ChatGPT

12-Hour Roadmap: Soccer Performance Analysis (PSG Data Scientist Prep)

1–3 hrs: Event Data Fundamentals

- **Get started with soccer data:** Install Python packages (pandas, statsbombpy, mplsoccer, scikit-learn, etc.) and download a sample event dataset. StatsBomb provides *open-data* (free event data for research) on GitHub [1]. Load the JSON files (competitions, matches, events) into Pandas or via `statsbombpy`. Examine event columns (passes, shots, locations) to understand structure.
- **Familiarize with common data tools:** Practice basic queries (e.g. filter all shots or passes in a match) and simple plots. Check PSG's job ad: it expects expertise with event data (StatsBomb/Opta/Wyscout) [2]. Use this time to read the StatsBomb documentation and try a quick script to count shots/goals. The StatsBomb *Data Champions* webinar materials are useful – they cover importing free event data and plotting on a pitch [3].
- **Explore an example:** For context, see resources like *"Working with StatsBomb Data in Python"* (e.g. Medium tutorials) and Edd Webster's FootballAnalytics repo for code examples. Try plotting a simple shot chart: extract shots from a match and scatter them on a half-pitch. Use `mplsoccer` for quick visuals [4] and verify understanding of the (0–100) coordinate system. This hands-on exploration solidifies how event logs translate into analysis.

3–5 hrs: Key Performance Metrics (xG, PPDA, Packing)

- **Learn Expected Goals (xG):** Study what xG means and how it's used. For example, StatsBomb explains that xG is *"the probability of a shot resulting in a goal"* (0 to 1) based on shot context [5]. Read about how features like distance, angle, assist type, and player positions feed into xG [6]. Then compute a basic xG chart: filter the events to shots and sum `shot_statsbomb_xg` by minute or by team. Try fitting a logistic regression with scikit-learn using features like shot location to predict goals – this mirrors a simple xG model building exercise.
- **Study PPDA (Pressing Intensity):** Read definitions of pressing metrics. For instance, PPDA (Passes Per Defensive Action) is defined as *"the number of passes an opposing team makes before a defensive action"* in the attacking third [7]. A lower PPDA indicates more aggressive pressing. Understand the concept and then implement it: count a team's opponent passes in the final 3/5 of the pitch and divide by the number of tackles/interceptions. Practice this on a match from the dataset.
- **Understand Packing:** Learn that packing measures passing/dribbling effectiveness: it counts how many defenders are bypassed by a pass or dribble [8]. This highlights line-breaking plays by players. Review the example in the Medium article and try calculating packing for your event data (compare passes that 'break lines' vs backward passes). These two metrics are popular in modern analysis and were even highlighted in a PSG job listing for tactical KPIs [9].

5–7 hrs: Tracking Data & Spatio-Temporal Analysis

- **Introduction to tracking data:** Tracking data gives x–y coordinates of *all players and the ball over time*. It enables deeper tactical analysis (formation changes, off-ball runs) [10]. Download a sample tracking dataset, such as the Metrica Sports open-data repository (2–3 games of synchronized tracking + events) [11]. Inspect the CSVs: learn the coordinate normalization (0–1 field) and time stamps.
- **Load and visualize tracking:** Use tools like Kloppy or plain pandas to read the data. Plot some raw tracking points: e.g. map a single player's movement or the ball's path on the pitch over time. Check out Laurie Shaw's *Friends of Tracking* tutorial for an example workflow [12]. As an exercise, compute simple stats: player speed/acceleration from successive coordinates, or distance covered.
- **Contextual metrics from tracking:** Try a basic spatial analysis: compute the team's "width" (max distance between leftmost and rightmost players) or "depth" (max distance front-back) over time. Plot heatmaps of player positions. These skills lay groundwork for PSG's focus on positional data (Hawkeye/Second Spectrum) and spatio-temporal modeling [2] [9].

7–9 hrs: Visualization Tools and Dashboards

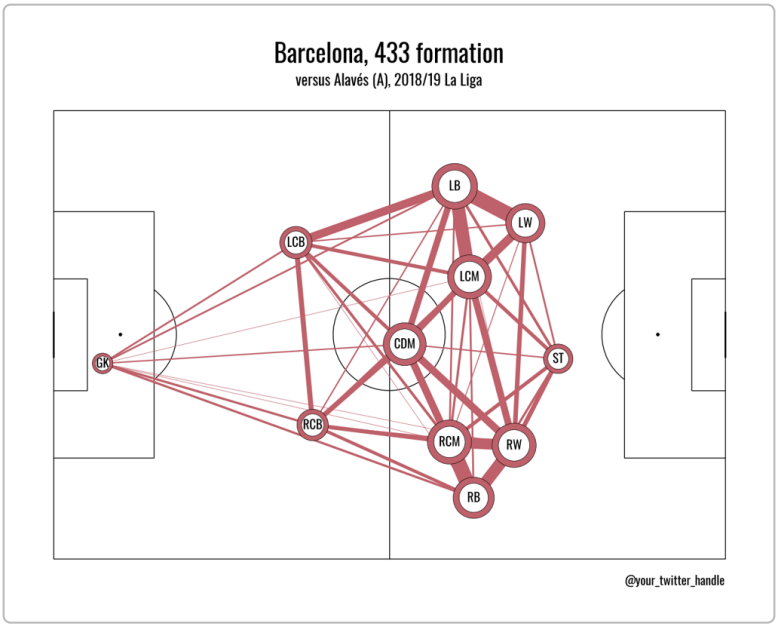

Figure: Example pass network for FC Barcelona (4-3-3 formation) – In soccer analysis, visualizing passing networks helps coaches see team structure. Nodes show players (size ~ involvement) and lines show passes (thickness ~ frequency). Creating such pitch-based diagrams (as above) is common practice [13]. Learn to use `mplsoccer` or Plotly to draw pitches, arrows, and nodes for passes; this turns raw data into intuitive images. PSG explicitly values tools like Plotly/Dash for storytelling [9]. - **Practice pitch plots:** Use `mplsoccer` or other libraries to recreate examples: shot charts, pass maps, possession zones. For instance, plot all shots of a team with circles sized by xG, or color-code passes by outcome. Follow the *mplsoccer gallery* examples (e.g. pass network, heatmap, shot freeze frames) to see code patterns. The StatsBomb webinar materials show how to make "pitch plots" with Matplotlib [3]. - **Interactive dashboards:** Spend an hour building a simple Streamlit or Dash app. Since you know Streamlit, try an interactive match report: e.g. user selects a match, and the app displays key visuals (shot map, xG timeline,

a press intensity gauge). Embed one of your mplsoccer plots or use Plotly for interactivity. This demonstrates "data storytelling" – turning analytics into actionable insights for coaches [9].

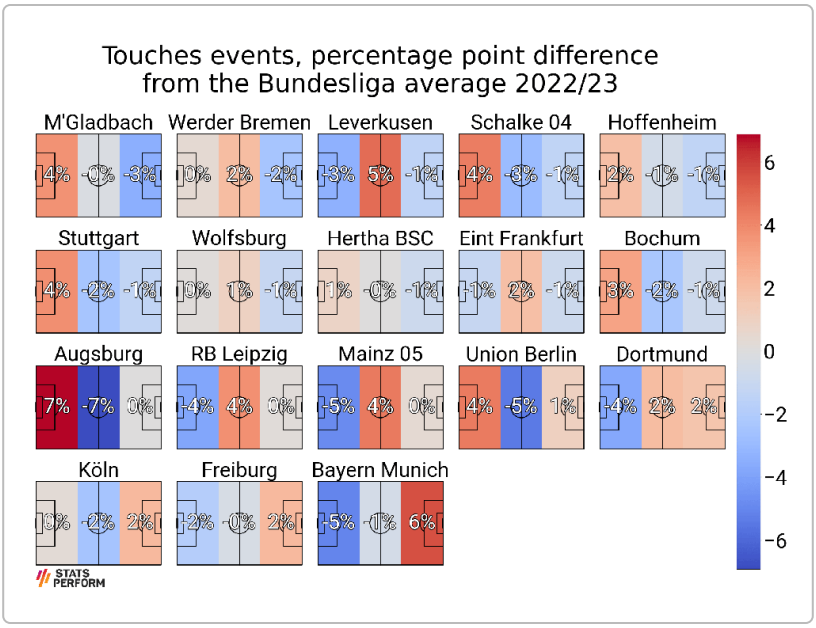

Figure: Heatmaps of team touch distribution by pitch third (percentage vs Bundesliga average) – Another useful viz is a comparative heatmap. The example above shows each Bundesliga team's touches (%) in defensive/mid/attacking third vs league avg. Blue/red indicates below/above average. You can create such stats by binning events (or tracking) into zones and plotting grids [3]. These visuals highlight team tendencies and can be built with `mplsoccer` binning or Matplotlib heatmaps. - **Try team heatmaps:** For each team or phase of play, compute a 3×3 grid statistic (e.g. total passes or touches in each third). Use `pitch.bin_statistic()` in mplsoccer or Plotly heatmap to display values on mini-pitches. This exercise shows how to aggregate event data spatially and present it clearly. It also mimics performance reports a data scientist might deliver to analysts or coaches.

9–12 hrs: Capstone Project & Data Sources

- **Project idea (GitHub):** Build an end-to-end soccer analysis project. For example, create a **"Match Analysis Dashboard"** using StatsBomb data. Steps: (1) Train a simple xG model (scikit-learn) on open data; (2) Compute match metrics (actual vs xG, PPDA, packing) for a specific game; (3) Develop an interactive Streamlit app showing a shot chart with xG, the xG progress timeline, and a pass network for each team【23†Image】. Host all code on GitHub with clear notebooks and a readme. This demonstrates modeling, Python skills, and visualization – exactly the deliverables a football data scientist would provide (models, KPIs, and dashboards) [9] [3].
- **Additional project angle:** You could also incorporate tracking analysis. For instance, use Metrica data to compute defensive line height over the match, or generate a "speed map" of player movements. Even a small animation of player trajectories or a heatmap of positions shows mastery of tracking data. Frame your GitHub repo as an analytics report (Markdown or slides) linking code, data, and insights.
- **Potential data sources:** Gather free soccer data to support your project. Key sources include **StatsBomb Open Data** (GitHub: event data for dozens of matches) [1], **Metrica Sports** sample data

(tracking + events CSVs) [11], and **Wyscout 2018 World Cup event data** on Figshare [14]. Kaggle also hosts soccer datasets (e.g. **StatsBomb Kaggle**, European league stats, World Cup JSON). For tracking, explore open sets like SkillCorner or LastRow (listed in the Metrica repo) [15]. Using these sources, cite them as you report results. By the end, you'll have shown end-to-end skills in processing data, building metrics, and communicating findings – exactly what PSG's Football Data Scientist role requires [2] [9].

Summary: This 12-hour plan prioritizes the essentials: exploring event data (StatsBomb), learning soccer-specific metrics (xG, PPDA, packing), getting a taste of tracking data (Metrica), and mastering visualization (mplsoccer, Plotly, Streamlit). Each block includes hands-on practice and links to free resources. By following it, you'll compile a showcase project and gain confidence in the key topics PSG highlights: advanced match analysis, player performance models, and compelling dashboards [2] [3].

Sources: See StatsBomb Open Data [1] and tutorial materials [3]; medium posts on PPDA/packing [7] [8]; Metrica sample data info [11] [12]; Hudl StatsBomb metrics guide [5] [6]; PSG job posting for context [2] [9]; mplsoccer examples [4] [3]. Images show example pitch plots as discussed.

---

[1] GitHub - statsbomb/open-data: Free football data from StatsBomb
https://github.com/statsbomb/open-data

[2] [9] Football Data Scientist - F/H at Paris Saint-Germain - Campus D1, France | aijobs.net
https://aijobs.net/job/1356336-football-data-scientist-fh/

[3] Using Hudl Statsbomb Free Data In Python - Hudl Statsbomb | Data Champions
https://statsbomb.com/articles/soccer/using-statsbomb-free-data-in-python/

[4] [13] Football Data Analytics — Let's start | by Maciej Gieparda | Medium
https://medium.com/@DataThinker/football-data-analytics-lets-start-4ad1a28ee357

[5] [6] What is xG? How is it calculated? | Hudl Statsbomb | Data Champions
https://statsbomb.com/soccer-metrics/expected-goals-xg-explained/

[7] [8] Football Stats: PPDA and Packing. Measuring Pressing with Passes Per... | by Building Blocks | Medium
https://medium.com/@buildingblocks/football-stats-ppda-and-packing-a750a0df18ef

[10] Advanced Player Tracking Data Analysis in Football | Medium
https://footsci.medium.com/tracking-data-the-most-detailed-and-accurate-information-about-players-actions-on-the-pitch-c9d9ec7e2e9d

[11] [12] [14] [15] GitHub - metrica-sports/sample-data: Metrica Sports sample tracking and event data
https://github.com/metrica-sports/sample-data