1. Good afternoon all. My name is Andrew Estes and my thesis is titled a Poisson Analysis of Formula 1 Pit Stops. Basically, I attempted to predict when a pit stop would be made during a Formula 1 race.
2. To start with, we need to understand data types. The type of data we have leads us to different models. My goal was to predict whether a pit stop would be made or not. This clearly indicates towards a Poisson Distribution in the probability that an event – pit stop in this case – occurs.
3. To keep it simple, a Poisson distribution involves data that is countable. Height based upon gender is a good example, there are 120 thousand women with the height of 160cm. Another type would be number of pit stops during a race.
4. Okay, now we should have an expert-level understanding of statistics at this point. You’re welcome and tips can be left in the jar on the way out. We now need to learn about Formula 1, aka F1. F1 cars are the fastest road-racing cars in the world. In 2020, a driver Lewis Hamilton averaged 164 mph through a lap and back in 2005, a driver hit a top speed of 232 mph. However, it isn’t a drag race. There are many corners that require significant braking – slowing to 60mph before re-accelerating up to 200mph. The impact of the braking is often measured between 4-6g force. To put that in perspective, Apollo 16 – the last mission to have men on the moon – re-entered the earth’s atmosphere at 7g forces. What we see here is the starting grid. There are 20 cars made up of 10 separate teams.
5. Those drivers are all competing with each other for placement. The higher you place, the more points you get. And a cumulative point total leads to a driver’s final placement at the season’s end. The team’s final placement depends on the cumulative point total from both drivers. As you can see in 2022, McLaren lost to Alpine even though McClaren’s top driver was better than Alpine’s top driver.
6. There are over 20 races per season and normally five of the six hospitable continents are utilized. Within each continent, many different countries have provided a course.
7. Within each country, there are multiple tracks that can be used. You can see the US has the highest variety of tracks and is the fourth most frequently visited country.
8. In our 5-year study, 27 countries were visited. Italy was, by far, the most frequently visited country. This is likely due to the heavy racing that took place in 2020, the year of Covid.
9. With the wide variety of continents and countries, there is a wide variety in climate. You can see the basic weather numbers here…
10. In addition to the geographic impact, each course has a unique layout. You can see the France and Chinese course layout here. The colors correspond to the gear shift of the driver who had the fastest lap in the respective race. China is obviously much curvier and requires more braking than France. Looking at the color-pattern, the fastest driver in China had to go down to 2nd gear meanwhile in France, the lowest gear was 3rd gear. Similarly, the highest gear in France was hit 3 times, in China, it was only hit once.
11. In addition to the layout changes, there are two types of tracks. There is a dedicated racetrack and there is a street track. Here is a snapshot of the Vegas race that will occur in November this year. It’s right on the strip. And as you can imagine, having a race on a regular road has its own challenges that dedicated tracks don’t provide.
12. Let’s look at Monaco for example. This is a steep decline with a tight turn.
13. Austin’s “tight” corner isn’t as angled for the turn, nor does it have the elevation change to consider.
14. My analysis was specifically on the pit stop aspect of the race. In this picture, you’ll see the pit stop lane on the left side – and can even make out a car leaving the pit stop area. All pit stop lanes are at the start/finish line. And while it looks clear-cut, it can be pandemonium.
15. Here is a photo of two cars leaving at once along with a third car getting the tires changed.
16. It’s quite crowded. In fact, there are 20 people for each car with specific roles when it comes to the pit stop.
17. Here is a real-life picture of a pit stop with the various crew members huddled in.
18. Now that we have an idea of what a pit crew responsibility is, we can actually look at a video of a pitstop…. Man, it would be great to be able to change the tires on my car in 2.1 seconds instead of making a 4 hour long appointment. This is completely unrelated to the analysis but if it took 2 seconds to change a vehicle’s tires, the entire population of Kirksville could have their tires changed in less than 10 hours.
19. Back to the video, I took a snippet 1 second apart. Excluding the slightly separate camera angle, can you tell me the difference here? It's the stripe on the tire. That indicates that a different compound is used.
20. In fact, there are 6 different types of hard tires, excluding intermediate tires and wet tires.
21. The type of tire used, and the decision to make a pit stop, is determined by a group of personnel including the race strategist, chief engineer, other crew members, and to an extent the driver. The staff has a massive influx of data coming from the various sensors on the car to help measure tire degradation and other information. And they use this information to make the decision to pit or stay out.
22. Other than the tires and timing, a consideration a strategists uses to make a pit stop decision involves flags. A yellow flag forces drivers to drive no faster than 80 kmh, around 50mph. If drivers on the course are only going 50mph, rather than 150mph, then the time loss suffered by making a pit stop is significantly reduced. So a 20 second loss can be transformed into a 15 second loss.
23. Now, let’s take a look at the F1 race in Austin. Any guesses where the start-point is?
24. Any questions for me?
25. The first step in any project is getting the data. The data for the F1 information can only be accessed via Python. Here is a snippet of the over 500 dataframes for the initial analysis. You’ll notice a wide variety of dimensions. Even the same race a year apart has different dimensions.
26. For the laps dataframe, there were 27 columns and rows ranging from 60 to over 1000 for each race.
27. The results dataframe had 16 rows and 20 columns for each race.
28. We merged those two dataframes by Driver, Grid Position, Position, and Points to create a “final” dataframe.
29. There are two other dataframes that we considered but did not use. The first was race control messages.
30. The second was weather.
31. Here are some snapshots of various transformations. As you can see, it took over 2400 lines of Pythonic code to get the data in an appropriate format. We tried several attempts to clean the data. The “Total Time” variable caused errors as drivers who were lapped or did not complete the race were intermingled in the final row. We could remove them from the dataframe but it would result in a 97% data loss. We then tried to create the pit stop time, but there were a bunch of issues with that. To begin with, each driver’s very first lap included a pit stop exit time. Obviously driver’s don’t begin in the pit lane so this was strange and fixed by forcing the pit stop exit time to 0 on the first lap. The second issue was the timing – the first record showed a pit exit time of 24 minutes 6 seconds and 41 miliseconds. Once again, an obvious error as it doesn’t take 20+ minutes for a pit stop. We then tried total sector time, but whenever there was a pit stop, there was no sector 1 time. Our next attempt was to use sector session time. Similar to sector time, session time is a slightly different time measurement but still came away with the same issue. Our next step was creating cumulative time per lap by driver. We used a groupby method and cumulative summing functions in Python to create the driver’s total lap time and total drive time. Sorting by total time, we know that the ending order should have points go in 25, 18, 15, 12, 10, etc. However, we had only 2 of the top ten places were accurate. If we remove pit stop times and sort by race time, we had 7 spots in the top ten accurate. If we remove all the rows with red flags, only 5 of the top ten spots were accurate. With all of these issues, we reverted back to the non-adjusted data where non-finishers and lapped drivers would be interspersed in the final results. There isn’t a way to measure the final 20 row accuracy without comparing actual results to the dataframe results for each race (103 races and 20 drivers would equate to a manual checking of 2060 rows). We then reduced the dimensionality from 32 columns to 16 and selected only lap numbers when pit stops were made, reducing the total number of rows from 110,124 to 3415. We also had to categorize, or codify, string and Boolean variables into the appropriate format.
32. Now that the data has been wrangled, we did some visualization in Shiny. First is the nicely colored one showing the frequency of pit stops by team at a specific course over the 5-year period. One thing we considered was seeing how different drivers and teams made pit stops. An issue that arose is teams such as Sauber and Renault are no longer active. Similarly, not all the drivers from 2018 are active in 2022.
33. Our next visualization is a histogram of tire life when pit stop was made. Pretty normal to right-skewed distribution. And here is the lap number when a pit stop is made – more right skewed. Generally speaking, tires last around 20 laps before a pit stop is made.
34. The next analysis is a histogram of lap numbers when a pit stop was made. This makes sense – right-skewed. There is a less likelihood of making a pit stop at the last lap of a race compared to the middle of a race. The main reason is time loss – a 20 second pit stop is not recoverable on the last lap of the race. It does appear someone took a pit stop on the last lap of the race with the 77 lap number having one count. This count was Monaco which has 78 laps and 161 miles. The Belgian grand prix is on the other side with 44 laps and is 191 miles. The massive influx at the first few lap numbers are the result of red flags and weather changes.
35. We then looked at the mean-variance for Saudi Arabia 2018 and all the courses over all the years. The mean variance assumption for Poisson regression was violated in both instances, although much worse in the full analysis.
36. Our next step was to create a pit stop distribution dataframe to look at the frequency of pit stops, tire life, and when they were made. What we see here is that driver #26 and driver #28 both took a first pit stop at lap 24. Driver 28 then took a second pit stop 4 laps later, whereas driver 26 did not take another pit stop the entire race. Knowing exactly why this happened in this instance is not within the scope of the study, although there can be several reasons why.
37. What we did with this data is visualize the actual and normalized number of pit stops taken per race. As you can see, well over 75% of the races have drivers taking only 1 or 2 pit stops. The IQR has all the pit stops at 1 or 2, the mean is 1.856 and the standard deviation is 1.07.
38. Time for analysis
39. First up was the Poisson regression. Sci\_kit had a value error due to array dimensionality. We found a way around that thru a package called statModels. One was a log-link function without offsets and the other a log-exposure function with offsets.
40. Since we had over-dispersion, we tried Consul’s Poisson, also known as Lagrangian Poisson Distribution. It’s essentially a generalization of the standard Poisson distribution and is referred to as GP1. We had some additional errors running this model, several variables had a maximum likelihood optimization failure to converge. Another variable had a divide by zero error, and another had a linear algebra singular matrix error.
41. Our next attempt was Famoye’s Poisson. It is a two-parameter discrete distribution which measures the location and dispersion. The distribution is unimodal and can be skewed but approaches normal distribution when the location parameter gets large. This seems to match the histogram of tire life when pit stop was made (slide 34). Every single variable resulted in a convergence variable.
42. With Poisson failing on multiple methods, we had two options to consider. A quasi-poisson process and a negative binomial regression. We started with negative binomial. There are two common negative binomial formulas and NB2 is the standard form of NB used to estimate over-dispersed Poisson data. This is typically the first model statisticians turn to when we find over-dispersed data per Joseph Hilbe who wrote a book only covering Negative Binomial regression. Hilbe was a fellow of the Royal Statistical Society and American Statistical Association and wrote books on Modeling Count Data as well as Predictive Analytics.
43. His book included an equation to estimate alpha. Alpha is the estimate of the dispersion parameter. To estimate alpha, we created a training and test set and added a lambda vector to the training set dataframe. After a few additional steps we were able to estimate alpha and lo and behold, the alpha value was negative. Alpha values should fall within 0 and 1 so something went awry.
44. This is how I felt. So with a deadline approaching, I wanted to show some data science-y things. What I did and what I am about to explain is not good statistics and would not be used in the real world but I wanted to exhibit understanding and knowledge
45. So I used the absolute value of the alpha value. I ran a t-test on it to determine it’s significance and then looked at its predictive power and then visualized the output.
46. Since this was bad statistics, I went to another model – random forest. I first looked at the random forest model on one particular race and received an accuracy score of 80%. Pretty stoked, there was a zero-division error for the classification report so the F-score was not available. Here is the breakdown of the variable importance as determined by the random forest. With some momentum I then applied the random forest to the full dataset and received an accuracy score of 42.6%
47. Now it’s time to throw some paint at the wall. First up was Quasi-Poisson. There was no model or package in python to run a quasi-poisson process. My first step was to backdoor R into Python by transforming the dataframe into an R dataframe and running R script on it. We were able to get it into R but “glm” was not recognized as an R command. At this point, we simply wrote the dataframe into a csv and accessed it via RStudio. However, that required some data cleaning replication such as making the driver name a factor instead of a string. After cleaning the data, we re-ran normal poisson with the AIC of 22343 and sandwich covariance estimator which can be used to estimate the variance when the underlying model is incorrect. The result was an analysis for every single race course and significant p-values on pretty much every single variable. Further cleaning would be needed to better analyze the results here. We then tried the quasi-poisson which had a non-applicable AIC. We ran a negative binomial as well with an AIC of 21984 – so slightly better than the normal Poisson. As part of this research I found several other tests that were attempted such as Generalized Estimated Equations which failed due to the model matrix being rank deficient. The Flex mix package did not work due to log likelihood error. The NLME package required a Gaussian distribution.
48. So finally, we moved to Bayesian Poisson analysis. The first step was to run a simulation for the normal model using the arguments provided in the book found online. It was written by two professors and a practicing data scientist. We followed the steps outlined by the authors using their code and fitting it to our model. This was done to make sure that the information put in the code was correct and could be fixed and updated at a later date. Some issues with following their exact code was using the family = gaussian argument. Poisson models do not have several auxiliary parameters that were used in the Bayesian set up, such as prior\_aux “auxiliary parameter, e.g. error SD (interpretation depends on the GLM).” Even following their code pretty closely there were several errors. First was sampling errors where the Bayesian Fraction of Missing Information was low. This was due to using only 2 chains and 500 iterations. We updated it to 4 chains and 1000 iterations and had Bulk Effective Sample Size was too low, indicating the posterior means and medians may be unreliable in addition to posterior variances and tail quantiles being unreliable due to Tail Effective Sample size being too low as well. We upped the iterations again and still came away with errors and an r-hat of 4.22. We continued as if we received a working or useful model so theoretically, this is how we would proceed. Ideally we would review the posterior simulation by looking at the MCMC trace, density, and autocorrelation plots. We could run another posterior predictive checks where the newly created histograms follow the actual histogram. We could run a confidence interval for the variables or filter variables down to driver or team.
49. Questions?