt. The number at the end of the variables corresponds to which card was dealt first in the sequence, if no card is dealt 0 takes the place of NA. Variables sumofcards and sumofdeal are numerics which total the value of the cards dealt to each player and the dealer, per hand, respectively. Blkjck is a simple character string with two unique values nowin and Win. Winloss describes each players outcome simplistically per hand. Character values Loss and Win are the two options. Variables plybustbeat and dl bustbeat describe the hand outcome in more detail with unique values Beat, Plwin, Bust, DlBust, and Push for plybustbeat with Dlwin, Beat, PlBust, Bust, and Push for dl bustbeat. Each of these variables are class character. Plwinamt and dlwinamt describe the hand outcome monetarily. These values are numeric and a base bet of \$10 is assumed for each player each hand. The final variable play2cardsum is interesting. It describes the starting position from which blackjack players make their decision. It represents the sum of card1 and card2 numerically.

Initial exploration and experience makes me look towards dealcard1 and ply2cardsum as two potential explanatory variables. This dataset is a wealth of information and I am sure that I will be able to analyze any angle of blackjack strategy and probability.

- Question 1: First, I want to see what the win probabilities are for a player based on the hand they are dealt, and see how these probabilities change through the course of a hand i.e when the dealers up card is factored in. For experimentation summary statistics will be calculated for critical values, which will then be visualized and compared using ggplot.
- Question 2: Second, based on these probabilities I intend to calculate an optimal strategy. To investigate this, tables that describe the probability of a win will be produced, visualizations will be produced that track the difference between an optimal and sub-optimal strategy.
- Question 3: Third, now that we know the best way to play the game, how can we maximize the expected value and profit of a given player? I will utilize various mathematical betting sequences such as the as the Fibonacci to determine the answer to this question. Visualizations will analyze how large of a difference a true understanding of a game makes in outcomes.

The overarching purpose of this analysis is to show how optimized strategy is essential to making the most out of any market or game. I was inspired by Micheal Lewis books Moneyball and The Undoing Project, which are based around this idea.

<https://www.kaggle.com/mojocolors/900000-hands-of-blackjack-results>

## S&P 500 Analysis

```
glimpse(dat2)
```

```
## Observations: 618,445
## Variables: 7
## $ date      <date> 2013-02-08, 2013-02-11, 2013-02-12, 2013-02-13, 2013-02-14, 2013-02-15, 2013-02-19, 2013-02-20, 2013-02-21, 2013-02-22, 2013-02-23, 2013-02-24, 2013-02-25, 2013-02-26, 2013-02-27, 2013-02-28, 2013-03-01, 2013-03-04, 2013-03-05, 2013-03-06, 2013-03-07, 2013-03-08, 2013-03-11, 2013-03-12, 2013-03-13, 2013-03-14, 2013-03-15, 2013-03-18, 2013-03-19, 2013-03-20, 2013-03-21, 2013-03-22, 2013-03-23, 2013-03-24, 2013-03-25, 2013-03-26, 2013-03-27, 2013-03-28, 2013-03-29, 2013-03-30, 2013-03-31, 2013-04-01, 2013-04-02, 2013-04-03, 2013-04-04, 2013-04-05, 2013-04-08, 2013-04-09, 2013-04-10, 2013-04-11, 2013-04-12, 2013-04-15, 2013-04-16, 2013-04-17, 2013-04-18, 2013-04-19, 2013-04-22, 2013-04-23, 2013-04-24, 2013-04-25, 2013-04-26, 2013-04-29, 2013-04-30, 2013-05-01, 2013-05-02, 2013-05-03, 2013-05-04, 2013-05-06, 2013-05-07, 2013-05-08, 2013-05-09, 2013-05-10, 2013-05-13, 2013-05-14, 2013-05-15, 2013-05-16, 2013-05-17, 2013-05-20, 2013-05-21, 2013-05-22, 2013-05-23, 2013-05-24, 2013-05-27, 2013-05-28, 2013-05-29, 2013-05-30, 2013-05-31, 2013-06-03, 2013-06-04, 2013-06-05, 2013-06-06, 2013-06-07, 2013-06-10, 2013-06-11, 2013-06-12, 2013-06-13, 2013-06-14, 2013-06-17, 2013-06-18, 2013-06-19, 2013-06-20, 2013-06-21, 2013-06-24, 2013-06-25, 2013-06-26, 2013-06-27, 2013-06-28, 2013-06-29, 2013-06-30, 2013-07-01, 2013-07-02, 2013-07-03, 2013-07-04, 2013-07-05, 2013-07-08, 2013-07-09, 2013-07-10, 2013-07-11, 2013-07-12, 2013-07-15, 2013-07-16, 2013-07-17, 2013-07-18, 2013-07-19, 2013-07-22, 2013-07-23, 2013-07-24, 2013-07-25, 2013-07-26, 2013-07-29, 2013-07-30, 2013-07-31, 2013-08-01, 2013-08-02, 2013-08-03, 2013-08-04, 2013-08-05, 2013-08-06, 2013-08-09, 2013-08-10, 2013-08-11, 2013-08-12, 2013-08-13, 2013-08-14, 2013-08-15, 2013-08-16, 2013-08-17, 2013-08-20, 2013-08-21, 2013-08-22, 2013-08-23, 2013-08-24, 2013-08-27, 2013-08-28, 2013-08-29, 2013-08-30, 2013-08-31, 2013-09-03, 2013-09-04, 2013-09-05, 2013-09-06, 2013-09-07, 2013-09-10, 2013-09-11, 2013-09-12, 2013-09-13, 2013-09-14, 2013-09-17, 2013-09-18, 2013-09-19, 2013-09-20, 2013-09-21, 2013-09-24, 2013-09-25, 2013-09-26, 2013-09-27, 2013-09-28, 2013-09-30, 2013-10-01, 2013-10-02, 2013-10-03, 2013-10-04, 2013-10-05, 2013-10-06, 2013-10-09, 2013-10-10, 2013-10-11, 2013-10-12, 2013-10-13, 2013-10-16, 2013-10-17, 2013-10-18, 2013-10-19, 2013-10-20, 2013-10-23, 2013-10-24, 2013-10-25, 2013-10-26, 2013-10-27, 2013-10-30, 2013-10-31, 2013-11-01, 2013-11-02, 2013-11-03, 2013-11-04, 2013-11-05, 2013-11-06, 2013-11-09, 2013-11-10, 2013-11-11, 2013-11-12, 2013-11-13, 2013-11-16, 2013-11-17, 2013-11-18, 2013-11-19, 2013-11-20, 2013-11-23, 2013-11-24, 2013-11-25, 2013-11-26, 2013-11-27, 2013-11-30, 2013-12-01, 2013-12-02, 2013-12-03, 2013-12-04, 2013-12-05, 2013-12-06, 2013-12-09, 2013-12-10, 2013-12-11, 2013-12-12, 2013-12-13, 2013-12-16, 2013-12-17, 2013-12-18, 2013-12-19, 2013-12-20, 2013-12-23, 2013-12-24, 2013-12-25, 2013-12-26, 2013-12-27, 2013-12-30, 2013-12-31, 2014-01-01, 2014-01-02, 2014-01-03, 2014-01-04, 2014-01-05, 2014-01-06, 2014-01-09, 2014-01-10, 2014-01-11, 2014-01-12, 2014-01-13, 2014-01-14, 2014-01-15, 2014-01-16, 2014-01-17, 2014-01-18, 2014-01-19, 2014-01-20, 2014-01-21, 2014-01-22, 2014-01-23, 2014-01-24, 2014-01-25, 2014-01-26, 2014-01-27, 2014-01-28, 2014-01-29, 2014-01-30, 2014-01-31, 2014-02-01, 2014-02-02, 2014-02-03, 2014-02-04, 2014-02-05, 2014-02-06, 2014-02-09, 2014-02-10, 2014-02-11, 2014-02-12, 2014-02-13, 2014-02-14, 2014-02-15, 2014-02-16, 2014-02-17, 2014-02-18, 2014-02-19, 2014-02-20, 2014-02-23, 2014-02-24, 2014-02-25, 2014-02-26, 2014-02-27, 2014-02-28, 2014-03-01, 2014-03-02, 2014-03-03, 2014-03-04, 2014-03-05, 2014-03-06, 2014-03-07, 2014-03-08, 2014-03-09, 2014-03-10, 2014-03-11, 2014-03-12, 2014-03-13, 2014-03-14, 2014-03-15, 2014-03-16, 2014-03-17, 2014-03-18, 2014-03-19, 2014-03-20, 2014-03-23, 2014-03-24, 2014-03-25, 2014-03-26, 2014-03-27, 2014-03-28, 2014-03-29, 2014-03-30, 2014-03-31, 2014-04-01, 2014-04-02, 2014-04-03, 2014-04-04, 2014-04-05, 2014-04-06, 2014-04-09, 2014-04-10, 2014-04-11, 2014-04-12, 2014-04-
```

For this analysis, I am examining a dataset that contains 5 years worth of data from the S&P 500 from 2013-2018. The data is stored in `all_stocks_5yr.csv` and it was downloaded from a repository on Kaggle that is linked below. The data consists of 618,445 observations of 7 variables. The variable `date` represents the trading day in question and is stored as a date. Variables `open`, `high`, `low`, and `close` represent the price in USD of the stock on one trading day. They are all stored as numeric values. `Open` is the price at market open, `close` at market close, and `high` and `low` represent the range of prices on that day. The variable `volume` represents the amount of shares bought and sold per ticker per day, it is a numeric. `Name` represents the stock ticker symbol, they are stored as characters and there are 506 unique values.

Through initial exploration, I stumbled onto the package TTR. It is a wealth of computational finance functions and information. I hope to use the tools available in this package to draw insights that would have previously been beyond my reach. For the purposes of this analysis I will select one ticker to analyze vs the market, and my starting capital will be \$100,000.

- Question 1: How can I predict the direction of stock price? First, I plan on using TTR to create new variables RSI, Bollinger Bands, Momentum, Volatility, and MACD. These variables will all be numeric values. These variables are all indicators used in financial analysis and fall into four categories: trend, momentum, volume, and volatility. I will examine these indicators, their meanings, and what value they bring to the table. For visualization I will plot the indicator values over time, as well as the profit signals.
- Question 2: Second, based on these probabilities of profit what is the best strategy? This strategy will essentially combine all the buying and selling signals of the chosen indicators. I can then create visualizations that show when the optimal time to buy and sell are.
- Task 3: Lastly, I hope log the individual transactions that will take place. I will be using an all-in stop-loss strategy where 100% of my portfolio value is invested each time a trade is placed. If I can muster it, I will create a function that takes a date range as an input and outputs transactions, and profit.

The overarching purpose of this analysis is to build a foundation for algorithmic trading. I hope to be able to build a live data algorithm by graduation and I believe this is an important step.

<https://www.kaggle.com/camnugent/sandp500>

## NBA Shot Analysis

```
glimpse(dat3)
```

```
## Observations: 200,966
## Variables: 24
## $ GRID TYPE      <chr> "Shot Chart Detail", "Shot Chart Detail", "Shot Chart Detail", "Shot Cha
```

[illegible]

```
## Observations: 219,458
## Variables: 24
## $ GRID_TYPE          <chr> "Shot Chart Detail", "Shot Chart Detail", "Shot Chart Detail", "Shot Cha
## $ GAME_ID            <chr> "0021800001", "0021800001", "0021800001", "0021800001", "0021800001", "00
## $ GAME_EVENT_ID      <dbl> 7, 10, 14, 17, 19, 21, 23, 25, 27, 29, 32, 35, 38, 42, 44, 46, 48, 50, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99, 101, 103, 105, 107, 109, 111, 113, 115, 117, 119, 121, 123, 125, 127, 129, 131, 133, 135, 137, 139, 141, 143, 145, 147, 149, 151, 153, 155, 157, 159, 161, 163, 165, 167, 169, 171, 173, 175, 177, 179, 181, 183, 185, 187, 189, 191, 193, 195, 197, 199, 201, 203, 205, 207, 209, 211, 213, 215, 217, 219, 221, 223, 225, 227, 229, 231, 233, 235, 237, 239, 241, 243, 245, 247, 249, 251, 253, 255, 257, 259, 261, 263, 265, 267, 269, 271, 273, 275, 277, 279, 281, 283, 285, 287, 289, 291, 293, 295, 297, 299, 301, 303, 305, 307, 309, 311, 313, 315, 317, 319, 321, 323, 325, 327, 329, 331, 333, 335, 337, 339, 341, 343, 345, 347, 349, 351, 353, 355, 357, 359, 361, 363, 365, 367, 369, 371, 373, 375, 377, 379, 381, 383, 385, 387, 389, 391, 393, 395, 397, 399, 401, 403, 405, 407, 409, 411, 413, 415, 417, 419, 421, 423, 425, 427, 429, 431, 433, 435, 437, 439, 441, 443, 445, 447, 449, 451, 453, 455, 457, 459, 461, 463, 465, 467, 469, 471, 473, 475, 477, 479, 481, 483, 485, 487, 489, 491, 493, 495, 497, 499, 501, 503, 505, 507, 509, 511, 513, 515, 517, 519, 521, 523, 525, 527, 529, 531, 533, 535, 537, 539, 541, 543, 545, 547, 549, 551, 553, 555, 557, 559, 561, 563, 565, 567, 569, 571, 573, 575, 577, 579, 581, 583, 585, 587, 589, 591, 593, 595, 597, 599, 601, 603, 605, 607, 609, 611, 613, 615, 617, 619, 621, 623, 625, 627, 629, 631, 633, 635, 637, 639, 641, 643, 645, 647, 649, 651, 653, 655, 657, 659, 661, 663, 665, 667, 669, 671, 673, 675, 677, 679, 681, 683, 685, 687, 689, 691, 693, 695, 697, 699, 701, 703, 705, 707, 709, 711, 713, 715, 717, 719, 721, 723, 725, 727, 729, 731, 733, 735, 737, 739, 741, 743, 745, 747, 749, 751, 753, 755, 757, 759, 761, 763, 765, 767, 769, 771, 773, 775, 777, 779, 781, 783, 785, 787, 789, 791, 793, 795, 797, 799, 801, 803, 805, 807, 809, 811, 813, 815, 817, 819, 821, 823, 825, 827, 829, 831, 833, 835, 837, 839, 841, 843, 845, 847, 849, 851, 853, 855, 857, 859, 861, 863, 865, 867, 869, 871, 873, 875, 877, 879, 881, 883, 885, 887, 889, 891, 893, 895, 897, 899, 901, 903, 905, 907, 909, 911, 913, 915, 917, 919, 921, 923, 925, 927, 929, 931, 933, 935, 937, 939, 941, 943, 945, 947, 949, 951, 953, 955, 957, 959, 961, 963, 965, 967, 969, 971, 973, 975, 977, 979, 981, 983, 985, 987, 989, 991, 993, 995, 997, 999, 1001, 1003, 1005, 1007, 1009, 1011, 1013, 1015, 1017, 1019, 1021, 1023, 1025, 1027, 1029, 1031, 1033, 1035, 1037, 1039, 1041, 1043, 1045, 1047, 1049, 1051, 1053, 1055, 1057, 1059, 1061, 1063, 1065, 1067, 1069, 1071, 1073, 1075, 1077, 1079, 1081, 1083, 1085, 1087, 1089, 1091, 1093, 1095, 1097, 1099, 1101, 1103, 1105, 1107, 1109, 1111, 1113, 1115, 1117, 1119, 1121, 1123, 1125, 1127, 1129, 1131, 1133, 1135, 1137, 1139, 1141, 1143, 1145, 1147, 1149, 1151, 1153, 1155, 1157, 1159, 1161, 1163, 1165, 1167, 1169, 1171, 1173, 1175, 1177, 1179, 1181, 1183, 1185, 1187, 1189, 1191, 1193, 1195, 1197, 1199, 1201, 1203, 1205, 1207, 1209, 1211, 1213, 1215, 1217, 1219, 1221, 1223, 1225, 1227, 1229, 1231, 1233, 1235, 1237, 1239, 1241, 1243, 1245, 1247, 1249, 1251, 1253, 1255, 1257, 1259, 1261, 1263, 1265, 1267, 1269, 1271, 1273, 1275, 1277, 1279, 1281, 1283, 1285, 1287, 1289, 1291, 1293, 1295, 1297, 1299, 1301, 1303, 1305, 1307, 1309, 1311, 1313, 1315, 1317, 1319, 1321, 1323, 1325, 1327, 1329, 1331, 1333, 1335, 1337, 1339, 1341, 1343, 1345, 1347, 1349, 1351, 1353, 1355, 1357, 1359, 1361, 1363, 1365, 1367, 1369, 1371, 1373, 1375, 1377, 1379, 1381, 1383, 1385, 1387, 1389, 1391, 1393, 1395, 1397, 1399, 1401, 1403, 1405, 1407, 1409, 1411, 1413, 1415, 1417, 1419, 1421, 1423, 1425, 1427, 1429, 1431, 1433, 1435, 1437, 1439, 1441, 1443, 1445, 1447, 1449, 1451, 1453, 1455, 1457, 1459, 1461, 1463, 1465, 1467, 1469, 1471, 1473, 1475, 1477, 1479, 1481, 1483, 1485, 1487, 1489, 1491, 1493, 1495, 1497, 1499, 1501, 1503, 1505, 1507, 1509, 1511, 151
```

Association for the 2009-2010, and 2018-2019. The data is stored in the `nba_shots_2010.csv` and the `nba_shots_2019.csv`. I compiled this data using an api based web scraper that uses the `requests` and `json` modules in `python`. The link to the web scraper template that I used is included below. The 2010 data consists of 200,966 observations of 24 variables. The 2019 data consists of 219,458 observations and 24 variables. The variables for each dataset are the same. `GRID_TYPE` is negligible, it is a class character with one unique value. `GAME_ID`, `GAME_EVENT_ID`, `PLAYER_ID`, and `TEAM_ID` are all either characters or numeric with the purpose of identifying some individual or event. `PLAYER_NAME` and `TEAM_NAME` are more useful versions of the ID variables, being stored as human readable characters. `PERIOD` identifies the current period when a given shot was taken, it is stored as a numeric. `MINUTES_REMAINING` and `SECONDS_REMAINING` in conjunction tell the amount of time remaining when the shot was taken and are stored as numerics. `EVENT_TYPE` tells whether the shot was a Missed Shot or a Made Shot and is stored as a character string. `ACTION_TYPE` describes the type of shot being taken, there are 45 different types of shots stored in the data, and they are stored as characters. Wow! `SHOT_TYPE` tells us whether a shot was a 3PT Field Goal, or a 2PT Field Goal stored as characters. `SHOT_ZONE_BASIC`, `SHOT_ZONE_AREA`, and `SHOT_ZONE_DISTANCE` quantify where the shot was taken in 7, 6, and 5 zones respectively. Stored as characters. `LOC_X` and `LOC_Y` are powerful variables that plot a shot's location on a coordinate plane. They are numeric, as are `SHOT_ATTEMPTED_FLAG`, and `SHOT_MADE_FLAG` which are 0 or 1 and indicate the same information as other statistics. `GAME_DATE`, `HTM`, and `VTM` give information about the game where the shot was taken, and are stored as a numeric and two characters respectively.

- Task 1: First, I will detail the process of data allocation. I will then examine the strategy shakeup beautifully explained by Kirk Goldsberry in his book *Sprawlball* by examining relationships such as FGP over distance from basket, points per shot over distance from basket, and the change in shooting distributions from 2009-2019. Ggplot will be used in conjunction with cowplot for visualizations.
- Task 2: Second, after giving the above relationships a good look, I will attempt to develop my very own basketball statistics. One statistic will be based solely on a teams efficiency in executing the *Sprawlball* strategy. It will resolve to a simple numeric value that can be evaluated on a continuous scale. The other statistic will attempt to predict outcomes at a better than random percentage (Tested by hypothesis test). I will do this by combining various elements such as points per possession, average FGP, average shot distance, and shot type.
- Task 3: If possible, I hope to use my twitter developer access to build a twitter bot. This bot will interact with the twitter API and live web data to publish when a *Sprawlball* inspired event occurs. If my statistic evaluates to higher than a certain value the bot will automatically publish a tweet describing the game in question.

The overarching purpose of this analysis is to again visited optimized strategy and its benefits.

<https://datavizardry.com/2020/01/28/nba-shot-charts-part-1/>