An Analysis of Probabilistic Factors in Association of Tennis Professionals Winners

Final Project

Name: Rares Finatan
Student Number: 685688202
Course: IST 687
Term: Fall/Winter 2022

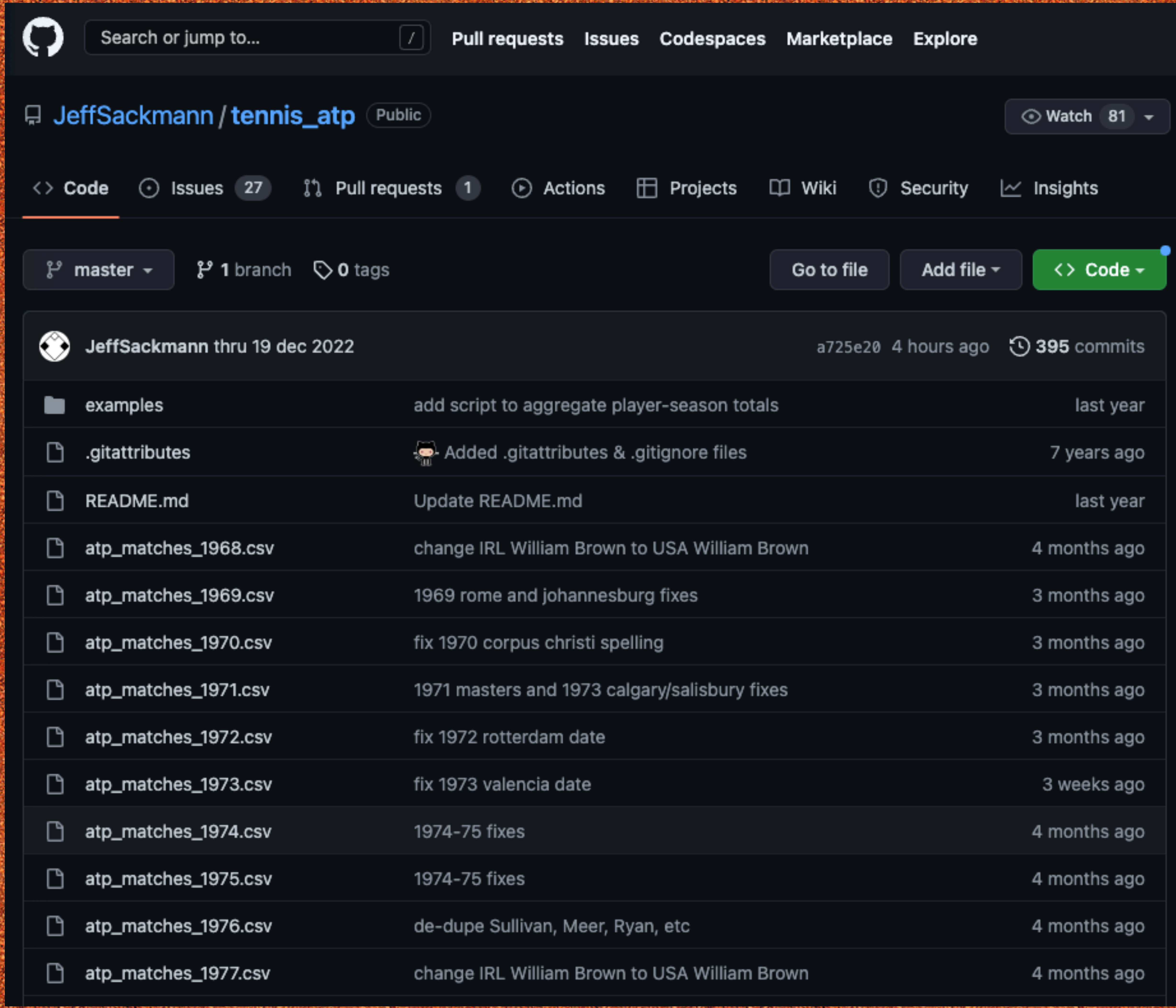
Abstract and Project Overview

An Analysis of Probabilistic Factors in Association of Tennis Professionals

- This study aims to identify the highest-contributing attributes of an Association of Tennis Professionals (ATP) winner's performance for the years 2000 to 2022.
- The study is aimed at tennis fans and sports betting enthusiasts looking to gain an understanding of a player's performance from a list of 589 ATP-registered players across 54,276 matches.
- The study will attempt to engineer features for modelling pertaining to player matchups, environmental scenarios, and tournament-specific performance.

Data and Methodology

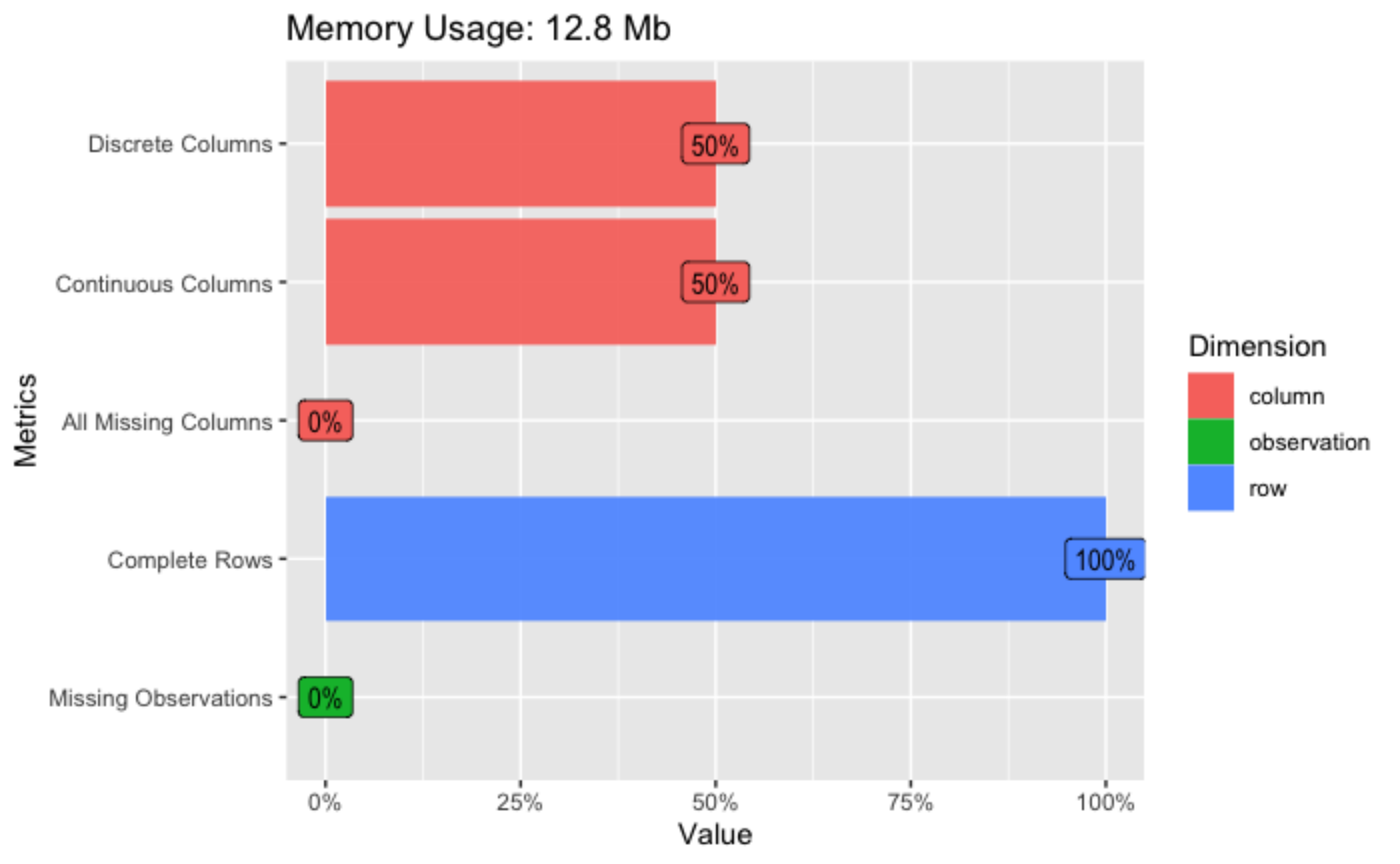
1. Hypothesis writing and initial problem framing
2. Data collection and data imports
3. Data exploration
4. Data cleansing and feature selection
5. Feature engineering
6. Model(s) creation
7. Model(s) evaluation
8. Conclusions



Preliminary Data Analysis

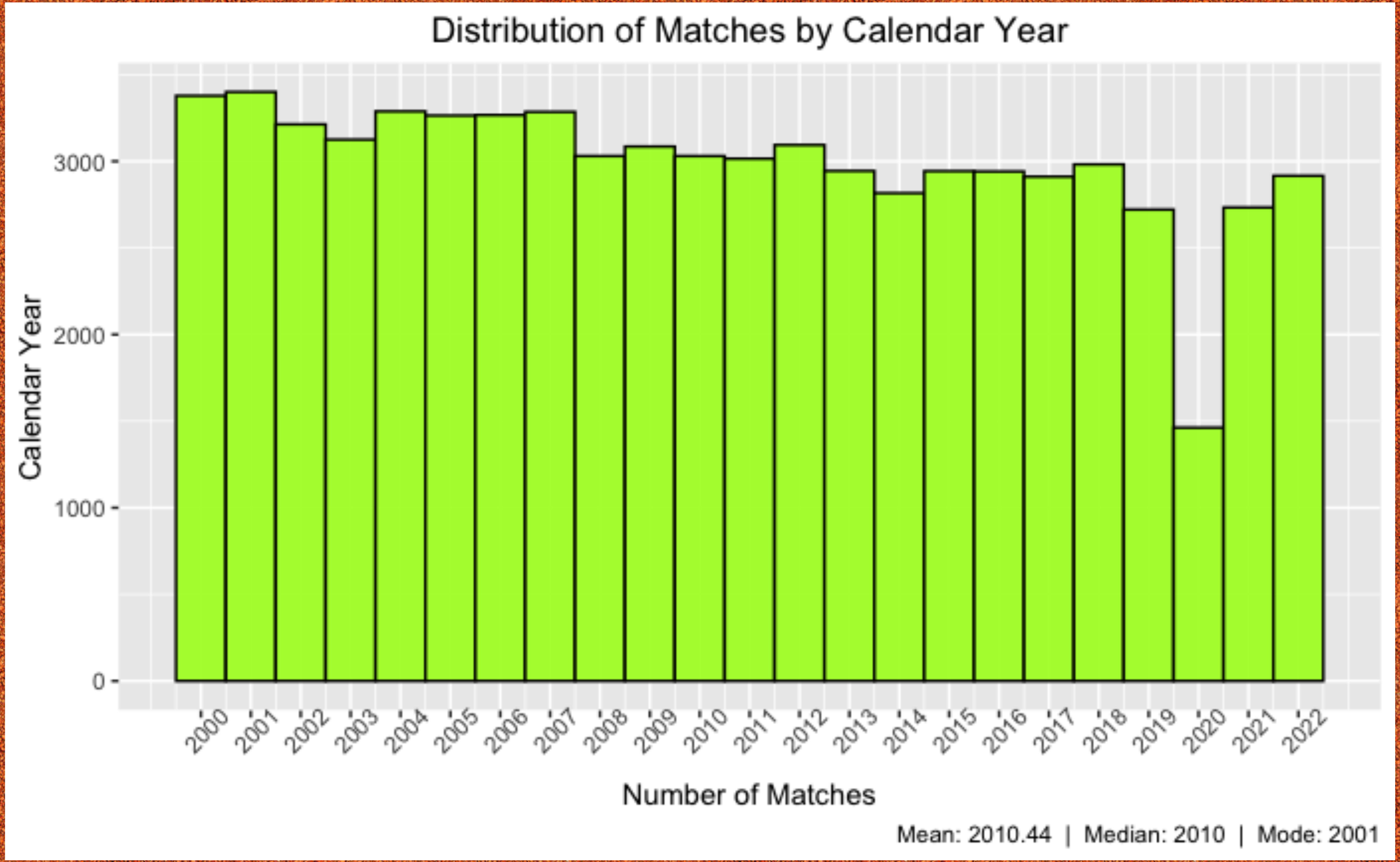
Summary statistics, data distribution

- Removing attributes with high NA%
- Removing highly correlated attributes
- Removing NAs and zero observations where imputation not possible
- Context-specific cleansing (COVID-19)



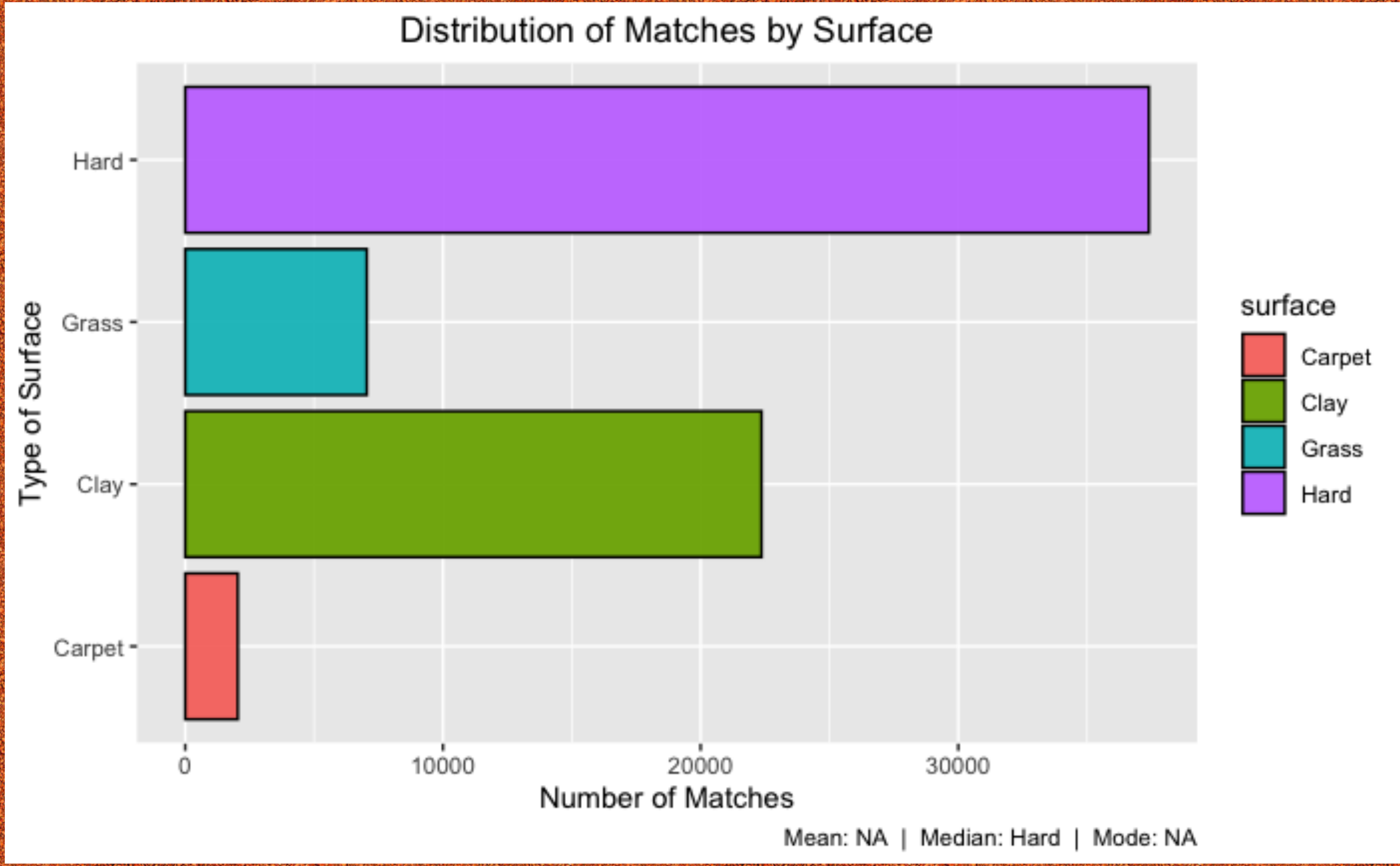
Preliminary Data Analysis

Summary statistics, data distribution



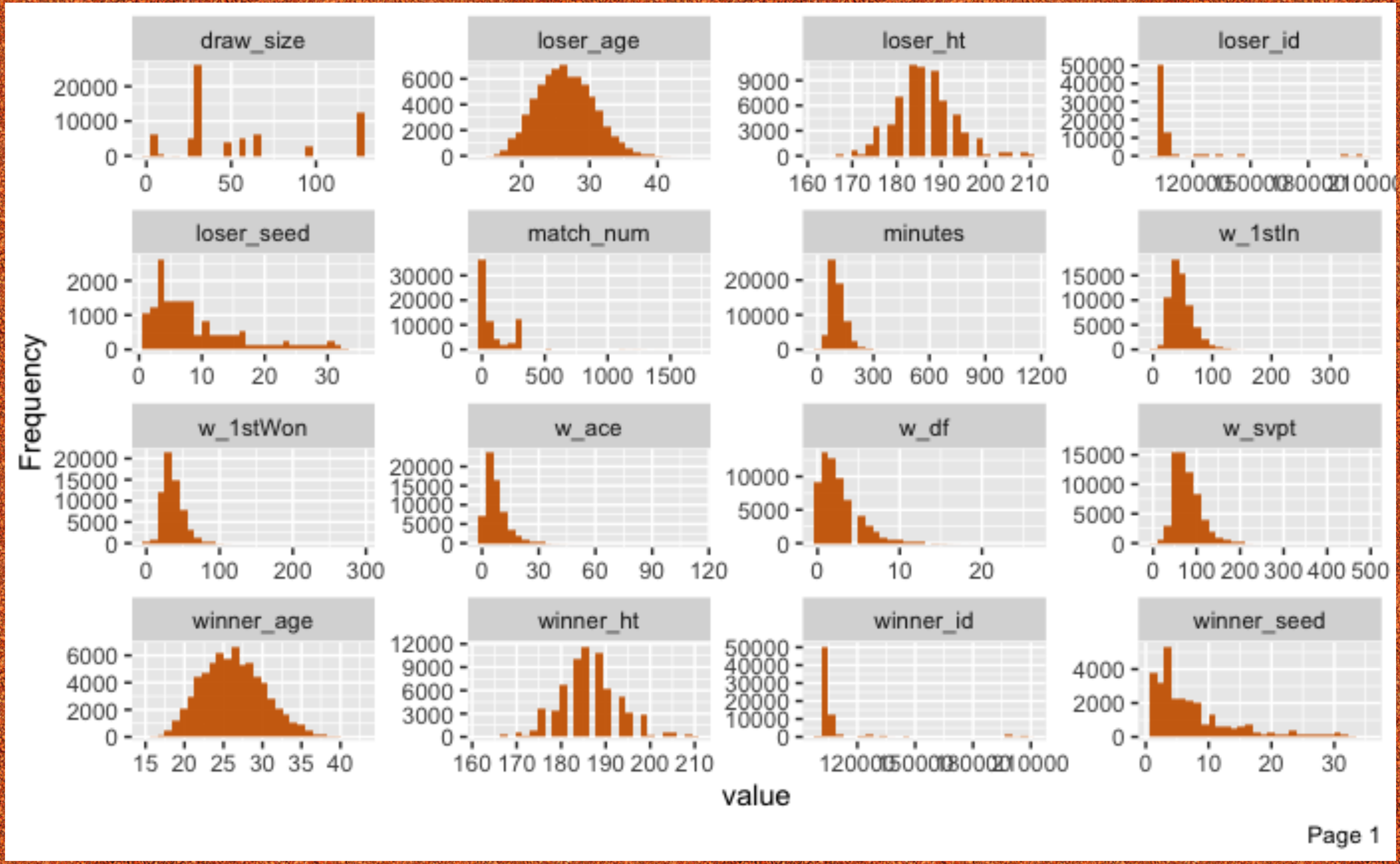
Preliminary Data Analysis

Summary statistics, data distribution



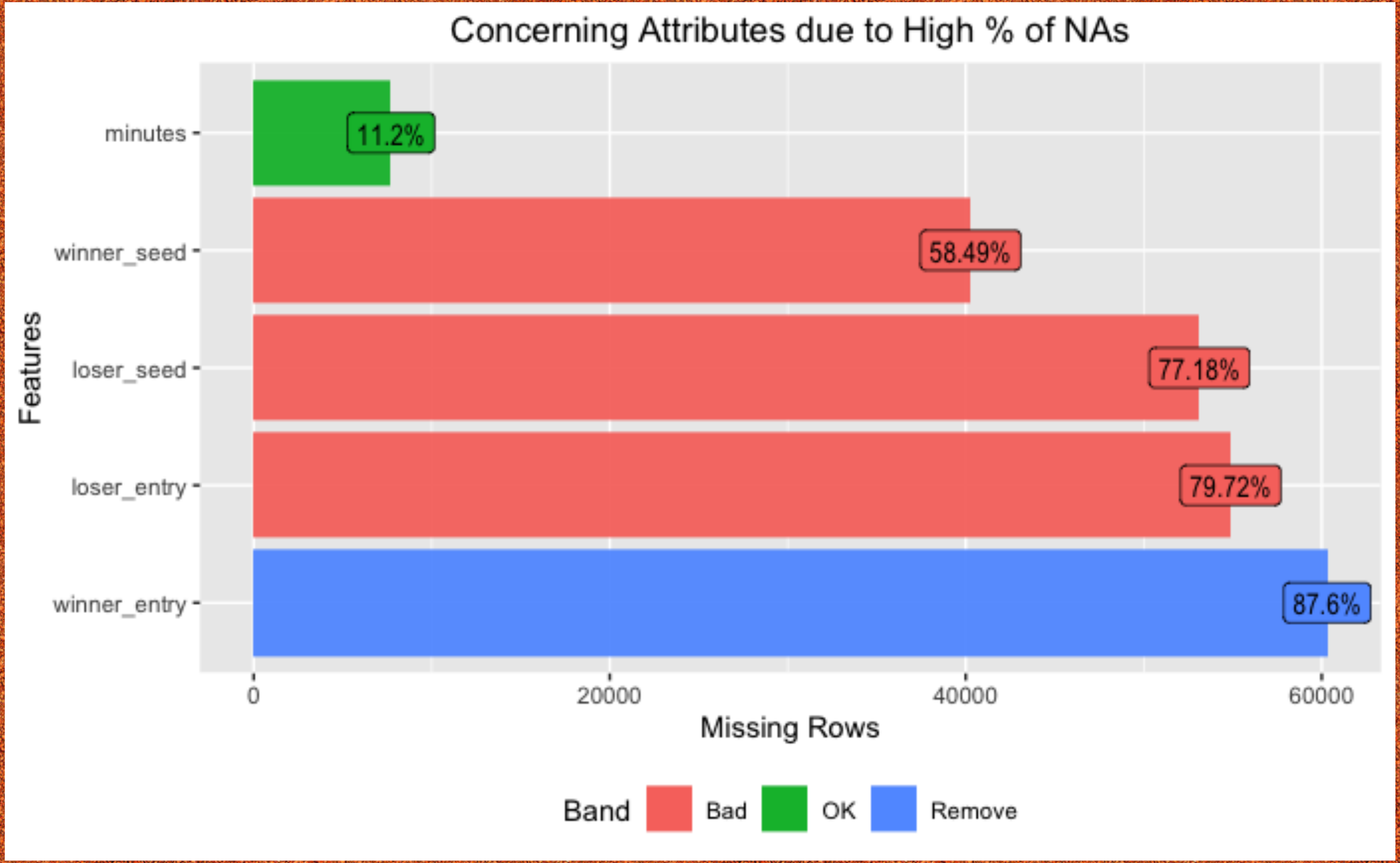
Preliminary Data Analysis

Summary statistics, data distribution



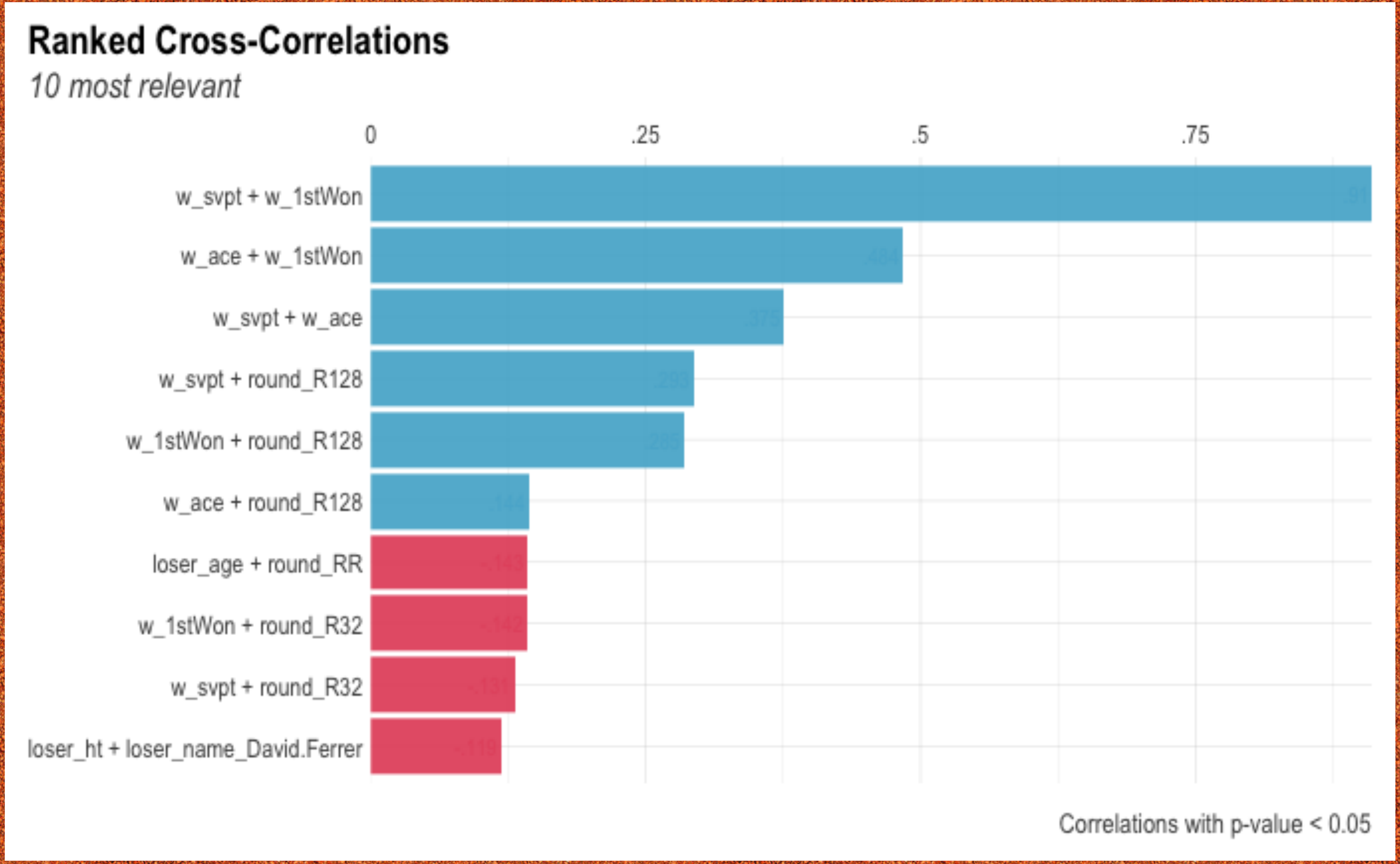
Data Exploration

Elementary Dimensionality Reduction



Data Exploration

Elementary Dimensionality Reduction



Feature Engineering

Newly Created Attributes

- Head-to-Head record for player pairings
- % Win on Surface
- % Win at Tournament Stage (Round)
- % Win at Tournament Level (Challenger, Grand Slam, etc.)
- % Win at Specific Tournament
    - % Win at Tournament Stage * % Win at Tournament Level

Data Splitting

70% Train, 30% Test

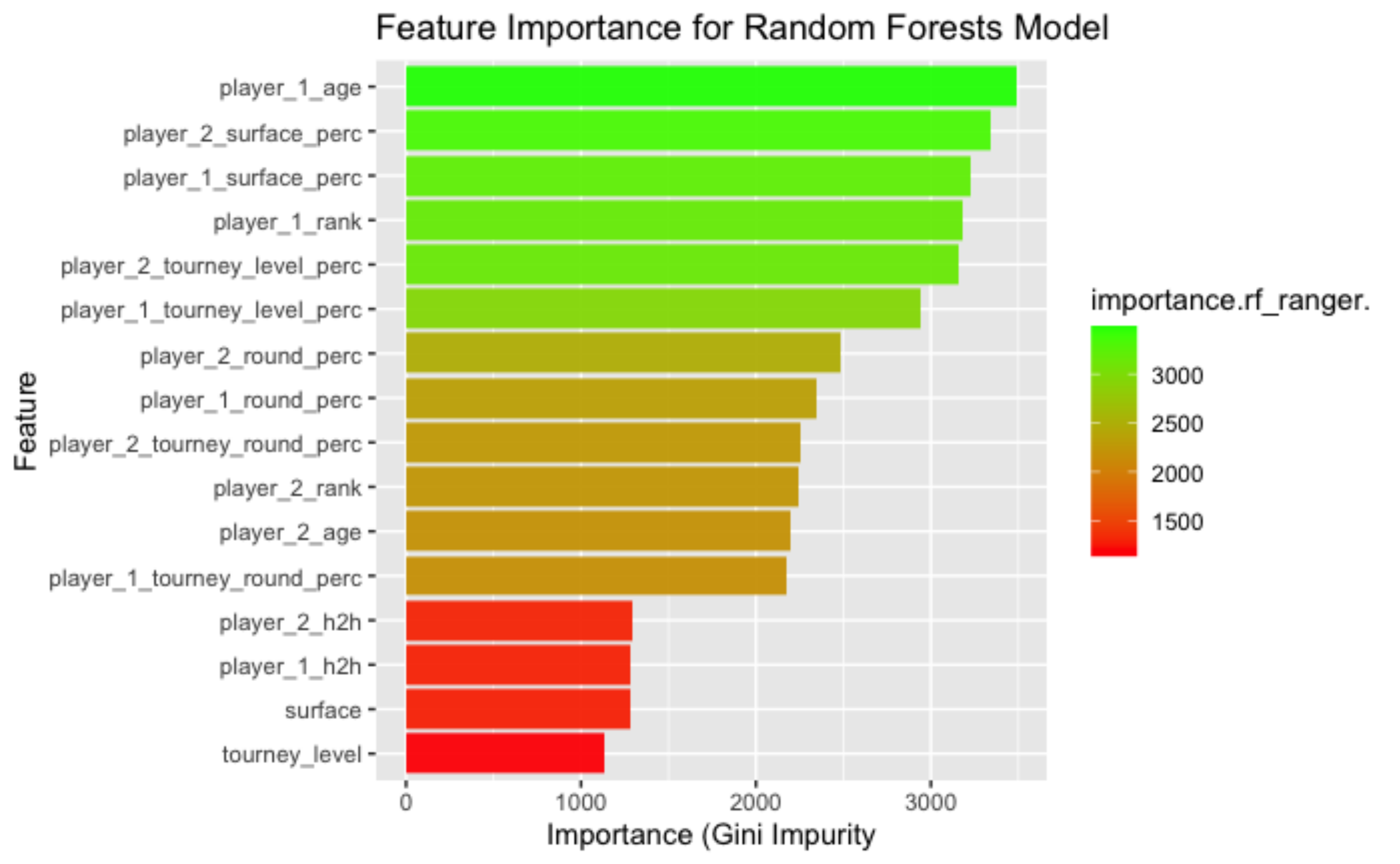
```
#Set the seed for reproducibility
set.seed(123)

#Set target variable as factor
merged_df$result <- as.factor(merged_df$result)

#Split the data into a training set (70%) and a testing set (30%)
train_idx <- createDataPartition(merged_df$result, p = 0.7, list =
FALSE)
train <- merged_df[train_idx, ]
test <- merged_df[-train_idx, ]
```

Model Building

Random Forest, Default Settings



Model Building

Random Forest, Grid-Search Optimized Settings

- Hyperparameters part of the grid search:
  - mtry: the number of variables randomly sampled as candidates at each split
  - min.node.size: the minimum number of observations at a terminal node
  - num.trees: number of trees in the forest

```
hyper_grid <- expand.grid(
  mtry = floor(n_features * c(.15, .25, .35)),
  min.node.size = c(1, 3, 5),
  num.trees = n_features * c(5, 10, 15)
)

for(i in seq_len(nrow(hyper_grid))) {
rf_ranger_opt <- ranger(
    formula         = result ~ .,
    data            = train,
    num.trees       = n_features * 10,
    mtry            = hyper_grid$mtry[i],
    min.node.size   = hyper_grid$min.node.size[i],
    verbose         = FALSE,
    seed            = 123,
    respect.unordered.factors = 'order',
  )
```

Model Building

Random Forest, Grid-Search Optimized Settings

hyper_grid

Filter

	mtry	min.node.size	num.tress	rmse	percentage_gain	default_rmse
1	5	1	40	0.5582334	-2.118771	0.5646339
2	5	1	80	0.5582334	-2.118771	0.5646339
3	5	1	120	0.5582334	-2.118771	0.5646339
4	5	1	160	0.5582334	-2.118771	0.5646339
5	5	3	40	0.5595660	-1.677616	0.5646339
6	5	3	80	0.5595660	-1.677616	0.5646339

Model Building

Random Forest, Manual and Truncated

- **Manual random forest:** manual adjustment to grid-search optimized hyperparameters
- **Truncated:** manual adjustment to grid-search optimized hyperparameters, but with truncated attributes based off the manual random forest
    - Removed low-gini importance attributes from the manual model, $tourney_level

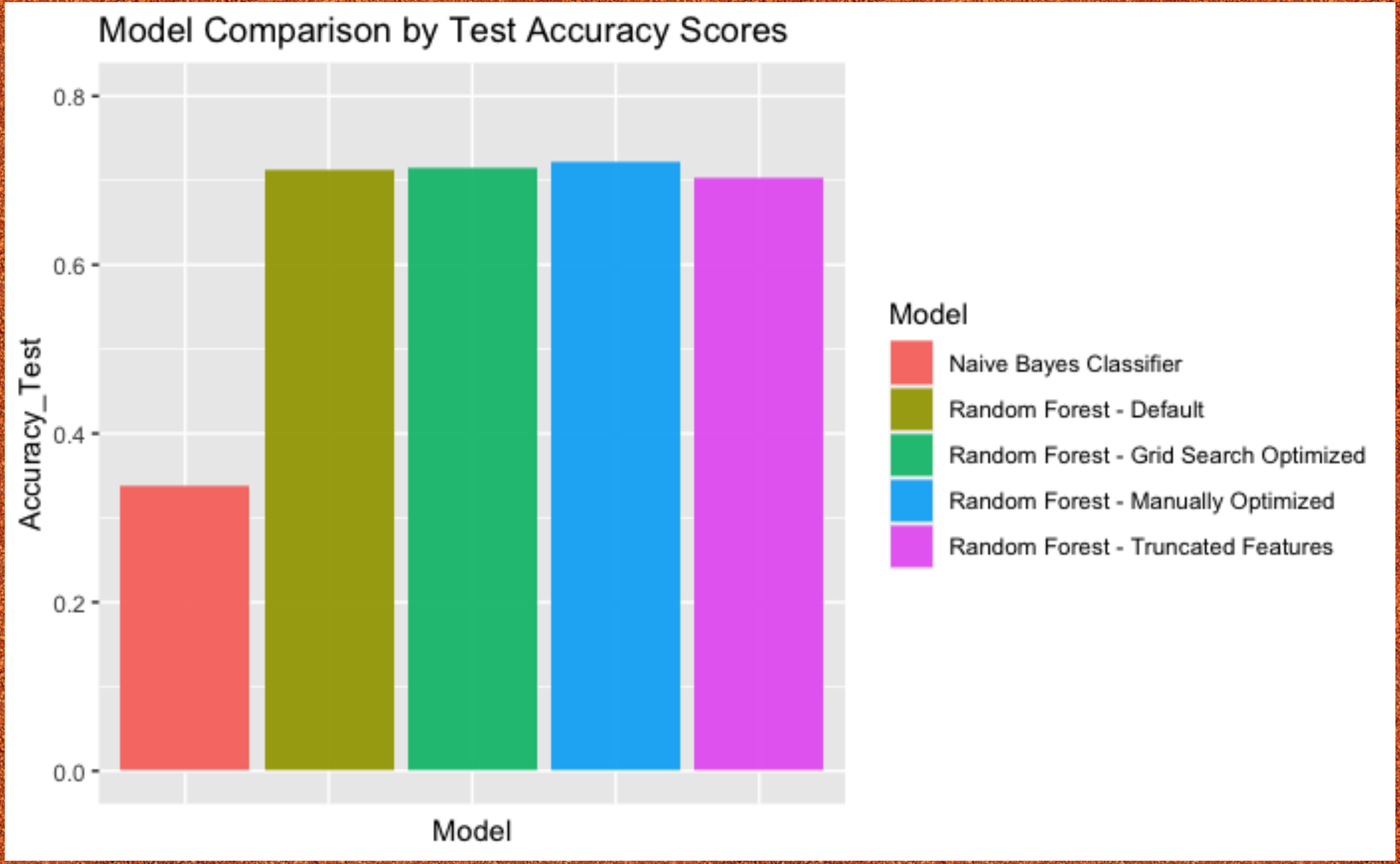
Model Evaluation

Random Forest (4 Variants), Naive Bayes

Model	OOB Error	RMSE	Accuracy_Train	Accuracy_Test
Random Forest – Manually Optimized	0.3033	0.5507	0.6967	0.7211
Random Forest – Grid Search Optimized	0.3115	0.5581	0.6885	0.7148
Random Forest – Default	0.3021	0.5496	0.6979	0.7136
Random Forest – Truncated Features	0.3195	0.5652	0.6805	0.7043
Naive Bayes Classifier	NA	NA	0.4311	0.3378

Model Evaluation

Random Forest (4 Variants), Naive Bayes



Project Summary

Changes Since Update #3

- Issues and Challenges:
  - R struggling to compute larger data sets and data objects as they accrue within local memory
  - XGBoost not as friendly in R due to xgb.matrix data type requirement
  - Grid-search exceptionally computationally expensive for randomforest package, and ranger package

- What to do differently next time?
  - Time-series sampling instead of simple stratified sampling
  - Reduce classification levels to increase model accuracy