Andrea Maver

Euroleague Shiny App

Summary

- Introduction
- What is Shiny? Learning Shiny with NBA data
- Euroleague Shiny app
- Creating a more complex application

About me

Andrea Maver

Currently studying Data Science master degree at Milano Biccoca Univeristy

Internship as assistant video and data analyst for Bocconi's basketball team

Mail

a.maver@campus.unimib.it

What is Shiny

We'll follow a great tutorial by Julia Wrobel, which explains Shiny basics and the steps to build a simple application.

You can find the link to the online tutorial in the course lecture page, or search the following link.

http://juliawrobel.com/tutorials/shiny_tutorial_nba.html

Quick recap

User interface

Input options to create the filters
Define the elements the users can interact with
Output the plots in the main panel

Server

Filter the datatabase based on the user's inputs
Create the plots
Consider putting long functions into a Helper file

Euroleague app - Overview

After learning some of the basics, we'll see how to create a more complex application, following the classic steps of a data science project, like obtaining the data, manipulating it to our needs and creating the final product

To check the application follow the link

https://andreamaver.shinyapps.io/EuroleagueApp/

Obtaining the data

Pre-made dataset

Sometimes organizations will provide you with some clean data. Otherwise, look online if someones has already created a dataset like the one you need. Websites like Kaggle have a huge amount of datasets available to download about basically any subject.

API (Application Programming Interface)

Set of functions that allow you to interact with the databases that a company might make available to download.
You can be sure about the origin and the quality of data, but there might be some limitations on the quantity of data to be downloaded and might require a (paid) subscription.

Scraping

Obtain non structured data through the source code of web pages (HTML).
It's the most time consuming way because a lot of cleaning is required to make the data structured and usable.

Euroleague APIs

Searching for a way to interact with the Euroleague APIs, I found some pages full of JSON files that contained the information I needed, but no official documentation.

Example API page: https://live.euroleague.net/api/points?gamecode=1&seasoncode=E2021

In the 'eurolig' R package there are some functions to interact with those pages, but some changes needed to be made, beacuse the old URLs aren't working anymore.
Github: https://github.com/solmos/eurolig

After obtaining the data we can start manipulating it, so filter out unwanted columns and creating new ones

Manipulating data

To create some useful filters, we need to design new columns using the ones we have available. Most of them require simple groupings or if-else conditions.

Made or missed shot
Distance from the basket
Court side (left or right)
Home or away
Win or loss
Game round (regular season, playoffs or final four)

Also, a new custom dataset is created to plot the network assist in a more efficient way.

Building the UI

Two tab panels, one for players and one for teams.

A sidebar containing all the filters available to modify your visualizations. The insertion of those filters was made possible by the manipulation of the dataframe in the previous step.

The most important part: the main panel, containing four different tabs, one for each type of plot.
In this section UI and server interact and we output the plot functions with a spinner loader for calculations.

The same code is repeated for the Teams page, but beware that all the elements you use must be distinct from one another.

Building the server

The file contains all the output functions for the different visualizations.
Each one of them is composed by a filtering of the dataset based on the UI inputs and the actual plotting function.

Some of them are simple ggplot or plotly functions integrated with the lines that form the court, other were taken from the book 'Basketball data science with applications in R' by Zuccolotto and Manisera and modified to fit Euroleague data instead of NBA's.

The same visualizations with minor tweaks are present in the Teams tab.

Helper file

To make the server code slimmer, some functions were instantiated in the helpers file.

A black theme for the heatmaps and the plot of the court lines.

Binning functions to group the data and create the dumbbell plots.

The adapted functions from the book.

Some useful links

Find inspiration:

https://github.com/toddwschneider/ballr (also some great code here!)

https://buckets.peterbeshai.com/app/#/playerView/201935_2015?colors=six

https://seriecgoldshiny.shinyapps.io/Final/

https://baslare.net/elChart.html

Implementation and some code:

http://juliawrobel.com/tutorials/shiny_tutorial_nba.html

https://github.com/solmos/eurolig

https://github.com/sndmrc/BasketballAnalyzeR

https://bodai.unibs.it/bdsports/wp-content/uploads/sites/2/2020/04/Basketball-Data-Science-code-April-2020.pdf

https://github.com/rstudio/cheatsheets/blob/main/data-visualization-2.1.pdf

https://shinyapps.dreamrs.fr/shinyWidgets/

Thank you for your kind attention

If you have any questions, please feel free to ask