Final Project

Ethan Page, Dawson MacLean, Kasey Smith, Ella Ewoldt, Maggie Wilder

2025-04-24

CATAN

Intro

Catan is a board game where players take on the role of settlers to obtain/trade resources and build roads/settlements. Resources include sheep, lumber, wheat, clay, and ore. A player's chance of winning the game is determined by both strategy and randomness (due to dice rolls).

We got our data from Kaggle, and online community for data scientists. Our data contains information for 50 games with 4 players (200 entrees). The Kaggle data has information about the games such as dice rolls, number of resources gained/lost, turn order, and more.

Data can be found here https://www.kaggle.com/datasets/lumins/settlers-of-catan-games (as well as in our project)

The goal of our project is to compare descriptive trends in the Catan data with what a randomforest model predicts are most important factors in who wins the game. Essentially, we are trying to find factors that predict a game win.

Our "lie" for this project will be showing how descriptives (resources, game turn, piece color, etc.) contribute to who wins the game. Our graphs will make it appear as if certain descriptives help players to win more games compared to other descriptives.

In reality, the randomforest predictor model gives us a better overall view of what might be important predictors of game wins. This model contains a lot more data than our data set, so it is more likely to be less affected by outliers than our data set with only 50 games.

Modifying Data

One factor of board games that a lot of people are passionate about is the color of their game piece. It is an easy thing to get superstitious over as a result of attachment to their preferred color. The Catan data set did not have color listed as a variable, but we figured that people might be willing to interpret data about whether a color was more associated with wins, particularly if their color was favored. So, we started by manipulating the data set to include a variable for color. We set the seed and then randomly assigned colors to each of the four players in each game. As it was randomly sampled, with no connection to the original data, there are no *real* correlations to be found.

```
# Reading in Catan data from data folder
CatanData <- read.csv(file = 'data/catanstats.csv', header = TRUE)

set.seed(333)
# Creating object that has all of the desired colors
colors <- c("red", "blue", "white", "orange")
```

```r
#applies all 4 colors per game with no repeats
CatanData$color <- unlist(
  lapply(
    split(CatanData, CatanData$gameNum),
    function(group)
    sample(colors))
  )
```

RandomForest Probability

We imported the libraries "randomForest" and "caret" from CRAN to help us create a model that could predict the likelihood of winning a Catan game given when provided with starting data. The randomForest library takes our data and does a bunch of "decision trees" based on our factors (has wood/has clay/average probability for both starting settlements/etc). It thenaverages across theses "trees" to try and get a general prediction of what factors are important for winning.

```r
#training RF_model


library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:randomForest':
##
##     combine

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(caret)
library(randomForest)




trainCatanModel <- function(full_data) {


  # Label winners per game
  full_data$win <- rep(FALSE, nrow(full_data))
  for (game in unique(full_data$gameNum)) {
    maybe_win <- full_data$player[full_data$points >= 10 & full_data$gameNum == game]
    if (length(maybe_win) > 0) {
```

```
      full_data$win[((game - 1) * 4) + min(maybe_win)] <- TRUE
    }
  }

  # Dice prob lookup based on 2 dice rolling
  prob_lookup <- c("0"=0, "2"=1/36, "3"=2/36, "4"=3/36, "5"=4/36,
                   "6"=5/36, "7"=0, "8"=5/36, "9"=4/36,
                   "10"=3/36, "11"=2/36, "12"=1/36)

  # preparing our sub info
  #     (data made from other data)
  full_data <- full_data %>%
    rowwise() %>%
    mutate(
      avg_prob = sum( #calcs avg probability of both tiles
        prob_lookup[as.character(X1settlenum1)],
        prob_lookup[as.character(X1settlenum2)],
        prob_lookup[as.character(X1settlenum3)],
        prob_lookup[as.character(X2settlenum1)],
        prob_lookup[as.character(X2settlenum2)],
        prob_lookup[as.character(X2settlenum3)]
      ), #calcs the resource diversity (has different resources)
      resource_diversity = n_distinct(c(
        X1settlersc1, X1settlersc2, X1settlersc3,
        X2settlersc1, X2settlersc2, X2settlersc3)),
      has_wood = as.integer("L" %in% c( # if has lumber -> TRUE
        X1settlersc1, X1settlersc2, X1settlersc3,
        X2settlersc1, X2settlersc2, X2settlersc3)),
      has_brick = as.integer("C" %in% c( #if has clay -> True
        X1settlersc1, X1settlersc2, X1settlersc3,
        X2settlersc1, X2settlersc2, X2settlersc3))
    ) %>%
    ungroup()

  #getting rid of factors we aren't tracking
  simplified_data <- full_data %>%
    select(win, avg_prob, resource_diversity, has_wood, has_brick, player)

  # Train model
  set.seed(42)
  simplified_data$win <- as.factor(simplified_data$win)
  model <- randomForest(
    win ~ avg_prob + resource_diversity + has_wood + has_brick + player,
    data = simplified_data
  )

  return(model)
}
```

```
source('TrainRFModel.R')
rf_model <- trainCatanModel(CatanData)
```

Then, once we have trained the model on our data, we create functions to easily use the model to calculate

the probability of winning. This way we don't have to go going all this every single time). One function is to predict for an entire set of 50 games, while the other is just for predicting based on one player's game stats.

```
#Predictor functions

library(dplyr)
library(caret)
library(randomForest)

#sourcing the train model function
#separating for readability in RMarkdown

source('TrainRFModel.R',local = knitr::knit_global())

predictCatan <- function(gameData, model) {
  prob_lookup <- c("0"=0, "2"=1/36, "3"=2/36, "4"=3/36, "5"=4/36,
                   "6"=5/36, "7"=0, "8"=5/36, "9"=4/36,
                   "10"=3/36, "11"=2/36, "12"=1/36)

  # Feature engineering
  processed_data <- gameData %>%
    rowwise() %>%
    mutate(
      avg_prob = sum(
        prob_lookup[as.character(X1settlenum1)],
        prob_lookup[as.character(X1settlenum2)],
        prob_lookup[as.character(X1settlenum3)],
        prob_lookup[as.character(X2settlenum1)],
        prob_lookup[as.character(X2settlenum2)],
        prob_lookup[as.character(X2settlenum3)]
      ),
      resource_diversity = n_distinct(c(
        X1settlersc1, X1settlersc2, X1settlersc3,
        X2settlersc1, X2settlersc2, X2settlersc3)),
      has_wood = as.integer("L" %in% c(
        X1settlersc1, X1settlersc2, X1settlersc3,
        X2settlersc1, X2settlersc2, X2settlersc3)),
      has_brick = as.integer("C" %in% c(
        X1settlersc1, X1settlersc2, X1settlersc3,
        X2settlersc1, X2settlersc2, X2settlersc3))
    ) %>%
    ungroup()

  # Keep only model-relevant columns
  pred_input <- processed_data %>%
    select(avg_prob, resource_diversity, has_wood, has_brick, player)


  # Predict
  probs <- predict(model, newdata = pred_input, type = "prob")[, "TRUE"]
  #return(probs)
}
```

```
predictCatanPlayer <- function(new_row, model) {
  prob_lookup <- c("0"=0, "2"=1/36, "3"=2/36, "4"=3/36, "5"=4/36,
                   "6"=5/36, "7"=0, "8"=5/36, "9"=4/36,
                   "10"=3/36, "11"=2/36, "12"=1/36)

  new_row <- new_row %>%
    mutate(
      avg_prob = sum(
        prob_lookup[as.character(X1settlenum1)],
        prob_lookup[as.character(X1settlenum2)],
        prob_lookup[as.character(X1settlenum3)],
        prob_lookup[as.character(X2settlenum1)],
        prob_lookup[as.character(X2settlenum2)],
        prob_lookup[as.character(X2settlenum3)]
      ),
      resource_diversity = n_distinct(c(
        X1settlersc1, X1settlersc2, X1settlersc3,
        X2settlersc1, X2settlersc2, X2settlersc3)),
      has_wood = as.integer("L" %in% c(
        X1settlersc1, X1settlersc2, X1settlersc3,
        X2settlersc1, X2settlersc2, X2settlersc3)),
      has_brick = as.integer("C" %in% c(
        X1settlersc1, X1settlersc2, X1settlersc3,
        X2settlersc1, X2settlersc2, X2settlersc3))
    )

  pred_input <- new_row %>%
    select(avg_prob, resource_diversity, has_wood, has_brick, player)

  prob <- predict(model, newdata = pred_input, type = "prob")[, "TRUE"]

  return(prob)
}



# Compare row 2
#cat("Row 2 via predictCatan = ", all_probs[2], "\n")
#cat("Row 2 via predictCatanPlayer = ", predictCatanPlayer(CatanData[2, ], rf_model), "\n")
```

The predicted probability of winning for every single player of every game (over 200 lines) is a lot, but the "importance" analysis of the model helps to understand what factors are relevant to predicting a win.

```
source('simulatingCatan.R')
imp<- importance(rf_model)
imp
```

```
##                    MeanDecreaseGini
```

```
## avg_prob                16.162768
## resource_diversity       5.360601
## has_wood                 2.438437
## has_brick                2.777573
## player                   7.324454
```

To simplify what these mean, a *higher number* basically means that it is *more important* to being able to predict whether or not the player wins. In this we see the "better numbers" (higher probability of rolling them on average) your settlements are on is the most important factor by far . This is followed by turn order. Resources, whether that be resource diversity or having specific resources at the start, did not seem to be as important predictors.

Descriptive Data Graphs

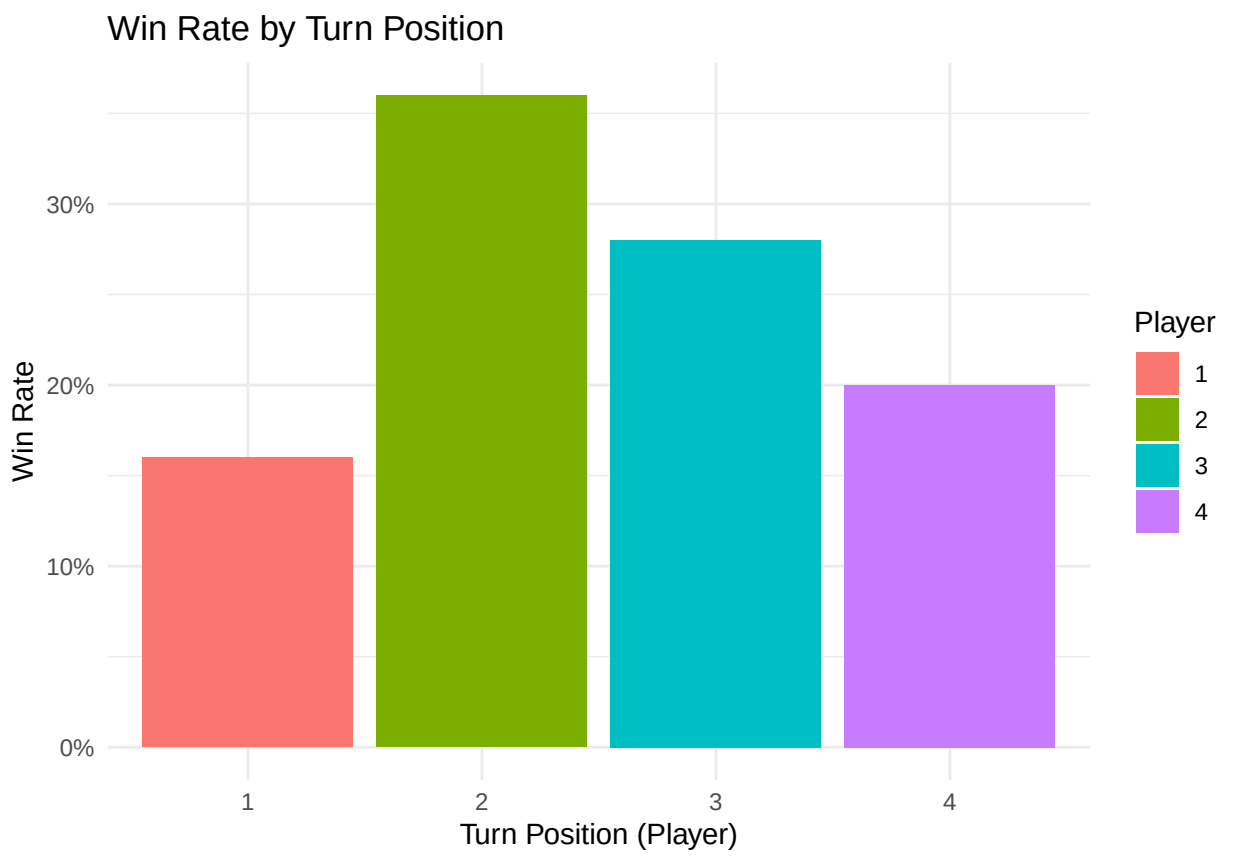
Here we take a different approach and graph the data to try and gleam information from that.

Win rate by turn order

Here it suggests that going 2nd and 3rd most often appeared in winning players

```
#determining winner based on the first player to have >=10 points
CatanData$winloss <- rep("Loss", nrow(CatanData))

for (game in unique(CatanData$gameNum)) {

    maybe_win <- CatanData$player[CatanData$points >= 10 & CatanData$gameNum == game]

    if (length(maybe_win) > 0) {
      CatanData$winloss[((game - 1) * 4) + min(maybe_win)] <- "Win"
    }
  }

library(tidyverse)

library(ggplot2)

##win rate by player turn

wins_only <- CatanData %>%
  filter(winloss == "Win")

win_rate_player <- CatanData %>%
  group_by(player) %>%
  summarise(
    total = n(),
    wins = sum(winloss == "Win"),
    win_rate = wins / total
  )

ggplot(win_rate_player, aes(x = factor(player), y = win_rate, fill = factor(player))) +
  geom_col() +
  labs(title = "Win Rate by Turn Position",
```

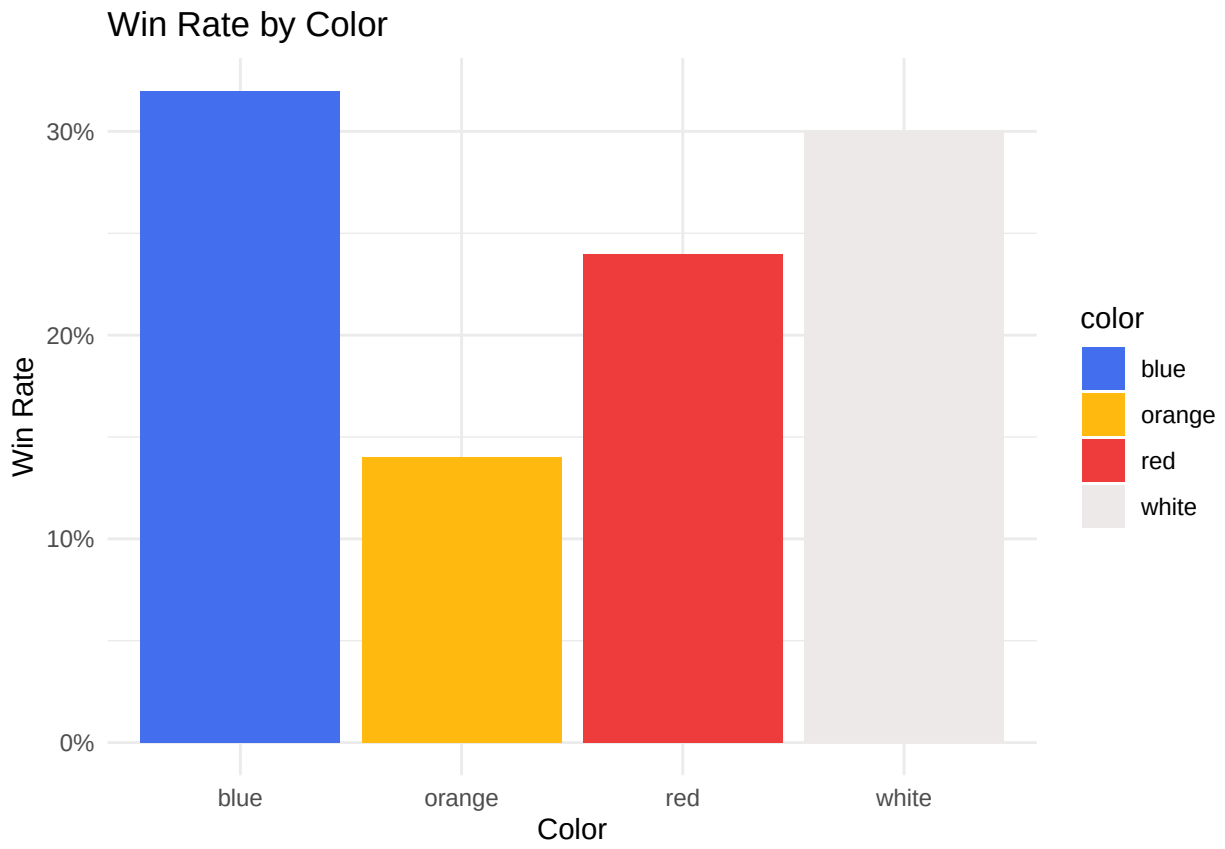
```
      x = "Turn Position (Player)",
      y = "Win Rate",
      fill = "Player") +
  scale_y_continuous(labels = scales::percent) +
  theme_minimal()
```



The second graph depicts the win rate by color. Based on the random data from the seed, more winners tend to pick blue, and the fewest winners tend to pick orange. Taking this data at face value, it seems that picking blue results in a significantly greater chance of winning a game of Catan than picking orange does. Those whose preferred color is blue might be inclined to believe these results, even though they are fully randomized within the 50 games of data and should have no real effect on game outcome.

```
##win rate by color
win_rate_color <- CatanData %>%
  group_by(color) %>%
  summarise(
    total = n(),
    wins = sum(winloss == "Win"),
    win_rate = wins / total
  )

ggplot(win_rate_color, aes(x = color, y = win_rate, fill = color)) +
  geom_col() +  # Use 'geom_col' to create a bar plot
  scale_fill_manual(values = c("royalblue2", "darkgoldenrod1", "brown2", "snow2")) +  # Assign custom c
  scale_y_continuous(labels = scales::percent_format()) +
  labs(title = "Win Rate by Color",
```

```
       x = "Color",
       y = "Win Rate") +
  theme_minimal()
```

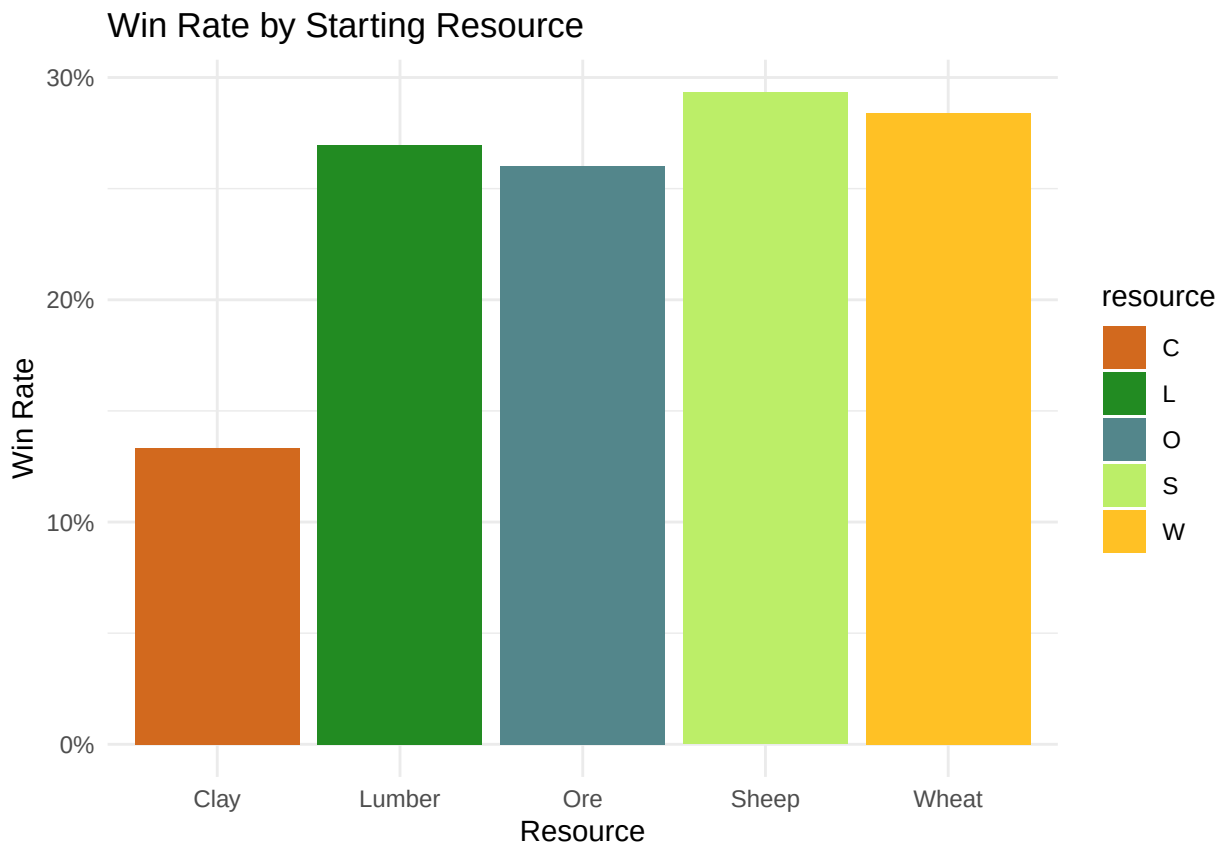


Win Rate by Starting Resource

This graph highlights the impact of starting resources on winning likelihood. Wheat clearly stands out as the most winning-associated resource, followed by ore, sheep, wood (logs), and finally brick (clay). If you're aiming for the best shot at victory, prioritizing wheat-heavy placements is a strong strategic move. The data also shows that starting with two of the same resource or building near the desert significantly decreases your chances — variety and access are key in early-game planning.

```
##win rate resource
Catan_long <- CatanData %>%
  pivot_longer(
    cols = matches("settlenum|settlersc"),
    names_to = c("settlement", ".value"),
    names_pattern = "(\\dsettle)(num\\d|rsc\\d)"
  ) %>%
  rename(number = num1, resource = rsc1)  # fix names for consistency

Catan_long <- Catan_long %>%
  mutate(
    number = as.numeric(number),
```

```r
    resource = str_trim(resource),
    resource = ifelse(str_detect(resource, "^[A-Z]$"), resource, NA)
  ) %>%
  drop_na(number, resource)

win_rate_resource <- Catan_long %>%
  group_by(resource) %>%
  summarise(
    total = n(),
    wins = sum(winloss == "Win"),
    win_rate = wins / total
  )

label_map <- c("L" = "Lumber", "C" = "Clay", "S" = "Sheep", "W" = "Wheat", "O" = "Ore")

# Generate the plot
ggplot(win_rate_resource, aes(x = resource, y = win_rate, fill = resource)) +
  geom_col() +
  scale_fill_manual(values = c("L" = "forestgreen",    # Lumber (L) = Green
                               "C" = "chocolate",   # Clay (C) = Brown
                               "S" = "darkolivegreen2",       # Sheep (S) = Blue
                               "W" = "goldenrod1",     # Wheat (W) = Yellow
                               "O" = "cadetblue4")) +       # Ore (O) = Gray
  scale_y_continuous(labels = scales::percent_format()) +
  scale_x_discrete(labels = label_map) +  # Replace abbreviations with full names
  labs(title = "Win Rate by Starting Resource",
       x = "Resource",
       y = "Win Rate") +
  theme_minimal()
```



Analysis

From both the descriptive graphs and the randomForest predictor model, we see a different emphasis of important factors.

The graphs appear to show that your choice of *piece color* and *starting resources* would make a large impact, but this not necessarily the case.

The *color* of tokens chosen was generated randomly by us at the start, therefore the variance we see in the graph is purely due to chance (you can see this effect in action by changing the set seed before sampling).

```
#copying data from row 1
data1 <- CatanData[1,]
#changing the color to blue
data1$color <- "blue"

#predicting chance of winning
predictCatanPlayer(data1,rf_model)
```

```
## [1] 0.214
```

```
#changing color to red
data1$color <- "red"

#predicting now
predictCatanPlayer(data1,rf_model)
```

```
## [1] 0.214
```

This short code above also reaffirms that color does not affect the probability of winning. The probability of winning is the same (0.214) regardless of the color "chosen".

The specific starting *resources* also appear to be important to distinguish, but the importance model shows that individual resources have a relatively low importance (factors has_clay and has_wood in the single digit levels). What appears to matter more is not the type of resource, but rather the average probability of obtaining your resource (whatever they might be) based on the number of the resource.

The graph showing wins by turn order does seem to be relevant. The randomForest model's second highest importance factor was that of turn_order. This suggests that the most common turn orders appearing in winning players (2nd and 3rd) may actually be quite important in determining your success in Catan.

```
#for reference
imp
```

```
##                     MeanDecreaseGini
## avg_prob                   16.162768
## resource_diversity          5.360601
## has_wood                    2.438437
## has_brick                   2.777573
## player                      7.324454
```