Professional Golfers' Association (PGA) Tour Data Visualizations

LinQuest Corporation

Anthony Brennan
Maxwell Lazzara
Cameron Meadows

Advisor: Dr. Michael Gorman

June 20, 2021

Table of Contents

1. Executive Summary

LinQuest has several responsibilities and a broad range of specialties; one of those responsibilities is providing analytical solutions on sports data to their client for users to interact with and use at their discretion. In May 2021, they helped launch a new sports segment focused on golf on their client's site, and our goal is to provide them with data visualizations that intuitively explain player performance to the site users. Our team has developed these visualizations from what we found to be the most statistically significant golf metrics to golfer performance and developed them in a way that the average person on the site could view and understand. We have gained a keen understanding of the metrics throughout the project and figured out statistically what can describe and predict a 'good' round on the PGA Tour. With those understandings in place, our team has developed visualizations that display this information to the users.

All visualizations are programmed using the Python programming language. Once implemented into the client's environment, they will provide the users with helpful knowledge at the tip of their fingers. A traditional sports site will provide a multitude of tables filled with numbers, and without a prior understanding of the sport, someone may not be able to decipher what is good from the bad. Most users are looking to quickly make well informed decisions that will make them the most money. Not all of them know the game well enough to look at tables and piece together which player would be the best bet in any given circumstance. Therefore, we have provided LinQuest with five visualizations that work with the data from those tables to paint a picture for the users of how a golfer has performed in the past and under certain conditions. These visualizations were thought of outside of the box to look at only the most impactful golf metrics along with the most impactful course and weather attributes. Our vision was to allow a user to look at important golf metrics, and with the attributes for the upcoming tournament, look to see how those metrics change with the change of attributes. This allows the user to form their own educated decision on how a player may perform, but we are leading them toward how we feel they will perform given past data. With the approval from LinQuest, we expect that some if not all our products will go into operation and draw the attention of outside users because our products will give them an edge over the competition.

2. Business Background

LinQuest is headquartered in Beavercreek, OH, and began operations in 2004 to provide technical solutions to government agencies. Many of LinQuest's primary customers are related to the government and national security, primarily in defense, intelligence, and other civil government departments. LinQuest follows a 4-step life cycle: “understanding the customer needs, engineering the product, integrating the system, and operations”. Much of its products revolve around space and locational software. LinQuest's mission statement is "To provide innovative and high-quality technologies, solutions and services to national security and industry customers focused on the convergence of C4ISR, information, and cyber systems." It has six main company values: People, Teamwork, Integrity, Excellence, Innovation, and Customer Satisfaction. The company vision is "To be a leader in the market for engineering, operational services, and technology solutions with an impeccable record for customers' mission success and where our employees excel at what they do." Perduco is a LinQuest company that Stephen Chambal co-founded, and they are focused on using data to help their clients make decisions through analytics. Within the realm of analytics, there is a group focused on sports analytics, which is what our project focuses on. One of the key products that they produced for their client is an optimization model that provides users with a lineup for daily fantasy sports competition that will give them the most *bang for their buck*. When mentioning their optimization model, they noted that they have perfected it so that their users are seeing results that are far above users of other gambling and fantasy sports sites.

3. Project Background

Our contact, Jacob Loeffelholz, works within the LinQuest corporation. Jacob and his team work on developing interactive tools for the users on one of the significant sports broadcasting company's web platforms. They currently provide those interactive tools for four of the five major sports (NBA, NFL, NHL, and MLB), and as of late May, they have moved into the Professional Golfers' Association (PGA). Golf has an overwhelming amount of data, from the type of grass to the number of strokes gained on a hole, that can be used to describe a golfer’s performance. This data is embedded within tables that provide absolutely nothing but numbers to a user. Without a deep knowledge of golf, it would be hard to understand or get a proper understanding of these tables. Our task is to help launch into this new sector (PGA Tour) and find new and unique ways to display golf metrics to their users in a way that takes little to no background knowledge of golf or the PGA Tour. With the idea in mind, these same visualizations could be used in other sports to allow users to gain an advantage.

Golf is an old but still growing game globally, and there are still new metrics for measuring performance being created. Strokes gained has been around for less than a decade, and few people understand what it means. Our client needs to provide users with interactive visualizations that will explain all these metrics to gamblers and daily fantasy sports users who currently do not understand all the metrics and their current table format. These visualizations will need to give the users an edge over their friends and the house odds when they are in competitions. Daily fantasy sports allow users to have a salary cap to select six golfers. Each golfer is given a dollar amount based on past performance and future predictions. Golfers accumulate points based on performance on each hole, along with bonuses based on the final finishing spot. Users are turning to analytics and more metrics to select the golfer that will give them the best chance to win. Strokes gained is a new metric that explains a player's performance the best, but that is not the only metric that should be considered in the development of our visualizations. Using the most basic of metrics creatively will still provide the users with an advantage in understanding a player's performance.

4. Project Methodology

Outline of our Approach/Methodology:

1. **Prepare and learn the data with descriptive statistics.** We worked through all the data to handle any missing data and other issues that arise. Once the data was in the correct format, we began merging data tables (i.e., golf_data, courses, and weather) based on their universal unique identifier (UUID) to perform a complete analysis. All data manipulation and merges were performed in a single Python script, *data.py*, to use in the study. Before developing statistical models, we ran simple descriptive statistics on all parts of the data as a quality check to make sure we had the data in a way that could be processed for statistical purposes and in our visualizations. We relied on statistical methods, like correlation and regression, to determine statistically significant attributes to the golfer's performance on a round-by-round basis. The goal of this phase was to prepare our data and begin to gain an understanding of which metrics we intend to explore in our visualizations.
2. **Finding key factors that correlate to a golfers' performance** to explain a player's round by round performance. With a lot of exploration in R-Notebooks and Jupyter Notebooks, we began understanding what attributes we should look at, which led to the first visualization, an interactive filter scatter plot visualization [Appendix A]. This filter visualization allowed further exploration of what made sense when explaining a player's performance. Statistics were then implemented using all the metrics to see how each metric correlated with a player's round performance. We later developed a k-Nearest Neighbor model that found the features (**Strokes Gained**, **Birdies Gained**, **Birdies**, **Bogeys**, and **Bogeys Avoided**) that were the most significant predictors for a player's performance. From a Gambling and Daily Fantasy sports standpoint, we presumed that 'Birdies,' 'Bogeys,' and 'Strokes Gained' were going to be critical features for nearly all visualizations created. We decided to look at how players performed in the past and with different conditions (weather and course) with various features. Using the same correlation idea, we developed a list of course and weather attributes (**Temperature**, **Wind Speed**, **Course Length**, **Course Rating**, and **Green Grass Type**) that greatly impacted a player's performance.
3. **Develop our best visualizations in Python code** for the client to use in their environment. Our final product is developed using Python programming language and libraries such as Bokeh, Plotly, Seaborn, Pandas, and NumPy. We found that visually in sports, one of the best visualizations is a heatmap, so we developed two of our five visualizations as such. These heatmaps explain how a player performs under particular weather or course conditions by looking at Strokes Gained, Birdies, or Bogeys Avoided. Our interactive bar chart was developed using our kNN model to select the best features for a user to then choose from and see how a player has done in every round with that feature. This bar chart also provides them with a line showing the player's average and the entire PGA average to indicate if a player is above or below the whole PGA Tour. The fourth visualization is a conditions bar chart that shows how a player has performed against their average with a given feature for all five of the weather and course

attributes we found to be the most impactful. The user can set the attributes to match the upcoming tournament and see how any given player has performed in the past with those conditions compared to their overall average.

5. Data

LinQuest provided us with three separate data files from 2018-Present, and all these data files contained Universally Unique Identifiers (UUID) that made it possible to link the files together.

Data Provided:

1. **Golf Data**: 50,063 records and 34 columns, containing data on a round by round basis for every tournament/course/player during the given period (2018-Present). Each record represents a single player's performance for *x* round in *y* tournament.
2. **Weather Data**: 618 records and six columns containing data for each tournament round during the given period (2018-Present).
3. **Course Data**: 94 records and ten columns contain data for each course that was played during the given period (2018-Present).

Data Manipulation:

In one Python script, *data.py*, we performed all our data manipulations so that we were able to use all the data as needed for our visualizations uniformly.

The golf data file had several instances that needed to be changed or accounted for, and the first being an empty row for each player in every tournament representing only the final overall score. That provided us no added information, so we dropped missing data on the course_id column because it would show NA on those rows. We also dropped on missing columns for the 'strokes' column because we found one tournament with zero scores, so that tournament was useless for our analysis. Since a player can have zero Birdies, Pars, and Bogeys in a round, we filled missing values in those columns with zero. As previously mentioned, Strokes Gained is a newer metric to the game of golf, and since it's new, it is still not recorded in every tournament. Therefore, we had to calculate for strokes gained by calculating the average score for each round of each tournament and then subtracting the player's score from the average for that round where the strokes gained were missing. To add value to the golf data, we added a birdie gained, and bogeys avoided metric to the data set and will go further into those metrics in the next section.

Most manipulation occurred with the weather data file because it was given to us with multiple times for each round in no uniform order. We had to collect the average of each weather metric for each date in every tournament because our golf data wasn't equipped to provide golf data for multiple times within the same date. Once we no longer had duplicate dates for each tournament, we needed to know the weekday for each date to link to a round (1, 2, 3, 4), and since tournaments are played Thursday-Sunday, it was easy to do. Then we found tournaments that started on Wednesday and some that went to Monday due to cancellation or another unique incident. We had to account for this by using “if-statements” if there was no Thursday but was a Wednesday, or No Sunday but was a Monday, etc. We were getting it down to have only rounds 1, 2, 3, and 4 for each tournament to tie back with the golf data.

There were final cosmetic changes to the data, such as renaming columns to match columns in another file or changing the player's name to a proper format. Finally, we merged golf, weather, and course data on like columns into one data frame consisting of 50,063 records and 52 columns. All visualizations were built from this combined data frame that can be called by using the *data_merge* function once you have imported the *data* python script.

6. Understanding the Metrics

Without prior knowledge about golf, learning what the metrics mean can be a task in and of itself, especially when multiple Strokes Gained metrics have only been around for less than a decade. Strokes Gained can be explained by *the number of strokes above or below the average strokes for the given round,* mathematically formulated by:

$$g_i = J(d_i, c_i) - J(d_{i+1}, c_{i+1}) - 1$$

Where:

d = distance from the hole
c = condition of current ball location
J = Average number of strokes for PGA player to get in the hole from c

While Strokes Gained is easy to think of and formulate, the subgroups of Strokes Gained (Tee to Green, Around the Green, Putting) are much more complex, which is why the advanced formula. Given the complexity and newness of these metrics, they are not kept for every tournament, which is why we had to calculate for missing overall Strokes Gained and ignore all other Strokes Gained metrics. Before getting into the following two metrics, we will describe the scoring in golf. Strokes can be described as the total number of swings at the golf ball in a round. Par is what an expert player would be expected to get on a hole, and a birdie is when a player finishes a hole one stroke better than par, while a bogey is when they finish with strokes above par.

With the same formulation of overall Strokes Gained, we developed two new metrics that turned out to be critical to a player's performance, Birdies Gained and Bogeys Avoided. Birdies Gained are a golfer’s total birdies for a round above or below the average birdies for that same round. Bogeys Avoided are a golfer’s total bogeys for a round above or below the average bogeys for that same round. These are not only statistically important to a player's performance, but also to a fantasy sports user where they want a player above everyone else in birdies and above everyone else in avoiding bogeys.

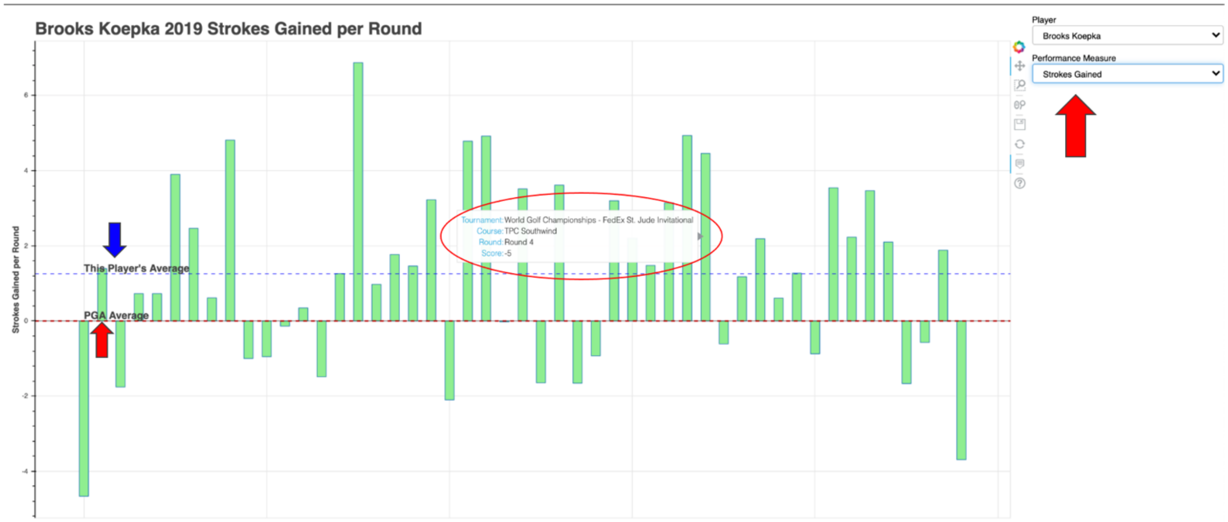
7. Visualizations

Interactive Bar Chart

The interactive bar chart has been developed using the Bokeh library in Python and is running on only data from tournaments played in 2019. This year was chosen since it was the last complete year that we had from our entire data set, but the visualization can run on all the provided data. The data for this visualization was also sorted by the date, and the x-axis was set to the index to show every round that was played throughout the year.

Key Features of visualization:

- Runs on a K-Nearest Neighbors (kNN) model that uses correlation filtering from 0.2-0.8 to choose the features from the data set that have the lowest Root Mean Square Error, which shows the highest predictive power. In this case, the features selected were Strokes Gained, Birdies Gained, Birdies, Bogeys, and Bogeys Gained, which had an R-squared value of 0.98 and RMSE value of 0.45.
    - K-Nearest Neighbors (kNN) is a data classification algorithm that estimates the likelihood a data point is to be a member of one group or another, which is dependent on the group of data points closest to it [1].
    - Due to features missing data and a feature selector wrapper only returning one feature (Strokes Gained), we had to use correlation filtering, which leaves features having multicollinearity. Still, for this visualization, it works fine since we are making actual predictions.
- Users can choose from the features found using the kNN model.
- Users can choose from any Golfer that played in 2019.
- Provides a Blue dashed line that shows the selected player's average for the selected metric.
- Provides a Red dashed line that shows the PGA Tour average for the selected metric.
- The hover tool is also active, showing the Tournament, Course, Round, and Score for each bar represented in the visualization. [Circled in Red in *Interactive Bar Chart* below]



Player: Brooks Koepka

Performance Measure: Strokes Gained

Tournament: World Golf Championships - FedEx St. Jude Invitational
Course: TPC Southwind
Round: Round 4
Score: -5

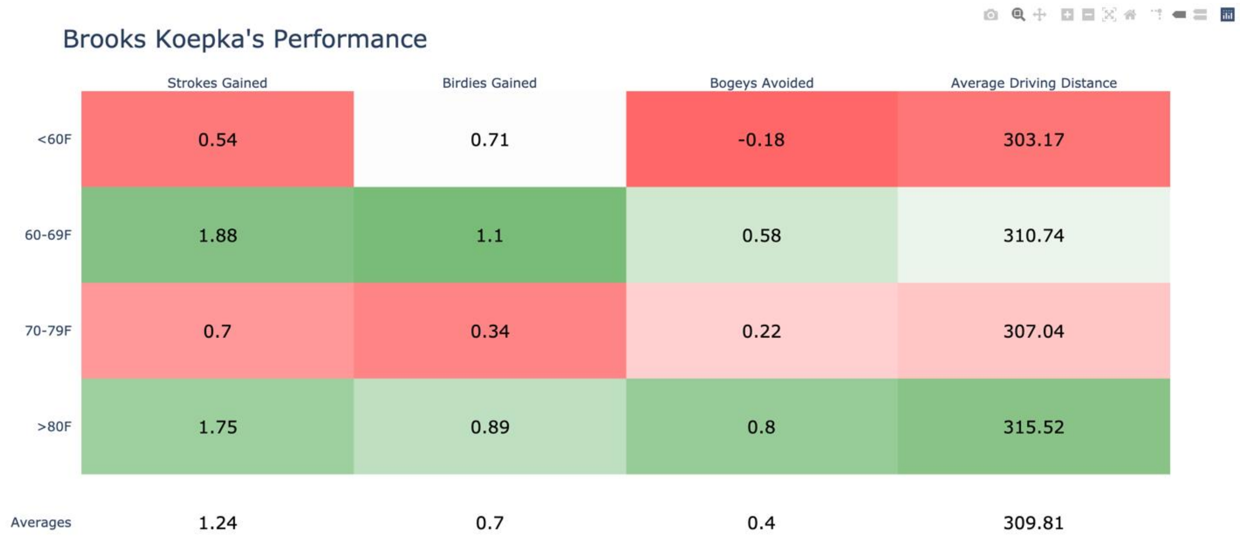
Interactive Bar Chart

Condition Heatmaps

Both condition heatmaps were developed using the Plotly library in Python, and they both run on all data provided for the project (2018-2021). Both heatmaps also allow you to change within the code which player you want to view. The data used in the heatmap has been standardized so that the color is based on the relative distance from the overall metric average, where Green represents performance above average, and Red represents performance below average. Though the data has been normalized, the heatmap still displays the actual average of each square, as shown in the *Single Condition Heatmap* below.

Key Features of Single Condition visualization:

- You can enter any course/weather attribute within the code that you want to see on the y-axis, i.e., Course Length, Course Rating, Wind Speed, etc.
- You can change the metrics to whatever metrics you find to be the most important within the code, i.e., Birdie, Bogey, GIR, etc.
- Filters out players that play less than six rounds.
- A user can quickly look at a heatmap like the one shown in the *Single Condition Heatmap* below and see that Brooks Koepka performs very well with temperatures above 80 degrees.



Brooks Koepka's Performance

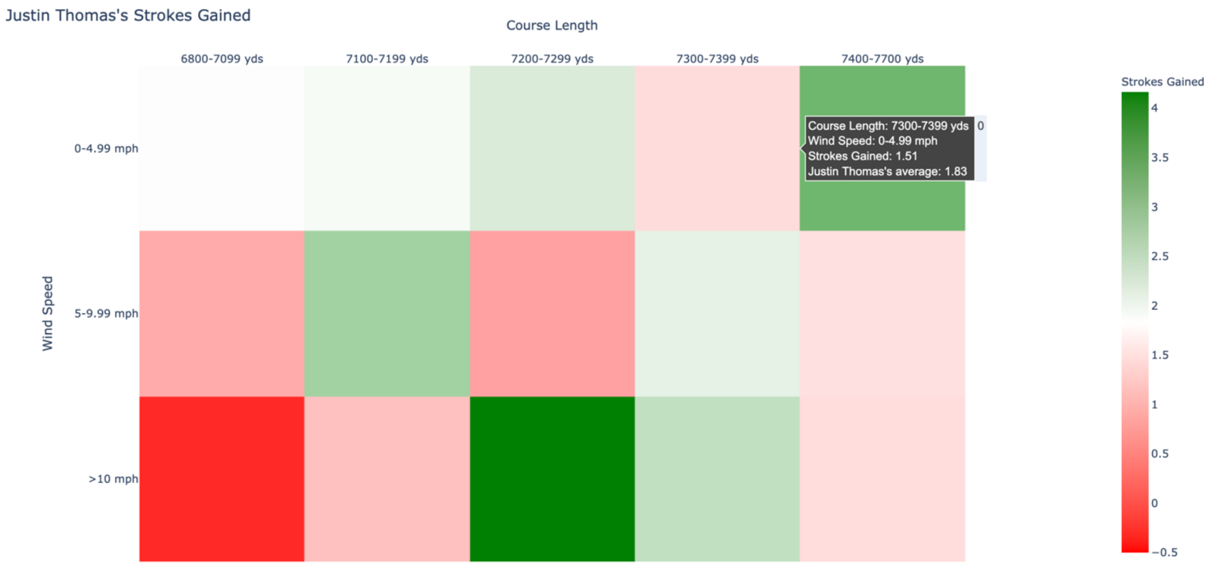
	Strokes Gained	Birdies Gained	Bogeys Avoided	Average Driving Distance
<60F	0.54	0.71	-0.18	303.17
60-69F	1.88	1.1	0.58	310.74
70-79F	0.7	0.34	0.22	307.04
>80F	1.75	0.89	0.8	315.52
Averages	1.24	0.7	0.4	309.81

Single Condition Heatmap

The multiple condition heatmap has been developed to show three separate heatmaps, representing a significant feature (Strokes Gained, Birdies Gained, Bogeys Avoided). This heatmap is colored from Green to Red, with the midpoint being the average for the given feature. Within the code, there is an if statement that makes sure there are at least three rounds played for each square represented so that there are not any outlier single round squares that are misrepresenting the data.

Key Features of Multiple Condition visualization:

- Allows the user to see weather or course attributes on each axis to see how a player performs under those conditions.
- The hover tool is active to show what each visualization is representing.
- Filters out players that play less than six rounds.
- Within the code, you can select any attribute against another, i.e., Temperature, Course Rating, Precipitation, Grass Type, etc.
- Users can quickly look at upcoming tournament conditions and see how a player has performed with those conditions in the past. i.e., Justin Thomas performs very poorly with short courses and high winds, as shown in the *Multiple Condition Heatmap* below.



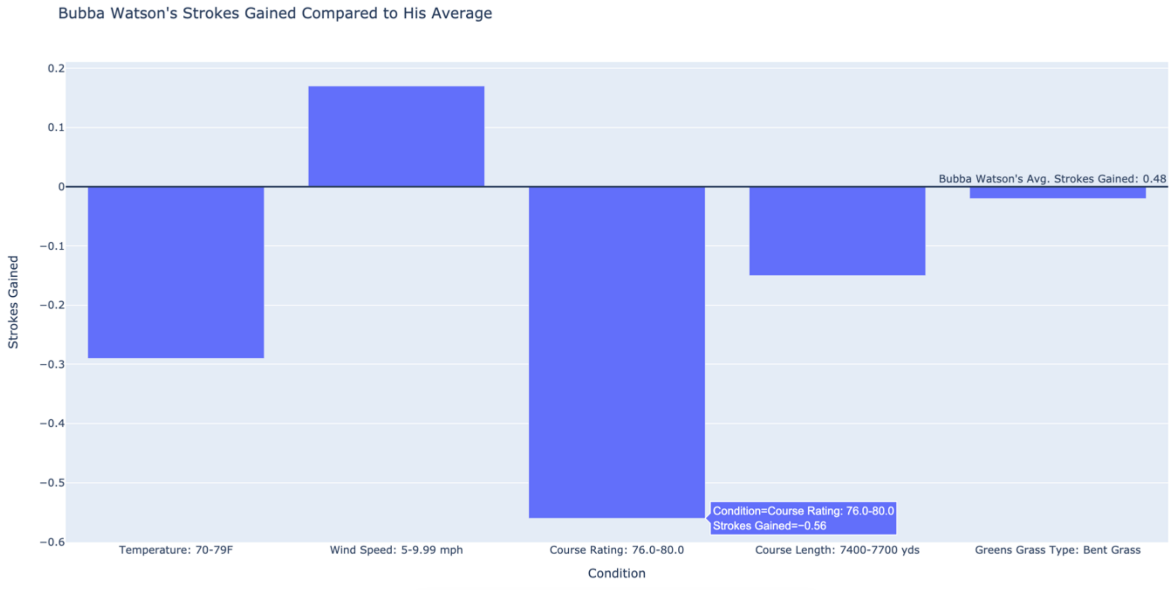
Multiple Condition Heatmap

Conditions Bar Chart

The conditions bar chart is developed using the Plotly library in Python, and it uses all the data provided from 2018-2021. The chart is created so that the zero line represents the player's average for the chosen y-axis metric. In the *Round Conditions Bar Chart* figure below, you can see the average that the zero line represents with the label shown on the far right of the graph; in this case, it is 0.48. This was created with the idea that a user could choose the conditions of an upcoming tournament and see how that player performs against their average. The *Round Conditions Bar Chart* figure below represents the forthcoming US Open at Torrey Pines.

Key Features of visualization:

- Within the code, you can choose any golfer to be represented on the graph.
- You can choose from any of the metrics within the code, i.e., Birdies Gained, Bogeys Avoided, Birdies, etc.
- Within the code, you can put in the specifications of any tournament and see how a player performs against their average.
- The hover tool is active to hover over any bar and show what that bar is representing.
- Users can quickly look at the *Round Conditions Bar Chart* figure below and see that Bubba Watson may not be a safe bet at the US Open in 2021.



Round Conditions Bar Chart

8. Recommendations

After completing all our visualizations and analysis we have the following recommendations moving forward.

- A wrapper method feature selector could be the best choice when choosing the features for any visualizations once all the Strokes Gained metrics were completed.
- Displaying the number of rounds represented by each square in the heatmaps would be beneficial to assure there are no outliers; in our code, we have only filtered out any square with less than three rounds.
- All our visualizations may be best used to display when a user-selected on a player's name, perhaps shown in a hover tool with the ability to be enlarged.
- A column that identified match play would be beneficial to make sure that data does not get mixed in with regular PGA Tour events.

9. Out of Scope

Predicting a player's score or the winner of any given professional tournament was labeled as out of this project's scope. We have developed visualizations that explain the data that we are given. It will provide the front-end user enough information to form an educated prediction on who they think will perform well given historical data.

10. Deliverables

Following the presentation, we have provided our entire code repository in one single zip file, and each script was embedded with comments along the way to explain every step. We have also provided our presentation slide show for the client to use at their discretion.

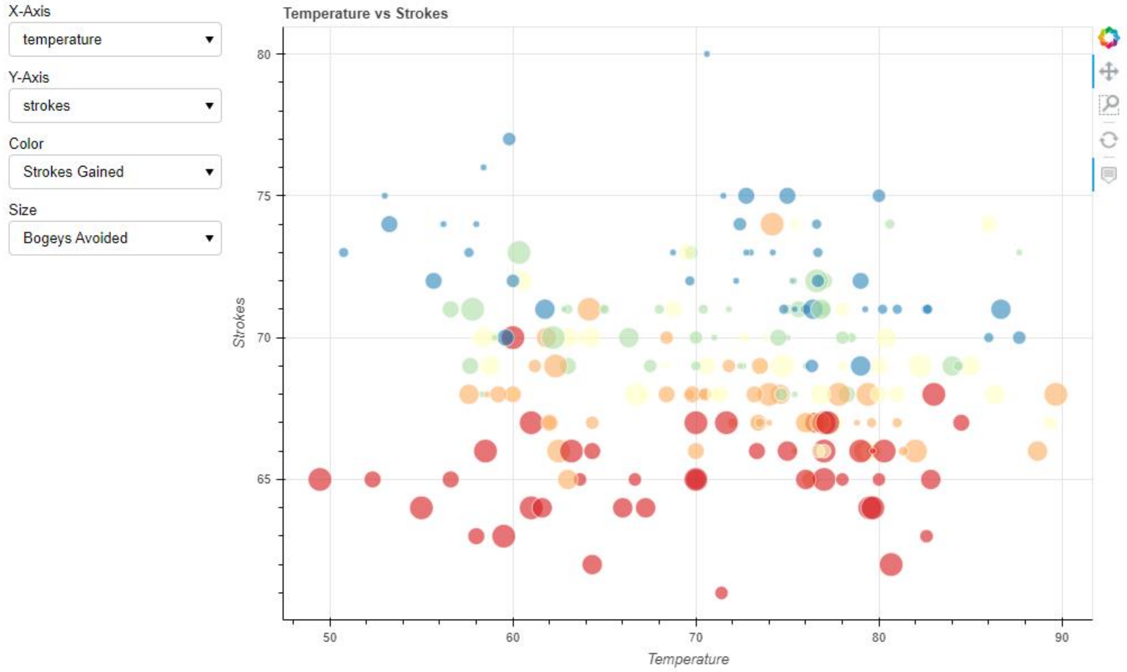
Code Repository:

- Final Visualizations [folder]
    - Data [folder]
        - golf_data [folder]
            - capstone_data_2018pt1.xlsx
            - capstone_data_2018pt2.xlsx
            - capstone_data_2019pt1.xlsx
            - capstone_data_2019pt2.xlsx
            - capstone_data_2020.xlsx
            - capstone_data_2021.xlsx
        - capstone_data – courses.xlsx
        - capstone_data – weather.xlsx
    - data.py
    - bar_chart.py
    - single_condition_heatmap.py
    - multi_condition_heatmap.py
    - conditions_barchart.py
    - filter_visualization.py

Special Thanks

It was an honor for all of us to work on this project, and while it was challenging at times, it was a pleasure for our team to work with you along the way. The skills that our team developed and sharpened are unmeasurable, and it was enjoyable to work with data that we are all fond of being sport's fans. We were forced to follow through on a real-world project from beginning to end, where we put all our learned skills together to make the very best visualizations in our minds. We hope you find these visualizations helpful, as well as finding a way to implement them into the environment for everyone to use.

Appendix A



X-Axis
temperature

Y-Axis
strokes

Color
Strokes Gained

Size
Bogeys Avoided

Interactive Filter Scatter Plot

References

[1] Techopedia. (2017, March 14). *What is K-Nearest Neighbor (K-NN)? - Definition from Techopedia*. Techopedia.com. https://www.techopedia.com/definition/32066/k-nearest-neighbor-k-nn.