assignment_03

2023-09-12

Preliminaries

Here, we load the data and merge them together into one data frame:

```
player_data <- read.csv("./data/player_data.csv")
salary_data <- read.csv("./data/salary_data.csv")

full_data <- merge(x = player_data, y = salary_data, by.x = "Player", by.y = "Name")
```

Reducing Players' Worth to Their Calculated Values

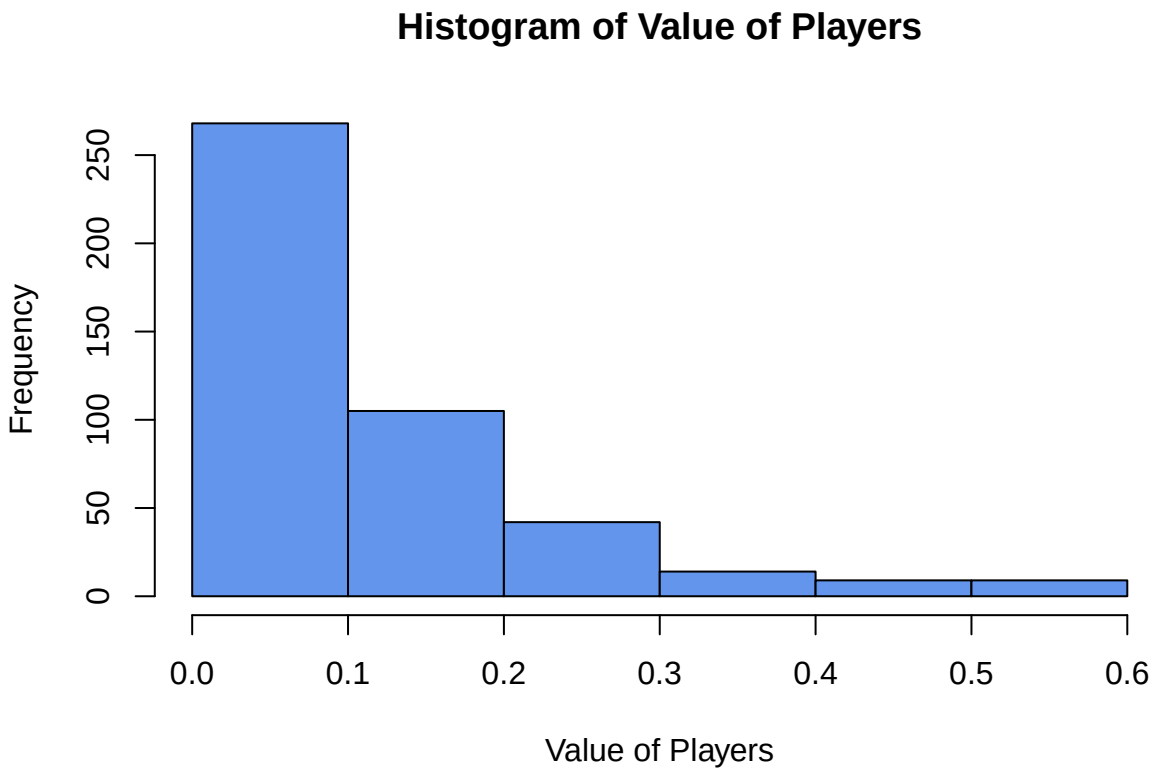
Here, we calculate the values of players with the formula given and remove the outliers in the data:

```
#calculating player value
Value <-(full_data$PTS - 2*full_data$X2P)/(full_data$Salary/100000)
full_data_with_value <- cbind(full_data, Value)

outliers <- 0.6

#removing outliers
full_data_with_value_no_outliers <- full_data_with_value[full_data_with_value$Value
                                                         <= outliers,]
```

Then, we produce the required histogram:



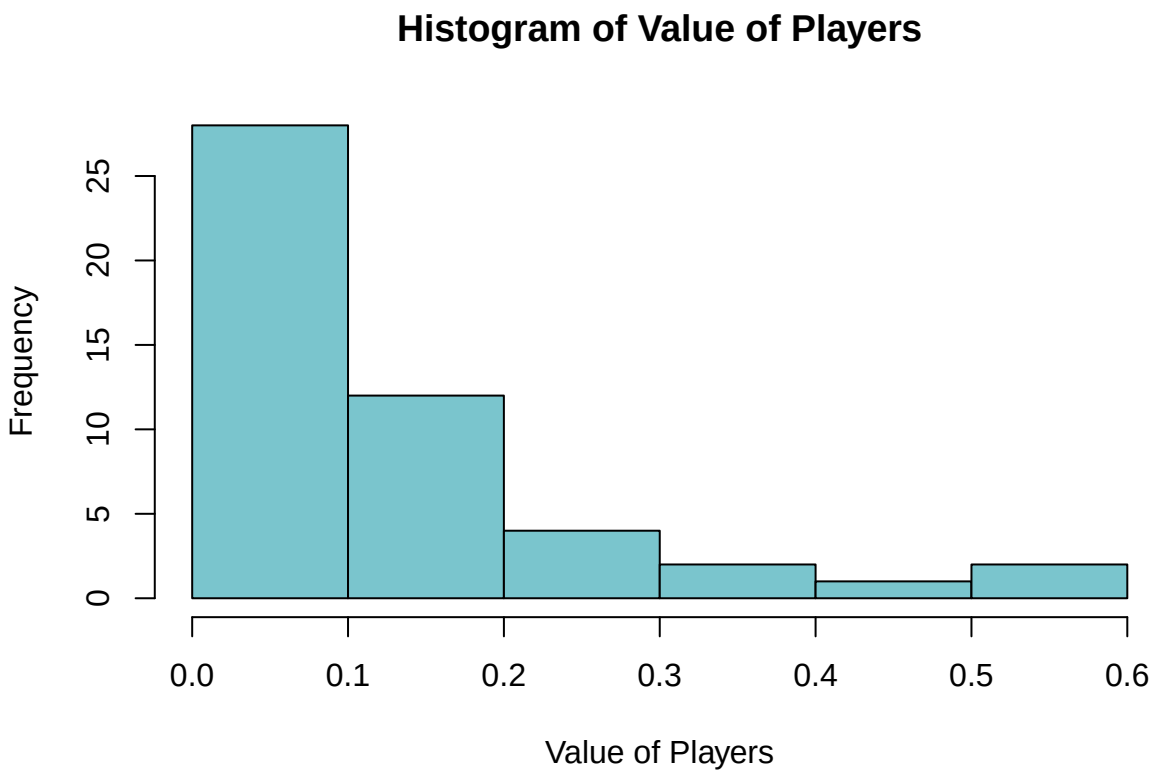
Sampling Players Instead (Sushi Buffet Style jk)

Dealing with the sample value data set in the same way as above:

```
#loading the data
sample_salary_data <- read.csv("./data/sample_salary_data.csv")
#merging the table
sample_data <- merge(x = player_data, y = sample_salary_data, by.x = "Player", by.y = "Name")
#calculating value
value_sample <-(sample_data$PTS - 2*sample_data$X2P)/(sample_data$Salary/100000)
#combining into one data frame
sample_data_with_value <- cbind(sample_data, value_sample)
#renaming column
colnames(sample_data_with_value)[6] = "Value"

#removing outliers
sample_data_with_value_no_outliers <- sample_data_with_value[sample_data_with_value$Value
                                                             <= outliers,]
```

The required histogram:



So Was Sampling Similar Enough?

The plot in question 3 better depicts the population's player values in the range 0-0.3 because the majority of players' value falls within this range. By sampling the data at random, it is possible that the sample contains larger proportion of players with higher value than the full data set, since it is random.

Playing Games with a (Potential?) Cheater

Here, we assign the model probabilities as required and choose the appropriate test statistic:

```
#probability of face card: 4/13, probability of numbered card: 9/13
deck_model_probabilities <- c(4/13, 9/13)

#results from the game
actual_face_cards <- 8
expected_face_cards <- 4

#choosing our test statistic
statistic_choice <- 3
```

Next, we create the required function:

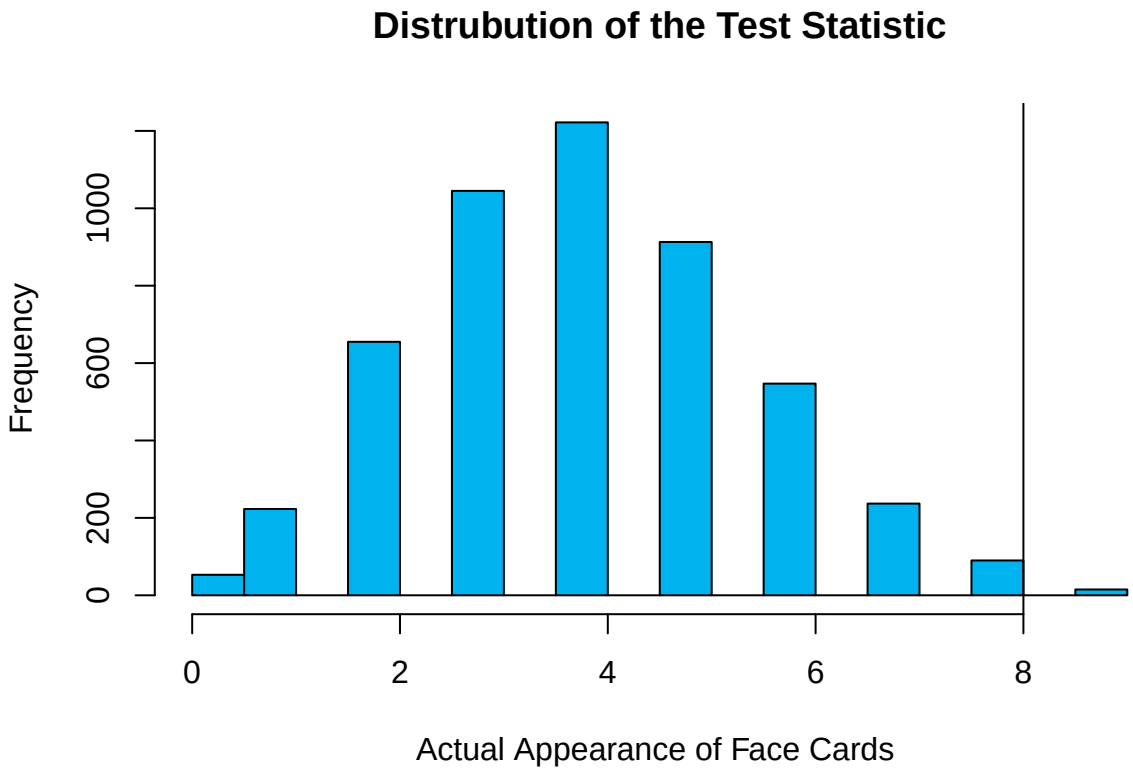
```
deck_simulation_and_statistic <- function(sample_size, model_proportions){
  sample(c("face_card", "number_card"),
         prob = model_proportions,
         size = sample_size,
         replace = TRUE)
}
```

Now, we draw 13 cards 5000 times as required, and take the required test statistic:

```r
deck_statistics <- numeric(5000)
for (i in 1:5000) {
  draw <- deck_simulation_and_statistic(13, deck_model_probabilities)
  deck_statistics[i] <- sum(draw == "face_card")
}
```

I tried to do this with `sapply` instead, but it got complicated very fast. Clearly, if there's a lot you're trying to fit into a for-loop, it's better to just write the loop.

Anyway, plotting the distribution of the test statistics, we get the following histogram:



Conclusion: Jade is a Cheater

As we can see from the distribution above, the frequency at which at least 8 face cards are drawn is very low compared to the other number of appearances. It isn't impossible for her to have drawn 8 face cards, but it is more likely that she is just a cheater.