EDA

Sam Lee

```
season_results = read_csv(str_c(data_dir, "MRegularSeasonDetailedResults.csv"))
teams = read_csv(str_c(data_dir, "MTeams.csv"))
tourney_results = read_csv("2023 Game Data.csv")

transform_data = function(t){
  t %>% pivot_longer(cols=c(WTeamID, LTeamID), names_to="WL", values_to = "TeamID") %>%
  select(TeamID, WL, WScore, LScore, WFGM, WFGA, LFGM, LFGA) %>%
  left_join(teams %>% select(TeamID, TeamName)) %>%
  pivot_longer(cols=c(WScore, LScore), values_to = "Score", names_to = "WLScore") %>%
  pivot_longer(cols=c(WFGM, LFGM), values_to="FGM", names_to = "WLFGM") %>%
  pivot_longer(cols=c(WFGA, LFGA), values_to="FGA", names_to = "WLFGA") %>%
  mutate(WL = sapply(WL, function(x)substr(x,1,1)),
         WLScore = sapply(WLScore, function(x)substr(x,1,1)),
         WLFGM = sapply(WLFGM, function(x)substr(x,1,1)),
         WLFGA = sapply(WLFGA, function(x)substr(x,1,1))
  ) %>% rowwise() %>%
  filter(all(c(WL, WLScore, WLFGM, WLFGA) ==
               first(c(WL, WLScore, WLFGM, WLFGA)))) %>%
  select(TeamID, WL, TeamName, Score, FGM, FGA)
}

#2022 Season
season.2022 <- season_results %>%
  filter(Season %in% 2022)

season.2022 <- transform_data(season.2022)

season.2022 %>% mutate(
    FGP = FGM/FGA
  ) -> season.2022
```

```r
calculate_prior_fgp = function(p, beta=1){
  #Returns alpha for a Beta(alpha, beta) such that alpha/(alpha+beta) = p (expected value)
  return(p*beta/(1-p))
}


#Calculate priors for the field goal percentage ~ Beta(alpha, beta)
#and for the field goal attempts ~ N(mu, sigma^2)
season.2022 %>% group_by(TeamID) %>%
  summarize(
    fgp.alpha.prior = calculate_prior_fgp(mean(FGP)),
    fgp.beta.prior = 1,
    fga.lambda.prior = mean(FGA),
    fga.tau.prior = (max(FGA)-min(FGA))/3,
    #Method of Moments https://arxiv.org/pdf/1605.01019.pdf
    fga.gamma.prior = mean(FGA)^2/var(FGA)+2,
    fga.phi.prior = mean(FGA)*(mean(FGA)^2/var(FGA)+1)
  ) -> season.2022.priors

#2023 Season we want to model
season.2023 <- season_results %>%
  filter(Season %in% 2023)

season.2023 <- transform_data(season.2023)

#Calculate posteriors for FGP for the 2023 season
season.2023.posteriors = season.2023 %>% left_join(season.2022.priors, by=join_by(TeamID))
  group_by(TeamID) %>%
  summarize (
    fgp.alpha.posterior = sum(FGM)+first(fgp.alpha.prior),
    fgp.beta.posterior = sum(FGA)-sum(FGM)+first(fgp.beta.prior),
    fga.lambda.prior = first(fga.lambda.prior),
    fga.tau.prior = first(fga.tau.prior),
    fga.gamma.prior = first(fga.gamma.prior),
    fga.phi.prior = first(fga.phi.prior)
  )

#Gibbs Sampling Method to Define Posterior
posterior.matrix = as.matrix(season.2023.posteriors[c("fga.lambda.prior", "fga.tau.prior",
                                                      "fga.gamma.prior", "fga.phi.prior")]
```

```r
iterations = 10000

#Matrices to store posterior distributions
posterior.normal.matrix = matrix(ncol=iterations, nrow=nrow(posterior.matrix))
posterior.invgamma.matrix = matrix(ncol=iterations, nrow=nrow(posterior.matrix))

#Calculate the Normal posterior distribution for each ith team via Gibbs sampling
for(i in 1:nrow(posterior.matrix)){
  ith_team = posterior.matrix[i,]
  data_i = season.2023[season.2023$TeamID == as.numeric(season.2023.posteriors[i,"TeamID"]
    unlist()

  #Gibbs sampling algorithm
  burn = 100
  iters <- iterations + burn
  mu.save <- rep(0, iters)
  mu.save <- ith_team["fga.lambda.prior"]
  sigma2.save <- rep(0, iters)
  sigma2 = ith_team["fga.phi.prior"]/(ith_team["fga.gamma.prior"]-1)
  sigma2.save[1] = sigma2

  lambda = ith_team["fga.lambda.prior"]
  tau = ith_team["fga.tau.prior"]
  gamma = ith_team["fga.gamma.prior"]
  phi = ith_team["fga.phi.prior"]
  n = length(data_i)

  if(any(is.na(ith_team))){
    posterior.normal.matrix[i,] = rep(NA_real_, iterations)
    posterior.invgamma.matrix[i,] = rep(NA_real_, iterations)
  } else {
    for(t in 2:iters){
      #Full conditional of mu
      lambda.p <- (tau^2*sum(data_i) + sigma2*lambda)/(tau^2*n + sigma2)
      tau2.p <- sigma2*tau^2/(tau^2*n + sigma2)

      #New value of mu
      mu <- rnorm(1, lambda.p, sqrt(tau2.p))
      mu.save[t] <- mu

      #Full conditional of sigma2
```

```
      gamma.p <- gamma + length(data)/2
      phi.p <- phi + sum((data_i - mu)^2)/2

      #New value of sigma2
      sigma2 <- rinvgamma(1, gamma.p, phi.p)
      sigma2.save[t] <- sigma2
    }

    posterior.normal.matrix[i,] = mu.save[-(1:burn)]
    posterior.invgamma.matrix[i,] = sigma2.save[-(1:burn)]
  }

  #print(i)

}


season.2023.posteriors$fga.mu.posterior = rowMeans(posterior.normal.matrix)
season.2023.posteriors$fga.sigma.posterior = sqrt(rowMeans(posterior.invgamma.matrix))

season.2023.posteriors %>%
  filter(!is.na(fga.mu.posterior)) -> season.2023.posteriors

#Monte Carlo Simulation to Simulate FGM
posterior.fgm.matrix = matrix(ncol=iterations, nrow=nrow(season.2023.posteriors))
for(i in 1:nrow(season.2023.posteriors)){
  #Randomly sample from p from the posterior beta distribution on Field Goal Percentage
  p = rbeta(iterations, as.numeric(season.2023.posteriors[i, "fgp.alpha.posterior"]),
            as.numeric(season.2023.posteriors[i, "fgp.beta.posterior"]))
  #Calculate distribution of mean FGM by multiplying p by a random sample of FGA by team i
  f = rnorm(iterations, as.numeric(season.2023.posteriors[i, "fga.mu.posterior"]),
            as.numeric(season.2023.posteriors[i, "fga.sigma.posterior"]))

  posterior.fgm.matrix[i,] = p*f
}

tourney_results[c("SEED", "TEAM...3")] %>%
  setNames(c("Seed", "TeamName")) -> tourney_results
clean_team_names = function(t){
  t$TeamName = sapply(t$TeamName, function(x){
    x = x %>% str_replace("[.]", "")
```

```
    x = x %>% str_replace("Florida", "FL")
    if(x == "Saint Mary's")x = "St Mary's CA"
    if(x == "College of Charleston")x = "Col Charleston"
    if(x == "Louisiana Lafayette")x = "Lafayette"
    if(x == "Fairleigh Dickinson")x = "F Dickinson"
    if(x == "Northern Kentucky")x = "N Kentucky"
    if(x == "Southeast Missouri St")x = "SE Missouri St"
    if(x == "Texas A&M Corpus Chris")x = "TAM C. Christi"
    if(x == "Texas Southern")x = "TX Southern"
    if(x == "Montana St")x = "Montana St"
    if(x == "Kennesaw St")x = "Kennesaw"
    if(x == "Kent St")x = "Kent"
    if(x == "North Carolina St")x = "NC State"
    return(x)
  })
  return(t)
}
tourney_results = clean_team_names(tourney_results)

tourney_results %>%
  left_join(teams[c("TeamID", "TeamName")]) -> tourney_results

#Omit the first four
tourney_results %>%
  filter(!TeamName %in% c("TX Southern", "Nevada",
                          "Mississippi St", "SE Missouri St")) %>%
  distinct() -> tourney_results

tourney_results$Region = rep(c("E", "S", "W", "M"), each=2, times=8)
```

```
#2023 Tournament Simulation

regions = c("E", "S", "W", "M")
tourney_results %>% group_by(Region) %>%
  mutate(
    Order = rep(LETTERS[1:(n()/2)], each=2)
  ) -> tourney_results

matchups = tibble()

compare_teams = function(k, l, alpha=0.25){
```

```
  k = which(season.2023.posteriors$TeamID == k)
  l = which(season.2023.posteriors$TeamID == l)
  list(
    p = mean(posterior.fgm.matrix[k,] > posterior.fgm.matrix[l,]),
    q = quantile(posterior.fgm.matrix[k,] - posterior.fgm.matrix[l,], alpha)
  )
}

tourney_results$Round = 1

for(round in 1:4){
 for(region in regions){
    t = tourney_results
    if(round > 1)t = matchups

    if(round < 5){
      #These are all the regional matches
      region.subset = t %>%
        filter(Region == region & Round == round)
    }

    region.subset$p = NA_real_
    region.subset$alpha.probability = NA_real_
    region.subset$Round = round+1
    if(round > 1){
      half = region.subset$Order[1:(length(region.subset$Order)/2)]
      region.subset$Order = c(half, rev(half))

      matchups[matchups$Region == region & matchups$Round == round,
               ]$Order =c(half, rev(half))
    }


    region.subset %>%
      arrange(Order) -> region.subset

    #Loop through every game
    i = 1
    while(i < nrow(region.subset)){
      p = compare_teams(region.subset[i,]$TeamID,
                        region.subset[i+1,]$TeamID)[["p"]]
```

```
      #Predictive probability distribution is a Bernoulli Distribution
      if(p > (1-p)){
        region.subset[i,]$p = p
        region.subset[i,]$alpha.probability =
          compare_teams(region.subset[i,]$TeamID,
                region.subset[i+1,]$TeamID)[["q"]] %>% as.vector() > 0
        matchups = rbind(matchups, region.subset[i,])
      } else {
        region.subset[i+1,]$p = 1-p
        region.subset[i+1,]$alpha.probability =
          compare_teams(region.subset[i+1,]$TeamID,
                region.subset[i,]$TeamID)[["q"]] %>% as.vector() > 0
        matchups = rbind(matchups, region.subset[i+1,])
      }
      i = i + 2
    }
  }
}

#Final Four and Champtionship
for(round in 5:6){
  t.subset = matchups %>%
    filter(Round == round)
  t.subset$Round = round+1

  #Loop through every game
  i = 1
  while(i < nrow(t.subset)){
    p = compare_teams(t.subset[i,]$TeamID,
                      t.subset[i+1,]$TeamID)[["p"]]
    #Predictive probability distribution is a Bernoulli Distribution
    if(p > (1-p)){
      t.subset[i,]$p = p
      t.subset[i,]$alpha.probability = compare_teams(t.subset[i,]$TeamID,
              t.subset[i+1,]$TeamID)[["q"]] %>% as.vector() > 0
      matchups = rbind(matchups, t.subset[i,])
    } else {
      t.subset[i+1,]$p = 1-p
      t.subset[i+1,]$alpha.probability = compare_teams(t.subset[i+1,]$TeamID,
              t.subset[i,]$TeamID)[["q"]] %>% as.vector() > 0
      matchups = rbind(matchups, t.subset[i+1,])
```

```
    }
    i = i + 2
  }
}
```

2023 Tournament Simulation

The column p indicates the predictive posterior probability of how likely that team was to make more field goals than their opposing team in the previous round.

First Round Matchups

Seed	TeamName	Region
1	Alabama	E
16	TAM C. Christi	E
1	Purdue	S
16	F Dickinson	S
1	Houston	W
16	N Kentucky	W
1	Kansas	M
16	Howard	M
2	Arizona	E
15	Princeton	E
2	Marquette	S
15	Vermont	S
2	Texas	W
15	Colgate	W
2	UCLA	M
15	UNC Asheville	M
3	Baylor	E
14	UC Santa Barbara	E
3	Kansas St	S
14	Montana St	S
3	Xavier	W
14	Kennesaw	W
3	Gonzaga	M
14	Grand Canyon	M
4	Virginia	E
13	Furman	E
4	Tennessee	S

Seed	TeamName	Region
13	Lafayette	S
4	Indiana	W
13	Kent	W
4	Connecticut	M
13	Iona	M
5	San Diego St	E
12	Col Charleston	E
5	Duke	S
12	Oral Roberts	S
5	Miami FL	W
12	Drake	W
5	St Mary's CA	M
12	VCU	M
6	Creighton	E
11	NC State	E
6	Kentucky	S
11	Providence	S
6	Iowa St	W
11	Pittsburgh	W
6	TCU	M
11	Arizona St	M
7	Missouri	E
10	Utah St	E
7	Michigan St	S
10	USC	S
7	Texas A&M	W
10	Penn St	W
7	Northwestern	M
10	Boise St	M
8	Maryland	E
9	West Virginia	E
8	Memphis	S
9	FL Atlantic	S
8	Iowa	W
9	Auburn	W
8	Arkansas	M
9	Illinois	M

Second Round Matchups

Seed	TeamName	Region	p
1	Alabama	E	0.5610
9	West Virginia	E	0.5820
16	F Dickinson	S	0.6691
8	Memphis	S	0.5597
1	Houston	W	0.7407
8	Iowa	W	0.6927
1	Kansas	M	0.6065
8	Arkansas	M	0.5306
2	Arizona	E	0.6771
7	Missouri	E	0.6101
2	Marquette	S	0.7380
10	USC	S	0.5101
15	Colgate	W	0.6078
10	Penn St	W	0.6872
2	UCLA	M	0.7059
10	Boise St	M	0.6492
14	UC Santa Barbara	E	0.5422
11	NC State	E	0.6153
3	Kansas St	S	0.6691
6	Kentucky	S	0.5080
3	Xavier	W	0.7640
11	Pittsburgh	W	0.5469
3	Gonzaga	M	0.8891
6	TCU	M	0.6947
13	Furman	E	0.7795
12	Col Charleston	E	0.6794
4	Tennessee	S	0.7305
12	Oral Roberts	S	0.7526
4	Indiana	W	0.6041
5	Miami FL	W	0.6554
13	Iona	M	0.5450
5	St Mary's CA	M	0.5609

Sweet 16

Seed	TeamName	Region	p
1	Alabama	E	0.6295
13	Furman	E	0.5327
8	Memphis	S	0.5729

Seed	TeamName	Region	p
12	Oral Roberts	S	0.7847
8	Iowa	W	0.5732
5	Miami FL	W	0.5415
1	Kansas	M	0.5339
13	Iona	M	0.6835
2	Arizona	E	0.5744
11	NC State	E	0.6942
2	Marquette	S	0.7470
6	Kentucky	S	0.6010
15	Colgate	W	0.7272
3	Xavier	W	0.7683
2	UCLA	M	0.6714
3	Gonzaga	M	0.7665

Elite 8

Seed	TeamName	Region	p
13	Furman	E	0.5177
2	Arizona	E	0.5228
12	Oral Roberts	S	0.5862
2	Marquette	S	0.6424
5	Miami FL	W	0.5317
3	Xavier	W	0.5339
13	Iona	M	0.5445
3	Gonzaga	M	0.7474

Final Four

Seed	TeamName	Region	p
2	Arizona	E	0.5980
12	Oral Roberts	S	0.5185
3	Xavier	W	0.5879
3	Gonzaga	M	0.7314

Championship

Seed	TeamName	Region	p
12	Oral Roberts	S	0.5308
3	Gonzaga	M	0.6118

Champion

Seed	TeamName	Region	p
3	Gonzaga	M	0.6309