**GameScraper**

Statistical Analysis and Automation of League of Legends Games

# Summary

## Summary

Many *League of Legends* players love to try new things and get better with their skills. Esports companies are wanting the most up-to-date information to analyze their team members performance to help win games or better their team’s chances in the different leagues across the world. There are many places to get information, but without automation it could take a lot of wasted time to pull the data and analyze them quickly.

GameScraper is a python application that let’s the end user input their *League of Legends* summoner’s name and opens an application to show statistical visualizations based on their recent play history. These statistics are based off the summoner’s Kills/Deaths/Assists and contain three different tabs: KDA Trends, KDA Scatter, and Champ Stats.

## Background

GameScraper is a culmination of the learnings throughout my first semester at the University of Arkansas and a desire to learn programming concepts to bring into the corporate world. The app brings together screen scraping tools and statistical analytics to bring the end user information they can use to better understand their performance.

# Requirements & Design

## Package Requirements

##### **Python v3 – 3.7 preferred**

* Files – GameScraper.py & main.py
  + Built into GameScraper folder – CSV file will create inside same folder

##### **Python Standard Library**

* tkinter

##### **Third-Party Packages**

* **Matplotlib -** <https://pypi.org/project/matplotlib/> **- pip install matplotlib**
* **Beautiful Soup -** <https://pypi.org/project/beautifulsoup4/> **- pip install bs4**
* **Selenium -** <https://pypi.org/project/selenium/> **- pip install selenium**
* **Pandas -** <https://pypi.org/project/pandas/> **- pip install pandas**

**ChromeDriver – contained in GameScraper folder**

**Install all packages necessary above before opening GameScraper.py**

## **Parser Logical Design**

# Instructions

# Current Progress and Opportunities

Results work as expected, though very basic and could have many improvements (stated below). The file depends on the user’s computer specifications for how fast it will work, but does run and provide the details I worked to achieve.

There are a few opportunities to improve the program. Chrome is a good browser, but to speed up the scraping on *op.gg*, a headless browser base would be a great option for computers without the processing speed, as well as removing the unnecessary visuals of a screen scraping program. For the main program, a better UI to make it more updated looking including possibly converting the program to the package “seaborn” instead of matplotlib’s 2D plots. Statistical developments as well would make the program better, including tables and descriptive statics on the front page and another page. Using Riot Games API could also incorporate more information in a quicker amount of time without parsing. Finally, though I tried, I could not add the Login to the main screen. The goal would be able to pull multiple users at once for team analysis as well as parse the information into their own csv files or tabs for further analysis.