World Cubing Association Data Analysis

Drew Shapiro

Colab link:
https://colab.research.google.com/drive/1Kce50MFo-IyPXCLWML3UGfdyzipYbUbd#scrollTo=GaS125qMlaCU

Research question

What factors affect or predict rubik's cube competition solve time?

Some specific questions I considered were:

- How does competitor country-of-residence impact cubing trends?
- How does the number of competitions attended affect a person's solve time?

Why I find this question interesting:
Answering this question can help provide insight into what factors most affect someone's solve times (potentially leading them to optimize their behavior).

The data

I will be analysing data from the World Cubing Association (WCA) database. The World Cubing Association, founded in 2004, has been holding competitions for twisty puzzles (namely the rubik's cube) since 2006. They have a database that keeps records of competitive solves that holds lots of pertinent data, such as competition ID, competitor name, each of their competition solve times, their average solve time in the competition, the exact scramble for each solve, and more. The data for every competitor throughout history can be found on their website:
https://statistics.worldcubeassociation.org/database-query

The entire database can be downloaded locally and takes up about 2gb of storage space. Several different files are part of this database. After my initial survey of the data, the most important file seemed to be the 'Results' file, which holds data from all competitors in all competitions, with one entry corresponding to one competitor's performance in a given competition. Another file called 'Competitions' holds the data for each competition, including its exact location, venue, date, etc. This is important because it's necessary in order to know what the exact date was for a given data entry in the 'Results' file. Data files:
https://drive.google.com/file/d/1ja996qL_jmm_1hdeTLuDLMRpIStguzFc/view?usp=drive_link
https://github.com/DrewShapiro5/WCA_Data_Analysis/tree/main/WCA_export

How does the data relate to my question?
The data set holds data from every competition, including date, location and time. This relates to the question because my question is in terms of location, date and other trends that are part of the data set.

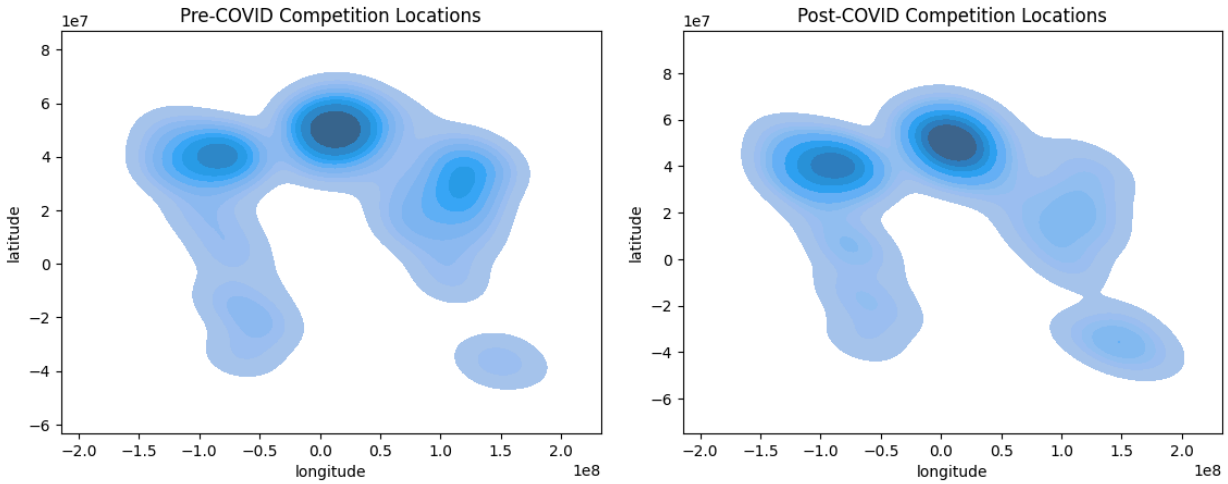
Visualization

To gain more insight into my research question, I performed a series of visualizations of the dataset. For data manipulation I used python's pandas library, and for data visualization I used matplotlib and seaborn.

Competition location changes over time

My first idea was to visualize how competition locations have changed over time. I decided to use COVID as a metric for this, visualizing competition location pre and post-covid. The 'Competitions' dataset (which I imported and named *comp_dates_df*) contained all competitions ever held and their exact geographical location. I decided to plot this using a 2D heatmap.
In order to add weight to each competition based on # of competitors, I had to make a new column using data from the 'Results' dataset (which I imported and named *cube_events_df*).

```
comp_dates_df['competitors'] =
comp_dates_df['id'].map(cube_events_df['competitionId'].value_counts())
```

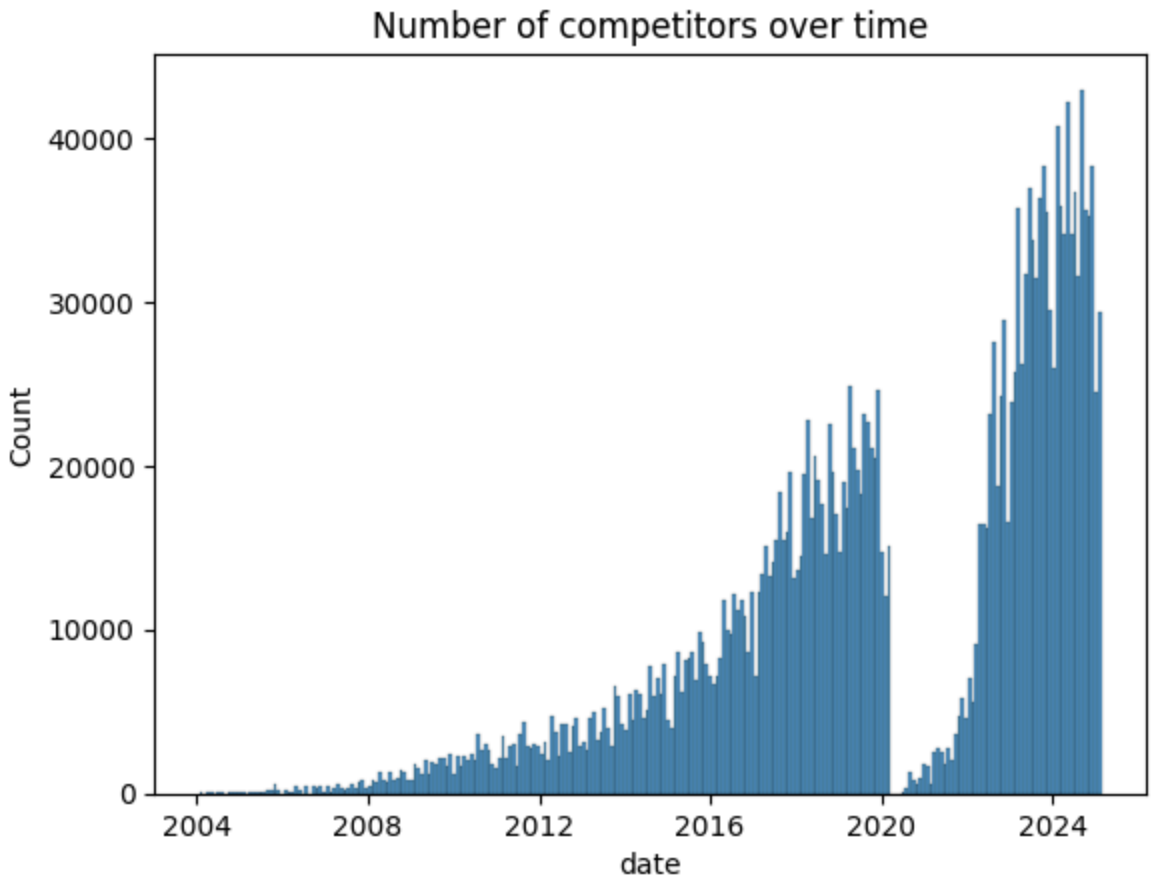


There doesn't appear to be much of a meaningful change between the two maps (except for some southward shift in the eastern hemisphere). I chose not to further pursue this line of inquiry because of this, and because it's not necessarily related to the question of solving time.

Number of competitors over time

Next, I wanted to visualize how the number of competitors changed over time. To do this, I needed to add a date to each entry in *cube_events_df*. I did this with a similar use of the map function (code shown below), taking the date from *comp_dates_df*. With this, a simple histogram can be made illustrating the growth of WCA competitions over time.

```
cube_events_df['date'] =
cube_events_df['competitionId'].map(comp_dates_df.set_index('id')['datetim
e'])
```



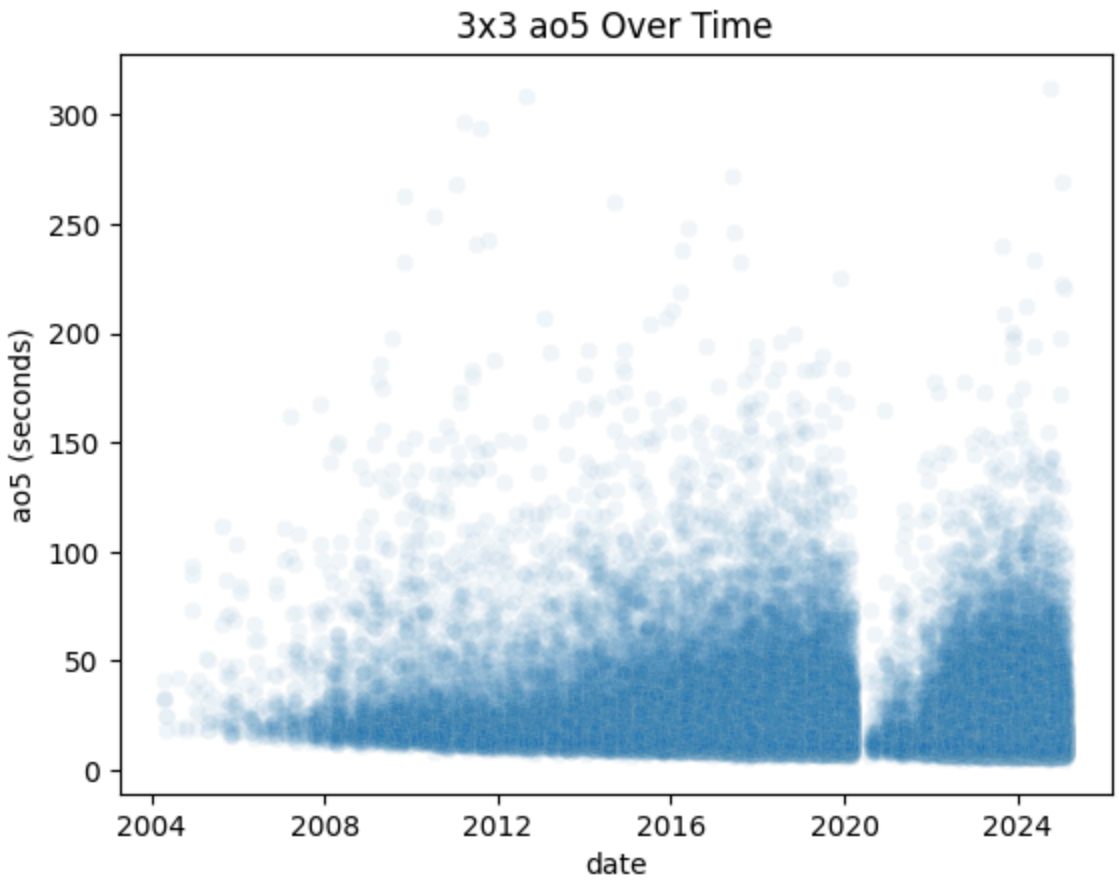
Note: each bin represents one month.

We can see that although there is a sharp upward trend, there was a break when COVID struck that dropped the competitor count down to 0.

3x3 ao5 over time

Important note: From now on, I will refer to a competitor's solve time in terms of ao5, defined by wikicube as: The average time of 5 solves in WCA competitions, calculated by removing the fastest and slowest solves and averaging the remaining three.

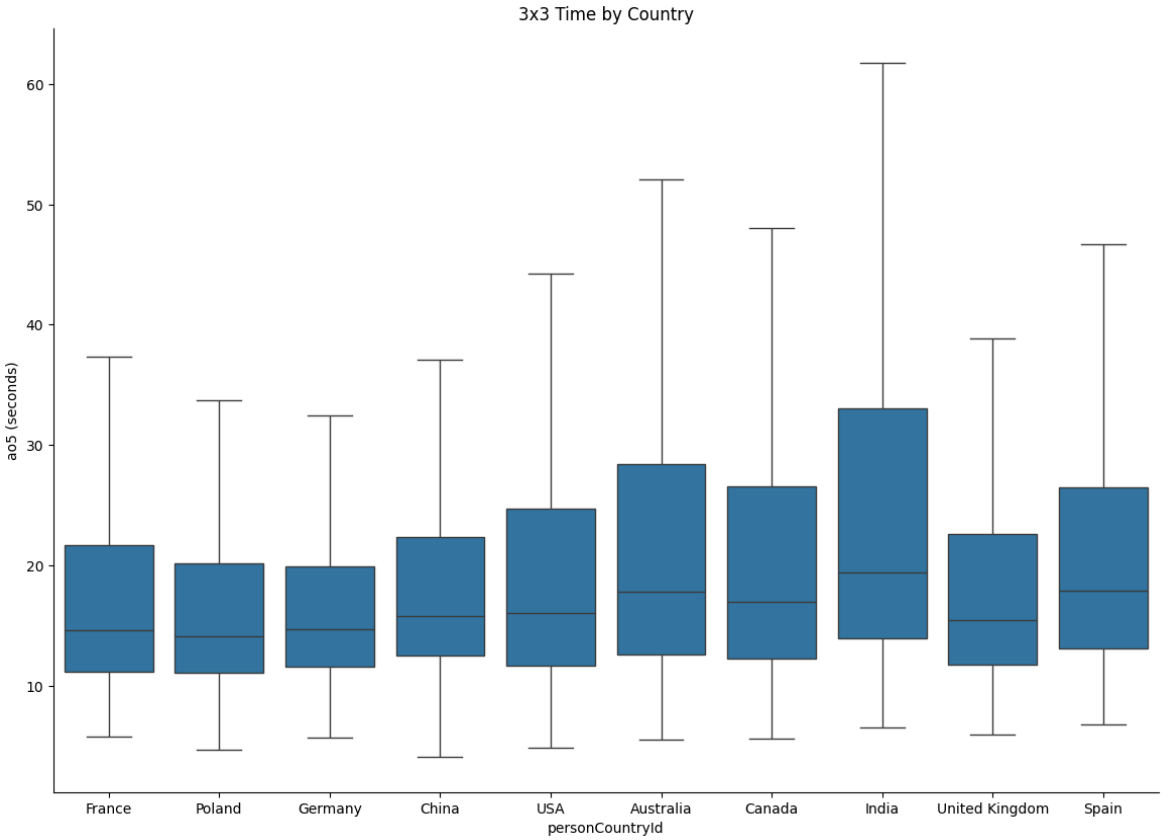
Rubik's cube ao5 over time can be plotted with a simple scatterplot, with each point representing one competitor at any competition. Since there are so many competitors, it's necessary to take a sample from the dataframe so that generating the plot happens in a reasonable amount of time. The following plot is generated from a random sample of 100,000 competitors.

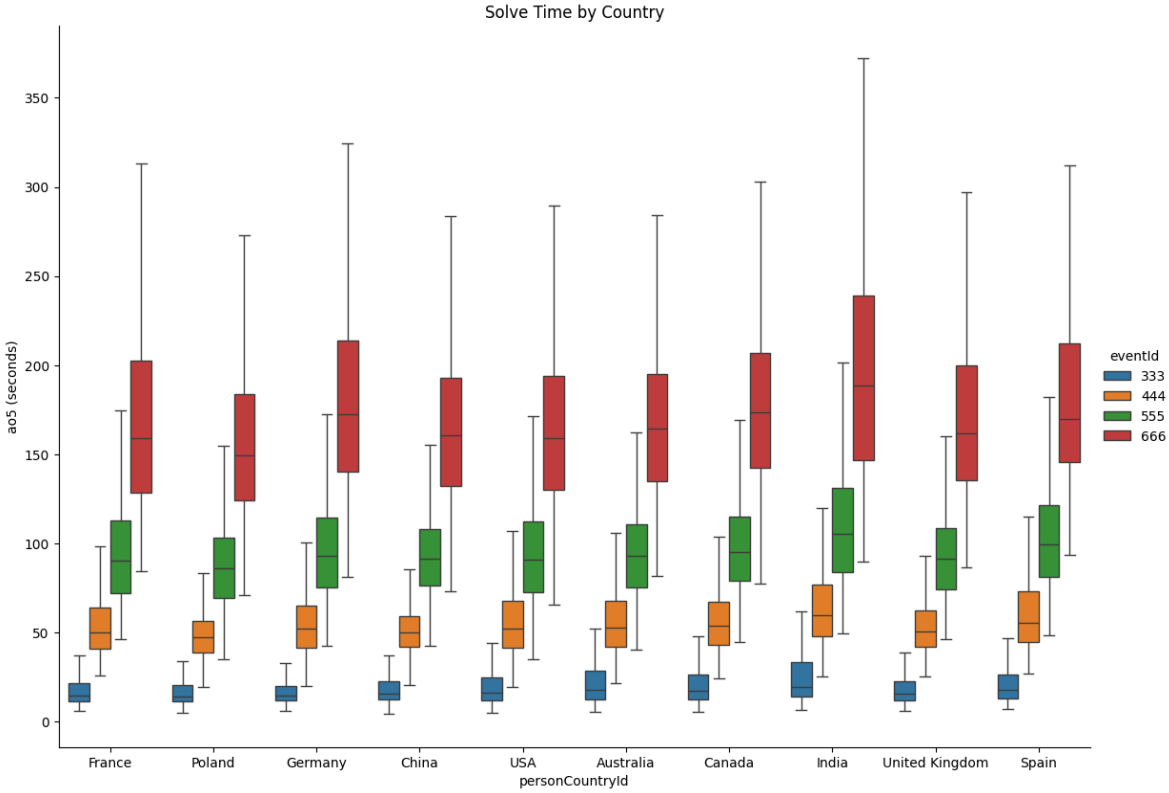


We can see that over time, the lowest ao5's drop closer and closer to 0.

Solve time by country

I wanted to visualize the difference in solve times between countries. To do this, I opted for a side-by-side box and whisker plot to visualize an overview of their distributions. I also decided to make two plots, one with only 3x3 times shown and the other with multiple events. I only plotted the 10 countries with the most competitors for visual clarity.





Note: personCountryId refers to the country that the competitor resides in.

Plotting the distribution of solve times by country gave me insight into the fact that there are clear differences between the distributions of ao5 times for each country.

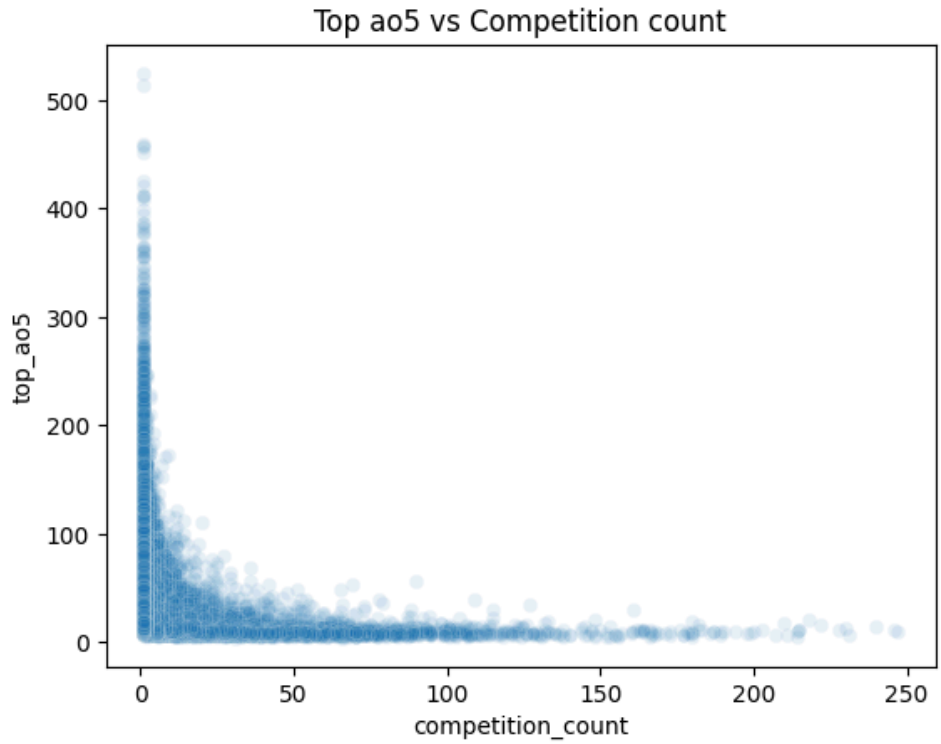
Top ao5 vs. number of 3x3 competitions

Next, I wanted to model how competition count affects a person's solve time. I thought that analyzing a competitor's top ao5 time would be best, because it best represents their growth as a solver over time, rather than their average ao5, which might be held back by their older solves. I needed to create a new dataframe with each competitor, their top average and their total number of competitions attended. I did this with the groupby function:

```
cube_3x3_solvers =
cube_events_completesolves_df[cube_events_completesolves_df['eventId'] ==
'333'].groupby('personId').agg(
    top_ao5=('average', 'min'),  # min because lower times are better
    competition_count=('competitionId', 'nunique')  # count unique
competitions
).reset_index()

cube_3x3_solvers = cube_3x3_solvers.set_index('personId')
```

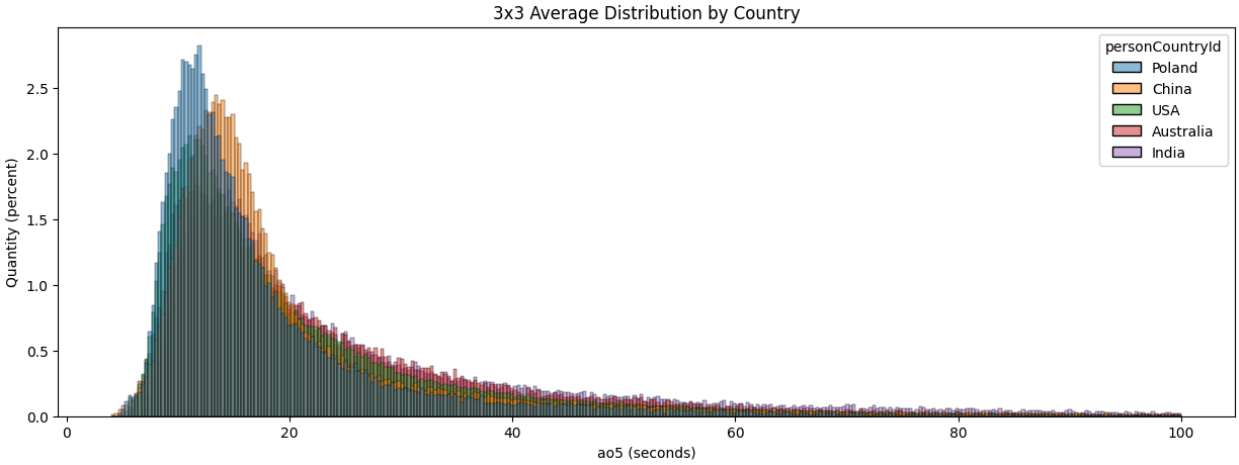
Making a scatterplot of this new dataframe shows the distribution of *top ao5* vs. *competition count*, where each point is a single competitor. We can see that there does appear to be a trend, but it's hard to read.



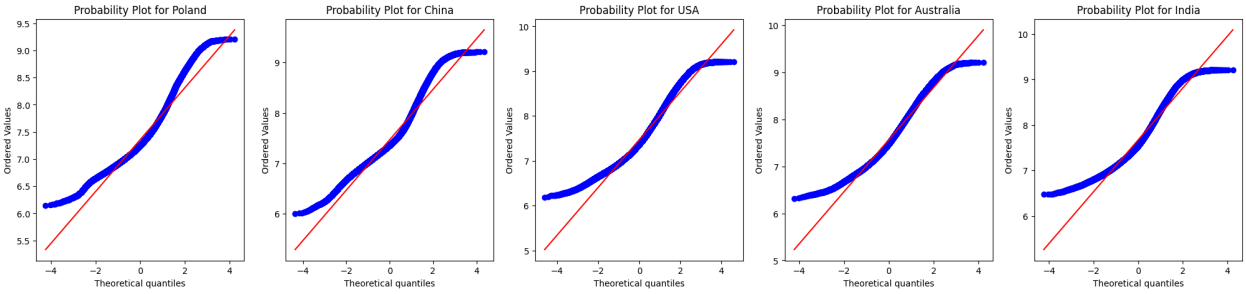
In order to model this trend, further data wrangling is required. I will return to this problem later.

Testing for significance between ao5 and Country

The goal was to perform a statistical test to see if a competitor's country has a significant impact on a ao5. I decided to use ANOVA to test this, because the independent variable is categorical and the dependent variable is numerical. To comply with the assumptions of an ANOVA test, I first needed to see if the distribution of ao5 was normal, so I plotted it (separated by country):



The distributions are clearly skewed right, though they do have a nice curve shape. To normalize the data, I applied a log function to 'ao5', creating 'log_average'. A check for normality is included below.



The plots below above the normality of the distribution AFTER log has been applied. The blue points are close to the line for the most part, so I decided to proceed with the ANOVA test since the distributions were approximately normal.

The null hypothesis for this test is:
There is no significant difference in ao5 between countries.

The following code produces the results of the ANOVA test:

```
m1 = smf.ols('log_average ~ C(personCountryId)',
data=top_countries_3x3_df).fit()
anova_table = sm.stats.anova_lm(m1, typ=2)
anova_table
```

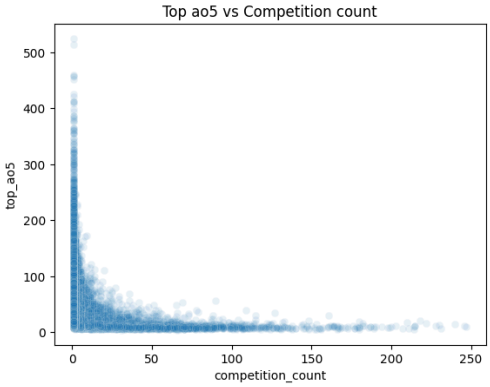
The results of the ANOVA test are:

	sum_sq	df	F	PR(>F)
C(personCountryId)	3733.506560	4.0	3207.524684	0.0

The test yielded a large F statistic, indicating a strong correlation. The p-value for this test was 0, giving us the confidence to reject the null hypothesis.

Modeling top ao5 against competition count

Recall the plot of competitors' top ao5 vs. their competition count:



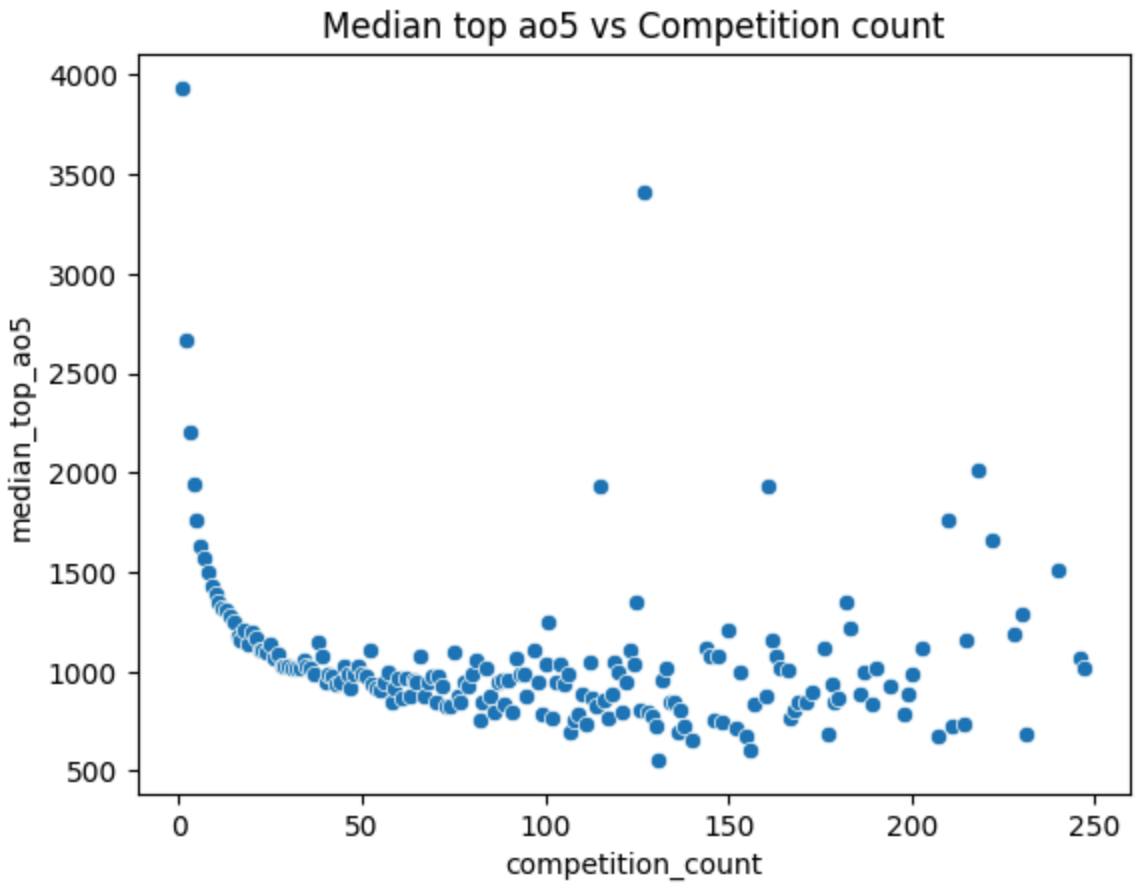
This plot shows a clear curve, but I knew it would be hard to model. The plot is completely solid and filled in below the main curve, and it's hard to interpret. I needed to find a way to further manipulate the data to get a clearer trend.

My solution was to take the median of all top ao5's for each value of competition count, creating a new dataframe where one entry corresponds to one value of competition count and its corresponding median top ao5.

```
median_time_for_compcount =
cube_3x3_solvers.groupby('competition_count').agg(
   median_top_ao5=('top_ao5', 'median'),
).reset_index()
```

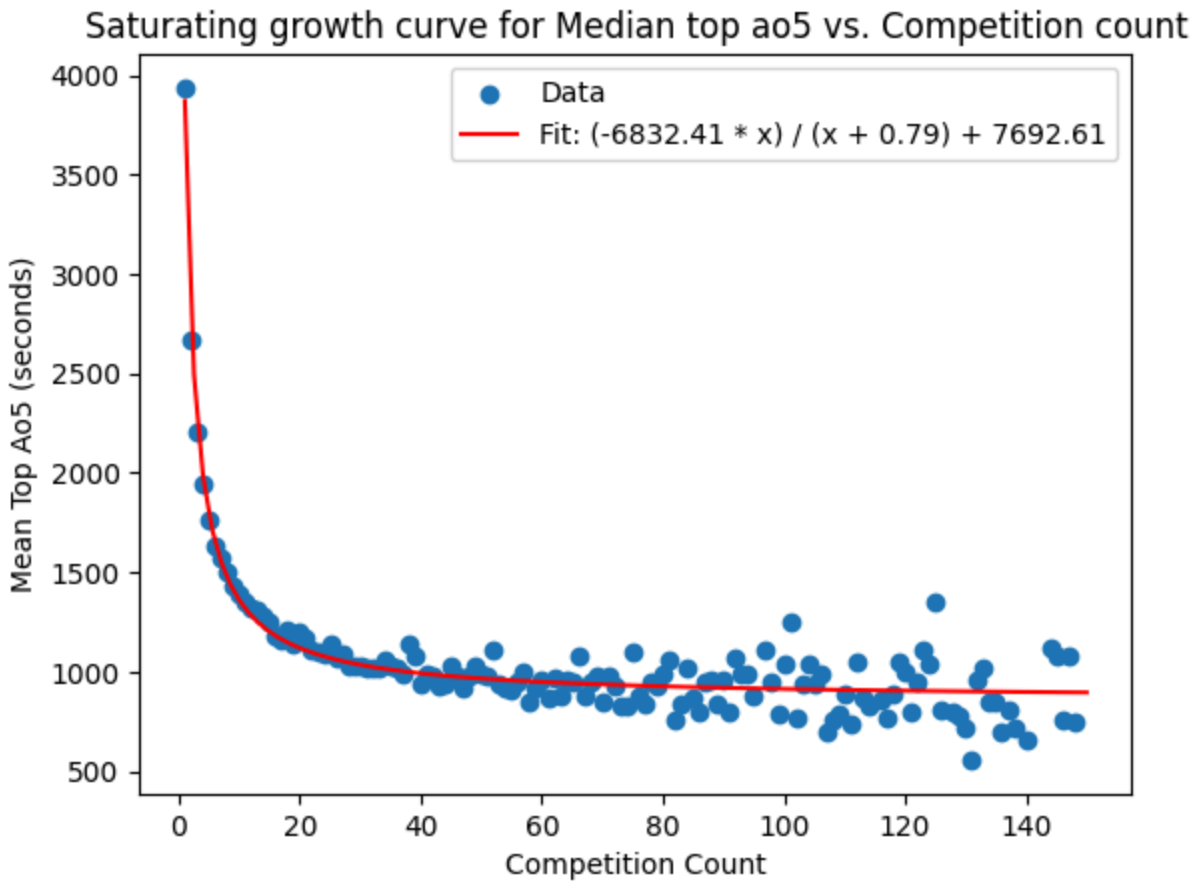
Plotting this new dataframe yielded the following distribution:

Note: ao5 is depicted in terms of hundredths of a second here. I did this because it worked better with the function fitting algorithm.

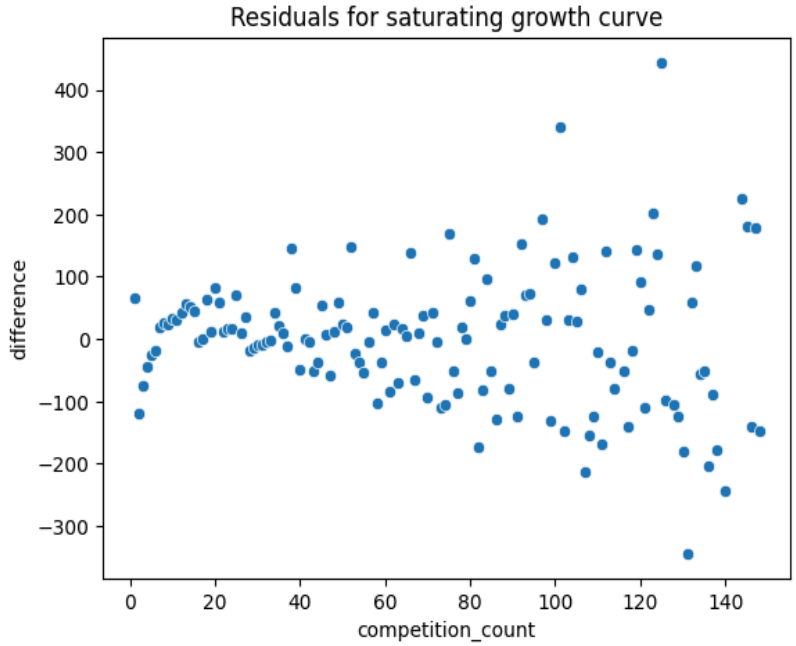


This shows a much more clear trend that resembles some type of decaying function. Since the distribution becomes very noisy towards higher values of competition_count, I chose to cut it off at a value of 150 for the sake of modeling. I believe that the reason for the noise is that there are less competitors with a large number of competitions behind them, which leads to more variation for higher values of competition_count.

At first glance the curve looked a lot like 1/x, but I've realized after further analysis that it most closely resembles a saturating growth curve of the form (a*x)/(b+x)+c. I used the curve_fit library from scipy to fit this curve:

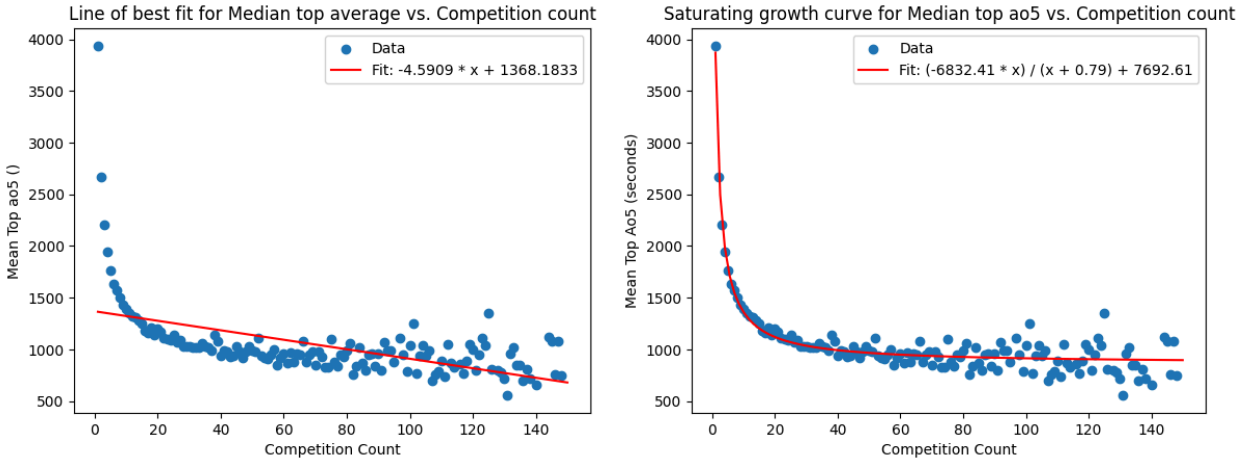


Visually, it seems like this curve is a good fit for this distribution. We can look at the residual plot to see if there appears to be any left-over residual curve:



'Difference' represents the difference between the predicted and true values for 'median top ao5'. There is a bit of a rise at the beginning of this plot, but for the most part there doesn't appear to be any left-over curves. The points spread out as 'competition_count' increases, but this is for the same reasons previously mentioned.

I decided to compare this model with a regular linear regression model. Here is a visual depiction of the two models side-by-side:



Visually, the saturating growth curve seems like a much better fit, but I wanted to quantify this difference. To do this, we can compare the R-squared values for each model, which represents the total amount of variation that can be accounted for by the model. For the linear regression model, the R-squared value was 0.282, which is not very high. I had to manually calculate the R-squared value for the growth curve (code below), and it came out to be 0.912, which is very high. Based on this metric, my optimized curve is a much better fit for this distribution.

```
y_predicted = x_data.apply(f)
ss_res = np.sum((y_data - y_predicted) ** 2) # Residual sum of squares
ss_tot = np.sum((y_data - np.mean(y_predicted)) ** 2) # Total sum of
squares
r_squared = 1 - (ss_res / ss_tot)
```

Conclusions

There are two significant conclusions that can be drawn from this analysis:

- There are significant differences in competitor ao5 between countries.
- Number of competitions attended has a negative impact on top average solve time (more competitions attended corresponds with faster times). I was able to model this distribution to much success. It would be reasonable to assume a causal relationship, i.e. attending more competitions leads to a faster top competitive ao5.

There are limitations to my analysis. The dataset itself is limited because it only holds competition data, and lots of solves happen outside of competitions. Competition trends may not reflect overall cubing trends, especially between different countries. While we can conclude that ao5 is different between countries, we can't conclude that the average solver is actually different between countries, because we don't have access to personal solve data.

I wasn't able to determine any casual relationship between country and ao5, which is something I could do in the future if I were to continue this project. However, for the reasons I mentioned, I'm not sure this would be valuable information.

Another limitation is in my analysis of competition count. There was a lot of data wrangling involved with turning the dataset into something useful. Fitting such an abstract set of data may not be totally indicative of reality. While it seems like higher competition count directly leads to lower ao5 times, it's worth noting that the curve was fit on points that represent the median value of many different solvers. The curve doesn't account for higher values of competition count very well because there is a lot of variation between ao5 of individual competitors.