Finding the Next Generations of Tennis Stars

Naoki Tsumoto, Roselyn Rozario, Ankit Gubiligari, Nakul Vadlamudi
Data Engineering Platforms for Analytics
Group 4

Contents

Business Use Case

- **Situation:** An apparel company is trying to find tennis players who will become the next generation of stars to take after greats like Roger Federer and Rafa Nadal before any other company can.
- **Business Problem:** The company does not know what kind of players will be successful in the tennis world in the future, and therefore are unsure of which players to sign. The company wants to find the next generation of players before other companies can and sign sponsorship deals with them.
- **Strategy:** The company has opted to take a data-driven approach. They have hired a group of data scientists to evaluate several years' worth of previous data to determine what factors or metrics drive player performance. The findings of this exploratory data analysis will help the company make strategic decisions, or rather, inform on which players to sign.

Project Goals/Research Objective(s)

How can the apparel company determine which players show talent similar to the greats of the game?

- Perform an exploratory data analysis on tennis data, which includes the following:
  - Player characteristics (e.g., height, dominant hand, country, etc.)
  - Match information (e.g., points won, aces, etc.)
  - Tournament information (e.g., surface type)
- Determine factors that will help inform on player performance.
  - For example, do left-handed players perform better overall? Or do players from a certain country have an advantage on certain types of courts (e.g., grass)?
- Present findings to apparel company so that they can use the factors to determine which players to sign.

Executive Summary

In response to the business problem, a dataset on tennis was found, which focuses on players, rankings, tournaments and matches.

Used Excel and SQL to work with the extracted data from Kaggle.

Conducted an exploratory data analysis was conducted to look at factors that drive player performance based on the dataset.

Showcased findings/insights as visuals created using Tableau and Python.

Methodology

Methodology Overview

Pipeline	Tools
Extract/Ingest	kaggle
Data Preparation (i.e. Pre-Processing)	MySQL
Store (i.e. Database Construction)	MySQL
Analyze	MySQL
Visualize/Report	tableau, python

Data Source

Dataset: ATP Matches

Kaggle serves as the data source hub in that a dataset was found on "ATP Matches," specifically of data of matches from 1968 until 2022.

CSV Files in Dataset:

1. ATP Matches
2. ATP Players
3. ATP Rankings

Per the description on Kaggle, under those listed to be given credits, Tennis Abstract was mentioned, which appears to be a location for tennis statistics/information and is likely where the dataset or its information was extracted from.

Dataset & Design Considerations

Category	Consideration
Data Preparation	• There are a lot of data types (e.g., INT, CHAR, etc.) in the CSV files. When creating the tables, the types will need to be checked to ensure they match the data.
Data Integrity	• Foreign keys are needed when creating relationships between tables. In some cases, a column needs to be created to serve as the primary key which in turn will be a foreign key. • By ensuring that the CONSTRAINT clause is being used in conjunction with the foreign key, it preserves referential integrity, and that the data is being added accurately.
Data Consistency	• The dataset spans over 50 years, during which tennis rules or formats might have changed. Such changes might affect data consistency.
Data Redundancy	• With multiple CSV files like match data, player data, and ranking data, there's potential for data redundancy.
Data Tools	• Running queries that return large amounts of data may impact the run time in MySQL and Tableau.
Complex Queries	• Complex queries might be needed to generate analyses or reports and to join data from the various tables that have been created from the multiple CSV files.
NULL Values	• Several columns have NULL values, which will have to be taken into consideration when running queries and analyses.

Relational Database Implementation Overview

Extract → Transform → Load

ATP Matches Dataset

1. ATP Matches
Tournament information, surface, winning and losing player information, and game information (e.g., points, aces, breakpoints, etc.)

2. ATP Players
Player information (e.g., name, dominant hand, country, DOB, height, etc.)

3. ATP Rankings
Rank, points, player and ranking date

Download CSVs

Pre-Processing

- Reviewed data in Excel.
- Determined which columns need cleaning and data types needed.
- Investigated amount of data available by reviewing NULL values.

Relational Database (Database Creation & Data Loading)

- Used SQL to create an EER model that is normalized and based off the Classical Model Schema.
- Created DML and DDL scripts to create a structured and normalized database and load extracted data from CSV files.

Extract, Transform and Load (ETL) Process

Data

- Downloaded data from Kaggle.
- 3 CSV files were extracted covering information on players, matches and rankings.

Data Preparation

- Pre-processing involved examining the columns, NULL values and data types.
- Normalization was done on the data.

Relational Modeling and Database Schema

- The Classical Model Diagram structure was used.
- The data first is represented as a Conceptual and Logical Model Design, which the the EER model is based off of.

Database Creation and Data Load

- DDL script created for database structures, such as tables.
- DML script created for loading data from extracted CSV files into relational tables created.

Relational Database

- Database contains at least 5 relational tables in 3NF+ form.

Normalization

1NF → 2NF → 3NF

ranking

ranking_id (PK)	ranking_date	rank	player_id (FK)	points
Auto Incremented	20100104	1	103819	10550

players

player_id (PK)	name_first	name_last	hand	dob	country_id (FK)	height	wikidata_id
100001	Gardnar	Mulloy	R	19131122	1	185	Q54544

matches

match_id (PK)	tourney_id (FK)	match_num	winner_id (FK)	winner_seed	winner_entry
Auto Incremented	1991-339	4	101889	8	

countries

country_id (PK)	ioc
1	USA

tournaments

tourney_id (PK)	tourney_name	surface	draw_size	tourney_level	tourney_date
Auto Incremented	Adelaide	Hard	32	A	19901231

players_matches

id (PK)	player_id (FK)	match_id (FK)
Auto Incremented	1	1

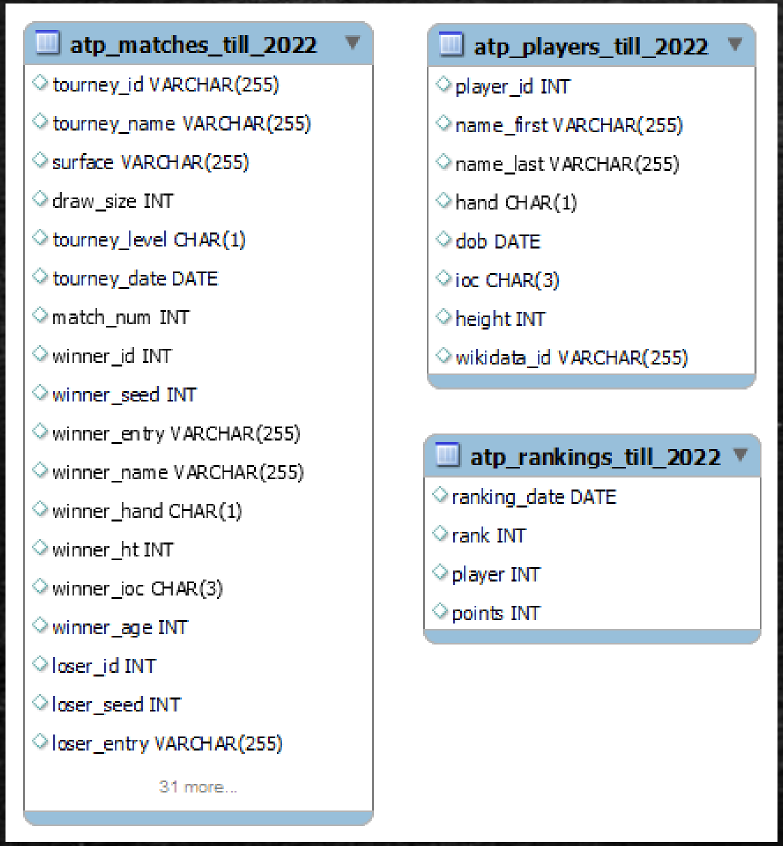
*Sample data from the CSV files used for normalization.

**Not all columns in each table shown in image due to space limitations.

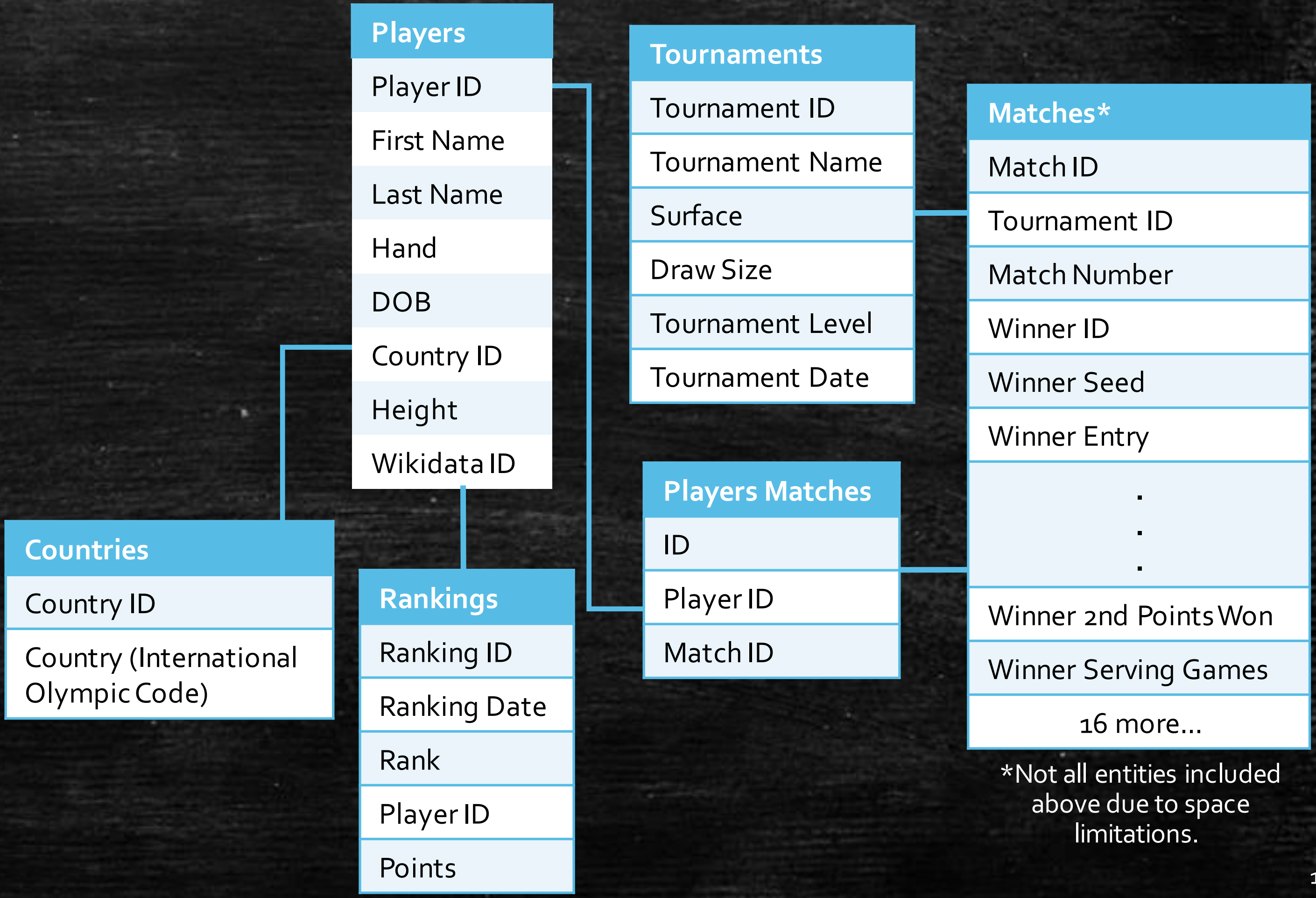
- The data from the 3 CSV files were used for normalization.
- The data was first put in 1NF then 2NF and finally in 3NF.
- The result of the normalization process is 6 relationship database tables.

Relational Modeling

Initial Form

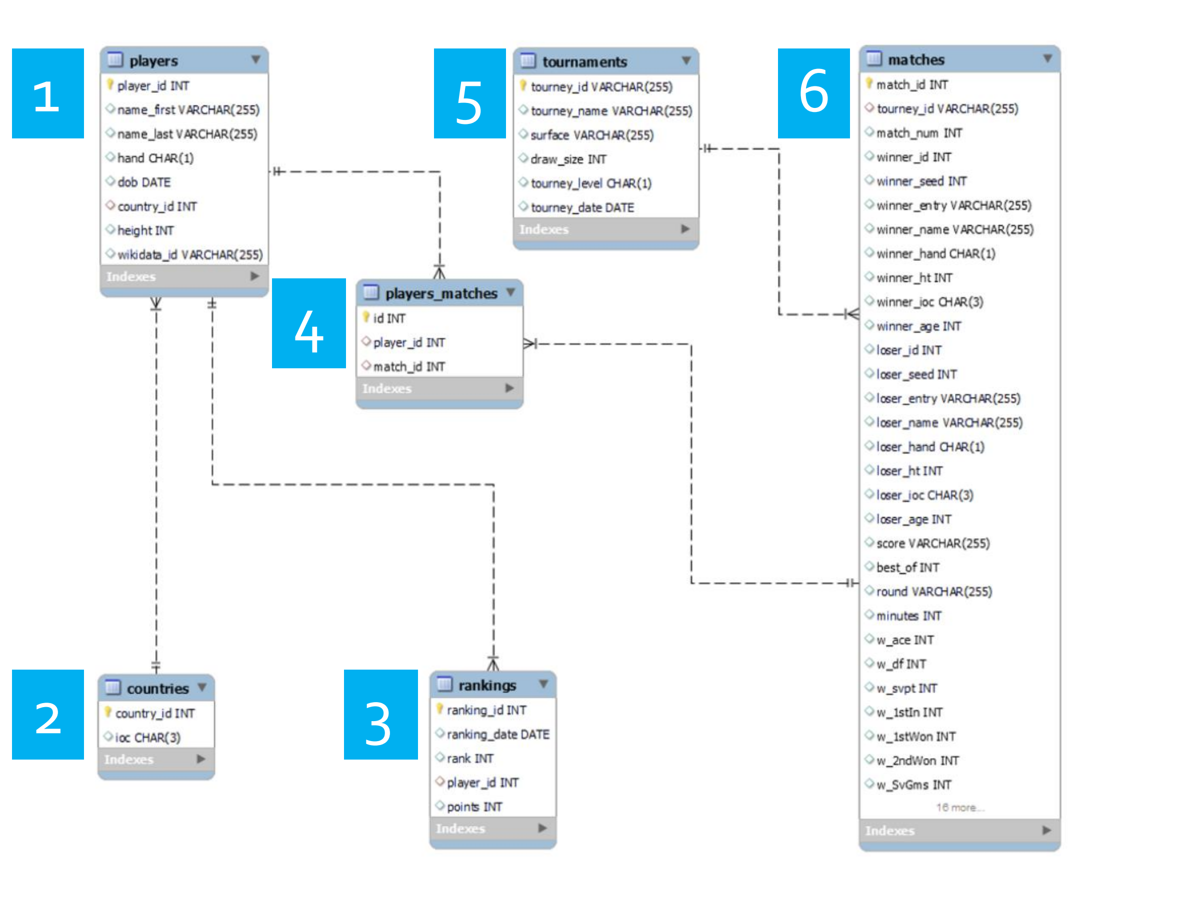


Conceptual and Logical Model Design



*Not all entities included above due to space limitations.

Enhanced Entity Relationship (EER) Model



Classical Model Schema

1. players
2. countries
3. rankings
4. players_matches
5. tournaments
6. matches

Database Creation & Data Loading

Examples from DDL and DML Scripts

DDL

- Created the database structure, such as by creating tables.
- Ensured data types appropriate for attribute.
- Preserved data integrity through primary and foreign keys and CONSTRAINT clauses.

```sql
-- ------------------------------------------------------
-- Table `Rankings`
-- ------------------------------------------------------
CREATE TABLE Rankings (
    ranking_id INT AUTO_INCREMENT PRIMARY KEY,
    ranking_date DATE,
    `rank` INT,
    player_id INT,
    points INT,
    CONSTRAINT fk_rankings_players FOREIGN KEY (player_id)
        REFERENCES Players(player_id)
        ON DELETE NO ACTION ON UPDATE NO ACTION
);
```

DML

- Loaded data from extracted CSV files into relational tables.
- Cleaned up and formatted data through SQL queries.

```sql
-- ------------------------------------------------------
-- Importing Data - `Rankings` Table
-- ------------------------------------------------------
LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\atp_rankings_till_2022_ranking_id.csv'
INTO TABLE Rankings
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES
(
    ranking_id,
    @ranking_date,
    `rank`,
    player_id,
    @points_value
)
SET
ranking_date = STR_TO_DATE(@ranking_date, '%Y%m%d'),
points = CASE
    WHEN @points_value = ' ' OR @points_value = '' OR @points_value REGEXP '^[^0-9]+$' THEN NULL
    ELSE CAST(@points_value AS SIGNED)
END;
```

Data Analysis, Reporting & Visualization

Data Analysis Overview

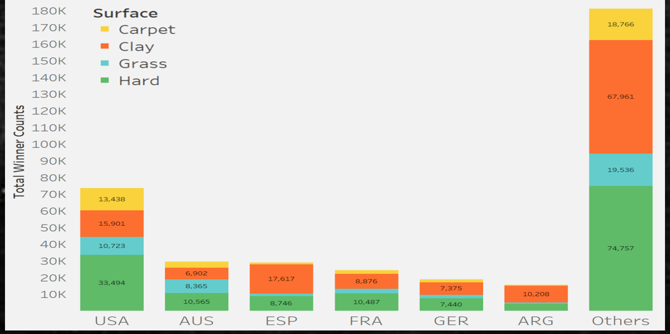
- Factors that drive player performance were approached in several folds:

Factor	Examples
Player Characteristics	• Age • Country • Dominant Hand
Players' Game Performance	• Aces • Service Points • Double Faults
External	• Surface Type • Entry • Seed
Combined (e.g., Player Characteristics and External)	• Player Winning Counts by Country

SQL (i.e., through moderately completely queries)

```sql
# Winning probability by age group

SELECT
    CASE
        WHEN m.winner_age BETWEEN 13 AND 19 THEN 'Teen'
        WHEN m.winner_age BETWEEN 20 AND 29 THEN '20s'
        WHEN m.winner_age BETWEEN 30 AND 39 THEN '30s'
        WHEN m.winner_age BETWEEN 40 AND 49 THEN '40s'
        WHEN m.winner_age BETWEEN 50 AND 59 THEN '50s'
        ELSE 'Other'
    END AS age_group,
    CONCAT(ROUND(100.0*COUNT(*)/(SELECT COUNT(*) FROM matches),2),'%') AS winningProbability
FROM players p
JOIN matches m
ON p.player_id=m.winner_id
GROUP BY age_group
ORDER BY COUNT(*)/(SELECT COUNT(*) FROM matches) DESC;
```

Tableau



Python

```python
# Filter player stats for selected countries
selected_countries = ['USA', 'ARG', 'GER', 'FRA', 'AUS', 'ESP']
filtered_player_stats = player_stats[player_stats['Country'].isin(selected_countries)]

# Prepare win/loss data from match data
match_data['Winner'] = 1  # Marking the winner
match_data['Loser'] = 0   # Marking the loser
win_data = match_data[['Winner Id', 'Winner']].rename(columns={'Winner Id': 'Player Id', 'Winner': 'Win'})
loss_data = match_data[['Loser Id', 'Loser']].rename(columns={'Loser Id': 'Player Id', 'Loser': 'Win'})
combined_results = pd.concat([win_data, loss_data])
win_rate_data = combined_results.groupby('Player Id').mean().reset_index()
```

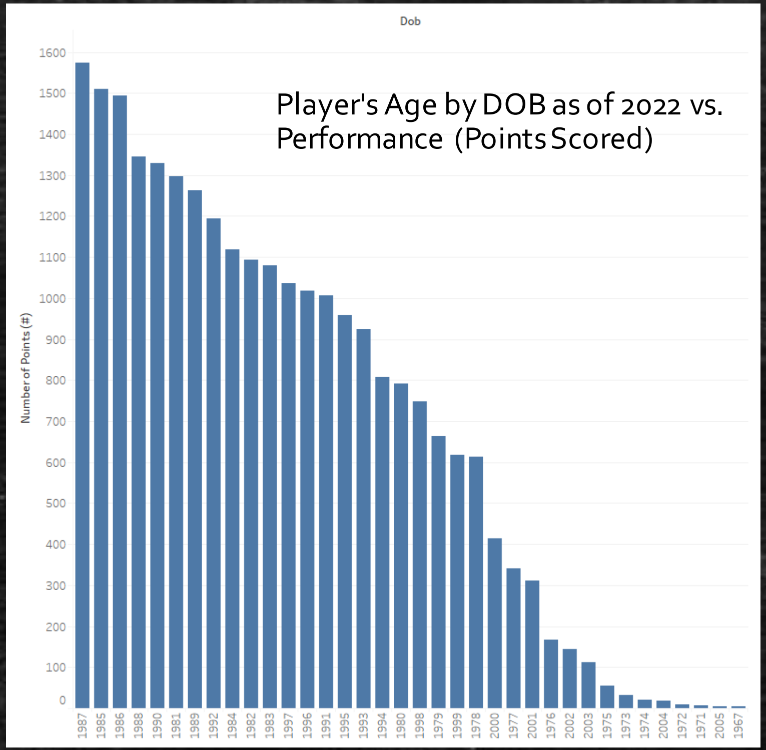
Player Characteristics

Age vs. Performance and Winning Probability

Winning Probability by Hand Used

Hand	Winning Probability
Left	14.46%
Right	84.48%

Right-handed players appear to be more likely to win. This is purely based on correlation and does not imply causation.

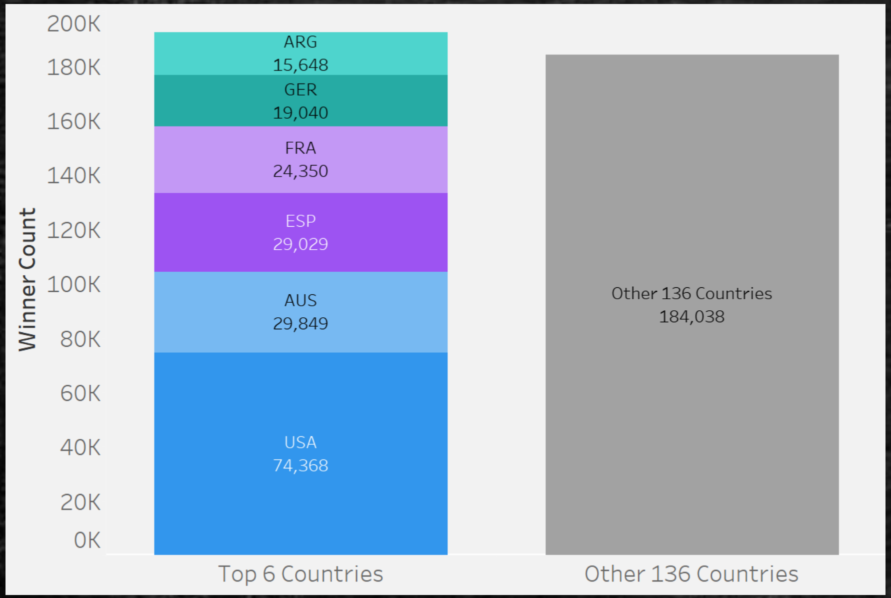


Age Group	Winning Probability
Teen	3.93%
20s	78.18%
30s	16.86%
40s	0.32%
50s	0.01%
Other	0.71%

- Players in their 20s have the highest probability to win followed by those in their 30s.
- Performance levels were suboptimal at the youngest age extremes.

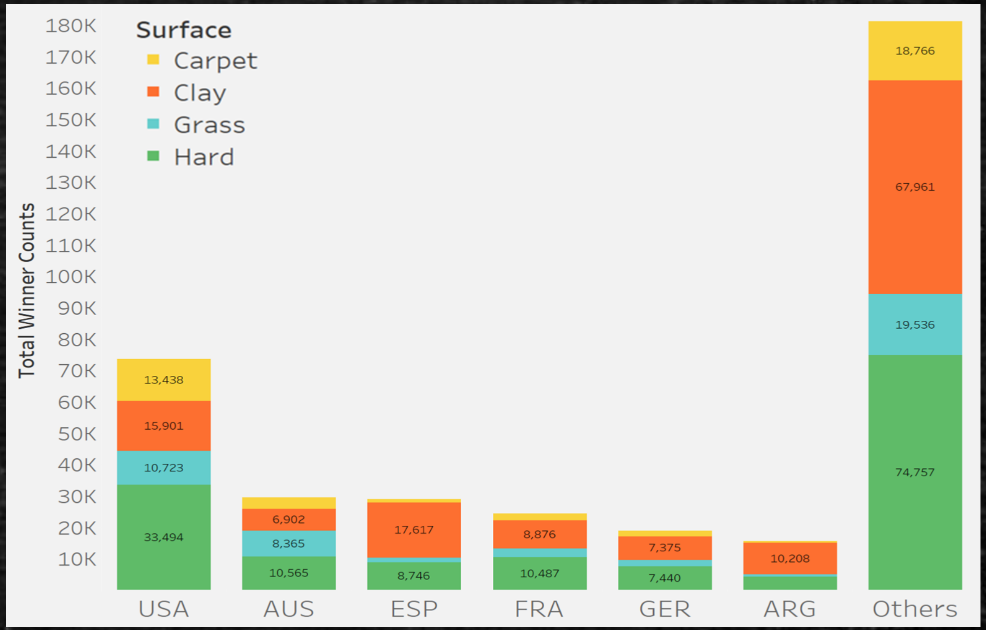
Player Country

Total Winner Counts by Country



European, Americas and Australia have the most wins in terms of countries compared to the rest of the world.

Total Winner Counts in Surface Types by Country



Clay and Hard surface types seem to be where players from most, if not all, countries perform the best.

External (Tournament) Factors

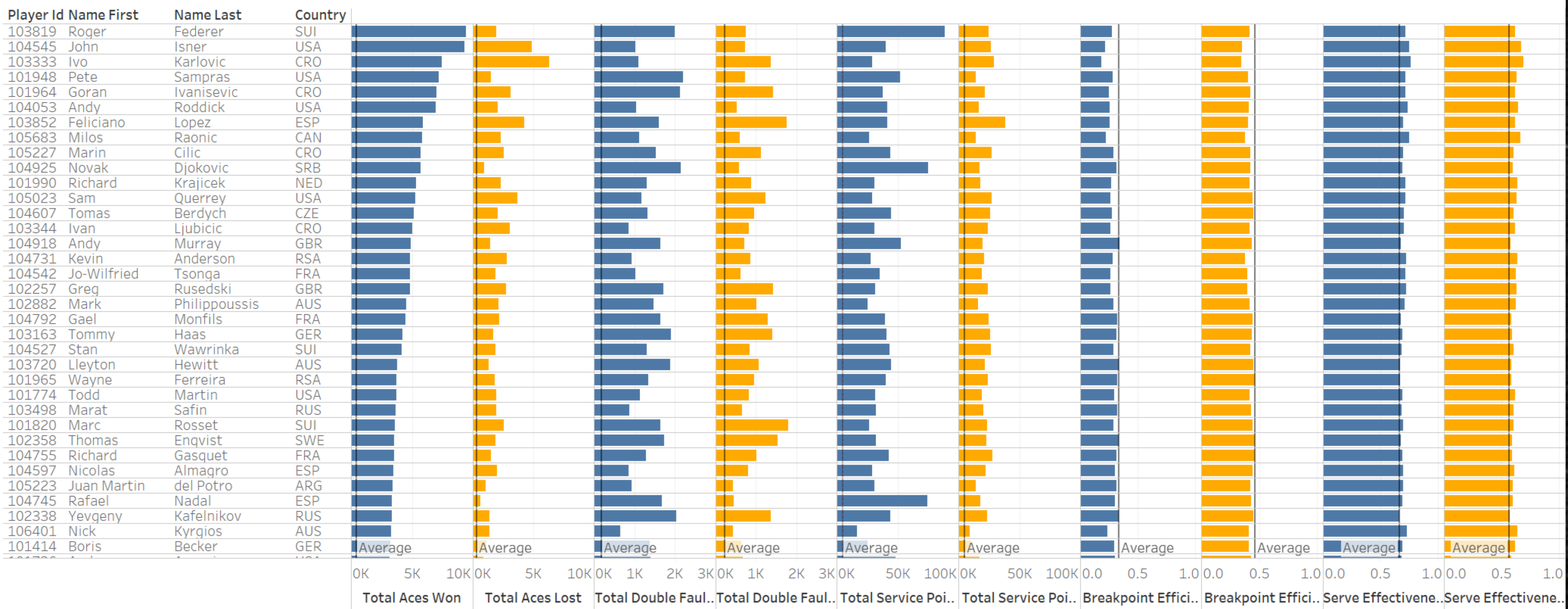
Entry Type	Winning Probability
Special Exempt	91.39%
Qualifier	5.39%
Wild Card	2.55%
Lucky Loser	0.57%
Protected Ranking	0.09%

Seed Group	Winning Probability
Low	63.15%
Lower Medium	0.96%
Upper Medium	5.14%
High	30.75%

Entry Type	Winning Probability (Loser)
Special Exempt	85.32%
Qualifier	9.06%
Wild Card	4.43%
Lucky Loser	1.09%
Protected Ranking	0.11%

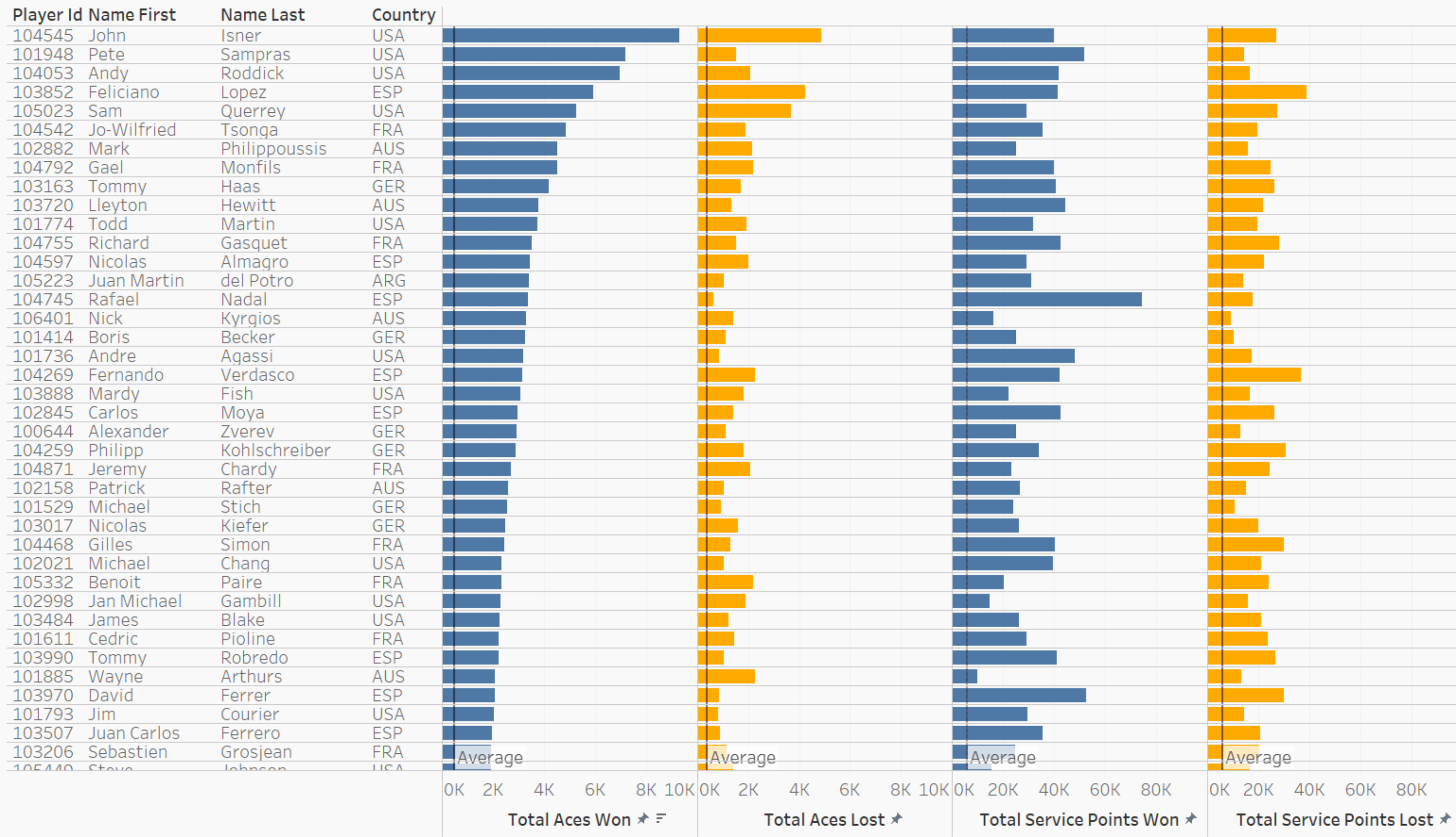
Seed Group	Winning Probability (Loser)
Low	81.38%
Lower Medium	0.73%
Upper Medium	3.46%
High	14.43%

Player Game Performance



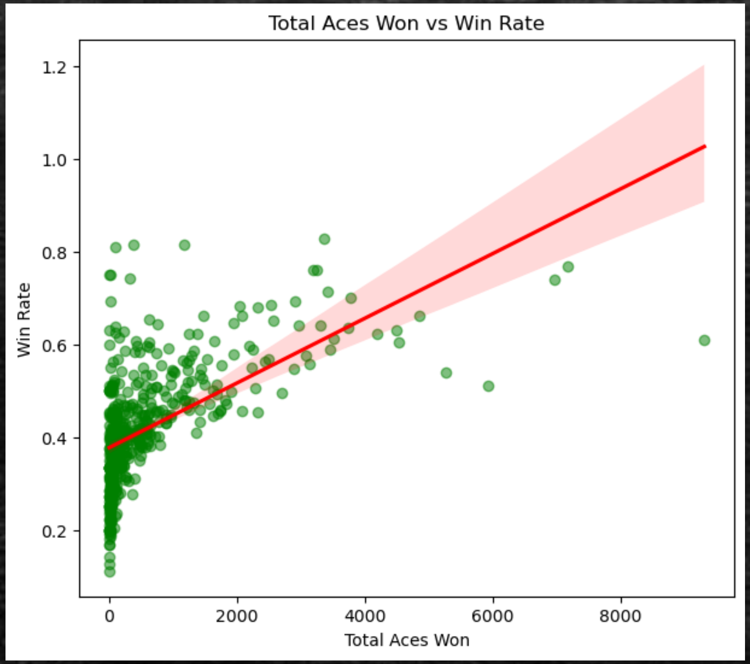
Not all factors were insightful in this visual form and that too told us something.

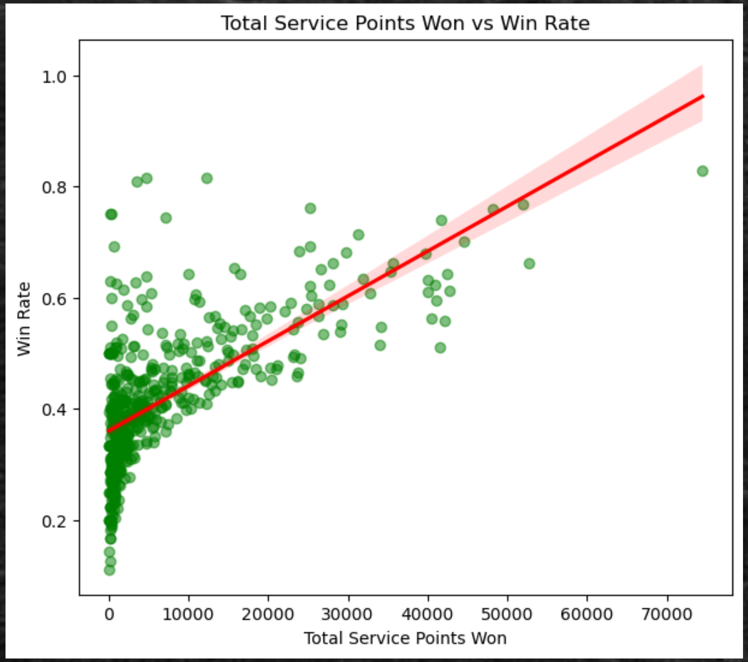
Players' Game Performance – Most Insightful Factors



- The system is overwhelmingly in favor of players with strong serves, indicating that they tend to win.
- The fact that aces are frequently scored suggests a clear distinction in skill level even among professionals.
- Therefore, monitoring specific indicators such as control and speed can help in identifying players with significantly stronger serves.

Player Game Performance – Top 6 Performing Countries

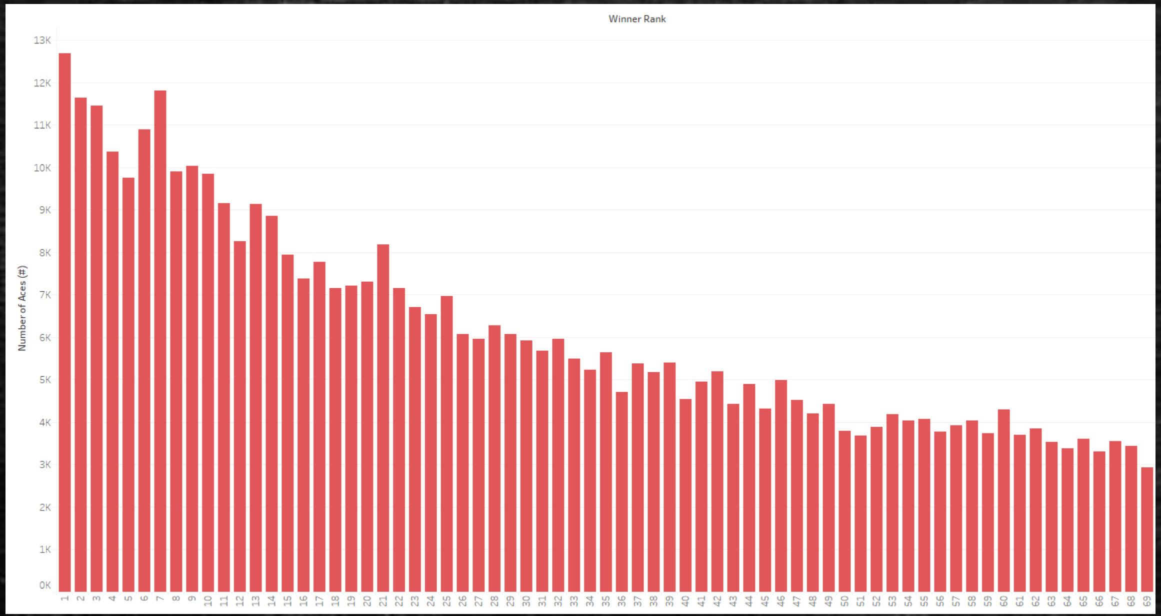




In general, there is a positive association between winning rate and total aces hit, as well as between winning rate and total service points won.

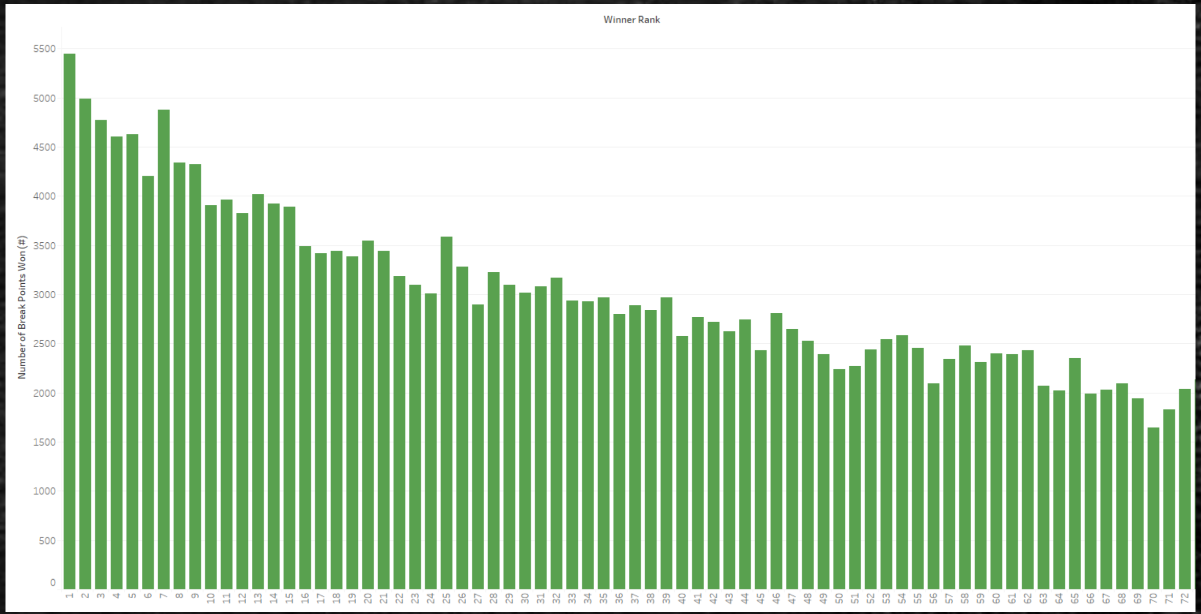
Player Ranking vs. Performance

Player's Rank vs. Performance (Aces)



- Number of aces scored exhibits a positive correlation with player rank, rising consistently as ranks increase.
- There is a noticeably consistent decline in the number of aces scored as ranks descend.

Player's Rank vs. Performance (Break Points Saved)



- Similarly, to aces scored, breakpoints won also steadily decreases as player rank decreases.

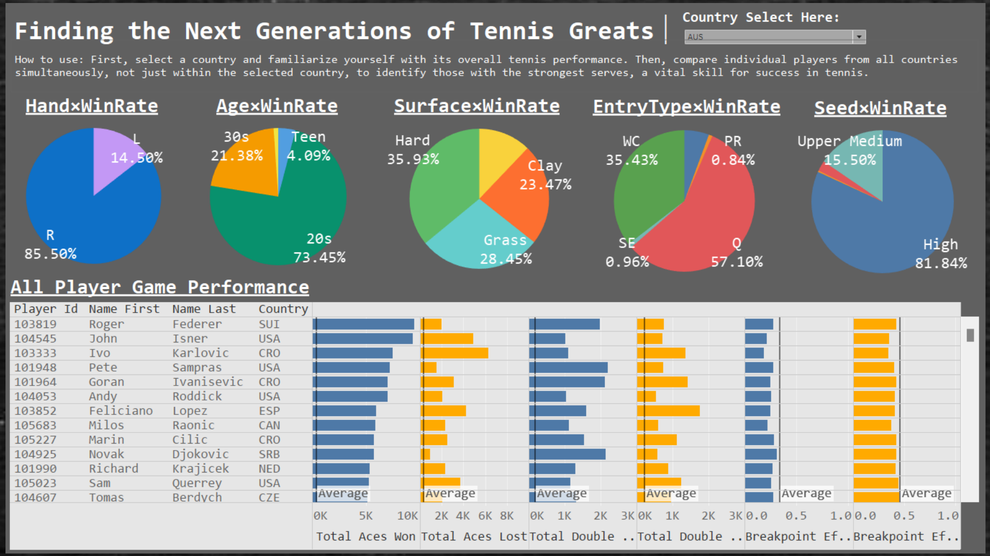


Tableau Dashboard Demo

Conclusion(s)

Recommendations

- Focus on players from USA, Australia, Spain, France, Germany and Australia.
  - Players from these countries play best on Clay and Hard surfaces. If players from the top 6 countries are signed, they should be encouraged to play in tournaments with these surfaces.
  - Observe a player's past game performance:
    - If they are more likely to win service points and hit aces, they have a positive association with winning and ranking higher.
    - More double faults mean their losing probability increases.
- Special Exempt players have the highest chance of winning.
- Players should be encouraged to play as many tournaments as they can, as earning any form of points will help their ranking. Higher ranks mean lower seeds, which means there is a higher probability of winning.
- Players are most in their prime in their 20s, so those players should be targeted. However, players in their 30s show success and can be considered too.
- Player's dominant hand is not a factor that should be considered as the findings do not point to causation.

Reflection

Scope for Improvement

- Given that the data was collected over a period of time, a historical time analysis could have been done to evaluate performance on a yearly basis.
- There were a lot of NULL values for certain attributes. Using other sources to fill the gaps could have been done.
- Using models and regression analysis to determine ideal predictors of interest to explain player performance.

Corrective Measures

- Using more interaction predictors, such as player hand by country.
- Using platforms such as OpenRefine or R for data preparation instead of SQL.

Lessons Learned

Group's Perspective

- Learning the intricacies of tennis datasets, tennis rules, and the data needed when it comes to analyzing tennis-related projects.
- Creating DDL and DML scripts and code based off extracted data.

Client Perspective

- Factors that they should look for when considering players to sponsor.
- The factors go beyond just looking at one specific aspect, and instead should include interactions/combined factors.

Appendix

Examples of SQL Code and Output*

SQL queries included using joins, CASE statements, functions sorting and grouping data, and created view.

```sql
SELECT
    c.ioc AS countryInitials,
    p.name_first AS firstName,
    p.name_last AS lastName,
    COUNT(a.winner_id) as winnerCount
FROM
    countries c
        INNER JOIN
    players p ON c.country_id = p.country_id
        INNER JOIN
    players_matches m ON p.player_id = m.player_id
        INNER JOIN
    matches a ON m.match_id = a.match_id
        INNER JOIN
    tournaments t ON a.tourney_id = t.tourney_id
GROUP BY
    c.ioc, p.name_first, p.name_last
ORDER BY
    c.ioc, COUNT(a.winner_id) DESC;
```

```sql
CREATE VIEW WinnerSeedWinningProbability AS
SELECT
    CASE
        WHEN m.winner_seed BETWEEN 1 AND 10 THEN 'High'
        WHEN m.winner_seed BETWEEN 11 AND 20 THEN 'Upper Medium'
        WHEN m.winner_seed BETWEEN 21 AND 30 THEN 'Lower Medium'
        ELSE 'Low'
    END AS seedGroup,
    CONCAT(ROUND(100.0*COUNT(*)/(SELECT COUNT(*) FROM matches),2),'%') AS winningProbability
FROM players p
JOIN matches m
ON p.player_id=m.winner_id
GROUP BY seedGroup
ORDER BY COUNT(*)/(SELECT COUNT(*) FROM matches) DESC;
```

player_id	name_first	name_last	WinnerName	WinnerAge	LoserName	LoserAge	TourneyLevel
100001	Gardnar	Mulloy	Mark Cox	54	Gardnar Mulloy	63	A
100002	Pancho	Segura	Torben Ulrich	50	Richard Gonzalez	53	G
100003	Frank	Sedgman	Tony Roche	47	Victor Eke	49	M
100004	Giuseppe	Merlo	Teimuraz Kakulia	46	Ray Keldie	46	M
100005	Richard	Gonzalez	Vijay Amritraj	48	Zeljko Franulovic	48	M
100006	Grant	Golden	Larry Turville	29	Grant Golden	49	M
100007	Abe	Segal	Zeljko Franulovic	41	Pieter Soeters	41	G
100009	Istvan	Gulyas	Zeljko Franulovic	42	Wilhelm Bungert	45	M
100010	Luis	Ayala	Tony Roche	39	Steve Turner	49	M
100011	Torben	Ulrich	Wilhelm Bungert	45	Torben Ulrich	53	M
100012	Nicola	Pietrangeli	Zeljko Franulovic	39	Thomas Lejus	44	M
100013	Neale	Fraser	Syd Ball	41	Takeshi Koura	42	G
100014	Trevor	Fancutt	Trevor Fancutt	34	Trevor Fancutt	40	G
100015	Sammy	Giammalva	Roy Emerson	32	Sammy Giammalva	37	G
100016	Ken	Rosewall	Zeljko Franulovic	46	Zeljko Franulovic	49	M
100017	Mal	Anderson	Vijay Amritraj	47	Yong Ho Chung	47	G
100018	Barry	Mackay	Tom Gorman	35	William Brown	39	M
100019	Wieslaw	Gasiorek	Zeljko Franulovic	40	Wieslaw Gasiorek	39	M
100020	Alejandro	Olmedo	Vladimir Zednik	47	Zan Guerry	41	M
100021	Ashley	Cooper	Raul Ramirez	32	Isao Watanabe	37	G
100022	Roy	Emerson	Zeljko Franulovic	41	Zeljko Franulovic	49	M
100023	Ramanat...	Krishnan	Wilhelm Bungert	40	Warren Jacques	40	M
100024	Jan Erik	Lundquist	Wilhelm Bungert	38	Zeljko Franulovic	38	M
100025	Barry	Phillips M...	Zeljko Franulovic	41	William Brown	64	M

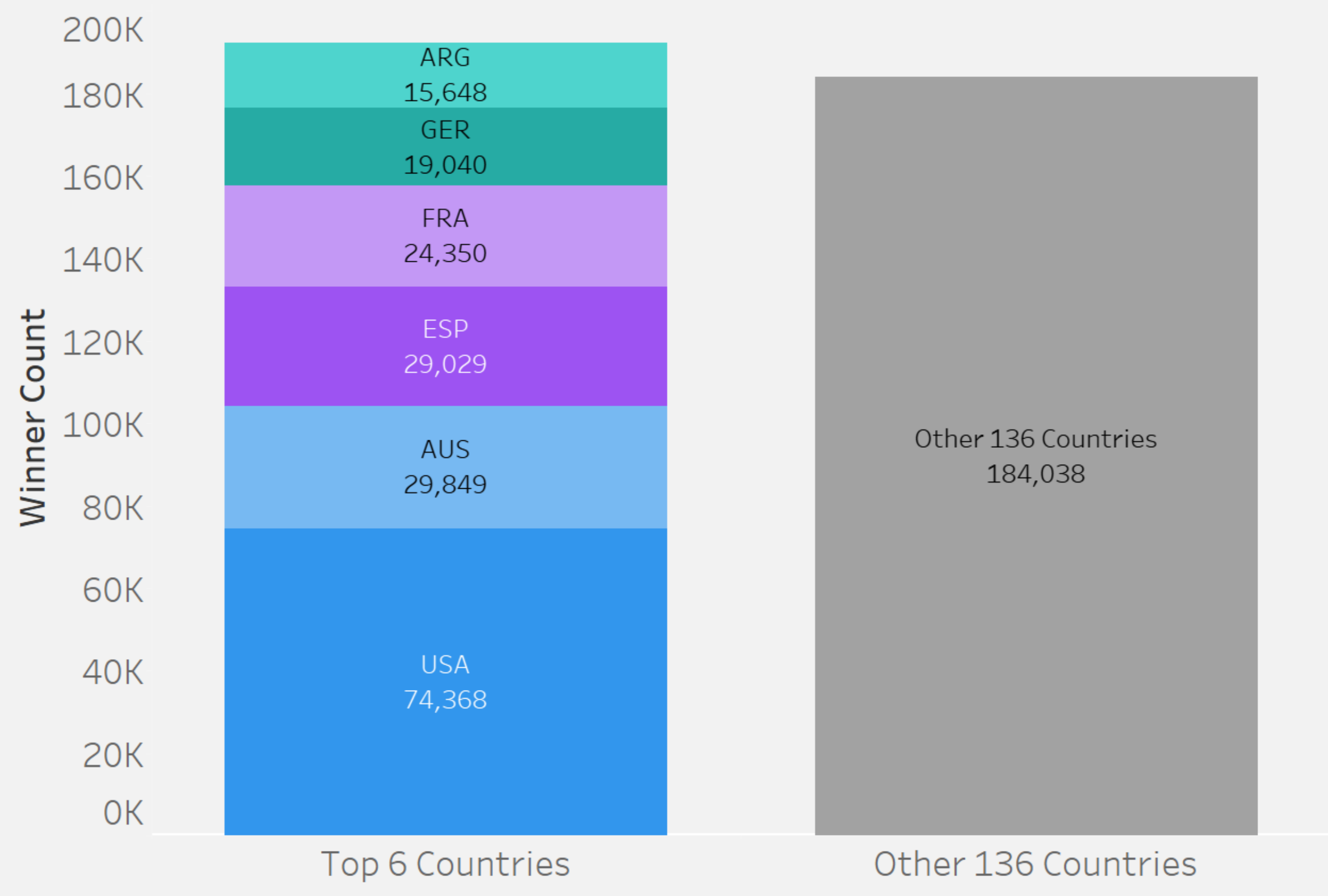
*Please review SQL files for full DML, DDL, and Queries.

Snapshot of Python Code for Data Analysis

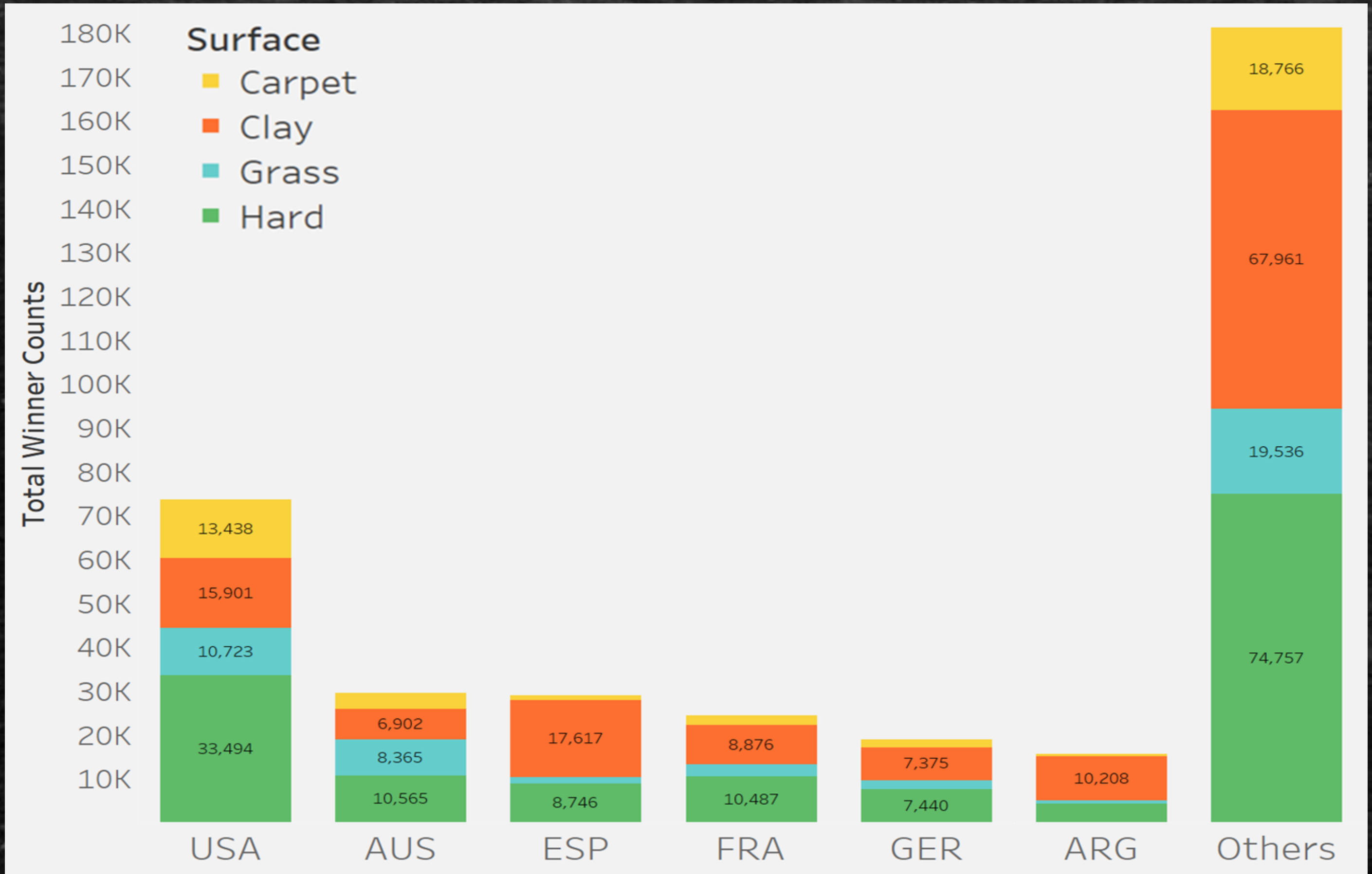
```
[5]: # Filter player stats for selected countries
selected_countries = ['USA', 'ARG', 'GER', 'FRA', 'AUS', 'ESP']
filtered_player_stats = player_stats[player_stats['Country'].isin(selected_countries)]

# Prepare win/loss data from match data
match_data['Winner'] = 1  # Marking the winner
match_data['Loser'] = 0   # Marking the loser
win_data = match_data[['Winner Id', 'Winner']].rename(columns={'Winner Id': 'Player Id', 'Winner': 'Win'})
loss_data = match_data[['Loser Id', 'Loser']].rename(columns={'Loser Id': 'Player Id', 'Loser': 'Win'})
combined_results = pd.concat([win_data, loss_data])
win_rate_data = combined_results.groupby('Player Id').mean().reset_index()

[6]: # Merge with the filtered player stats data
merged_data = pd.merge(filtered_player_stats, win_rate_data, on='Player Id', how='inner')

[7]: # Apply thresholds to filter out potential non-active players or incomplete records
threshold_aces = 10
threshold_service_points = 50
filtered_data = merged_data[(merged_data['Total Aces Won'] >= threshold_aces) &
                            (merged_data['Total Service Points Won'] >= threshold_service_points)]

[8]: # Perform linear regression analysis
X_aces_filtered = sm.add_constant(filtered_data[['Total Aces Won']])
X_service_points_filtered = sm.add_constant(filtered_data[['Total Service Points Won']])
y_filtered = filtered_data['Win']
model_aces_filtered = sm.OLS(y_filtered, X_aces_filtered).fit()
model_service_points_filtered = sm.OLS(y_filtered, X_service_points_filtered).fit()

[12]: # Total Aces Won vs Win Rate plot with green scatter points and a red regression line
fig1, ax1 = plt.subplots(figsize=(7, 6))
sns.regplot(x='Total Aces Won', y='Win', data=filtered_data, ax=ax1,
            scatter_kws={'color': 'green', 'alpha': 0.5}, line_kws={'color': 'red'})
ax1.set_title('Total Aces Won vs Win Rate')
ax1.set_xlabel('Total Aces Won')
ax1.set_ylabel('Win Rate')
plt.show()
```

1. Extracted data from SQL
2. Used Python to clean data up further for regression analysis
3. Created visuals based on analysis

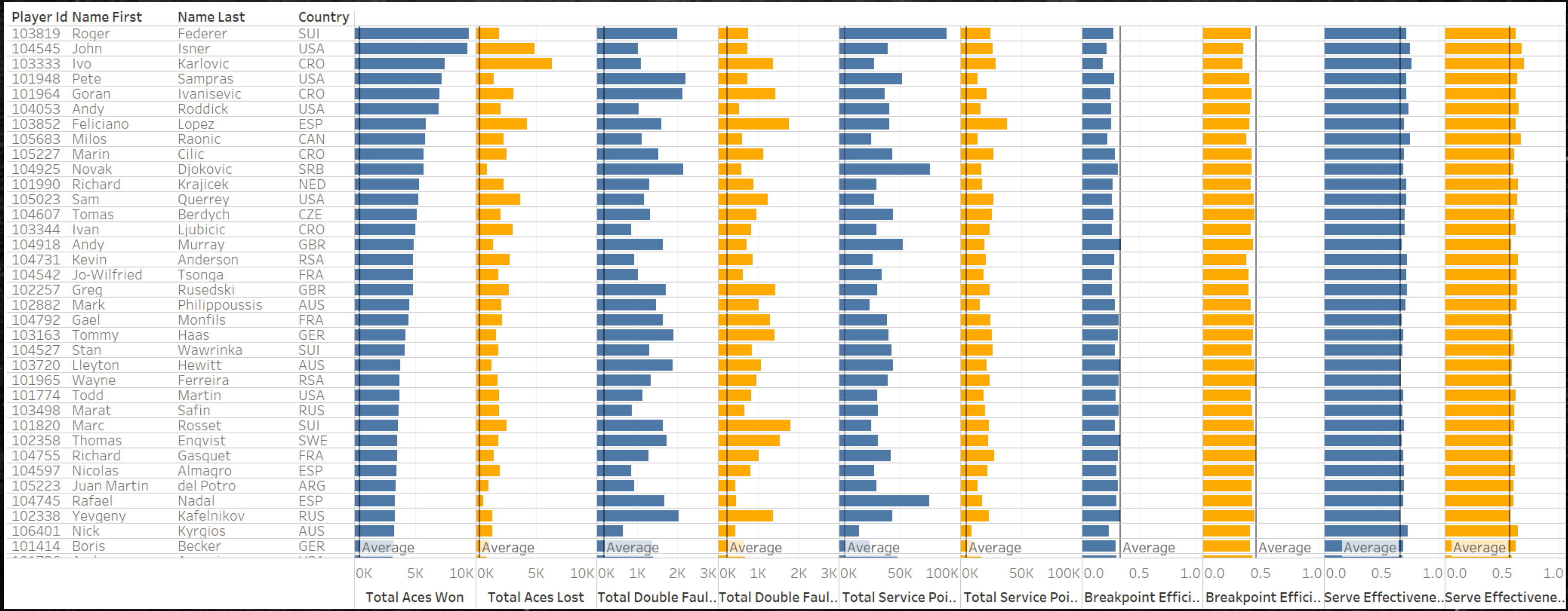
Visualization 1 (Tableau)
Winner Counts by Country



Visualization 2 (Tableau)
Winner Counts by Surface

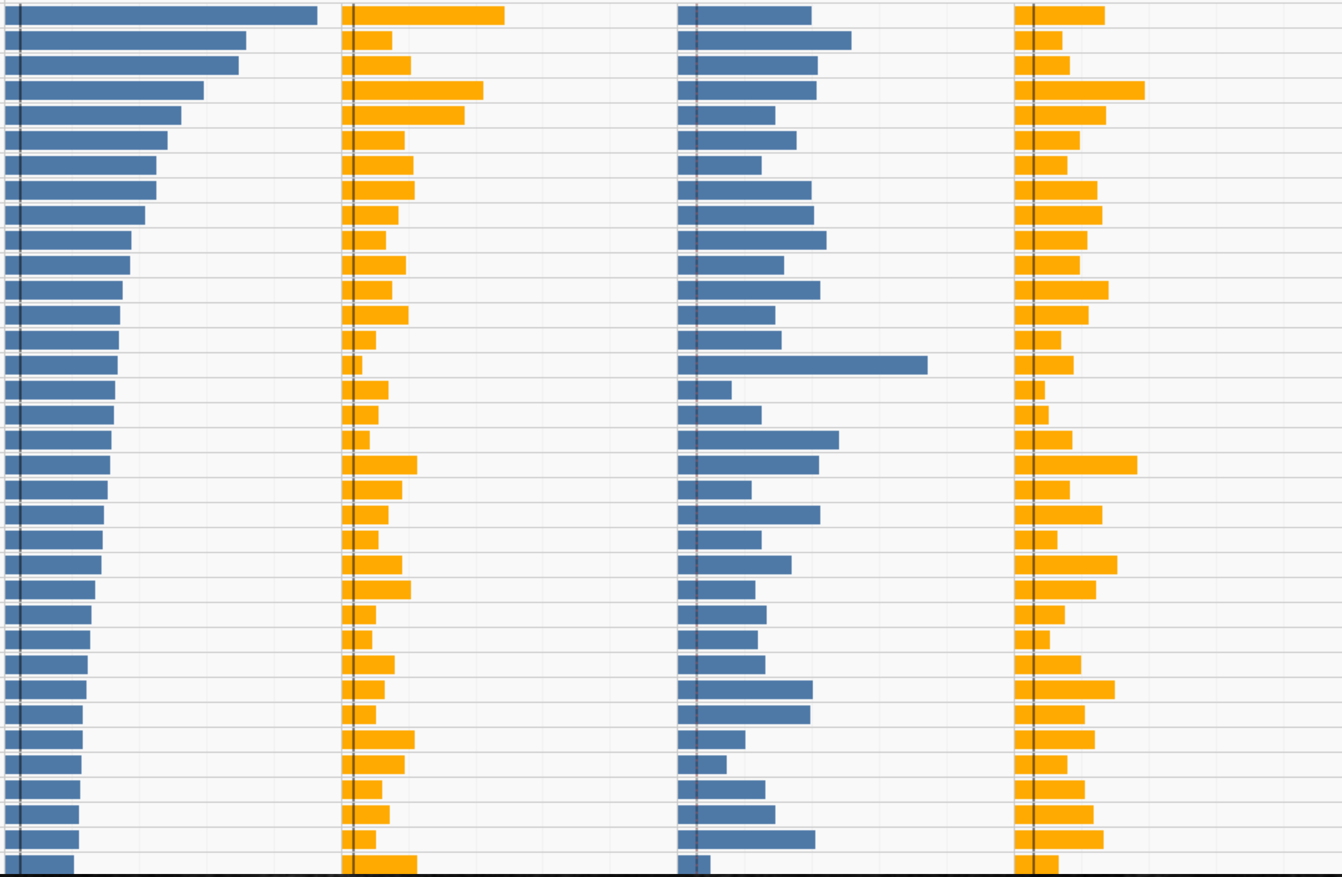


Visualization 3 (Tableau)
Player's Game Performance

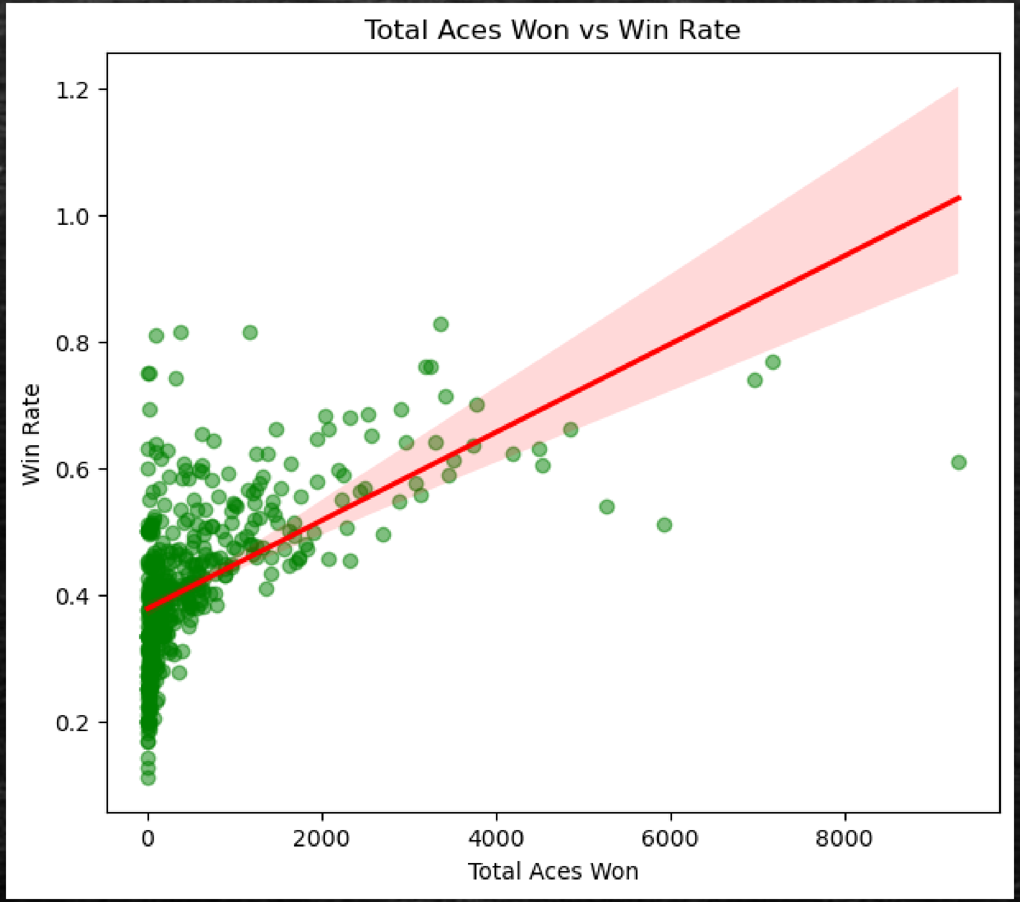


Visualization 4 (Tableau)
Players' Game Performance – Most Insightful Factors

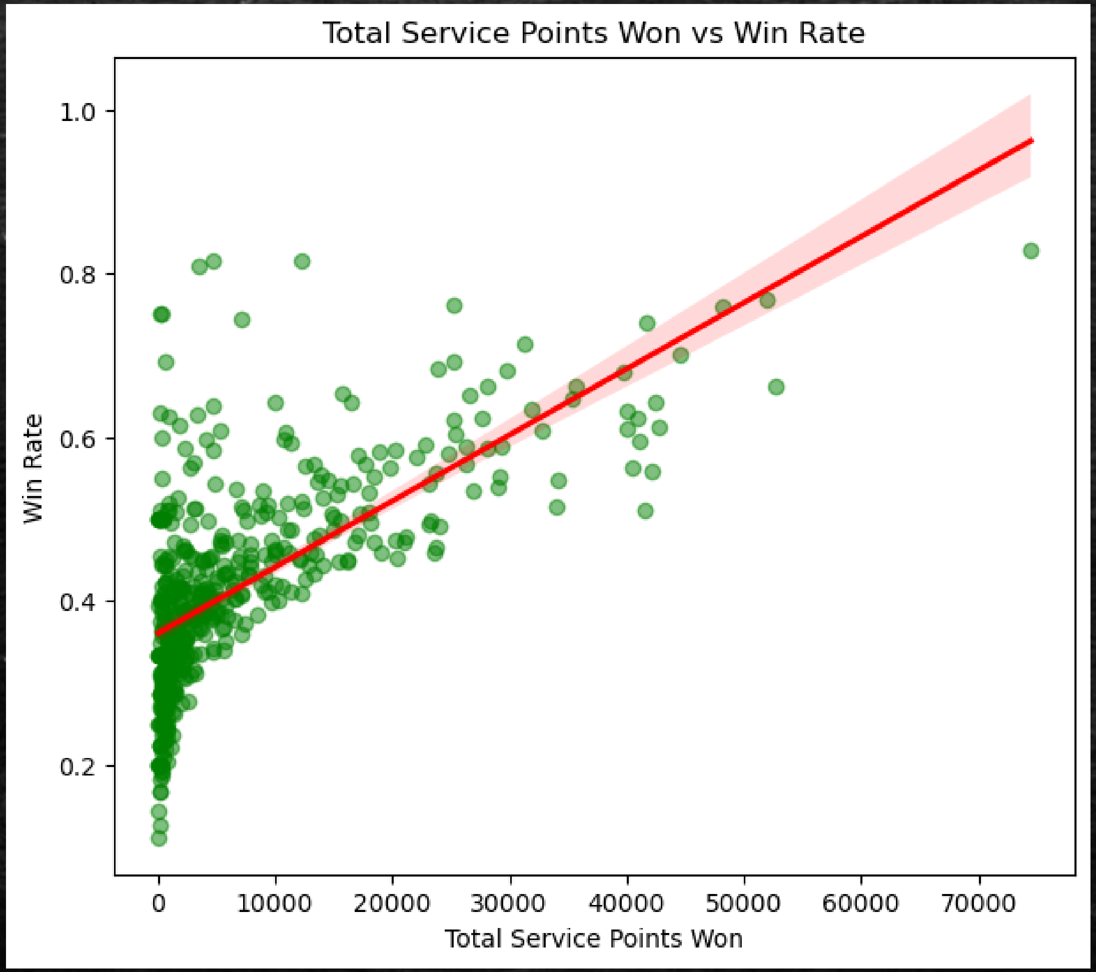
Player Id	Name First	Name Last	Country
104545	John	Isner	USA
101948	Pete	Sampras	USA
104053	Andy	Roddick	USA
103852	Feliciano	Lopez	ESP
105023	Sam	Querrey	USA
104542	Jo-Wilfried	Tsonga	FRA
102882	Mark	Philippoussis	AUS
104792	Gael	Monfils	FRA
103163	Tommy	Haas	GER
103720	Lleyton	Hewitt	AUS
101774	Todd	Martin	USA
104755	Richard	Gasquet	FRA
104597	Nicolas	Almagro	ESP
105223	Juan Martin	del Potro	ARG
104745	Rafael	Nadal	ESP
106401	Nick	Kyrgios	AUS
101414	Boris	Becker	GER
101736	Andre	Agassi	USA
104269	Fernando	Verdasco	ESP
103888	Mardy	Fish	USA
102845	Carlos	Moya	ESP
100644	Alexander	Zverev	GER
104259	Philipp	Kohlschreiber	GER
104871	Jeremy	Chardy	FRA
102158	Patrick	Rafter	AUS
101529	Michael	Stich	GER
103017	Nicolas	Kiefer	GER
104468	Gilles	Simon	FRA
102021	Michael	Chang	USA
105332	Benoit	Paire	FRA
102998	Jan Michael	Gambill	USA
103484	James	Blake	USA
101611	Cedric	Pioline	FRA
103990	Tommy	Robredo	ESP
101885	Wayne	Arthurs	AUS



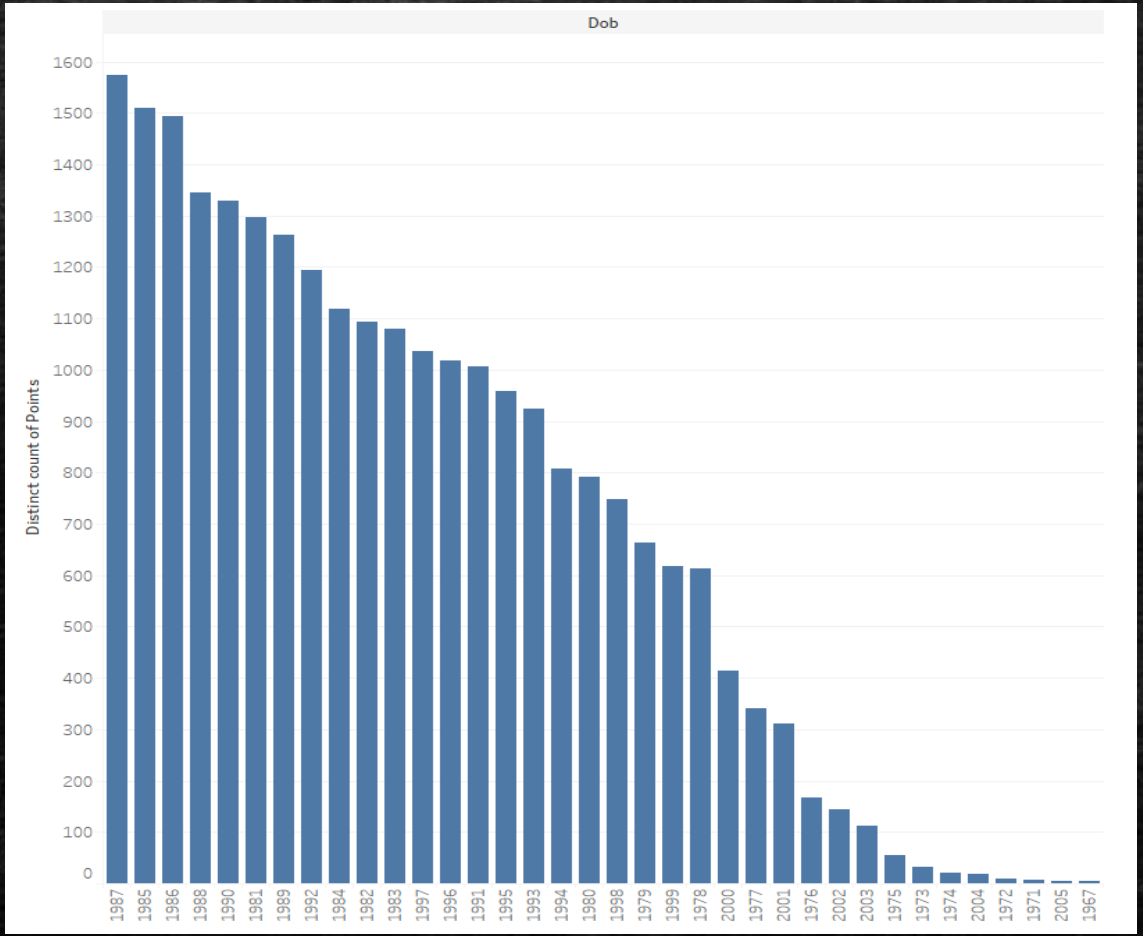
Visualization 5 (Python)
Total Aces Won and Winning Rate



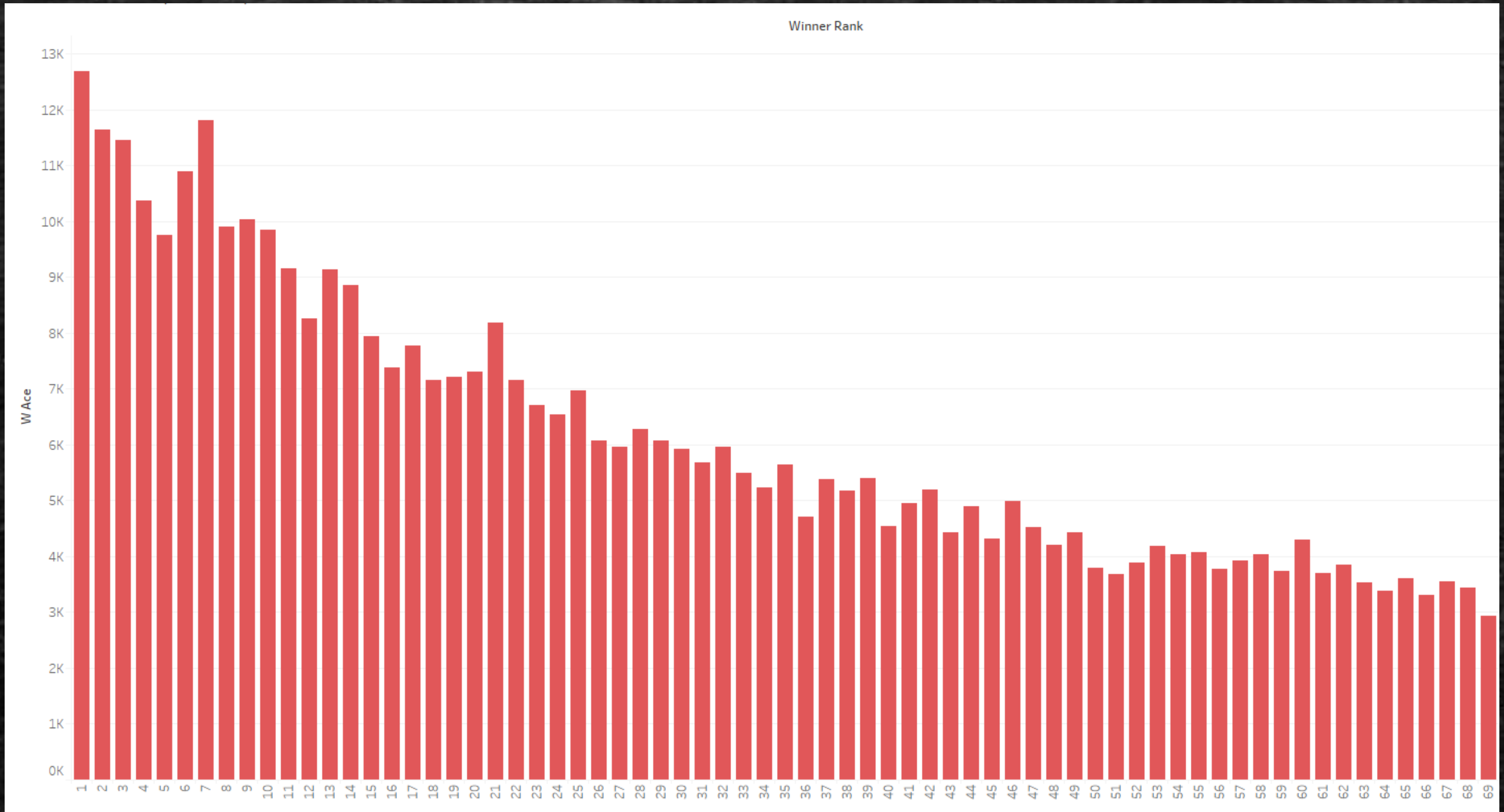
Visualization 6 (Python)
Total Service Points Won and Winning Rate



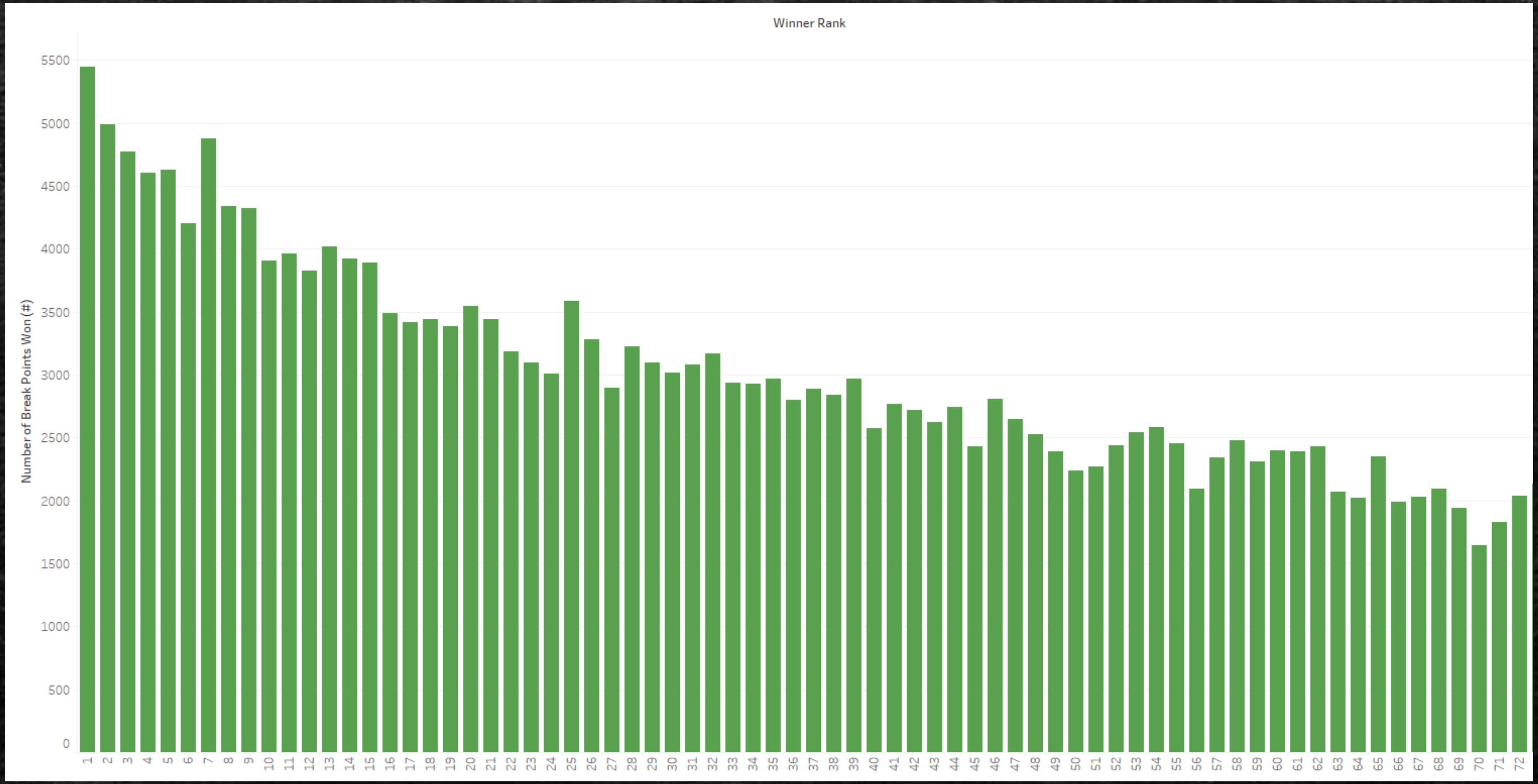
Visualization 7 (Tableau): Total Points Earned by Year of Birth



Visualization 8 (Tableau): Evaluation of Player Rank vs. Performance (Number of Aces Hit)



Visualization 10 (Tableau): Evaluation of Player Rank vs. Performance (Number of Breakpoints Won)



Dashboard 1 (Tableau)