Statistics 143 — Spring 2024 — Assignment 3
Due Monday March 4, 2024

Homework is to be uploaded on Gradescope by 10:00pm on Monday evening.

Please make sure on your assignment you indicate clearly other students with whom you collaborated, as well as any assistance you received from generative AI tools.

Written assignment

1. The Bayesian approach to summarizing model predictions is to average over the uncertainty of the model parameters. This is in contrast to the classical approach which substitutes parameters with point estimates. Generally, the Bayesian approach is more careful to incorporate uncertainty into predictions than classical approaches. This problem examines how to determine the predictive distribution of a binary game outcome factoring in the uncertainty of the model parameters.

   Suppose two teams, labeled A and B, are about to compete in a head-to-head game. Assume the respective team strengths are θ_A and θ_B, and that the probability team A defeats team B follows the Thurstone-Mosteller model, that is,

   $$\Pr(Y_{AB} = 1|\theta_A, \theta_B) = \Phi(\theta_A - \theta_B),$$

   where $Y_{AB} = 1$ if A defeats B and $Y_{AB} = 0$ if A loses to B, and $\Phi(\cdot)$ is the cumulative distribution function of a standard normal distribution. Also assume that from previous competition among other teams, the posterior distribution of the strength parameters are, independently, $\theta_A \sim \mathrm{N}(\mu_A, \sigma_A^2)$ and $\theta_B \sim \mathrm{N}(\mu_B, \sigma_B^2)$. The goal of this problem is to obtain the posterior mean probability that team A defeats team B. That is, we want to determine a simplified expression for

   $$\mathrm{E}(\Phi(\theta_A - \theta_B)) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \Phi(\theta_A - \theta_B)\varphi(\theta_A|\mu_A, \sigma_A^2)\varphi(\theta_B|\mu_B, \sigma_B^2)d\theta_A d\theta_B,$$

   where $\varphi(\cdot|\mu, \sigma^2)$ is the density function of a normal distribution with mean μ and variance σ^2.

   For the remainder of the problem, it may be helpful to recall the following properties about normal densities.

   - $\varphi(\theta|\mu, \sigma^2) = \varphi(\mu|\theta, \sigma^2)$.
   - $\varphi(\theta|\mu, \sigma^2) = \varphi(-\theta| - \mu, \sigma^2)$.
   - $\varphi(\theta + c|\mu, \sigma^2) = \varphi(\theta|\mu - c, \sigma^2)$ for real-valued c.
   - $\varphi(\theta|\mu_0, \sigma^2 + \sigma_0^2) = \int_{-\infty}^{\infty} \varphi(\theta|\mu, \sigma^2)\varphi(\mu|\mu_0, \sigma_0^2)d\mu$

   You may use these properties without proof.

   (a) By definition of a cdf,

   $$\Phi(\theta_A - \theta_B) = \int_{-\infty}^{\theta_A - \theta_B} \varphi(z|0, 1)dz.$$

Show that this expression is equivalent to

$$\Phi(\theta_A - \theta_B) = \int_{-\infty}^{0} \varphi(z| - (\theta_A - \theta_B), 1) dz.$$

Hint: This problem does not rely on the above normal density properties.

Solution:

$\boxed{\text{Approach1}}$

Let $z' = z - (\theta_A - \theta_B)$, then

$$\begin{aligned} dz' &= d(z - (\theta_A - \theta_B)) = dz \\ \Phi(\theta_A - \theta_B) &= \int_{-\infty}^{\theta_A - \theta_B} \varphi(z|0,1)dz = \int_{-\infty}^{\theta_A - \theta_B} \varphi(z - (\theta_A - \theta_B)| - (\theta_A - \theta_B), 1)dz \\ &= \int_{-\infty}^{0} \varphi(z'| - (\theta_A - \theta_B), 1)dz' = \int_{-\infty}^{0} \varphi(z| - (\theta_A - \theta_B), 1)dz \end{aligned}$$

$\boxed{\text{Approach2}}$

Note that

$$\Phi(\theta_A - \theta_B) = \int_{-\infty}^{0} \varphi(z| - (\theta_A - \theta_B), 1)dz = \Pr(X < 0),$$

where $X \sim \mathrm{N}(-(\theta_A - \theta_B), 1)$. We can *standardize* X to get standard normal distribution

$$Z = X + \theta_A - \theta_B, \ Z \sim \mathrm{N}(0, 1)$$

Then,

$$\begin{aligned} \Pr(X < 0) = \Pr(Z - (\theta_A - \theta_B) < 0) &= \Pr(Z < \theta_A - \theta_B) \\ &= \int_{-\infty}^{\theta_A - \theta_B} \varphi(z|0,1)dz = \Phi(\theta_A - \theta_B) \end{aligned}$$

(b) Replace $\Phi(\theta_A - \theta_B)$ in the integrand of the double-integral at the start of the problem with $\int_{-\infty}^{0} \varphi(z| - (\theta_A - \theta_B), 1)dz$. Rearrange the order of integration, and simplify the inner integral in black font

$$\mathrm{E}(\Phi(\theta_A - \theta_B)) = \int_{-\infty}^{0} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \varphi(z| - (\theta_A - \theta_B), 1)\varphi(\theta_A|\mu_A, \sigma_A^2)d\theta_A \varphi(\theta_B|\mu_B, \sigma_B^2)d\theta_B dz$$

so that the resulting expression is $\varphi(z|\mu_*, \sigma_*^2)$. What are μ_* and σ_*^2 in terms of θ_B, μ_A, and σ_A^2?

We can apply the properties at the start of the problem to simplify the integral:

$$\begin{aligned} &\int_{-\infty}^{\infty} \varphi(z| - (\theta_A - \theta_B), 1)\varphi(\theta_A|\mu_A, \sigma_A^2)d\theta_A \\ =&\int_{-\infty}^{\infty} \varphi(-z + \theta_B|\theta_A, 1)\varphi(\theta_A|\mu_A, \sigma_A^2)d\theta_A \\ =&\varphi(-z + \theta_B|\mu_A, 1 + \sigma_A^2) = \varphi(z|\theta_B - \mu_A, 1 + \sigma_A^2) \end{aligned}$$

Thus, $\mu_* = \theta_B - \mu_A$ and $\sigma_*^2 = 1 + \sigma_A^2$.

(c) Having integrated θ_A out of the original problem's integrand, now let us integrate θ_B out of the expression that remains. Simplify the integral in black font

$$E(\Phi(\theta_A - \theta_B)) = \int_{-\infty}^{0} \int_{-\infty}^{\infty} \varphi(z|\mu_*, \sigma_*^2)\varphi(\theta_B|\mu_B, \sigma_B^2) d\theta_B dz$$

so that the resulting expression is $\varphi(z|\mu_z, \sigma_z^2)$. What are μ_z and σ_z^2 in terms of μ_A, μ_B, σ_A^2 and σ_B^2?

The integral in black font is equivalent to

$$\begin{aligned}
&\int_{-\infty}^{\infty} \varphi(z|\mu_*, \sigma_*^2)\varphi(\theta_B|\mu_B, \sigma_B^2) d\theta_B \\
=&\int_{-\infty}^{\infty} \varphi(z|\theta_B - \mu_A, 1 + \sigma_A^2)\varphi(\theta_B|\mu_B, \sigma_B^2) d\theta_B \\
=&\int_{-\infty}^{\infty} \varphi(z + \mu_A|\theta_B, 1 + \sigma_A^2)\varphi(\theta_B|\mu_B, \sigma_B^2) d\theta_B \\
=&\varphi(z + \mu_A|\mu_B, 1 + \sigma_A^2 + \sigma_B^2) = \varphi(z|\mu_z, \sigma_z^2)
\end{aligned}$$

where

$$\begin{aligned}
\mu_z &= \mu_B - \mu_A \\
\sigma_z^2 &= 1 + \sigma_A^2 + \sigma_B^2
\end{aligned}$$

(d) Finally, express the integral in part (c) as $\Phi(D)$, where $\Phi(\cdot)$ is the standard normal cdf, and D is a function of the known parameters μ_A, μ_B, σ_A^2 and σ_B^2. Interpret this expression in comparison to the original Thurstone-Mosteller probability if you had substituted in point estimates for θ_A and θ_B.

From part (a),

$$\begin{aligned}
E(\Phi(\theta_A - \theta_B)) &= \int_{-\infty}^{0} \varphi(z|\mu_B - \mu_A, 1 + \sigma_A^2 + \sigma_B^2) dz \\
&= \Phi\left(\frac{-(\mu_B - \mu_A)}{\sqrt{1 + \sigma_A^2 + \sigma_B^2}}\right) \\
&= \Phi\left(\frac{\mu_A - \mu_B}{\sqrt{1 + \sigma_A^2 + \sigma_B^2}}\right)
\end{aligned}$$

If we have a point estimation of θ_A and θ_B,

$$\theta_A = \mu_A, \ \theta_B = \mu_B$$

Then we see the Thurstone-Mosteller probability

$$\Pr(Y_{AB} = 1|\theta_A, \theta_B) = \Phi(\theta_A - \theta_B) = \Phi(\mu_A - \mu_B)$$

Compared to the Thurstone-Mosteller probability, the expected win probability has $\sqrt{1 + \sigma_A^2 + \sigma_B^2}$ in the denominator, which is larger than 1, so its value is "shrunk" closer to 0.5 compared to the version that ignores the posterior variances.

2. The rating system attributed to Wesley Colley (see https://www.colleyrankings.com/method.html and the document linked within) has been used by some analysts to estimate college football teams. The system assumes that, for each team $i = 1, \ldots, J$,

$$r_i = \frac{\sum_{j=1}^{n_{tot,i}} r_j^{(i)} + 1 + (n_{w,i} - n_{\ell,i})/2}{2 + n_{tot,i}}$$

where

r_i is the unknown rating (strength) of team i,

$r_j^{(i)}$ is the unknown rating of the j-th opponent of team i,

$n_{w,i}$ is the total number of wins by team i,

$n_{\ell,i}$ is the total number of losses by team i, and

$n_{tot,i}$ is the total number of games played by team i, where $n_{w,i} + n_{\ell,i} = n_{tot,i}$

The goal of the Colley system is to estimate the r_i across all teams.

(a) Let $\boldsymbol{r} = (r_1, \ldots, r_J)^\mathsf{T}$ be the vector of ratings across the J teams. Now define vector $\boldsymbol{b} = (b_1, \ldots, b_J)^\mathsf{T}$ and $J \times J$ matrix $\boldsymbol{C}$ such that, for all $i = 1, \ldots, J$,

$$b_i = 1 + (n_{w,i} - n_{\ell,i})/2,$$

and

$$\begin{aligned} c_{ii} &= 2 + n_{tot,i} \\ c_{ij} &= -n_{j,i} \end{aligned}$$

where $n_{j,i}$ is the number of times team i has played team j. Show that the equation at the start of the problem for r_i can be expressed in matrix form as

$$\boldsymbol{C}\boldsymbol{r} = \boldsymbol{b}.$$

The matrix $\boldsymbol{C}$ is called the "Colley matrix" (at least by Colley). This equation, therefore, implies that $\boldsymbol{r} = \boldsymbol{C}^{-1}\boldsymbol{b}$ assuming $\boldsymbol{C}$ is invertible.

Solution:

Let $J = \{1, \cdots, J\}$ be the set of all teams. For each component of $\boldsymbol{b}$, we have

$$\sum_{j \in [J]} c_{ij} r_j = b_i$$

$$\sum_{j \in [J],\ j \neq i} -n_{j,i} r_j + (2 + n_{tot,i}) r_i = 1 + (n_{w,i} - n_{\ell,i})/2$$

$$(2 + n_{tot,i}) r_i = 1 + (n_{w,i} - n_{\ell,i})/2 + \sum_{j \in [J],\ j \neq i} n_{j,i} r_j$$

Note that $\sum_{j \in [J],\ j \neq i} n_{j,i} r_j$ is nothing more than a more compact expression of $\sum_{j=1}^{n_{tot,i}} r_j^{(i)}$, in which we group team i's opponent *in each game*, $r_j^{(i)}$, by the opponent team (imagining that team i played against team j in $n_{j,i}$ games, then $\sum_{j=1}^{n_{tot,i}} r_j^{(i)}$ would add r_j for

$n_{j,i}$ times.) Thus,

$$(2 + n_{tot,i})r_i = 1 + (n_{w,i} - n_{\ell,i})/2 + \sum_{j=1}^{n_{tot,i}} r_j^{(i)}$$

$$r_i = \frac{1 + (n_{w,i} - n_{\ell,i})/2 + \sum_{j=1}^{n_{tot,i}} r_j^{(i)}}{2 + n_{tot,i}}$$

This is the same equation at the start of the problem.

(b) Show that the solution to part (a) is, in fact, a ridge-penalized estimate of $\boldsymbol{r}$ by showing that $\boldsymbol{r} = \boldsymbol{C}^{-1}\boldsymbol{b}$ can be re-expressed as

$$\boldsymbol{r} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \mathbf{I})^{-1}(\boldsymbol{X}^\top \boldsymbol{y} + \lambda \gamma),$$

where

$\boldsymbol{X}$ is the usual pairing matrix in which row k, representing the pairing of team i_k and j_k, consists of 1 as the i_k element and -1 as the j_k in the element, and 0 everywhere else,

$\boldsymbol{y}$ is a vector of length n of game outcomes, in which y_k is assigned the value 0.5 if team i_k wins and -0.5 if team j_k wins,

λ, the ridge penalty parameter, is set to 2,

$\boldsymbol{\gamma}$, the centering vector, consists of elements $\gamma_j = 0.5$ for all $j = 1, \ldots, J$, and

$\mathbf{I}$ is the $J \times J$ identity matrix.

Let $(\boldsymbol{X}^\top \boldsymbol{X})_{ii}$ denote the i-th diagonal element of $\boldsymbol{X}^\top \boldsymbol{X}$, then we can compute it by

$$(\boldsymbol{X}^\top \boldsymbol{X})_{ii} = \boldsymbol{X}_{\cdot i} \cdot \boldsymbol{X}_{\cdot i} = \sum_k x_{ki} x_{ki}$$
$$= \sum_k x_{ki}^2 = n_{tot,i}$$

Similarly, we can compute the (i, j)-th element of $(\boldsymbol{X}^\top \boldsymbol{X})$ by

$$(\boldsymbol{X}^\top \boldsymbol{X})_{ij} = \boldsymbol{X}_{\cdot i} \cdot \boldsymbol{X}_{\cdot j} = \sum_k x_{ki} x_{kj}$$

By the construction of $\boldsymbol{X}$, $x_{ki}x_{kj} = -1$ if team i and j played in the game k, and 0 otherwise. So

$$(\boldsymbol{X}^\top \boldsymbol{X})_{ij} = \sum_k x_{ki} x_{kj} = -n_{j,i} = c_{ij}$$

With $\lambda = 2$, we see the i-th diagonal element of $(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \mathbf{I})$,

$$(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \mathbf{I})_{ii} = n_{tot,i} + 2 = c_{ii}$$

Thus we see $(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \mathbf{I}) = \boldsymbol{C}$.
Moreover, the i-th element of $\boldsymbol{X}^\top \boldsymbol{y}$ is

$$(\boldsymbol{X}^\top \boldsymbol{y})_i = \sum_k x_{ki} y_k$$

- If team i is the home team in game k and the home team won in the game k, or team i is the visitor team in game k and the visitor team won in the game k, $x_{ki}y_k = 0.5$.
- If team i lost in the game k, no matter played at home or away, $x_{ki}y_k = -0.5$.
- If team i not played in the game k, $x_{ki}y_k = 0$.

Thus

$$(\boldsymbol{X}^\top \boldsymbol{y})_i = \sum_k x_{ki}y_k = 0.5 \times (n_{w,i} - n_{l,i})$$

$\lambda\gamma$ is a constant vector of 1, so the i-th element of $(\boldsymbol{X}^\top \boldsymbol{y} + \lambda\gamma)$ is

$$(\boldsymbol{X}^\top \boldsymbol{y} + \lambda\gamma)_i = 0.5 \times (n_{w,i} - n_{l,i}) + 1 = b_i$$

We then conclude that the estimation of $\boldsymbol{r}$ with ridge regularization

$$\boldsymbol{r} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \mathbf{I})^{-1}(\boldsymbol{X}^\top \boldsymbol{y} + \lambda\gamma) = \boldsymbol{C}^{-1}\boldsymbol{b}$$

(c) The Colley system essentially assumes that $\mathrm{E}(y_{ij}) = r_i - r_j$, where the y_{ij} can be 0.5 or -0.5, and where the r_i and r_j are estimated using a ridge penalty approach. Given this correspondence to the ridge-penalized linear model, does Colley's choice of $\gamma_j = 0.5$ for all $j = 1, \cdots, J$ seem like a natural choice to center the ratings? If you were designing the system, where would you center the ratings?

No, the choice of $\boldsymbol{\gamma} = 0.5$ essentially centers all r_i to 0.5, which is not a natural choice. Because Colley models $\mathrm{E}(y_{ij})$ as the difference of strength estimates, centering all r_i's at which value does not make too much difference. Without any additional information, there is no good reason to center r_i's to 0.5 rather than 0. If I were designing the system, I might choose $\gamma_j = 0$ for all $j = 1, \cdots, J$ instead.

3. The file `nfl_score_2023.csv` in the Data Sets folder on the course Canvas site contains the game results of the 2023-24 NFL regular season. The data set consists of 272 games. The following variables are included in this data set.

`Week`: Week of the season

`Home`: Home team

`Away`: Visiting team

`Pts.Diff`: Home team score minus visiting team score for each game

`AtHome`: 1 if played at home team's stadium, 0 if played at a neutral site or not the usual home stadium

The goal of this problem is to answer questions about team strength during the 2023 NFL season. No games during the 2023 NFL regular season ended in a tie.

(a) Create a binary outcome variable (call this `y`), and fit the Bradley-Terry model with a home advantage parameter on the NFL data. Display the estimated strengths and standard errors of the teams in order from best to worst. What is the estimated home field advantage? For two teams of the same strength, what is the estimated probability that one defeats the other at the first team's home stadium?

Solution:

```
> library("glmnet")
> set.seed(143)  # for replicability
> nfl23 = read.csv("nfl_score_2023.csv")
> nfl23$y = ifelse(nfl23$Pts.Diff>0, 1, 0)
> Teams = unique(nfl23$Home)
> n_teams = length(Teams)
> # Bradley-Terry model
> # Construct design matrix
> X = outer(nfl23$Home, Teams, "==") - outer(nfl23$Away, Teams, "==")
> W = rbind(diag(n_teams-1), rep(-1, n_teams-1))
> X.star = X %*% W
> nfl23$X.star = X.star
> # Fit logit model
> ability.bt = glm(y~X.star+AtHome+0, family='binomial', data=nfl23)
> W_h = rbind(cbind(W,0),c(rep(0,n_teams-1),1))  # block matrix
> Teams.ability.bt = data.frame("Est"=W_h %*% ability.bt$coefficients)
> V.bt = W_h %*% vcov(ability.bt) %*% t(W_h) # covariance matrix
> Teams.ability.bt["StdErr"] = sqrt(diag(V.bt))
> row.names(Teams.ability.bt) <- c(Teams,"HFA")
> dimnames(V.bt) = list(c(Teams,"HFA"), c(Teams,"HFA"))
> Teams.rank = Teams.ability.bt[1:n_teams,]
> Teams.rank[order(-Teams.rank$Est),]
                              Est    StdErr
Baltimore Ravens       1.69947991 0.6026412
San Francisco 49ers    1.29218881 0.5707904
Cleveland Browns       1.11510426 0.5579844
Pittsburgh Steelers    0.86523115 0.5463375
Dallas Cowboys         0.82470356 0.5826912
Detroit Lions          0.79792320 0.5773370
Los Angeles Rams       0.77675581 0.5415526
Cincinnati Bengals     0.69695032 0.5280465
Philadelphia Eagles    0.63357858 0.5499420
Seattle Seahawks       0.49304184 0.5524143
Kansas City Chiefs     0.48310089 0.5293532
Buffalo Bills          0.47868415 0.5451046
Miami Dolphins         0.37673414 0.5683399
Jacksonville Jaguars   0.35425337 0.5300409
Houston Texans         0.34804339 0.5406641
Indianapolis Colts     0.14374758 0.5372967
Tampa Bay Buccaneers  -0.06719715 0.5342274
Green Bay Packers     -0.09913570 0.5225345
New Orleans Saints    -0.26627529 0.5312069
Las Vegas Raiders     -0.30742793 0.5270345
Denver Broncos        -0.31593102 0.5343791
Minnesota Vikings     -0.49017951 0.5356007
Tennessee Titans      -0.51528243 0.5624906
New York Jets         -0.53388382 0.5433629
New York Giants       -0.66086621 0.5552147
```

```
Chicago Bears         -0.69328167 0.5395600
Atlanta Falcons       -0.84048508 0.5341092
Arizona Cardinals     -0.86982930 0.6135577
Los Angeles Chargers  -0.94914218 0.5578082
Washington Commanders -1.28666447 0.6026627
New England Patriots  -1.31406028 0.6045930
Carolina Panthers     -2.16987894 0.7626631
```

Bradley-Terry model estimate

$$\Pr(y_i = 1) = F(\theta_a - \theta_b + \beta_h) = \frac{1}{1 + \exp\left(-(\theta_a - \theta_b + \beta_h)\right)}$$

For two teams with the same strength estimate,

$$\Pr(y_i = 1) = \frac{1}{1 + \exp(-\beta_h)}$$

This model estimate $\beta_h = 0.2603$, so $\Pr(y_i = 1) = 0.5647$

```
> 1 / (1 + exp(-Teams.ability.bt['HFA', 'Est']))
[1] 0.5647102
```

(b) Is the Ford (1957) requirement a concern for these data? Why or why not? *Hint: What might you expect as a result of fitting the Bradley-Terry model if the Ford condition were violated?*

No, this data **does not** violate Ford criterion. If the Ford condition were violated by having one subset of teams always defeating the remaining set of teams, then the strength estimates would be nonsensically large (or positive or negative). This does not happen, so there is no concern about the Ford condition being violated. Also, it is worth mentioning that with the NFL schedule, there is no partitioning the teams into two subsets where no team in one subset does not play teams in the other.

(c) Fit a ridge-regularized version of the Bradley-Terry model selecting the ridge penalty by 10-fold cross-validation, again including a home advantage parameter in the model.

i. Arguably the home field effect should not be shrunk via the ridge penalty. Explain why one should not regularize the home-field effect.

Regularization is used here to address the collinearity of the columns of the X matrix, which would lead to the unidentifiability of the strength parameters. Because the HFA parameter is not involved in the unidentifiability, it does not need to undergo a penalty.

ii. Use the `penalty.factor` argument of `cv.glmnet` and `glmnet` to prevent the home field parameter from being shrunk via ridge penalization. Display the estimated strengths of the teams in order from best to worst. Comment on the differences in the estimated team strengths between the unregularized Bradley-Terry estimates and the ridge-regularized estimates. Also comment on the difference in the home field parameter between the two fitted models.

<u>Code:</u>

```
> # 3(c) Model with ridge regularization
> # K-fold cross-validation
> K = 10
> ability.ridge.cv =
+   cv.glmnet(x=cbind(X, nfl23$AtHome), y=data.matrix(nfl23$y),
+             family="binomial",alpha=0, intercept=FALSE,
+             standardize=FALSE,penalty.factor=c(rep(1,n_teams),0))
> # Fit ridge regression model with best lambda
> best.lambda = ability.ridge.cv$lambda.min
> best.lambda
[1] 0.01357242
> ability.ridge.best =
+   glmnet(x=cbind(X, nfl23$AtHome), y=data.matrix(nfl23$y),
+          family="binomial", lambda = best.lambda, alpha=0,
+          intercept=FALSE, standardize=FALSE,
+          penalty.factor=c(rep(1,n_teams),0))
> Teams.ability=data.frame('BT.Est' = round(Teams.ability.bt$Est,3),
+                          'Ridge.Est' =
+                            round(ability.ridge.best$beta[,1],3))
> rownames(Teams.ability)=c(Teams,'HFA')
> Teams.rank = Teams.ability[1:n_teams,]
> Teams.rank[order(-Teams.rank$Ridge.Est),]
                      BT.Est Ridge.Est
Baltimore Ravens       1.699     0.621
San Francisco 49ers    1.292     0.481
Detroit Lions          0.798     0.426
Dallas Cowboys         0.825     0.401
Cleveland Browns       1.115     0.368
Philadelphia Eagles    0.634     0.306
Kansas City Chiefs     0.483     0.289
Buffalo Bills          0.479     0.280
Miami Dolphins         0.377     0.248
Pittsburgh Steelers    0.865     0.247
Los Angeles Rams       0.777     0.244
Houston Texans         0.348     0.158
Cincinnati Bengals     0.697     0.153
Seattle Seahawks       0.493     0.105
Jacksonville Jaguars   0.354     0.095
Indianapolis Colts     0.144     0.043
Tampa Bay Buccaneers  -0.067     0.039
Green Bay Packers     -0.099     0.034
New Orleans Saints    -0.266    -0.010
Denver Broncos        -0.316    -0.091
Las Vegas Raiders     -0.307    -0.091
Minnesota Vikings     -0.490    -0.182
New York Jets         -0.534    -0.206
Chicago Bears         -0.693    -0.229
Atlanta Falcons       -0.840    -0.274
```

```
Tennessee Titans       -0.515    -0.292
New York Giants        -0.661    -0.302
Los Angeles Chargers   -0.949    -0.426
Arizona Cardinals      -0.870    -0.493
Washington Commanders -1.287    -0.559
New England Patriots   -1.314    -0.562
Carolina Panthers      -2.170    -0.824
> Teams.ability['HFA',]
    BT.Est Ridge.Est
HFA   0.26     0.236
```

We observe that rankings predicted by both model are fairly consistent, meaning that the regularized and un-regularized model largely agrees on the *relative strength* of all teams. However, the *absolute value* of strength estimates by ridge-regularized model is smaller than those from ordinary BT model, which is expected as ridge-regularization centers model parameters to 0. The value of HFA, on the other hand, does not change much in the regularized model.

(d) Fit a regularized Bradley-Terry model with extra pseudo-games optimized via 10-fold cross-validation, again including a home field advantage parameter in the model.

i. Teams in the same division play each other twice during the regular season, once at one team's home field and once at the other team's home field. If the home-field advantage is a parameter in the model, why should we not combine these two games as one binomial outcome in the Y matrix (see the regularization lecture notes, slide 28)?

As in the unregularized BT model, we estimate

$$y_{ij} \sim \text{Binom}(n_{ij}, p_{ij})$$
$$p_{ij} = F(\theta_i - \theta_j + h\beta_h)$$

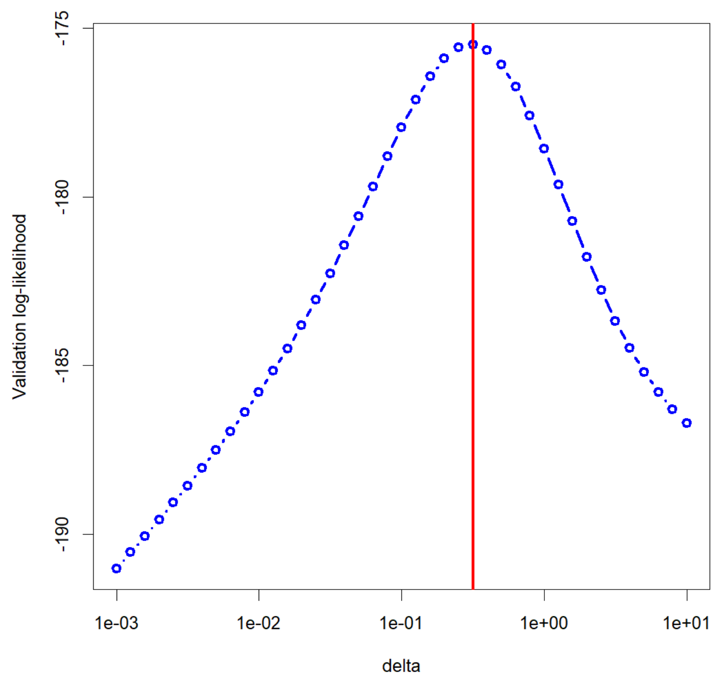
Although these two games are played between the same pair of teams, the estimated p_{ij} still depends on whether $h = 1$ or -1. So we may not combine the result of these two games into one binomial response variable.

ii. From the logic above, the vector of binary game outcomes should be transformed into a binomial outcome with values `c(y+delta, 1-y+delta)` for pseudo-game parameter `delta`. Perform 10-fold cross-validation using this binomial outcome representation, and fit the model. What is the optimized number of pseudo-games to add to each actual game? With this value of `delta`, summarize the estimated team strengths in order from best to worst, and compare the estimated strengths to the ordinary Bradley-Terry results and the ridge-regularized results. Also comment on the estimated home-field advantage parameter and how it compares to the estimates from the previous parts.

```
> # 3(d) Regularization via pseudo-games
> # Select delta by cross-validation
> K=10
> ls.delta = 10^((-30:10)/10)
> pseudo.gm.cv.log.likelihood = rep(0, length(ls.delta))
> # Split train and val dataset for each fold
> dev.val.split = function(i, df.orig, K=10){
```

```
+   df = df.orig[sample(nrow(df.orig)),]  # randomly permute the rows
+   start = (i-1) * (nrow(df) %/% K)+1
+   end = i * (nrow(df) %/% K)
+   if (i==1){
+     dev = df[(end+1):nrow(df),]
+     val = df[1:end,]
+   }else if(i==K){
+     dev = df[1:(start-1),]
+     val = df[start:nrow(df), ]
+   } else{
+     dev = rbind(df[1:(start-1),], df[(end+1):nrow(df),])
+     val = df[start:end,]
+   }
+   return (list(dev, val))
+ }
> # Cross-validation
> for (i in 1:K){
+   print(paste("K =",i))
+   split = dev.val.split(i,nfl23)
+   dev = as.data.frame(split[1])
+   val = as.data.frame(split[2])
+   for (j in 1:length(ls.delta)){
+     delta = ls.delta[j]
+     model = glm(cbind(dev$y+delta, (1-dev$y)+delta)~X.star+AtHome+0,
+           family = "binomial", data=dev)
+     p = predict(model, val, type="response")
+     ll = sum(val[,"y"]*log(p) + (1-val[,"y"])*log(1-p))
+     pseudo.gm.cv.log.likelihood[j]=pseudo.gm.cv.log.likelihood[j]+ll
+   }
+ }
> best.delta =
+   ls.delta[which.max(pseudo.gm.cv.log.likelihood)]
> best.delta
[1] 0.3162278

> plot(ls.delta, pseudo.gm.cv.log.likelihood,type='b',col="blue",lwd=3,
+      xlab="delta",ylab="Validation log-likelihood",log='x')
> abline(v=best.delta,lwd=3,col="red")
```




```
> # Refit the model with best delta
> ability.pseudo.gm.logit.best =
  glm(cbind(nfl23$y+best.delta, (1-nfl23$y)+best.delta)~
        X.star+AtHome+0,family = "binomial", data=nfl23)
> pseudo.gm.est=(W_h%*%ability.pseudo.gm.logit.best$coefficients)[,1]
> names(pseudo.gm.est) = c(Teams, 'HFA')
> Teams.ability$PseudoGame.Est =round(pseudo.gm.est,3)
> Teams.rank = Teams.ability[1:n_teams,]
> Teams.rank[order(-Teams.rank$PseudoGame.Est),]
                      BT.Est Ridge.Est PseudoGame.Est
Baltimore Ravens       1.699     0.621          0.911
San Francisco 49ers    1.292     0.481          0.685
Cleveland Browns       1.115     0.368          0.594
Pittsburgh Steelers    0.865     0.247          0.455
Detroit Lions          0.798     0.426          0.425
Dallas Cowboys         0.825     0.401          0.424
Los Angeles Rams       0.777     0.244          0.409
Cincinnati Bengals     0.697     0.153          0.375
Philadelphia Eagles    0.634     0.306          0.337
Kansas City Chiefs     0.483     0.289          0.276
Buffalo Bills          0.479     0.280          0.264
Seattle Seahawks       0.493     0.105          0.239
Miami Dolphins         0.377     0.248          0.204
Houston Texans         0.348     0.158          0.197
Jacksonville Jaguars   0.354     0.095          0.183
Indianapolis Colts     0.144     0.043          0.080
```

```
Tampa Bay Buccaneers  -0.067     0.039            -0.038
Green Bay Packers     -0.099     0.034            -0.049
New Orleans Saints    -0.266    -0.010            -0.141
Las Vegas Raiders     -0.307    -0.091            -0.171
Denver Broncos        -0.316    -0.091            -0.174
Minnesota Vikings     -0.490    -0.182            -0.270
Tennessee Titans      -0.515    -0.292            -0.292
New York Jets         -0.534    -0.206            -0.295
New York Giants       -0.661    -0.302            -0.364
Chicago Bears         -0.693    -0.229            -0.374
Arizona Cardinals     -0.870    -0.493            -0.453
Atlanta Falcons       -0.840    -0.274            -0.458
Los Angeles Chargers  -0.949    -0.426            -0.531
Washington Commanders -1.287    -0.559            -0.683
New England Patriots  -1.314    -0.562            -0.704
Carolina Panthers     -2.170    -0.824            -1.060

> Teams.ability['HFA',]
    BT.Est Ridge.Est PseudoGame.Est
HFA   0.26     0.236          0.138
```

Again, we see that the team rankings predicted by this model does not change too much, showing a consistency in the predicted relative strength. The magnitude of the strength estimation is close to the ridge-regularized model, but smaller than the un-regularized model. Unlike the ridge-regularized model, the estimation of HFA is also shrunk in this part.

(e) Finally, fit a regularized Bradley-Terry model with a phantom team known to have a strength parameter of 0, and with the weight of each game played by the phantom team optimized by 10-fold cross validation. Again, include a home-field advantage parameter in the model.

i. For the situation with the inclusion of a home-field advantage parameter, the data can be analyzed as ordinary binary outcomes as you would with an unregularized Bradley-Terry model. However, you will need to include 64 phantom games (32 teams $\times$ 2 games per team) between the phantom team and each actual team. Why should these games all be assumed to be played on neutral site?

Recall from lecture slides, the likelihood function of the model is

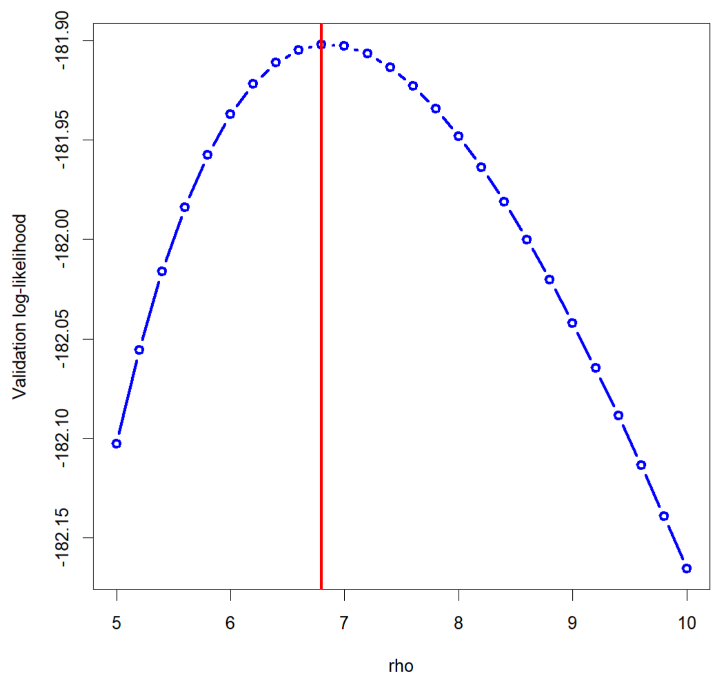
$$\log \mathcal{L}(\boldsymbol{\theta}|\boldsymbol{y}, \rho) = \log \mathcal{L}(\boldsymbol{\theta}|\boldsymbol{y}) + \rho \sum_{j=1}^{J} (\log F(\theta_j) + \log(1 - F(\theta_j)))$$

The second term, which effectively centers $\boldsymbol{\theta}$ at 0, directly comes from the additional games against phantom players. Note that this is simplified because we assumes the phantom player has strength 0 and the game is on neutral court. Otherwise we would have $F(\theta_j + \beta_h)$ or $F(\theta_j - \beta_h)$ instead. Again, we only want to regularize the teams strength estimate, $\boldsymbol{\theta}$, so no home field factor should be involved here.

ii. Fit the phantom team model and estimate the weights for the games against the phantom team. What is the optimized weight, and how do you interpret the value?

As before, summarize the estimated team strengths in order from best to worst, and compare the estimated strengths to the ordinary Bradley-Terry results, the ridge-regularized results, and the pseudo-game regularized results. Comment on the estimated home-field advantage parameter and how it compares to the estimates from the previous parts.

```
> # 3(e) Regularization with phantom player
> phantom.player.regularization = function(dev, rho){
+   X = outer(dev$Home, Teams, "==") - outer(dev$Away, Teams, "==")
+   X = rbind(X, diag(n_teams))
+   tmp = data.frame('AtHome' = c(dev[,'AtHome'], rep(0, n_teams)))
+   tmp$y = cbind(c(dev$y, rep(1, n_teams)),
+                 c(1-dev$y, rep(1, n_teams)))
+   tmp$X = X
+   w = c(rep(1, nrow(dev)), rep(rho, n_teams))
+   model = glm(y~X+AtHome+0, family="binomial", weights = w, data=tmp)
+   return (model)
+ }
> # Cross-validation
> K=10
> ls.rho = (25:50)/5
> pp.cv.log.likelihood = rep(0, length(ls.rho))
> for (i in 1:K){
+   print(paste("K =",i))
+   split = dev.val.split(i,nfl23)
+   dev = as.data.frame(split[1])
+   val = as.data.frame(split[2])
+   for (j in 1:length(ls.rho)){
+     model = phantom.player.regularization(dev, ls.rho[j])
+     val = val[,c('Home','Away','AtHome','y')]
+     val$X=outer(val$Home, Teams, "==")-outer(val$Away, Teams, "==")
+     p = predict(model, newdata=val, type="response")
+     ll = sum((val$y)*log(p) + (1-val$y)*log(1-p))
+     pp.cv.log.likelihood[j] = pp.cv.log.likelihood[j] + ll
+   }
+ }
> best.rho = ls.rho[which.max(pp.cv.log.likelihood)]
> best.rho
[1] 6.8
> plot(ls.rho, pp.cv.log.likelihood,type='b',col="blue",lwd=3,
+      xlab="rho",ylab="Validation log-likelihood")
> abline(v=best.rho,lwd=3,col="red")
```




```
> # Refit model with best rho
> ability.pp.best = phantom.player.regularization(nfl23, best.rho)
> Teams.ability$Phantom.Est =
+   round(ability.pp.best$coefficients,3)
> Teams.rank = Teams.ability[1:n_teams,]
> Teams.rank[order(-Teams.rank$Phantom.Est),]
                     BT.Est Ridge.Est PseudoGame.Est Phantom.Est
Baltimore Ravens      1.699     0.621          0.911       0.673
San Francisco 49ers   1.292     0.481          0.685       0.516
Detroit Lions         0.798     0.426          0.425       0.451
Dallas Cowboys        0.825     0.401          0.424       0.423
Cleveland Browns      1.115     0.368          0.594       0.397
Philadelphia Eagles   0.634     0.306          0.337       0.323
Kansas City Chiefs    0.483     0.289          0.276       0.304
Buffalo Bills         0.479     0.280          0.264       0.293
Pittsburgh Steelers   0.865     0.247          0.455       0.267
Los Angeles Rams      0.777     0.244          0.409       0.263
Miami Dolphins        0.377     0.248          0.204       0.258
Cincinnati Bengals    0.697     0.153          0.375       0.170
Houston Texans        0.348     0.158          0.197       0.165
Seattle Seahawks      0.493     0.105          0.239       0.114
Jacksonville Jaguars  0.354     0.095          0.183       0.103
Indianapolis Colts    0.144     0.043          0.080       0.044
Tampa Bay Buccaneers -0.067     0.039         -0.038       0.037
Green Bay Packers    -0.099     0.034         -0.049       0.032
New Orleans Saints   -0.266    -0.010         -0.141      -0.018
```

```
Las Vegas Raiders      -0.307    -0.091          -0.171       -0.098
Denver Broncos         -0.316    -0.091          -0.174       -0.099
Minnesota Vikings      -0.490    -0.182          -0.270       -0.194
New York Jets          -0.534    -0.206          -0.295       -0.220
Chicago Bears          -0.693    -0.229          -0.374       -0.248
Atlanta Falcons        -0.840    -0.274          -0.458       -0.299
Tennessee Titans       -0.515    -0.292          -0.292       -0.308
New York Giants        -0.661    -0.302          -0.364       -0.321
Los Angeles Chargers   -0.949    -0.426          -0.531       -0.453
Arizona Cardinals      -0.870    -0.493          -0.453       -0.522
Washington Commanders  -1.287    -0.559          -0.683       -0.600
New England Patriots   -1.314    -0.562          -0.704       -0.604
Carolina Panthers      -2.170    -0.824          -1.060       -0.903

> Teams.ability['HFA',]
    BT.Est Ridge.Est PseudoGame.Est Phantom.Est
HFA   0.26     0.236          0.138       0.236
```

The best ρ is estimated to be 6.8, which means that we imagine each team have played about $2 \times 6.8 \approx 14$ games against a "standard" phantom player, whose strength is 0, and won and lost exacly half of these games. Effectively, these games against phantom players pushes the strength estimate of each team toward 0.

The strength estimates from this phantom player model is very close to the previous two *regularized* models, and the rankings are consistent across all four models. The estimated home field factor from the phantom player model is close to the ridge-regularized and ordinary BT model, but larger than the pseudogame model.

4. In 1898, a double-round robin tournament was held in Vienna among the 20 best chess players at the time, including players such as Steinitz (the first official world champion), Tarrasch, Pillsbury, and others, who are familiar names to today's competitive chess players. One player, Adolf Schwarz, dropped out after 8 games, but these game results were retained for analysis. The data set consists of 350 games among the 20 players, with all but Schwarz playing 38 games each. No information about who played white versus black is in this data set.

   The file `vienna1898-played.csv` in the Data Sets folder on the course Canvas site contains the game results of the tournament. The following variables are included in this data set.

   `ID1`: ID (1 to 20) of Player 1

   `Player1`: Player 1 in the game

   `Result`: Result of game relative to Player 1 (1 for a win, 0.5 for a draw, 0 for a loss)

   `Player2`: Player 2 in the game

   `ID2`: ID (1 to 20) of Player 2

   The goal of this problem is to answer questions about the players' strengths.

   (a) What is the frequency of decisive games versus games ending in ties? Is it sensible to remove games ending in ties and analyzing the remaining games using a Bradley-Terry model? Why or why not?

Solution:

```
> library(cmdstanr)
> library("bpcs")
> vienna1898 = read.csv("vienna1898-played.csv")
> Players = unique(vienna1898$Player1)
> n_players = 20
> # 4a
> sum(vienna1898$Result!=0.5)/nrow(vienna1898)
[1] 0.6828571
> sum(vienna1898$Result==0.5)/nrow(vienna1898)
[1] 0.3171429
```

About 68.3% games are decisive, while 31.7% games end in ties. We **may not** simply discard all tied games, because there a a considerable amount of tied games in the data. Those games also provide information for players strength estimate. We would lose some information if simply discarding tied games.

(b) Some authors feel that it is okay to analyze paired comparison data through binary response models (e.g., Bradley-Terry) but treating a tie as equivalent to half a win and half a loss.

i. In this setting, if the (multiplicative) contribution to the likelihood function of a game in which player 1 wins is p, and in which player 2 wins is $1-p$, what is the contribution of a tie if it is treated as half a win and half a loss? *Hint: What would be the contribution to the likelihood of a win followed by a loss between the same two players? That contribution is worth two games. What modification should be made so that it is worth a total of one game?*

A natural thought is to assign the outcome variable

$$y_{ij} = 0.5$$

for tied games.
Then its contribution to the likelihood is

$$L(\boldsymbol{\theta}|y_{ij}) = p^{y_{ij}}(1-p)^{1-y_{ij}} = p^{1/2}(1-p)^{1/2} = \sqrt{p(1-p)}$$

ii. What is a reason that one might shy away from using binary response models for paired comparison data, treating ties as half a win and a loss?

One shortage of binary outcome model is that it can never predict a tie. For games in which a tie is possible, like chess in this question, we would like the model to predict some non-zero probability for a tie.

(c) Using the `bpc` function in the `bpcs` library, fit the Davidson extention of the Bradley-Terry model that includes a tie as a third outcome. For the `priors` argument, use

```
priors = list(prior_lambda_std=5,
              prior_nu_mu=0,
              prior_nu_std=1)
```

This will keep the impact of the prior distribution relatively uninformative. Determine the posterior mean and standard deviations of the strength parameters, as well as for the tie parameter, ν. Display the strength parameters in order from best to worst according to the estimated strengths.

```
> ability.bayesian = bpc(vienna1898, player0 = 'Player1',
+                         player1 = 'Player2',
+                         result_column = 'Result',
+                         model_type = 'davidson',
+                         solve_ties = 'none',
+                         priors = list(prior_lambda_std=5,
+                                       prior_nu_mu=0,
+                                       prior_nu_std=1))
> Ability.est = as.data.frame(
+   ability.bayesian$fit$summary()[2:22, c('mean', 'sd')])
> rownames(Ability.est) = c(Players, 'Tie Parameter')
> Teams.rank = Ability.est[1:n_players,]
> Teams.rank[order(-Teams.rank$mean),]
                  mean       sd
Pillsbury   2.80335527 1.280408
Tarrasch    2.53405819 1.295848
Janowski    2.30690702 1.281253
Steinitz    1.99399681 1.278062
Schlechter  1.12432808 1.256367
Burn        1.07374071 1.249174
Tschigorin  1.07366412 1.256404
Marzoczy    0.61248185 1.241578
Lipke       0.22323454 1.238714
Marco       0.11937474 1.256596
Blackburne -0.04680970 1.231365
Alapin     -0.05112413 1.242441
Schiffers  -0.19231292 1.251885
Showalter  -0.35625307 1.265607
Walbrodt   -0.64869985 1.257995
Halprin    -0.68042044 1.268661
Caro       -1.21244721 1.259833
Baird      -2.74058336 1.303264
Trenchard  -3.77044867 1.344732
Schwarz    -4.38402249 2.260085
```

(d) What are the estimated probabilities that Steinitz would defeat, draw, and lose to Pillsbury, based on the model fit? There are two ways to compute these probabilities.

i. First, insert the posterior means of the strength parameters of Steinitz and Pillsbury as well as the tie parameter ν into the Davidson probability formulas for a win, tie and a loss. Report these values.

Recall from lecture the Davidson probability formulas

$$\Pr(y_{ij} = 0) = \frac{\exp(\theta_j)}{\exp(\theta_i) + \exp(\theta_j) + \exp(\nu)\exp(\frac{\theta_i+\theta_j}{2})}$$
$$\Pr(y_{ij} = 0.5) = \frac{\exp(\nu)\exp(\frac{\theta_i+\theta_j}{2})}{\exp(\theta_i) + \exp(\theta_j) + \exp(\nu)\exp(\frac{\theta_i+\theta_j}{2})}$$
$$\Pr(y_{ij} = 1) = \frac{\exp(\theta_i)}{\exp(\theta_i) + \exp(\theta_j) + \exp(\nu)\exp(\frac{\theta_i+\theta_j}{2})}$$

Plug in the posterior mean of Steinitz and Pillsbury,

$$\theta_{\text{Steinitz}} = 1.97609231$$
$$\theta_{\text{Pillsbury}} = 2.78976283$$
$$\nu = 1.51012585$$

and obtain

$$\Pr(\text{Steinitz win}) = 0.099$$
$$\Pr(\text{Steinitz lose}) = 0.224$$
$$\Pr(\text{Tie}) = 0.676$$

Code:

```
> # 4d prediction
> theta.i = Ability.est['Pillsbury', 'mean']
> theta.j = Ability.est['Steinitz', 'mean']
> nu = Ability.est['Tie Parameter', 'mean']
> denom = exp(theta.i)+exp(theta.j)+exp(nu+(theta.i+theta.j)/2)
> Steinitz.win = exp(theta.j) / denom
> Steinitz.lose = exp(theta.i) / denom
> Steinitz.tie = exp(nu+(theta.i+theta.j)/2) / denom
> Steinitz.win
[1] 0.09943881
> Steinitz.lose
[1] 0.2243513
> Steinitz.tie
[1] 0.6762099
```

ii. Another way, which is more consistent with the Bayesian approach, is to compute the posterior expected values of the probabilities of a win, tie and a loss. As with the first problem on this homework, this approach averages over the uncertainty of the strength parameters rather than inserting point estimates into the probability formulas. You should use the `get_probabilities_df` function within the `bpcs` package to answer this question, as this function will perform the proper Bayesian computation numerically (via a Monte Carlo analysis). Unfortunately, however, there is a small bug in the function in which the results of the probability of a win and the probability of a loss are reversed – please make sure you note the corrected values in your solutions. The reversed values should be similar to your results in part (d)i above. How do these (corrected) probabilities differ from the ones by inserting the posterior mean point estimates?

```
> pred = get_probabilities_df(ability.bayesian, n=1000)
> pred[(pred$i=='Pillsbury')
+      & (pred$j=='Steinitz'),]
            i        j i_beats_j j_beats_i i_ties_j
150 Pillsbury Steinitz     0.094      0.25    0.656
```

After reversing the posterior expectation of winning probability for Steinitz and Pillsbury, we see these values are fairly close to the prediction by inserting posterior mean, though the posterior expectation of the probability of a tie is a little lower, and the probability of Steinitz's loss is a little higher than the other approach.