EDA

Sam Lee

```r
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, posterior.normal.matrix[i,], sqrt(posterior.invgamma.matrix[i,]))

  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")
```

```r
    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)
```

```r
#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
13	Lafayette	S

Seed	TeamName	Region
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	0.5819
16	F Dickinson	S	0.6660
8	Memphis	S	0.5599
1	Houston	W	0.7338
8	Iowa	W	0.6909
1	Kansas	M	0.6064
8	Arkansas	M	0.5266
2	Arizona	E	0.6768
7	Missouri	E	0.6118
2	Marquette	S	0.7366
10	USC	S	0.5108
15	Colgate	W	0.6044
10	Penn St	W	0.6866
2	UCLA	M	0.7010
10	Boise St	M	0.6466
14	UC Santa Barbara	E	0.5439
11	NC State	E	0.6150
3	Kansas St	S	0.6667
6	Kentucky	S	0.5107
3	Xavier	W	0.7621
11	Pittsburgh	W	0.5499
3	Gonzaga	M	0.8864
6	TCU	M	0.6941
13	Furman	E	0.7750
12	Col Charleston	E	0.6777
4	Tennessee	S	0.7272
12	Oral Roberts	S	0.7479
4	Indiana	W	0.6007
5	Miami FL	W	0.6501
13	Iona	M	0.5426
5	St Mary's CA	M	0.5615

Sweet 16

Seed	TeamName	Region	p
1	Alabama	E	0.6316
13	Furman	E	0.5318
8	Memphis	S	0.5688

Seed	TeamName	Region	p
12	Oral Roberts	S	0.7803
8	Iowa	W	0.5709
5	Miami FL	W	0.5379
1	Kansas	M	0.5350
13	Iona	M	0.6794
2	Arizona	E	0.5714
11	NC State	E	0.6915
2	Marquette	S	0.7452
6	Kentucky	S	0.6005
15	Colgate	W	0.7255
3	Xavier	W	0.7661
2	UCLA	M	0.6695
3	Gonzaga	M	0.7658

Elite 8

Seed	TeamName	Region	p
13	Furman	E	0.5188
2	Arizona	E	0.5239
12	Oral Roberts	S	0.5835
2	Marquette	S	0.6368
5	Miami FL	W	0.5287
3	Xavier	W	0.5322
13	Iona	M	0.5431
3	Gonzaga	M	0.7436

Final Four

Seed	TeamName	Region	p
2	Arizona	E	0.5973
12	Oral Roberts	S	0.5199
3	Xavier	W	0.5870
3	Gonzaga	M	0.7321

Championship

Seed	TeamName	Region	p
12	Oral Roberts	S	0.5297
3	Gonzaga	M	0.6144

Champion

Seed	TeamName	Region	p
3	Gonzaga	M	0.629

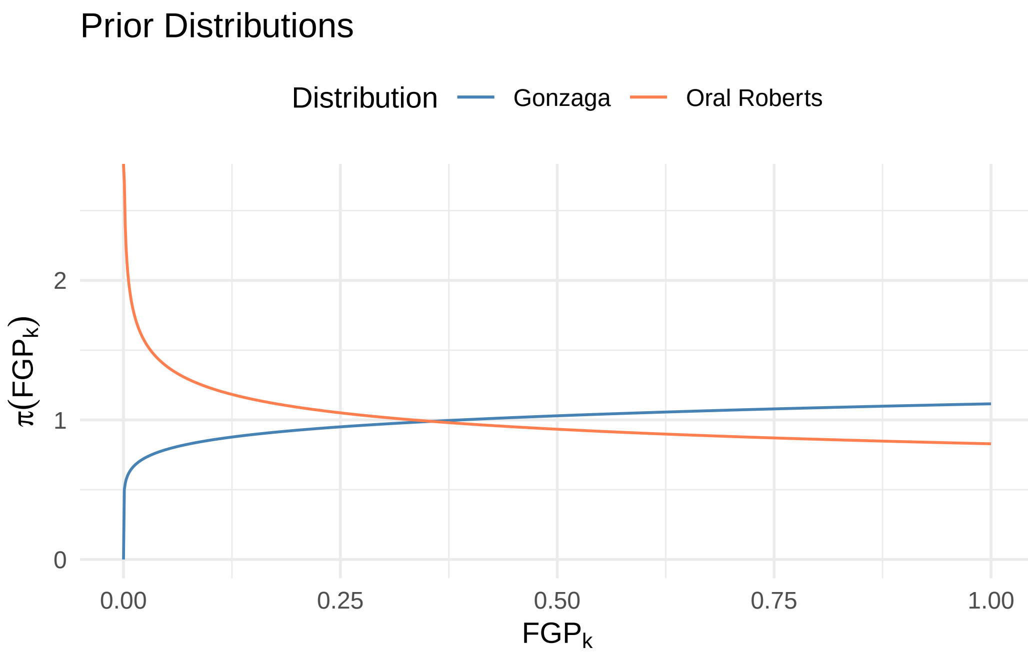
Prior Distributions for Oral Roberts and Gonzaga

Field Goal Percentage

$\text{FGP}_k \sim \text{Beta}(\alpha_k, \beta_k) \quad \forall k \in \text{Teams}$

```
k = 1211 #Gonzaga
l = 1331 #Oral Robers

k.alpha = season.2022.priors %>%
  filter(TeamID == k) %>% pull(fgp.alpha.prior)
k.beta = season.2022.priors %>%
  filter(TeamID == k) %>% pull(fgp.beta.prior)

l.alpha = season.2022.priors %>%
  filter(TeamID == l) %>% pull(fgp.alpha.prior)
l.beta = season.2022.priors %>%
  filter(TeamID == l) %>% pull(fgp.beta.prior)

ggplot(data = data.frame(x = c(0, 1)), aes(x)) +
  stat_function(fun = dbeta, n = 1001,
                args = list(shape1 = k.alpha, shape2 = k.beta),
                aes(color = "Gonzaga"),
                show.legend=T) +
  stat_function(fun = dbeta, n = 1001,
                args = list(shape1 = l.alpha, shape2 =l.beta),
                aes(color = "Oral Roberts"), show.legend=T) +
  ylab(expression(pi(FGP[k]))) +
  xlab(expression(FGP[k])) +
  ggtitle("Prior Distributions") +
```

```r
  theme_minimal() +
  labs(color = "Distribution") +
  scale_color_manual(
    values = c("Gonzaga" = "steelblue", "Oral Roberts" = "coral")) +
  theme(legend.position = "top")
```

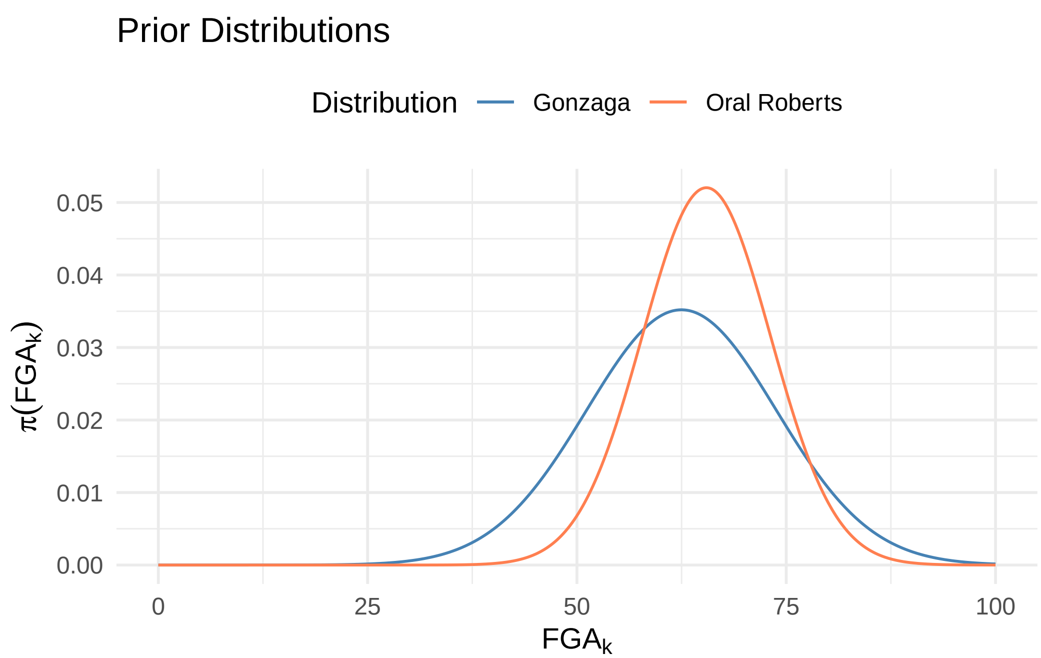


Field Goal Attempts

$\text{FGA}_k \sim N(\lambda_k, \tau_k^2)$

```r
k = 1211 #Gonzaga
l = 1331 #Oral Robers

k.lambda = season.2022.priors %>%
  filter(TeamID == k) %>% pull(fga.lambda.prior)
k.tau = season.2022.priors %>%
  filter(TeamID == k) %>% pull(fga.tau.prior)

l.lambda = season.2022.priors %>%
  filter(TeamID == l) %>% pull(fga.lambda.prior)
l.tau = season.2022.priors %>%
```

```
  filter(TeamID == 1) %>% pull(fga.tau.prior)

ggplot(data = data.frame(x = c(0, 100)), aes(x)) +
  stat_function(fun = dnorm, n = 1001,
                args = list(mean = k.lambda, sd = k.tau),
                aes(color = "Gonzaga"),
                show.legend=T) +
  stat_function(fun = dnorm, n = 1001,
                args = list(mean = l.lambda, sd =l.tau),
                aes(color = "Oral Roberts"), show.legend=T) +
  ylab(expression(pi(FGA[k]))) +
  xlab(expression(FGA[k])) +
  ggtitle("Prior Distributions") +
  theme_minimal() +
  labs(color = "Distribution") +
  scale_color_manual(
    values = c("Gonzaga" = "steelblue", "Oral Roberts" = "coral")) +
  theme(legend.position = "top")
```

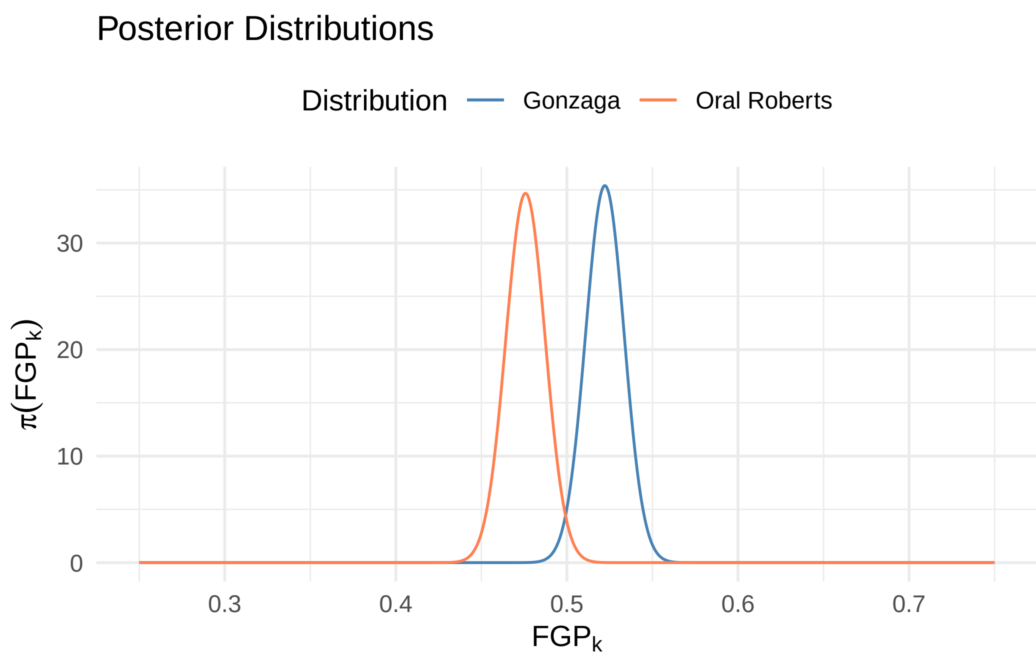


Posterior Distributions for Oral Roberts and Gonzaga

Field Goal Percentage

$\text{FGP}_k|\text{Data}_k \sim \text{Beta}(\alpha_k, \beta_k) \quad \forall k \in \text{Teams}$

```
k = 1211 #Gonzaga
l = 1331 #Oral Robers

k.alpha = season.2023.posteriors %>%
  filter(TeamID == k) %>% pull(fgp.alpha.posterior)
k.beta = season.2023.posteriors %>%
  filter(TeamID == k) %>% pull(fgp.beta.posterior)

l.alpha = season.2023.posteriors %>%
  filter(TeamID == l) %>% pull(fgp.alpha.posterior)
l.beta = season.2023.posteriors %>%
  filter(TeamID == l) %>% pull(fgp.beta.posterior)

ggplot(data = data.frame(x = c(0.25, 0.75)), aes(x)) +
  stat_function(fun = dbeta, n = 1001,
              args = list(shape1 = k.alpha, shape2 = k.beta),
              aes(color = "Gonzaga"),
              show.legend=T) +
  stat_function(fun = dbeta, n = 1001,
              args = list(shape1 = l.alpha, shape2 =l.beta),
              aes(color = "Oral Roberts"), show.legend=T) +
  ylab(expression(pi(FGP[k]))) +
  xlab(expression(FGP[k])) +
  ggtitle("Posterior Distributions") +
  theme_minimal() +
  labs(color = "Distribution") +
  scale_color_manual(
    values = c("Gonzaga" = "steelblue", "Oral Roberts" = "coral")) +
  theme(legend.position = "top")
```



We estimated the following the posterior distributions for their Field Goal Percentage:

$\mathrm{FGP}_{\mathrm{Gonzaga}}|\mathrm{Data}_{\mathrm{Gonzaga}} \sim \mathrm{Beta}(1026.116, 939)$

$\mathrm{FGP}_{\mathrm{Oral\ Roberts}}|\mathrm{Data}_{\mathrm{Oral\ Roberts}} \sim \mathrm{Beta}(896.829, 988)$

Hence,

$E(\mathrm{FGP}_{\mathrm{Gonzaga}}|\mathrm{Data}_{\mathrm{Gonzaga}}) = 0.522$

$E(\mathrm{FGP}_{\mathrm{Oral\ Roberts}}|\mathrm{Data}_{\mathrm{Oral\ Roberts}}) = 0.476$

$V(\mathrm{FGP}_{\mathrm{Gonzaga}}|\mathrm{Data}_{\mathrm{Gonzaga}})$=$1.2690439 \times 10^{-4}$

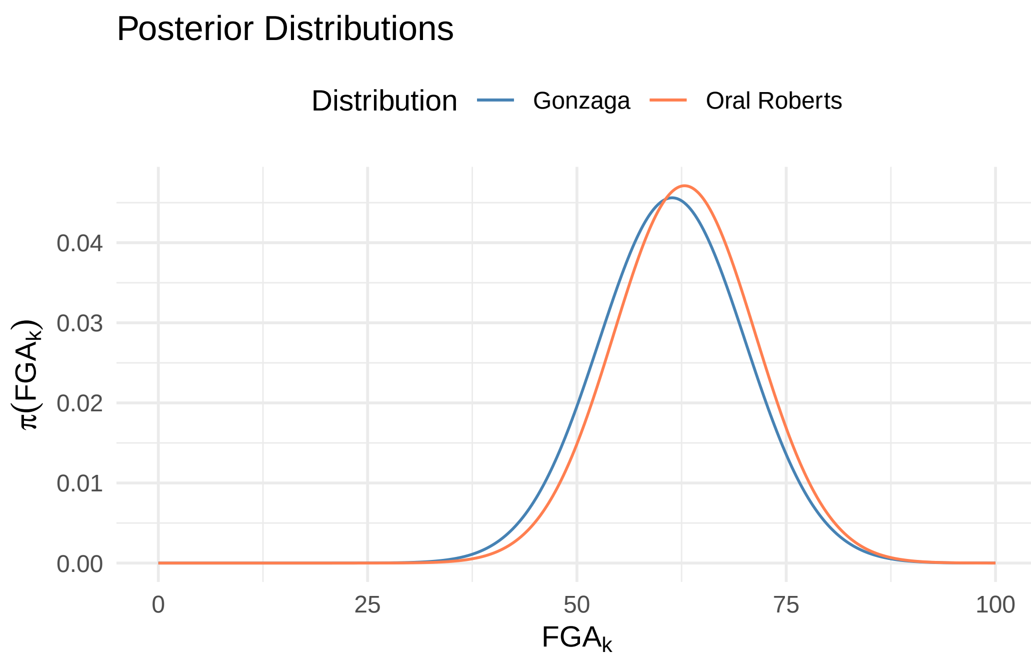
$V(\mathrm{FGP}_{\mathrm{Oral\ Roberts}}|\mathrm{Data}_{\mathrm{Oral\ Roberts}})$=$1.3225751 \times 10^{-4}$

Field Goal Attempts

$\mathrm{FGA}_k|\mathrm{Data}_k \sim N(\mu_k, \sigma_k^2)$‘

```
k = 1211 #Gonzaga
l = 1331 #Oral Robers

k = which(season.2023.posteriors$TeamID == k)
l = which(season.2023.posteriors$TeamID == l)
```

```
k.mu = mean(posterior.normal.matrix[k,])
k.sigma = sqrt(mean(posterior.invgamma.matrix[k,]))

l.mu = mean(posterior.normal.matrix[l,])
l.sigma = sqrt(mean(posterior.invgamma.matrix[l,]))

ggplot(data = data.frame(x = c(0, 100)), aes(x)) +
  stat_function(fun = dnorm, n = 1001,
                args = list(mean = k.mu, sd = k.sigma),
                aes(color = "Gonzaga"),
                show.legend=T) +
  stat_function(fun = dnorm, n = 1001,
                args = list(mean = l.mu, sd =l.sigma),
                aes(color = "Oral Roberts"), show.legend=T) +
  ylab(expression(pi(FGA[k]))) +
  xlab(expression(FGA[k])) +
  ggtitle("Posterior Distributions") +
  theme_minimal() +
  labs(color = "Distribution") +
  scale_color_manual(
    values = c("Gonzaga" = "steelblue", "Oral Roberts" = "coral")) +
  theme(legend.position = "top")
```



We estimated the following for the posterior distributions for Gonzaga and Oral Roberts for their Field Goal Attempts:

$\text{FGA}_{\text{Gonzaga}}|\text{Data}_{\text{Gonzaga}} \sim N(61.37, 8.45^2)$

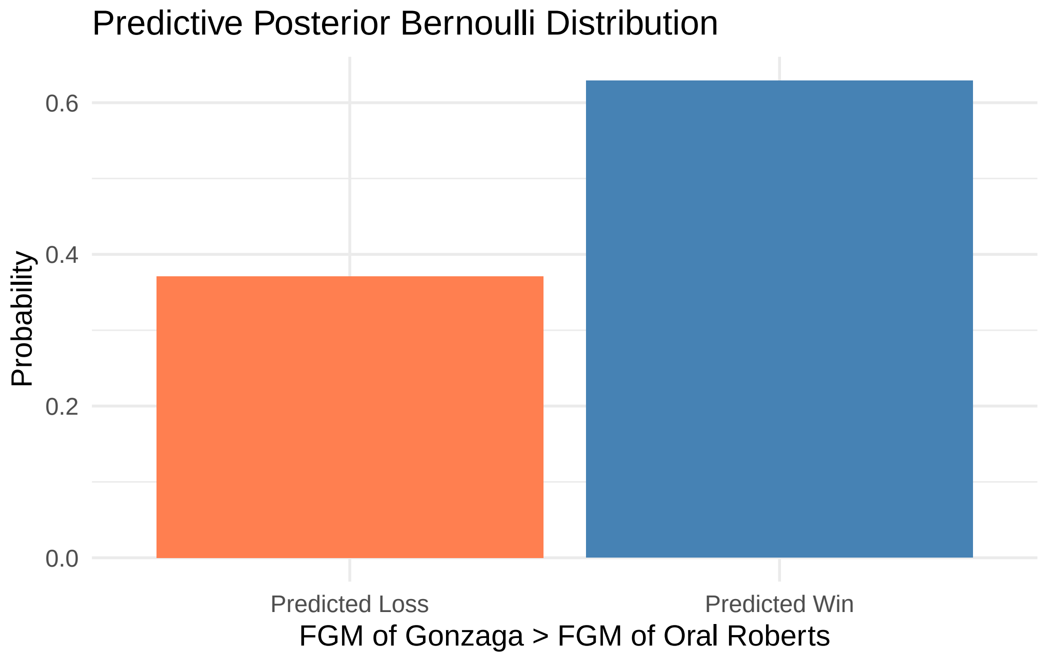
$\text{FGA}_{\text{Oral Roberts}}|\text{Data}_{\text{Oral Roberts}} \sim N(62.86, 8.44^2)$

Posterior Predictive Distribution on FGM

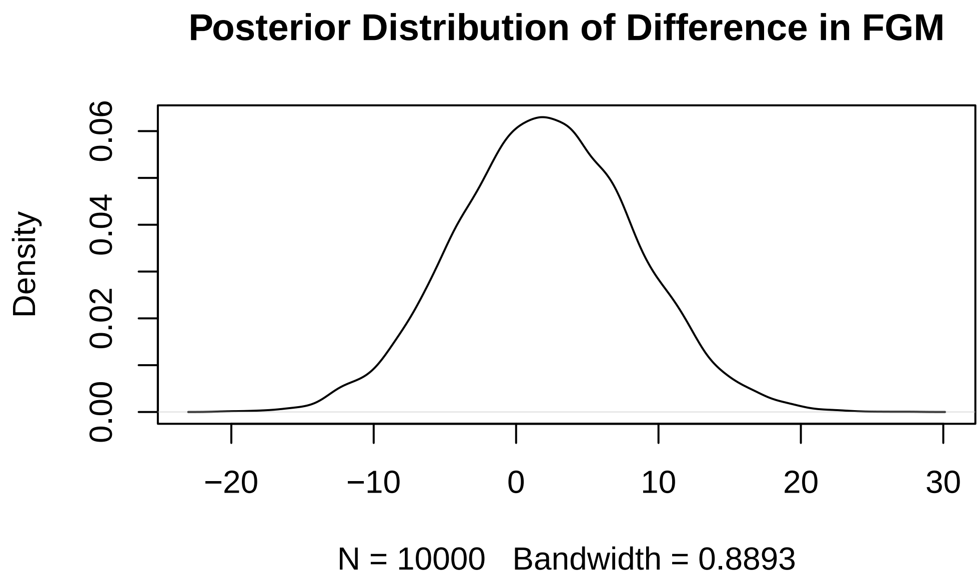
$\text{FGM}_k|\text{Data}_k > \text{FGM}_l|\text{Data}_k \sim \text{Bernoulli}(p_{kl})$

```
k = 1211 #Gonzaga
l = 1331 #Oral Robers

k = which(season.2023.posteriors$TeamID == k)
l = which(season.2023.posteriors$TeamID == l)

p.kl = data.frame(Event = c("Predicted Win", "Predicted Loss"),
                  Probability = c(mean(posterior.fgm.matrix[k,] > posterior.fgm.matrix[l,]
                                          1 - mean(posterior.fgm.matrix[k,]

ggplot(p.kl, aes(x = Event, y = Probability, fill = Event)) +
  geom_bar(stat = "identity") +
```

```
  labs(title = "Predictive Posterior Bernoulli Distribution", x = "FGM of Gonzaga > FGM of
  scale_fill_manual(values = c("Predicted Win" = "steelblue", "Predicted Loss" = "coral"))
  theme_minimal()+
  theme(legend.position = "none")
```




```
plot(density(posterior.fgm.matrix[k,] - posterior.fgm.matrix[l,]), main="Posterior Distrib
```



If we assume that the FGA is independent of FGP, then,

$E(\text{FGM}_{\text{Gonzaga}}|\text{Data}_{\text{Gonzaga}}) = E(\text{FGP}_{\text{Gonzaga}}|\text{Data}_{\text{Gonzaga}})E(\text{FGA}_{\text{Gonzaga}}|\text{Data}_{\text{Gonzaga}})$ $=32.0470887$

$E(\text{FGM}_{\text{Oral Roberts}}|\text{Data}_{\text{Oral Roberts}}) = E(\text{FGP}_{\text{Oral Roberts}}|\text{Data}_{\text{Oral Roberts}})E(\text{FGA}_{\text{Oral Roberts}}|\text{Data}_{\text{Roberts}})$ $=29.908236$

Hence,

$E(\text{FGM}_{\text{Gonzaga}}|\text{Data}_{\text{Gonzaga}} - \text{FGM}_{\text{Oral Roberts}}|\text{Data}_{\text{Oral Roberts}})=E(\text{FGM}_{\text{Gonzaga}}|\text{Data}_{\text{Gonzaga}}) - E(\text{FGM}_{\text{Oral Roberts}}|\text{Data}_{\text{Oral Roberts}})=2.1388527$

Where the expected values from FGA were approximated using Gibbs sampling techniques.

Our estimated posterior predictive mean using Monte Carlo Approximation is 2.1009086.

Our estimated posterior predictive variance using Monte Carlo Approximation is 38.8676893

95% Credible Interval for Difference in FGM

```
     2.5%     97.5%
-10.03584  14.42016
```

Given our data and prior knowledge, the there is a 95% probability that the difference in FGM between Gonzaga and Oral Roberts will be between -9.84 and 14.