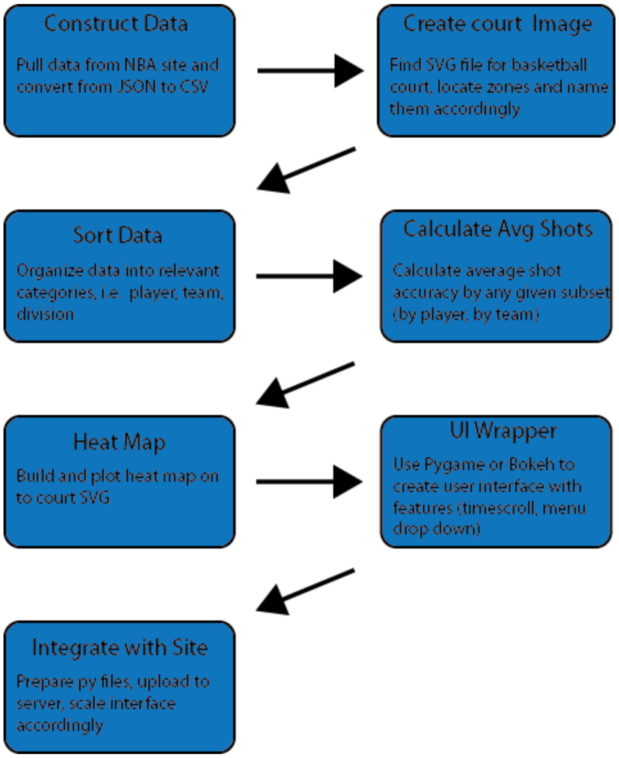
1. **Background and context** What information about your project does the audience need to participate fully in the technical review? You should share enough to make sure your audience understands the questions you are asking, but without going into unnecessary detail.

For this project we have decided to create a comprehensive, interactive data visualization tool that will allow users to intuitively parse through shot position data in NBA games. We want to provide a user control application that will illustrate the concentration of shots taken as a heat map in games since the beginning of the sports stats era. We will then proceed to explore visualization techniques beyond a court heatmap, potentially with wider variety of statistics. Lastly, we hope to be able to integrate our project into a our project website to make the tool more accessible to outside sources.



Construct Data

Pull data from NBA site and convert from JSON to CSV

Create court Image

Find SVG file for basketball court, locate zones and name them accordingly

Sort Data

Organize data into relevant categories, i.e. player, team, division

Calculate Avg Shots

Calculate average shot accuracy by any given subset (by player, by team)

Heat Map

Build and plot heat map on to court SVG

UI Wrapper

Use Pygame or Bokeh to create user interface with features (timescroll, menu drop down)

Integrate with Site

Prepare py files, upload to server, scale interface accordingly

2. **Key questions** What do you want to learn from the review? What are the most important decisions your team is currently contemplating? Where might an outside perspective be most helpful? As you select key questions to ask during the review, bear in mind both the time limitations and background of your audience.

Risk ID and Mitigation:

- Integrating code (after working separately) could be problematic, our team doesn't have much software development background, or much time to code together. If you have experience integrating code, what has been an effective workflow?

- Designing from the get go to integrate with a web app could be a challenge. Aside from data storage, what if anything do we need to plan ahead for?

Software architecture discussion:

- We need to make a critical decision about how and how often our code will re-compile data (that's constantly updating) from the NBA stats site. We know that pulling and parsing all the data will be computationally expensive, and only want to pull new data when we re-compile. How?
- We will sort our data into relavent categories i.e Player data, Team data, etc. Should we make these into Classes? What happens with players who change team midseason? Is there an alternative way to sort and store our data?
- What file format can store large amounts of data, and can also scale/adapt well to web application?

3. **Agenda for technical review session** Be specific about how you plan to use your allotted time. What strategies will you use to communicate with your audience?

5 minutes: Most important thing to get out of discussion is about software architecture. We need to know how to sort and store our data into categories (Classes maybe?) and also how to handle our large dataset efficiently with data updates

2 minutes: What do we need to do/avoid in order to have a final product that can be easily integrated with our project website?
3 minutes: Methods for integrating code from different team members.

Rest: More data visualizations we could do with this data set.

4. **Feedback form** Create a Google form that folks in the review will use to provide you with feedback or answers to various questions you pose to your audience. Since, at least for the first review, the time you have to present will be very short you should expect most of the feedback you get to come from this form rather than thoughts expressed orally during your session. Please submit a link to your Google form using this other Google form! (you must have this submitted no less than 2 hours before class so we have time to post a link on the course website).