PV251 Project Report

Tamara Babálová 485550

Motivation

I chose this topic because I like gaming and spent a lot of time playing League of Legends as well. While browsing relevant games datasets I found an interesting stats dataset for the whole last season, so I decided to work with that.

Data sources + Data preprocessing

I used the latest version of this dataset: https://www.kaggle.com/datasets/vivovinco/league-of-legends-stats-s13. The author also has an additional dataset for season 12 which would be relevant to use if I picked a different sheet to visualize but for the one I chose, this single dataset was sufficient.

The dataset consists of 12 CSV files, each file for one patch. It doesn't contain all patches from season 13, but more than half of them.

I started parsing all of the CSVs, preprocessing their attributes, such as stripping '%' from numerical values and adding a 'Patch' column. After that, I joined all the files into one data frame and sorted them by patches. I had to make a custom sort, so the patch 13.10 will go after 13.9, not 13.1, etc.

After that, I created another data frame, which had additional columns for averages for the original columns and each champion a distinct color. It was a bit problematic with the colors since I had 245 champions and many of them look similar, but when we're displaying a reasonable amount of champions in the graph, it's not as bad.

Explanation of the design choices

I chose to implement my #2 sheet. Initially, I wanted to implement a stream graph but unfortunately plotly doesn't support that. However, I decided to go with a line chart and edit it so it conveys the same idea as I sketched.

The main line graph consists of patches on the x-axis and user-selected attributes on the y-axis. This can be either a pick, win, or ban rate. So it shows the trend of selected rate over the patches of the season for selected champions. The champions can be categorized and filtered by either role or class (or just all). There's another dimension added to the graph by being able to choose a secondary rate attribute (same options as primary) and this affects the thickness of the lines and size of markers. I normalized the secondary attribute values to interval 0.0-1.0 so the difference in thickness is more noticeable but often they look very similar, this is due to the nature of the data.

On top of the graph for each patch there's the name of the champion with the highest primary rating for the patch. This stays even when selecting other champions or deselecting the top champions so we can easily compare.

When loading the graph initially, there are always 10 selected champions first and the user can use the slider to add or lower the number of champions shown. Users can also directly click on the legend to add specific champions by name. The champions for showing are sorted based on the primary attribute mean from the whole season (so for pick rate it's the champions that were picked the most throughout the whole season). I chose not to include a slider for amount of patches, since I only have 12 patches anyway and the slider wouldn't be very functional.

For changes to be applied (menu selection/slider) user has to click on the update button.

When clicking on a line/marker in the graph, it instantly highlights to full opacity and creates a border around the selected marker. It also shows a more detailed tooltip on the side with specific numbers for the selected attributes and patches. It also automatically updates the champion analysis part of the visualization.

The champion analysis part consists of 4 subplots containing smaller pie charts for each patch and a bigger one for the overall season stats. This part is focused on a single champion only and their pick, win, and ban rate and the roles they played during the year. For the roles part, it also shows the champion's tier for corresponding role for each patch and average tier in the last chart.

Users can also manually search for any champion they want to be shown in this section.

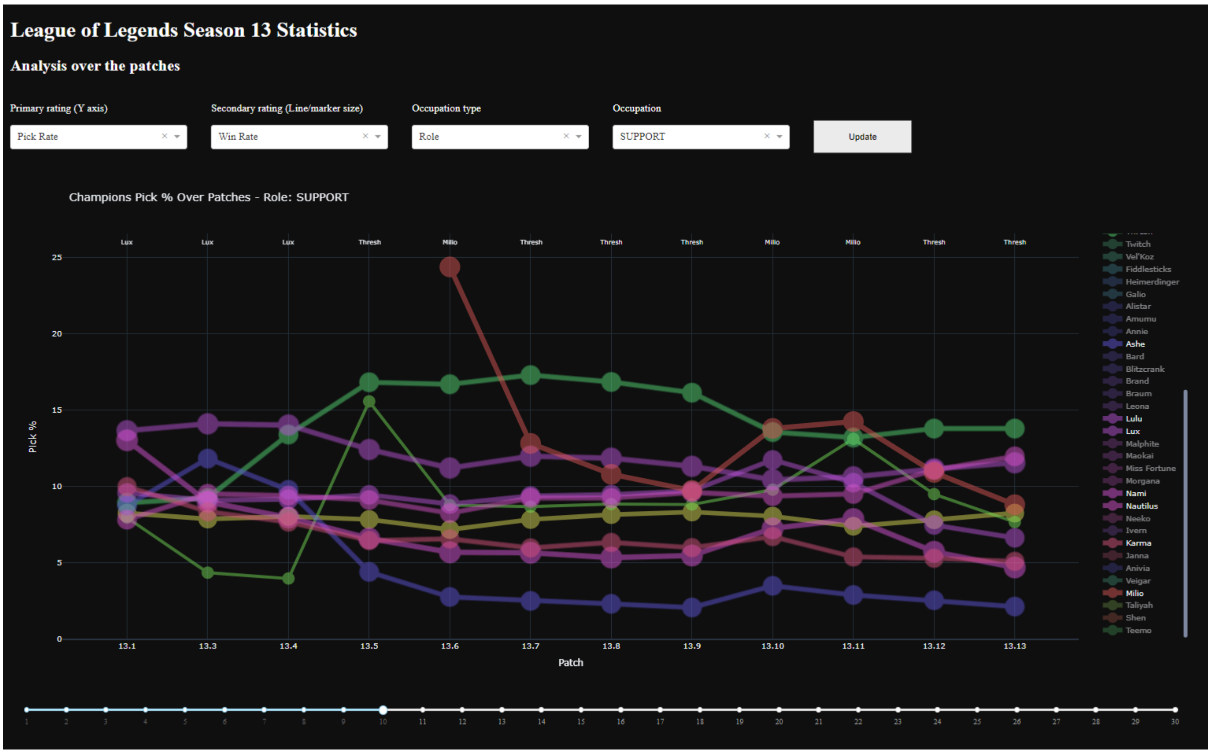
Interesting observations in your visualization

I found it interesting that most of the data is very similar to each other and there are very small differences for champions between the patches. However, it makes it all more interesting to see outliers and they are also more noticeable.

I also noticed that sometimes the top-rated champions for Patch are not top-rated in the first 10 overall throughout the year. I was wondering how to tackle this issue and upon initial load of the graph, I added the trace for any champions that got the highest spot for any patch along with the first 10 (or any number of champions chosen).

Screenshots

1. Main graph displaying the analysis over patches (dropdown menus to customize at the top and slider at the bottom)



League of Legends Season 13 Statistics

Analysis over the patches

Primary rating (Y axis): Pick Rate

Secondary rating (Line/marker size): Win Rate

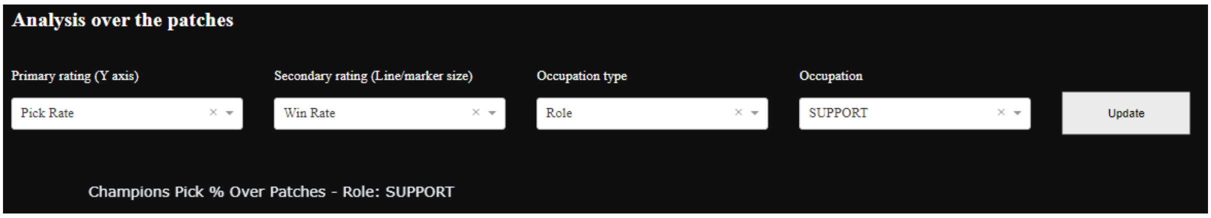
Occupation type: Role

Occupation: SUPPORT

Update

Champions Pick % Over Patches - Role: SUPPORT

2. Dropdown menus detail



Analysis over the patches

Primary rating (Y axis): Pick Rate

Secondary rating (Line/marker size): Win Rate

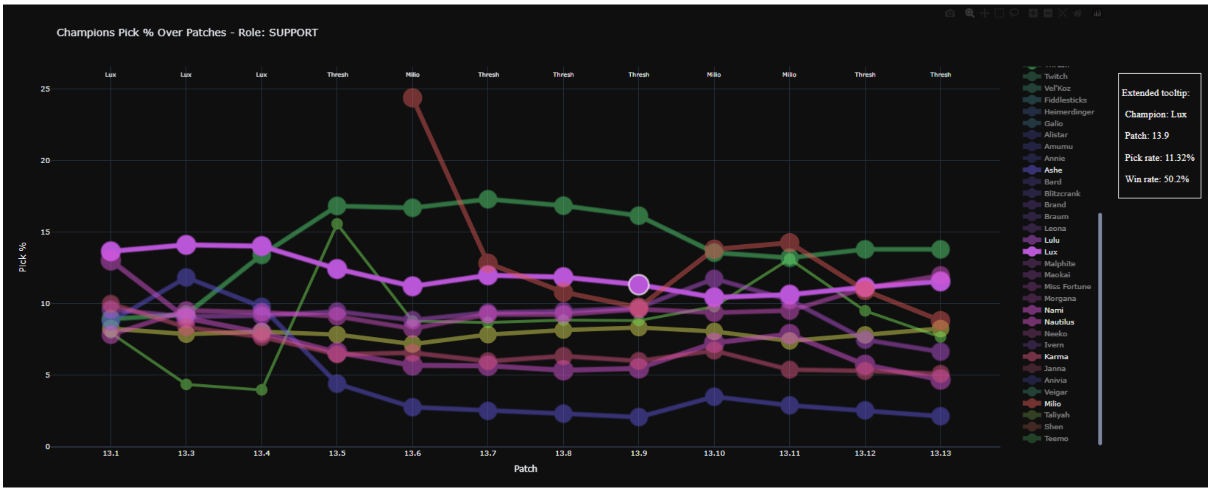
Occupation type: Role

Occupation: SUPPORT

Update

Champions Pick % Over Patches - Role: SUPPORT

3. Main graph after clicking on a trace marker (I clicked on Lux for patch 13.9)



Champions Pick % Over Patches - Role: SUPPORT

Extended tooltip:

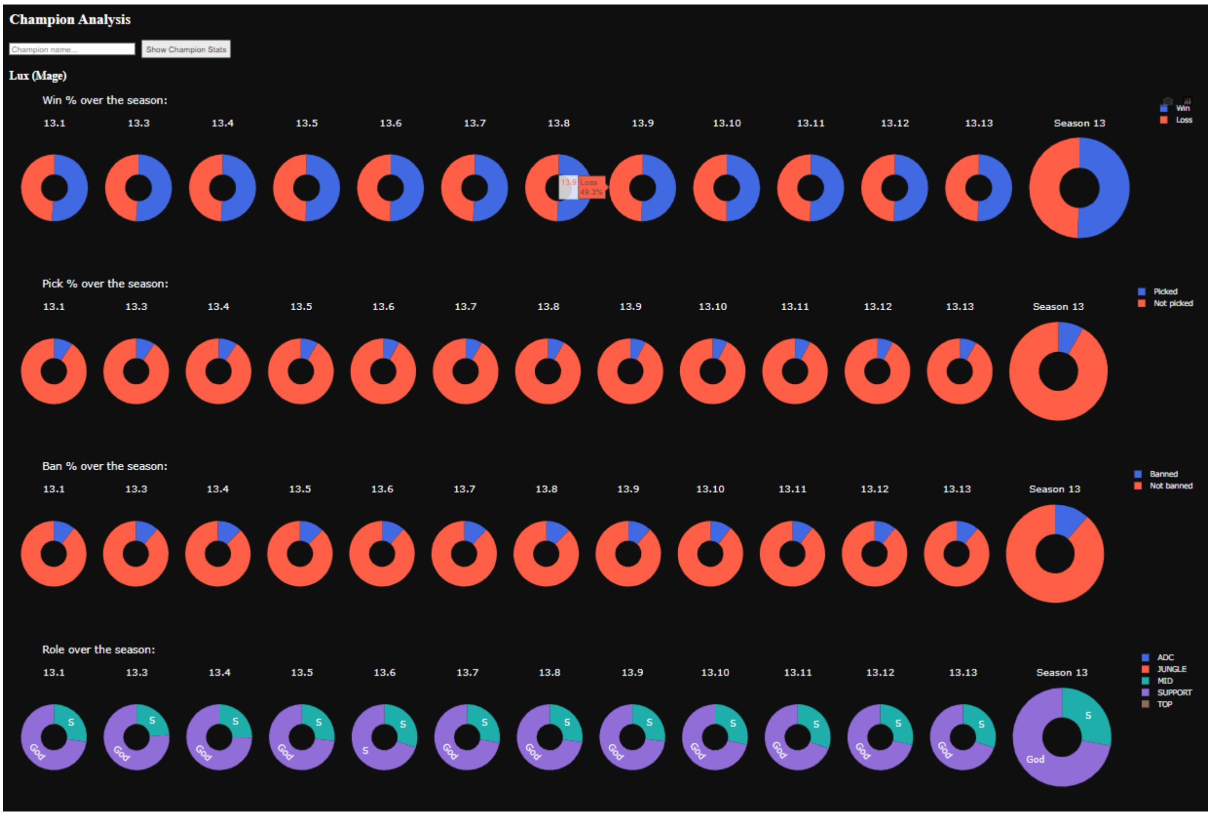
Champion: Lux

Patch: 13.9

Pick rate: 11.32%

Win rate: 50.2%

4. Clicking also triggers displaying second section of visualization, the champion analysis



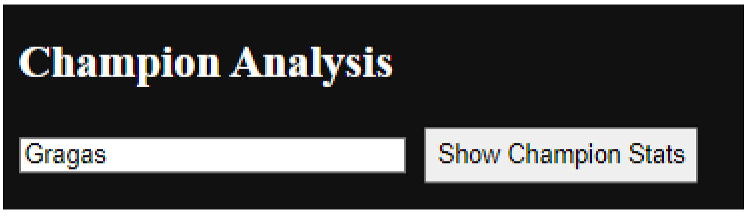
Champion Analysis

Champion name...

Show Champion Stats

Lux (Mage)

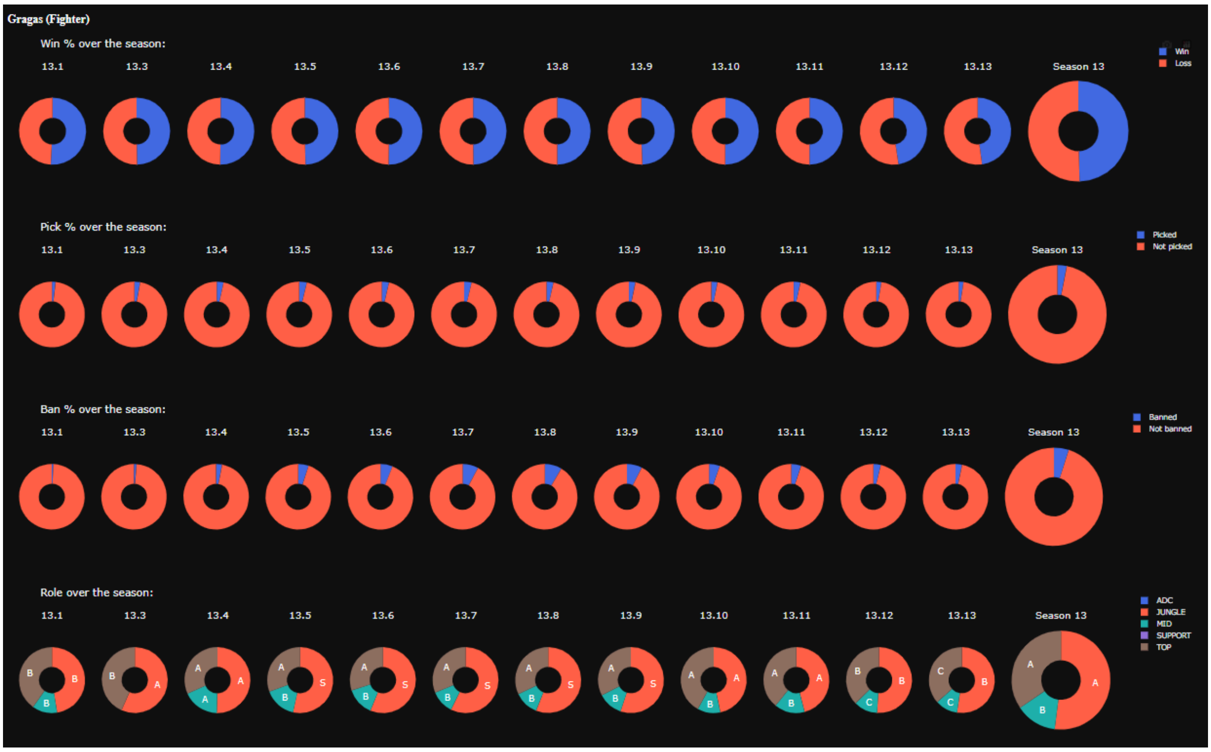
5. We can also search for any other existing champion to see their statistics



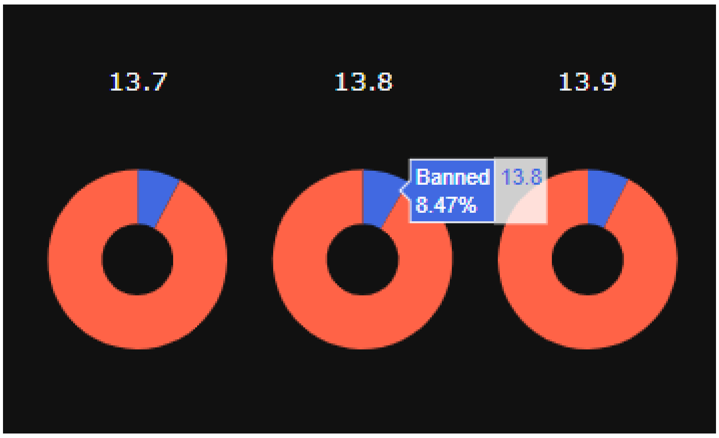
Champion Analysis

Gragas

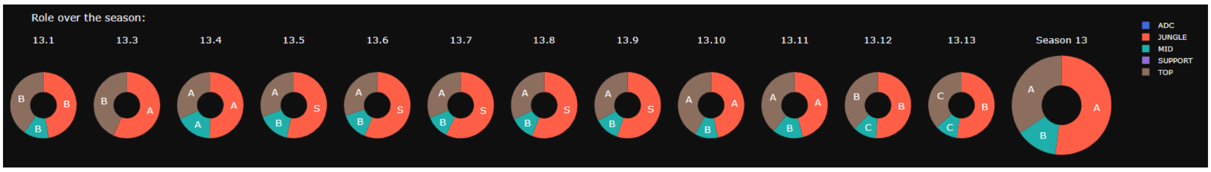
Show Champion Stats

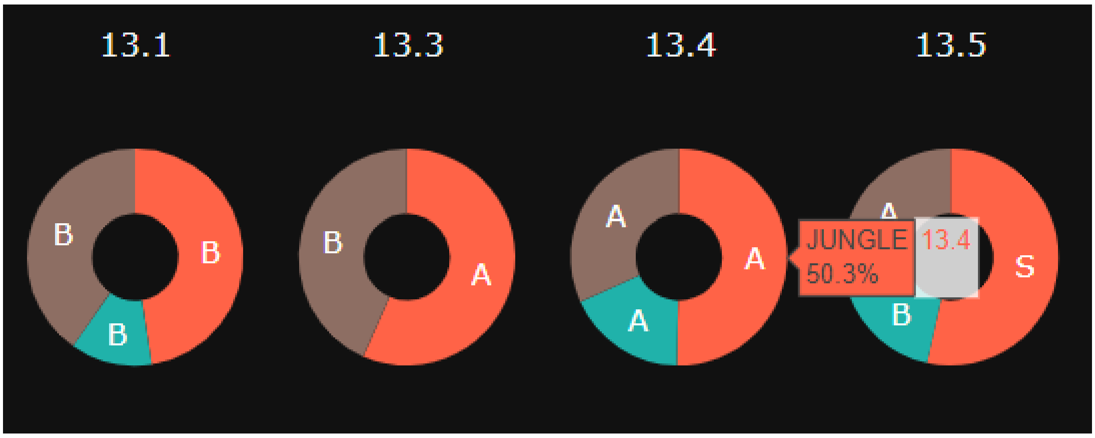


6. I hid the percentage labels, because they were distracting in my opinion, since the ratios are usually very similar, only differing in few percentages at most, but we can still see them hovering over desired the chart part

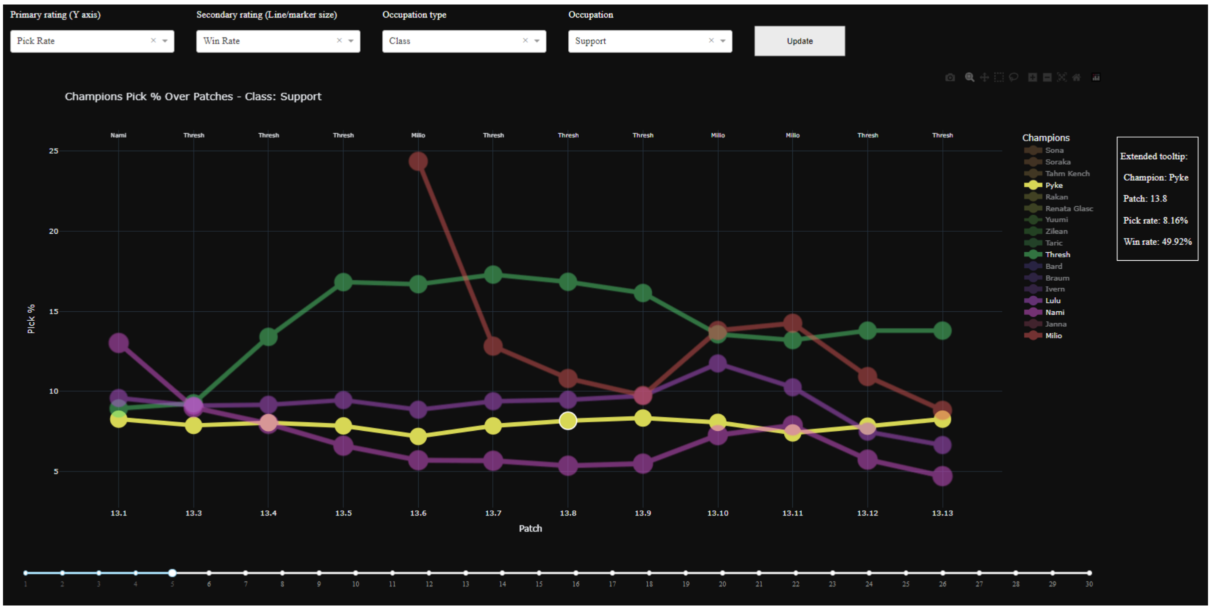


7. For the last set of charts displaying the Role, we have the Tiers for corresponding Role/Patch as well as labels





8. More examples of the main graph using the slider options



Primary rating (Y axis): Pick Rate

Secondary rating (Line/marker size): Win Rate

Occupation type: Class

Occupation: Support

Update

Champions Pick % Over Patches - Class: Support

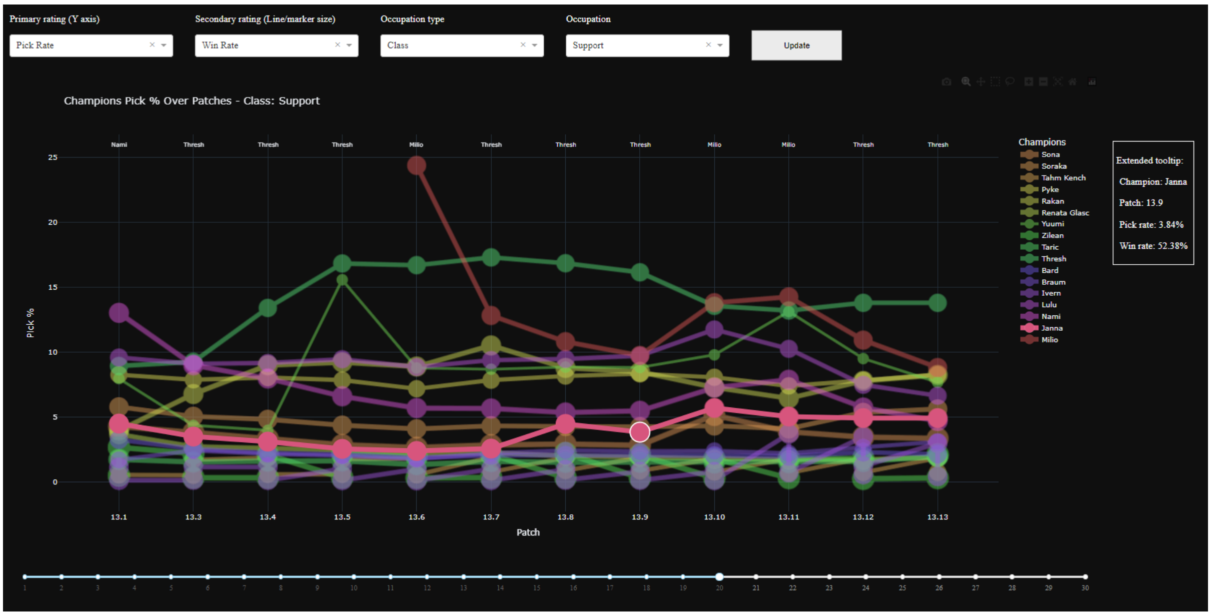
Extended tooltip:

Champion: Pyke

Patch: 13.8

Pick rate: 8.16%

Win rate: 49.92%



Primary rating (Y axis): Pick Rate

Secondary rating (Line/marker size): Win Rate

Occupation type: Class

Occupation: Support

Update

Champions Pick % Over Patches - Class: Support

Extended tooltip:

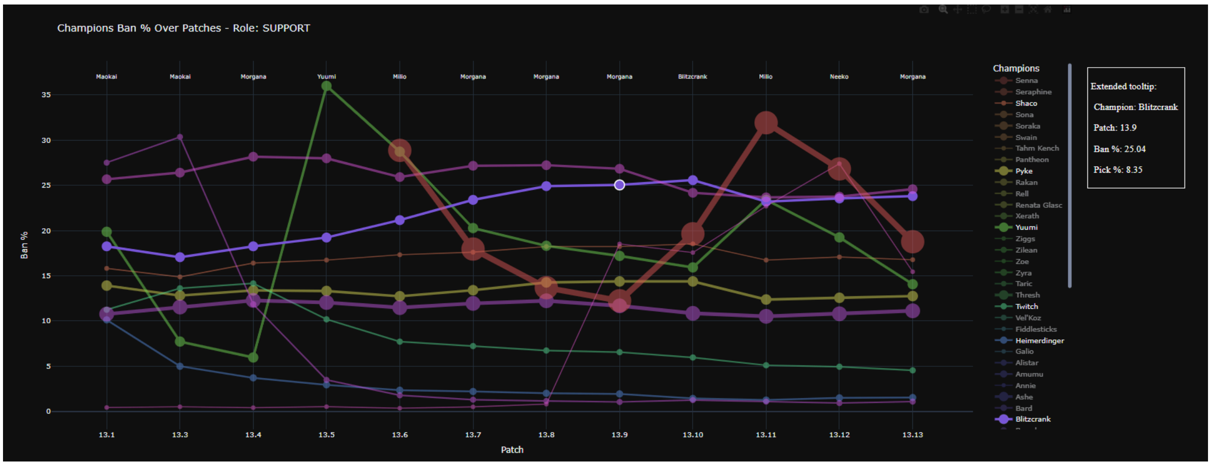
Champion: Janna

Patch: 13.9

Pick rate: 3.84%

Win rate: 52.38%

9. Ban rate over pick rate graph for supports role example



Champions Ban % Over Patches - Role: SUPPORT

Extended tooltip:

Champion: Blitzcrank

Patch: 13.9

Ban %: 25.04

Pick %: 8.35

Used technologies

I used Python, specifically, Jupyter lab and notebook we worked with in the seminars, and in there mainly the plotly and pandas libraries. The application is made with Dash.

Lessons learned

Regarding skills I mostly learned how to use plotly and dash effectively. I worked with pandas and google colaboratory before, which isn't too different from jupyter notebook, so I was already familiar with that.

I also learned a lot about visualization. I was creating the sheets and planning the designs not taking into consideration the technology I would be using and the difficulty level and in the end I had to change it up a little. I also had to think about all of the elements I designed initially and it they really convey the information efficiently and clearly and if I want to keep them (I ditched the patches slider for example). When it came to the stream graph I initially drew, I had to think about ways to mimic displaying multidimensional data in a single linechart, but it worked out in the end.