PROGRAMMING FOR DATA SCIENCE

PROJECT DOCUMENTATION

Comprehensive Analysis of NFL Player Metrics

SUBMITTED TO-

PROF. RAJESHKANNAN R.

SUBMITTED BY-

CHOKKAM SAMIKSHA (22BCE0596)

SEJAL CHAAJED (22BCE0137)

BANDARU DHANYA DEEPIKA (22BCE3693)

DATE : 13/02/25

INTRODUCTION

Sports analytics has revolutionized how player performance and team strategy are analysed. In the NFL, data-driven analysis enables thorough analysis of player ability, team alignment, and game flow. RStudio is used in this study to analyse and visualize player performance data, giving actionable insights to analysts, coaches, and fans.

ABSTRACT

This research, entitled Comprehensive Analysis of NFL Player Metrics Using RStudio, offers an extensive analysis of NFL play-by-play data in order to measure player performance, positional traits, and team balance through sophisticated analysis in RStudio using Shiny Web App. Through combining various functionalities, this analysis renders detailed information about individual and aggregate dynamics on the field. Key analytical elements consist of:

- **Position-Specific Insights:** Measuring height (in inches) and weight (in pounds) by position to see the physical characteristics of certain positions in the NFL.
- **Sports Analytics:** Comparing player and team statistics, such as individual performance metrics and team makeup.
- **Performance Benchmarks**: Determining top players through maximum speed and acceleration.
- **Player Spatial Analysis:** Mapping player locations on the field over certain time periods.
- **4-Hour Trends:** Monitoring player performance changes over time using aggregated speed data.

This data-driven approach empowers coaches, analysts, and fans with nuanced perspectives on player and team dynamics, enhancing strategic decision-making in the NFL.

METHODOLOGY

- **Data Collection**
  The study uses four datasets from Kaggle:
  - **Games Dataset:** Provides information on game dates, times, and participating teams.
  - **Players Dataset:** Contains player attributes such as height, weight, age, and position.
  - **Plays Dataset:** Includes play-by-play descriptions, down count, yardage details, and team formations.
  - **Week Data Dataset:** Records player movement data, including speed, acceleration, position coordinates, and play events.

- **Data Preprocessing**
  - **Cleaning & Transformation:**
    - Removing rows with missing data.
    - Converting all column names to lower case.
    - Age calculated based on date of birth.
    - Converting to suitable datatypes.
  - **Merging Datasets:**
    - Player metrics computed by linking movement data with player attributes.
    - Game-specific analytics linked with corresponding play data.

- **Feature Engineering**
  Several key performance indicators were computed:
  - **Total Distance Covered**: Summation of distances travelled by each player.
  - **Average Speed**: Mean speed values per player.
  - **Plays Participated**: Number of plays each player was involved in.
  - **Peak Performance Analysis**: Calculation of peak acceleration and speed moments.
  - **Spatial Visualization:** Mapping player locations on the field for set time period.
  - **Position Insights:** Height vs weight distribution by team position.

- **Data Visualization and Analysis**
  RStudio was utilized to create visualizations for analysing patterns in player and team performance. Methods included:
  - Scatter plots for positional attributes (height, weight, age).
  - Bar charts for top-performing players.
  - Line charts for 4-hourly performance trends.

PSEUDOCODE

1. Data Cleaning and Merging

```
_____________________________________
Players Data Cleaning:
    LOAD "players.csv" INTO players_dataframe
    CONVERT all column names in players_dataframe to lowercase
    REMOVE rows where "nflid" is missing
    IF "age" column does NOT exist AND "birthdate" column EXISTS:
        COMPUTE age by subtracting birth year from the current year
        ADD computed "age" column to players_dataframe
    CLEAN the "height" column:
        CONVERT height values to character type
        IDENTIFY valid numeric heights using regex matching ("^\\d+$")
        CONVERT valid height entries to numeric values
        CALCULATE the mean of valid numeric heights (ignoring NA)
        REPLACE non-numeric height entries with the computed mean height
        CONVERT the "height" column back to numeric

NFL Data Cleaning:
    LOAD "week_data.csv", "plays.csv", and "games.csv" into respective dataframes
    CONVERT all column names in each dataframe to lowercase
    REMOVE rows from week_data where "nflid" is missing

Data Merging for Team Metrics:
    MERGE week_data with games dataframe on "gameid"
    CREATE a new column "actualteam":
        IF "team" equals "home", ASSIGN "hometeamabbr" to "actualteam"
        ELSE ASSIGN "visitorteamabbr" to "actualteam"
    LEFT JOIN the merged dataframe with players_dataframe on "nflid"
    CREATE a new column "teamtype":
        IF "team" equals "home", SET "teamtype" to "Home Team"
        ELSE SET "teamtype" to "Away Team"
    IF "birthdate" column exists in the merged dataframe:
        COMPUTE "age" from "birthdate" by subtracting birth year from the current year
        ADD computed "age" column to the merged dataframe

Data Cleaning for Performance Metrics:
    GROUP week_data by "nflid"
    FOR each player:
        COMPUTE "Peak_Speed" as the maximum of "s" values (ignoring NA)
        COMPUTE "Peak_Acceleration" as the maximum of "a" values (ignoring NA)
```

```
    ARRANGE the resulting data in descending order of "Peak_Speed"
    LEFT JOIN the performance metrics with players_dataframe (selecting "nflid" and "displayname")

Data Cleaning for Weekly Trends:
    IF "week" column does NOT exist in week_data:
        LEFT JOIN week_data with games dataframe on "gameid" to retrieve "week" values
    GROUP week_data by "week"
    COMPUTE the mean of speed ("s") and acceleration ("a") for each week (ignoring NA)
    ARRANGE the grouped data by week

Data Cleaning for 4-Hour Trends:
    CLEAN the "time" column in week_data:
        REMOVE "T" and "Z" characters from "time"
        CONVERT the cleaned "time" string to datetime format using ymd_hms() with tz = "UTC"
    FILTER out rows where the cleaned time is NA
    MUTATE a new column "time_4hr" by flooring "time_clean" to the nearest 4-hour interval using
floor_date()
    GROUP the data by "time_4hr"
    COMPUTE "Total_Distance" as the sum of "dis" for each interval (ignoring NA)
    COMPUTE "Avg_Speed" as the mean of "s" for each interval (ignoring NA)
    ARRANGE the grouped data by "time_4hr"

Data Cleaning for Spatial Analysis:
    CHECK if required columns "playid", "nflid", "x", "y" exist in week_data
    IF any required column is missing:
        THROW an error indicating missing columns
    EXTRACT unique play IDs from week_data and STORE them in unique_plays
_____________________________________
```

2. Position-Specific Insights

```
_____________________________________
UI Panel: Position-Specific Insights
    Dropdown: Select Position (Choices = Unique positions from "players" dataframe)

    Tab 1: Scatter Plot (Height vs Weight)
        X-axis: Weight
        Y-axis: Height
        Color: Age
        Tooltip: Displays Age, Height, and Weight on hover
```

```
    Tab 2: Bar Chart (Average Attributes by Position)
        Attributes: Avg Height, Avg Weight, Avg Age
        X-axis: Position
        Y-axis: Attribute Values

Server Panel: Position-Specific Insights
    FILTER players data based on selected position (Reactive Function)
    RENDER Scatter Plot:
        PLOT Weight vs Height with Age as color scale using ggplot2
        CONVERT plot to interactive Plotly plot with tooltips
    RENDER Bar Chart:
        COMPUTE average Height, Weight, and Age per position using dplyr
        TRANSFORM data into long format for grouped bar chart using tidyr
        PLOT grouped bar chart using ggplot2
        CONVERT plot to interactive Plotly plot


_____________________________________
```

3. Sports Analytics

```
_____________________________________
UI Panel: Sports Analytics
    Dropdown: Select Analysis Type
        - Player Metrics
        - Team Metrics

    IF Player Metrics is selected:
        Dropdown: Select Player (Choices = Unique player names from "week_data")
        Display:
            Table: Total Distance Covered, Average Speed, Number of Plays
            Line Chart: Speed vs FrameID (Time)
                X-axis: FrameID (Time)
                Y-axis: Speed

    IF Team Metrics is selected:
        Dropdown: Select Team Type (Choices = "Home Team", "Away Team", "All Teams")
        Dropdown: Select Specific Team (Choices depend on "team_type" selection)
        Dropdown: Select Distribution Type (Choices = "Age", "Height", "Weight")

        Display:
            Bar Chart: Distribution by selected attribute
            Table: Average Metrics per Team
```

```
Server Panel: Sports Analytics
    DYNAMIC Dropdown for Team Selection:
        UPDATE choices based on selected "team_type"

    IF Player Metrics:
        FILTER "week_data" for selected player (Reactive Function)
        COMPUTE total distance, average speed, and number of plays
        RENDER Table displaying computed metrics
        RENDER Line Chart of Speed vs FrameID using ggplot2
    IF Team Metrics:
        IF Analysis Type is "Average Metrics":
            DISPLAY precomputed team averages from nfl_team_metrics
            RENDER Bar Chart using ggplot2
        ELSE:
            FILTER "week_data" for selected team (Reactive Function)
            PLOT Histogram of selected attribute (Age, Height, or Weight) using ggplot2


_____
```

4. Performance Benchmarks

```
_____
UI Panel: Performance Benchmarks
    Dropdown: Select Performance Metric (Choices = "Peak Speed", "Peak Acceleration")

    Display:
        Table: Top 10 Players based on selected metric
        Bar Chart: Horizontal Bar Graph ranking players by performance metric

Server Panel: Performance Benchmarks
    FILTER top 10 players by selected performance metric from performance_metrics (Reactive Function)
    RENDER Table displaying top 10 players
    RENDER Horizontal Bar Chart using ggplot2 and convert to Plotly for interactivity


_____
```

5. Sports Analytics (Spatial Analysis)

```
_____
UI Panel: NFL Spatial Analysis
    Dropdown: Select Game (Choices = Unique game IDs from "games" dataframe)
    Dropdown: Select Duration (Choices = Set duration options)
    Checkbox Group: Select Players to Display
```

```
    Display:
        Scatter Plot: Player positions on field
        X-axis: Field X coordinate
        Y-axis: Field Y coordinate
        Color: Player name

Server Panel: NFL Spatial Analysis
    FILTER "week_data" for selected game and duration (Reactive Function)
    JOIN filtered data with "players" to retrieve player names
    RENDER Scatter Plot of player positions using ggplot2
    CONVERT plot to interactive Plotly plot allowing selection/deselection of players


_____________________________________
```

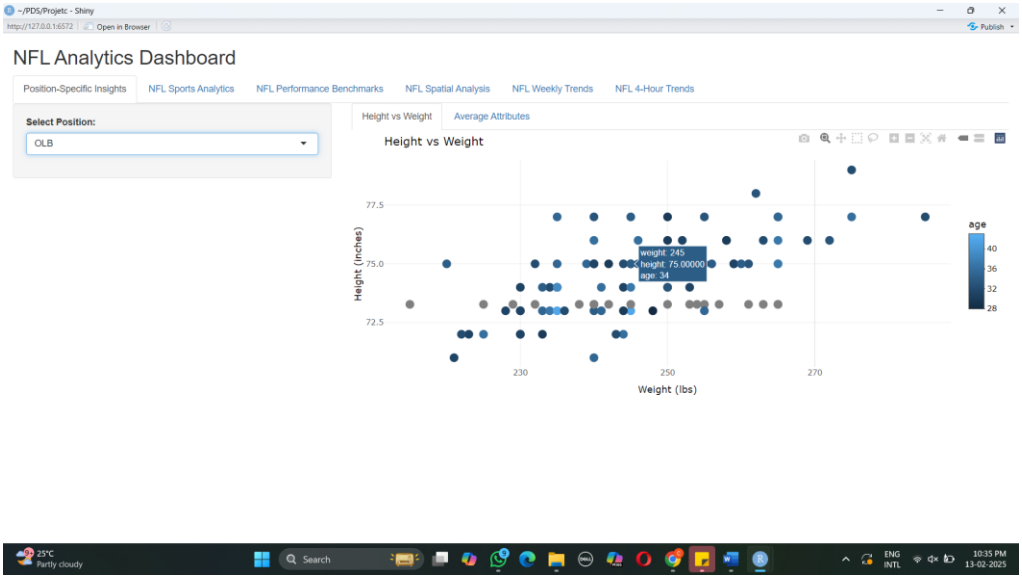
6. 4-Hour Trends

```
_____________________________________
UI Panel: NFL 4-Hour Trends
    Display:
        Line Chart: Average Speed over time (4-hour intervals)
        X-axis: 4-hour time interval
        Y-axis: Average Speed
        Tooltip: Shows Date and Time of Recording on hover

Server Panel: 4-Hour Trends
    CLEAN "time" column in "week_data":
        REMOVE "T" and "Z" characters from timestamp
        CONVERT cleaned timestamps to datetime format using ymd_hms with tz = "UTC"
    FILTER out invalid timestamps
    ROUND each timestamp to nearest 4-hour interval using floor_date
    GROUP data by 4-hour interval and COMPUTE total distance and average speed
    RENDER Line Chart of Average Speed over 4-hour intervals using ggplot2
    CONVERT plot to interactive Plotly plot with tooltips displaying Date and Time


_____________________________________
Run the Shiny App
_____________________________________
CALL shinyApp with UI and Server components
```

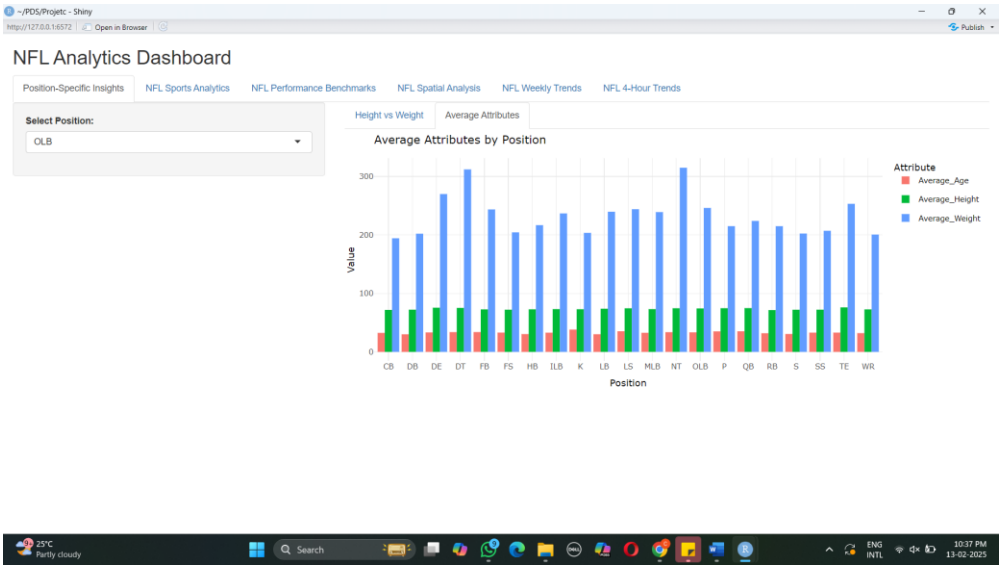
ANALYSIS AND RESULTS

1. Position-Specific Insights

- **Height vs. Weight by Football Team Position**
  A scatter plot visualizes height vs. weight for each position. On hovering over a data point, details such as player age, height, and weight are displayed, helping to identify football team positional trends in player physique.

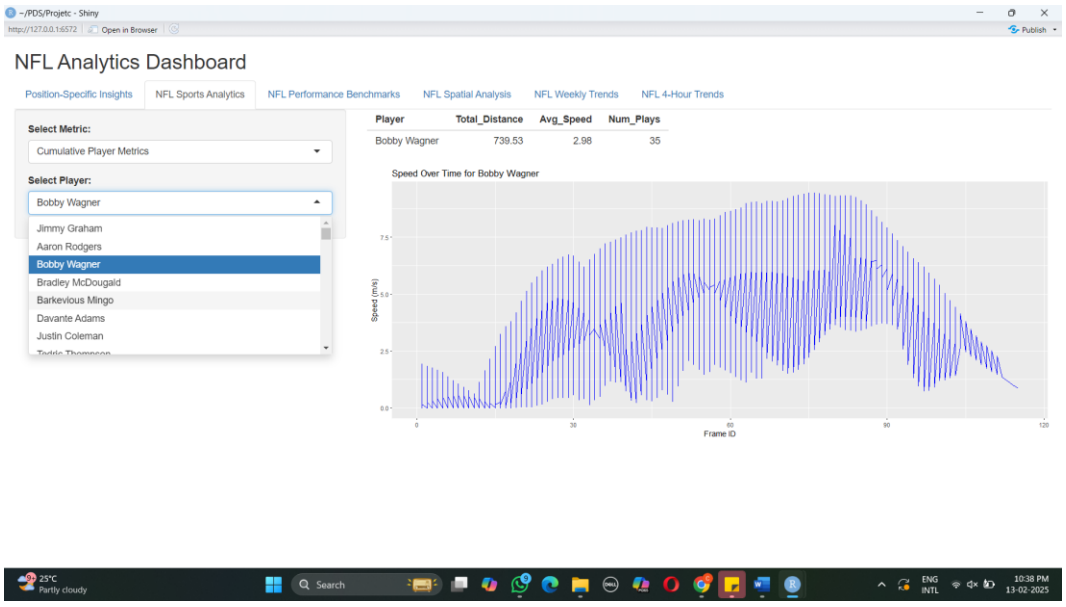


- **Average Attributes per Position**
  A bar graph represents the average age, height, and weight for each position, providing insights into the physical requirements for different roles in the NFL.

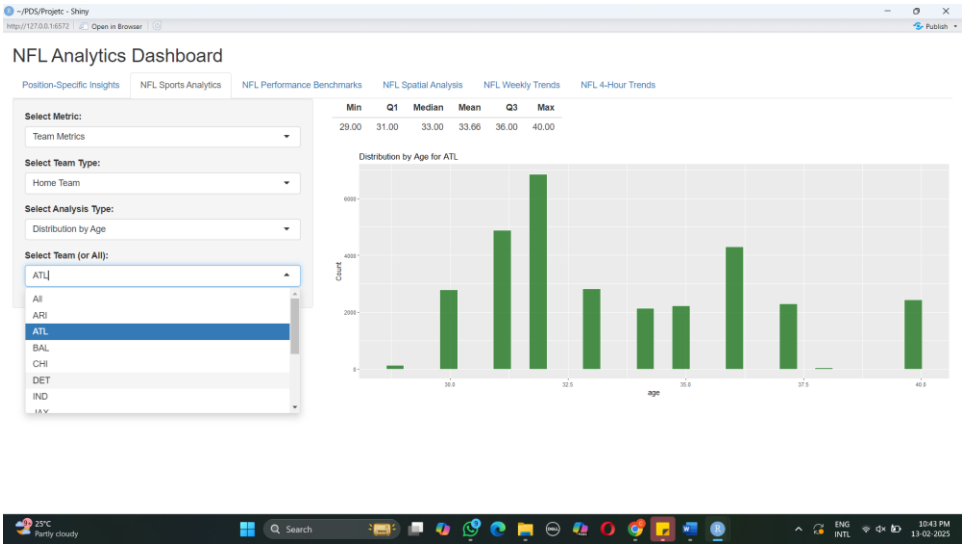


2. Sports Analytics

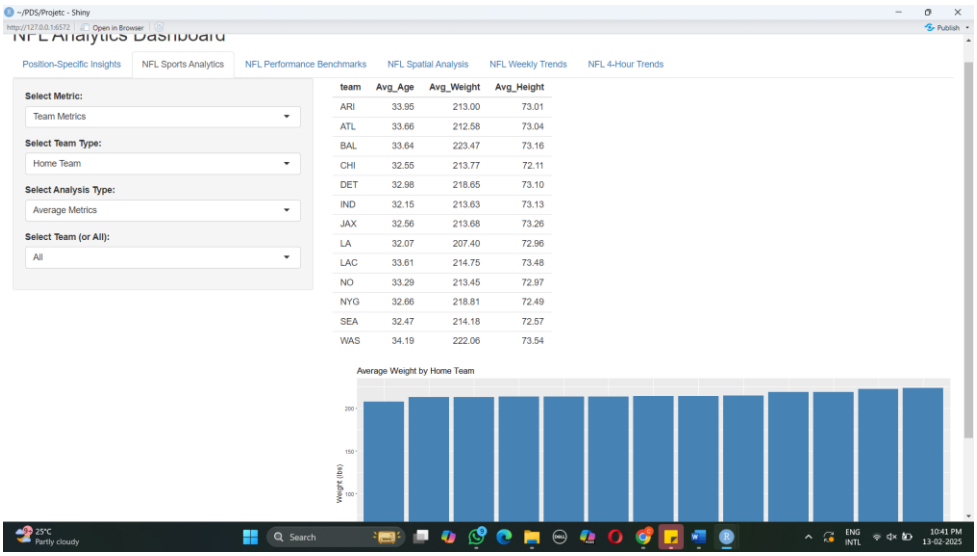
- **Player Metrics**
  For a selected player, total distance covered, average speed, and the number of plays participated in are displayed. A graph plotting speed against frame ID (time) provides a visual representation of player movement patterns over time.



- **Team Metrics**
  Team metrics are analysed using:
  - **Distribution Graphs:** Showing age, height, and weight distributions for a chosen team or all teams (home, away, or both).

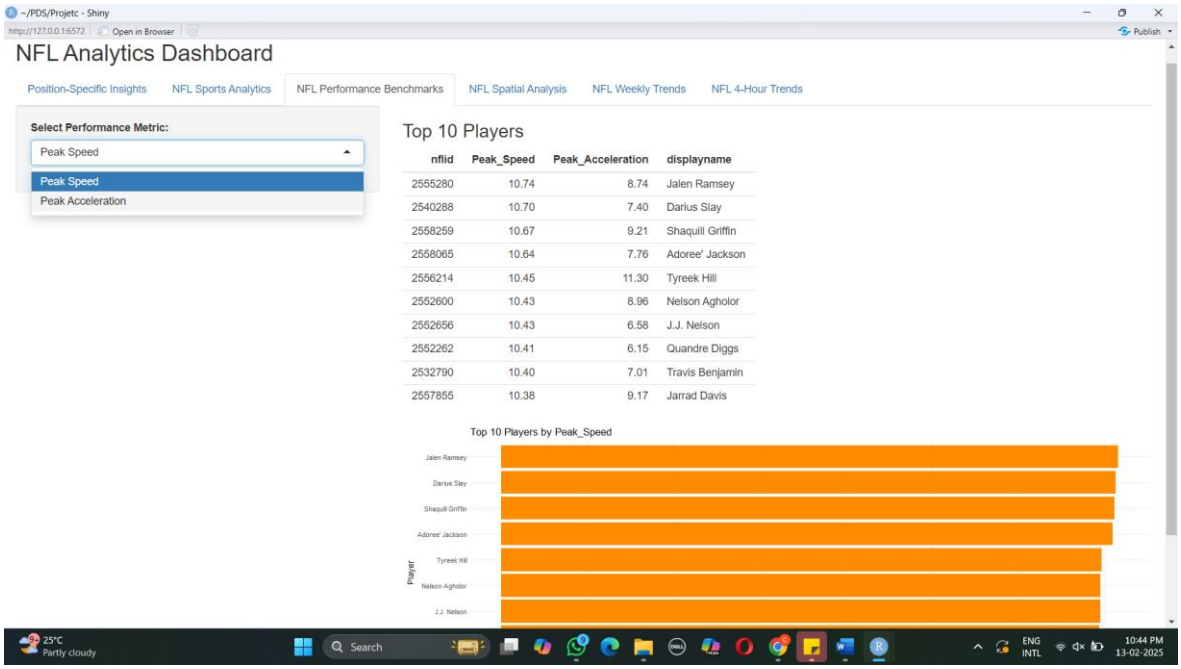


- **Average Metrics:** Displaying per-team averages for age, height, and weight, either for one team at a time or all teams collectively.



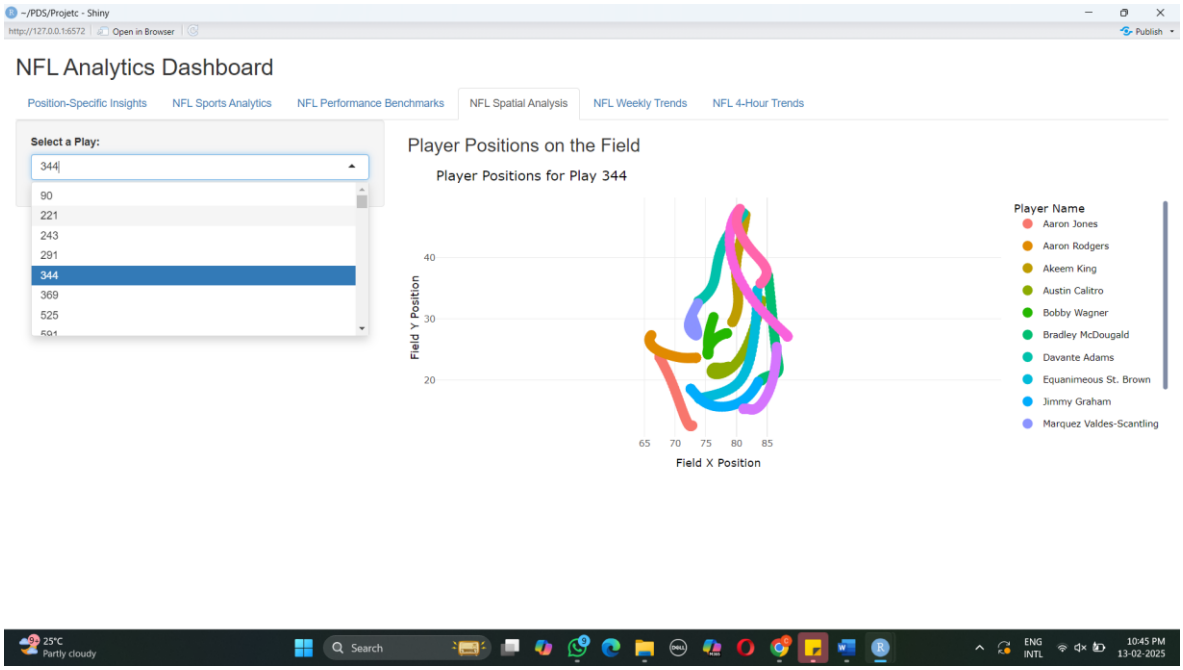
3. Performance Benchmarks

Top 10 players are ranked based on peak speed and acceleration. A horizontal bar graph compares these players, providing benchmarks for elite athleticism.



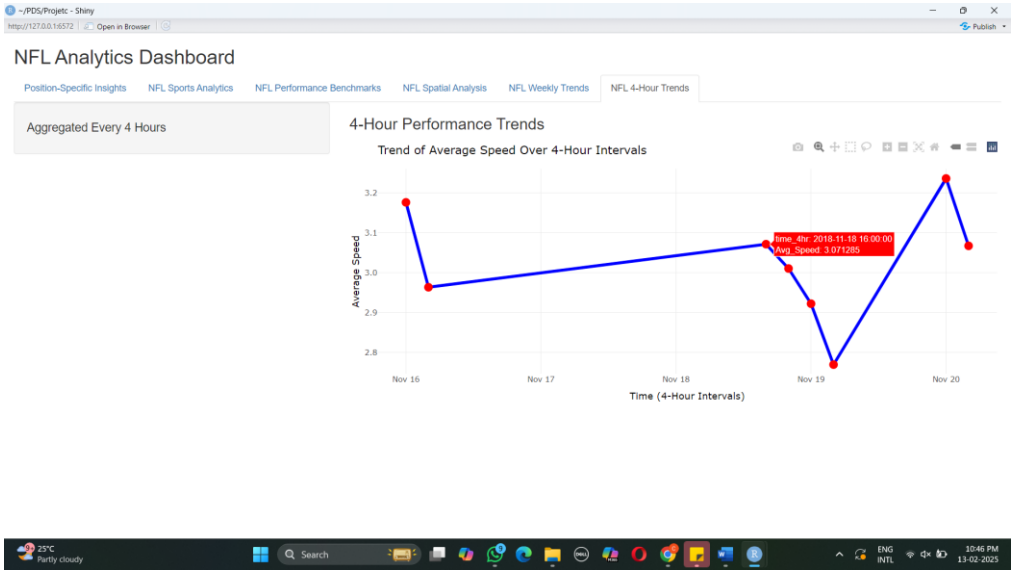
4. Player Spatial Analysis

For a selected game, player positions are visualized on a field over a specific time duration using a scatter plot. Users can focus on chosen players by deselecting others, enabling targeted analysis of player movements.



5. 4-Hour Trends

Average speed is calculated for all players every 4 hours. A line graph depicts trends over time, with date and time displayed on hovering over data points. This visualization helps in understanding variations in player performance throughout a given period.



CONCLUSION

This project illustrates how RStudio can be utilized to explore intricate NFL data sets, extracting valuable information about player performance, team makeup, and game play. Through the use of statistical measures, visualization, and trend analysis, this methodology supports strategic decision-making by coaches, analysts, and enthusiasts. Future work can build upon these findings through predictive modelling and machine learning algorithms to improve player scouting, injury prevention, and game strategy optimization.

REFERENCES

- NFL Player Tracking Data: Kaggle
- RStudio Documentation: Shiny Web App