Statistics 143 — Spring 2024 — Assignment 5
Due Friday April 19, 2024

Homework is to be uploaded on Gradescope by 10:00pm on Friday 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

This problem set is concerned with analyzing regular-season NRL (National Rugby League of Australia) games played between 2009 to 2022. The file `nrl-games.csv` in the Data Sets folder on the course Canvas site contains information on 16 teams. The data set consist of the results of 2656 games. The following variables are included in this data set.

`Date`: The date on which the game was played (YYYY-MM-DD)

`Home.Team`: Home team

`Away.Team`: Away team

`Home.Score`: Score by home team

`Away.Score`: Score by away team

`outcome`: 1 if home team won, 0 if home team lost, and 0.5 for a tie

`season`: season in which game was played (ranging from 2009 to 2023)

You will need to load the `PlayerRatings` library in R to perform the following problems.

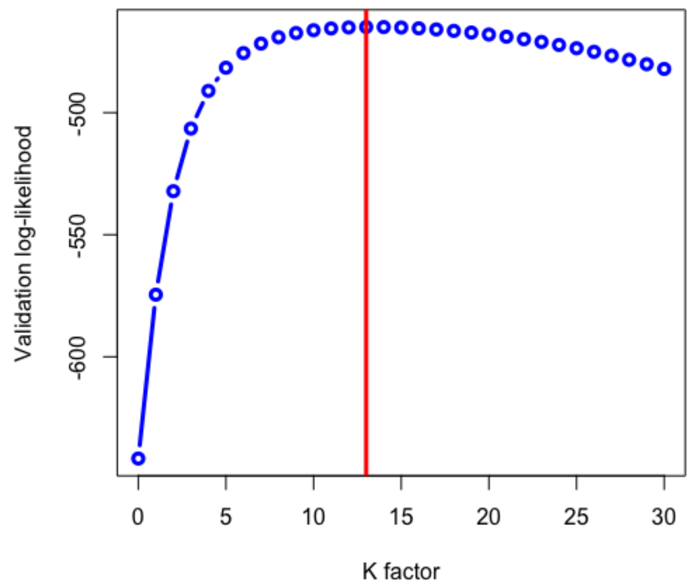
1. This problem involves developing an Elo rating system for NRL teams.

   (a) The first goal is to optimize the K-factor of the Elo system. Let the K-factor for the remaining seasons be a value over a choice set between 0 and 30. Use seasons 2009 through 2018 as training data to initialize the rating system using an initial rating of 1500 and an initial K-factor of 30 for the first season. Use one of the candidate choices as the K-factor for the remaining seasons. For seasons 2019 through 2022, alternately evaluate the predictive log-likelihood followed by updating the ratings, and then sum the season-specific log-likelihoods. What value of K is the best choice? Plot the relationship between the candidate K values and the validation log-likelihood, highlighting the optimal K.

   Solution:

```
> library("PlayerRatings")
> x = read.csv("nrl-games.csv")
>
> df.train = df.val =
+   data.frame(season = x$season, home=x$Home.Team,
```

```
+               visitor=x$Away.Team, outcome=x$outcome)
> df.train = df.train[df.train$season %in% 2009:2018,]
> df.val = df.val[df.val$season %in% 2019:2022,]
>
> # Elo model
> ls.k = 0:30
> log.likelihood.elo = rep(0, length(ls.k))
> hfa=0
> kfac.init = 30
>
> for (j in 1:length(ls.k)){
+     ratings.elo <- elo(df.train[df.train$season==2009,],
+                        init=1500, kfac = kfac.init,
+                        gamma=hfa, sort=FALSE)
+     ratings.elo <- elo(df.train[df.train$season!=2009,],
+                        kfac = ls.k[j],
+                        status=ratings.elo$ratings,
+                        gamma=hfa, sort=FALSE)
+     # Validation on val set
+     ll.tmp = 0
+     for (val.year in 2019:2022){
+       pred = predict(ratings.elo, df.val[df.val$season==val.year,],
+                      gamma = hfa)
+       # Log-likelihood
+       ll.tmp <- ll.tmp+
+           sum(df.val[df.val$season==val.year,]$outcome*log(pred)+
+           (1-df.val[df.val$season==val.year,]$outcome)*log(1-pred))
+       # Update model
+       ratings.elo <- elo(df.val[df.val$season==val.year,],
+                          kfac = ls.k[j],
+                          status = ratings.elo$ratings,
+                          gamma=hfa, sort=FALSE)
+       }
+     log.likelihood.elo[j] = ll.tmp
+ }
>
> best.k.ind = which(log.likelihood.elo == max(log.likelihood.elo))
> best.k = ls.k[best.k.ind]
> best.k
[1] 13
> plot(ls.k,log.likelihood.elo,type='b',col="blue",lwd=3,
+      xlab="K factor",ylab="Validation log-likelihood")
> abline(v=best.k,lwd=3,col="red")
```



The best K value is 13.

(b) Refit the Elo system using a rating of 1500 for all teams in 2009 and a K-factor of 30 (as in part (a)), and now use the optimized K-factor for seasons 2010 onward. Show the ratings of the teams, sorted from best to worst, in year 2022, along with the number of games, wins, draws and losses (these are part of the output of the fitted rating system output). Who are the top 3 teams in 2022?

Solution:

```
> ratings.elo <- elo(df.train[df.train$season==2009,],
+                    init=1500, kfac = kfac.init,
+                    gamma=hfa, history=TRUE, sort=FALSE)
> ratings.elo.best <- elo(rbind(df.train[df.train$season!=2009,], df.val),
+                         kfac = best.k,
+                         status=ratings.elo$ratings,
+                         gamma=hfa, history = TRUE, sort=FALSE)
> oo.elo <- order(ratings.elo.best$ratings$Rating,
+                 decreasing=T)
> ratings.elo.best$ratings[oo.elo,
+                 !names(ratings.elo.best$ratings) %in% c("Lag")]
                    Player   Rating Games Win Draw Loss
12        Penrith Panthers 1719.310   332 190    2  140
7          Melbourne Storm 1695.803   332 235    2   95
15         Sydney Roosters 1620.850   332 190    1  141
13   South Sydney Rabbitohs 1616.178   332 191    1  140
11          Parramatta Eels 1600.867   332 155    2  175
4          Cronulla Sharks 1565.022   332 162    2  168
2         Canberra Raiders 1533.328   332 160    1  171
10 North Queensland Cowboys 1485.421   332 161    1  170
```

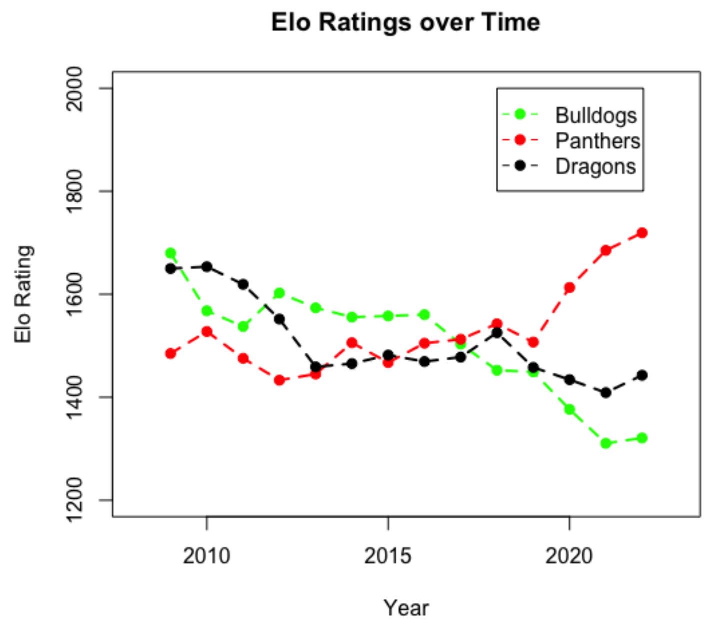
```
6          Manly Sea Eagles 1472.788   332 177    1  154
1          Brisbane Broncos 1458.294   332 174    2  156
14        St George Dragons 1442.735   332 161    1  170
9         Newcastle Knights 1398.081   332 129    4  199
8      New Zealand Warriors 1374.449   332 138    3  191
5         Gold Coast Titans 1361.119   332 131    1  200
16             Wests Tigers 1334.811   332 138    0  194
3       Canterbury Bulldogs 1320.944   332 152    0  180
```

The top 3 teams in 2022 are Penrith Panthers, Melbourne Storm and Sydney Roosters.

(c) Plot the Elo rating trajectories (on the same graph) for the Canterbury Bulldogs, the Penrinth Panthers, and the St George Dragons. Describe in a few sentences how each team has been performing over time.

Solution:

```
> ratings.elo.best.history =
+   cbind(ratings.elo$history[,,1], ratings.elo.best$history[,,1])
> dimnames(ratings.elo.best.history)[[2]] = 2009:2022
>
> plot(2009:2022, ratings.elo.best.history["Canterbury Bulldogs",],
+      type="n", ylab="Elo Rating", xlab="Year",
+      main="Elo Ratings over Time",ylim=c(1200,2000), xlim=c(2008,2023))
> points(2009:2022, ratings.elo.best.history["Canterbury Bulldogs",],
+        col="green", pch=19)
> lines(2009:2022, ratings.elo.best.history["Canterbury Bulldogs",],
+       col="green",lty=2,lwd=2)
> points(2009:2022, ratings.elo.best.history["Penrith Panthers",],
+        col="red", pch=19)
> lines(2009:2022, ratings.elo.best.history["Penrith Panthers",],
+       col="red",lty=2,lwd=2)
> points(2009:2022, ratings.elo.best.history["St George Dragons",],
+        col="black", pch=19)
> lines(2009:2022, ratings.elo.best.history["St George Dragons",],
+       col="black",lty=2,lwd=2)
> legend(2018, 2000, legend=c("Bulldogs","Panthers","Dragons"),
+        col=c("green","red","black"),
+        pch=19,lty=2, ncol=1)
```



The Canterbury Bulldogs began with the strongest performance among the three teams in 2009, yet it gradually declined over time and became the weakest after 2019. The Penrith Panthers maintained relatively stable performance from 2009 to 2019, but experienced a surge since 2019 and became the strongest. The St George Dragons' performance saw a rapid decline from 2010 to 2013, and stabilized thereafter with a slight decreasing trend.

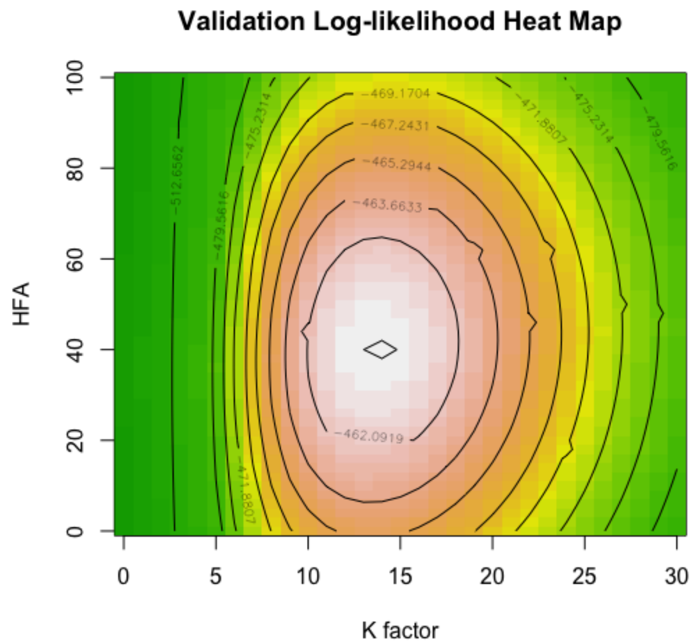
(d) The Elo system can be extended to include a home-field advantage, which adds a value `gamma` to the rating difference between the home and away ratings to account for the advantage of playing at home. The `elo` function has a argument "`gamma`" that sets the value of HFA. Using the same division in part (a) of seasons into training and validation periods, optimize both the K-factor and `gamma` by selecting a range of candidate K-factors (the ones you used in part (a)) and a candidate range of `gamma` values (you might try 0 to 100), and optimize the predictive log-likelihood over all combinations of K and `gamma`.

i. What are the best choices of K and `gamma`? Create a heat map that displays the validation log-likelihood as a function of K and `gamma` over the values in your candidate sets (the most straightforward way to do this is to use the `image` function in R).

Solution:

```
> ls.gamma = (0:50)*2
> log.likelihood.elo2 = matrix(0, length(ls.k), length(ls.gamma))
> kfac.init = 30
>
> for (j in 1:length(ls.k)){
+   for (i in 1:length(ls.gamma)){
+     ratings.elo <- elo(df.train[df.train$season==2009,],
+                        init=1500, kfac = kfac.init,
+                        gamma=ls.gamma[i], sort=FALSE)
```

```
+     ratings.elo <- elo(df.train[df.train$season!=2009,],
+                        kfac = ls.k[j],
+                        status=ratings.elo$ratings,
+                        gamma=ls.gamma[i], sort=FALSE)
+     # Validation on val set
+     ll.tmp = 0
+     for (val.year in 2019:2022){
+       pred = predict(ratings.elo, df.val[df.val$season==val.year,],
+                      gamma=ls.gamma[i])
+       # Log-likelihood
+       ll.tmp <- ll.tmp+
+         sum(df.val[df.val$season==val.year,]$outcome*log(pred)+
+         (1-df.val[df.val$season==val.year,]$outcome)*log(1-pred))
+       # Update model
+       ratings.elo <- elo(df.val[df.val$season==val.year,],
+                          kfac = ls.k[j],
+                          status = ratings.elo$ratings,
+                          gamma=ls.gamma[i], sort=FALSE)
+     }
+     log.likelihood.elo2[j,i] = ll.tmp
+   }
+ }
>
> best.ind = which(log.likelihood.elo2 == max(log.likelihood.elo2),
+                  arr.ind = T)
> best.k = ls.k[best.ind[1]]
> best.gamma = ls.gamma[best.ind[2]]
> best.k
[1] 14
> best.gamma
[1] 40
>
> image(ls.k, ls.gamma,
+       log.likelihood.elo2,
+       col=terrain.colors(50),
+       breaks=quantile(log.likelihood.elo2, (0:50)/50),
+       xlab="K factor",
+       ylab="HFA",
+       main="Validation Log-likelihood Heat Map")
> contour(ls.k, ls.gamma, log.likelihood.elo2, add=T,
+         levels=quantile(log.likelihood.elo2, (0:10)/10))
```



The best combination of K and `gamma` is 14 and 40.

ii. For the optimized value of `gamma`, what is the estimated probability that one team defeats another on its home field, assuming both teams have the same Elo rating?

<u>Solution:</u>

The probability that team a defeats team b with $r_{at} = r_{bt}$ on team a's home field can be calculated as:

$$P(y_{abt} = 1) = \frac{1}{1 + 10^{-(\frac{r_{at} - r_{bt} + \gamma}{400})}} = \frac{1}{1 + 10^{-\frac{\gamma}{400}}}$$

```
> prob = 1/(1+10^(-best.gamma/400))
> prob
[1] 0.5573116
```

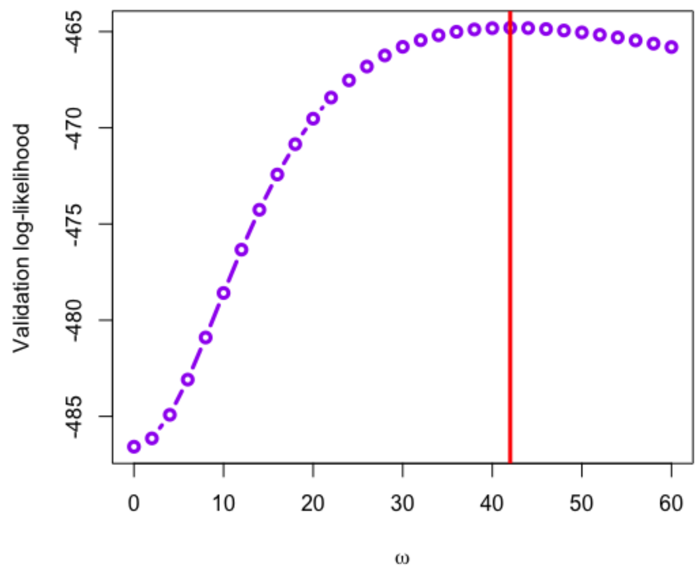
The estimated probability that one team defeats another team of equal rating on its home field is 0.56.

2. This next problem involves developing a Glicko rating system for NRL teams.

(a) You will first optimize the innovation standard deviation (ω) for this model. Consider values of ω between 0 and 60. As with the Elo model, use seasons 2009 through 2018 as training data to initialize the rating system using an initial rating of 1500 and an initial standard deviation (σ) of 350. For seasons 2019 through 2022, alternately evaluate the predictive log-likelihood followed by updating the ratings, and then sum the season-specific log-likelihoods. The probabilities of game outcomes should use the Glicko expected game outcome formula that accounts for the uncertainty in teams' ratings. Use the Glicko probability prediction function defined in the lecture notes R code, not the version implemented in the `PlayerRatings` library (which does not appear to be correctly implemented). What value of ω is the best choice? Plot the relationship between the candidate ω values and the validation log-likelihood, highlighting the optimal ω, as you did with the Elo model.

Solution:

```
> # Glicko model
> # The function computes the prob of team 1 win given Glicko ratings
> predict.fn.glicko <- function(t1, t2, ratings){
+   rd = ratings[t1, "stderr"]^2+ratings[t2, "stderr"]^2
+   g = 1/sqrt(1+0.00001007252*rd)
+   E = 1/(1+10^(-g * (ratings[t1,"est"]-ratings[t2,"est"])/400))
+   return (E)
+ }
>
> ls.omega =(0:30)*2
> log.likelihood.glicko = rep(0, length(ls.omega))
>
> for (j in 1:length(ls.omega)){
+     ratings.glicko = glicko(df.train, init = c(1500, 350),
+             cval=ls.omega[j],rdmax=15000, history = FALSE, sort=FALSE)
+     ll.tmp = 0
+     for (val.year in 2019:2022){
+         pred = predict.fn.glicko(df.val[df.val$season==val.year,]$home,
+                df.val[df.val$season==val.year,]$visitor,
+                data.frame("est"=ratings.glicko$ratings[,2],
+                     "stderr"=sqrt(ratings.glicko$ratings[,3]^2
+                                  + ls.omega[j]^2),
+                    row.names = ratings.glicko$ratings[,1]))
+       # Log-likelihood
+       ll.tmp <- ll.tmp+
+         sum(log(pred^df.val[df.val$season==val.year,]$outcome
+         *(1-pred)^(1-df.val[df.val$season==val.year,]$outcome)))
+       # Update the model
+       ratings.glicko <- glicko(df.val[df.val$season==val.year,],
+                                status = ratings.glicko$ratings,
+                                cval=ls.omega[j], gamma=hfa, sort=FALSE)
+       }
+     log.likelihood.glicko[j] = ll.tmp
+ }
> best.omega.ind = which(log.likelihood.glicko
+                         == max(log.likelihood.glicko))
> best.omega = ls.omega[best.omega.ind]
> print(best.omega)
[1] 42
> plot(ls.omega,log.likelihood.glicko,type='b',col="purple",lwd=3,
+      xlab=expression(omega),ylab="Validation log-likelihood")
> abline(v=best.omega,lwd=3,col="red")
```



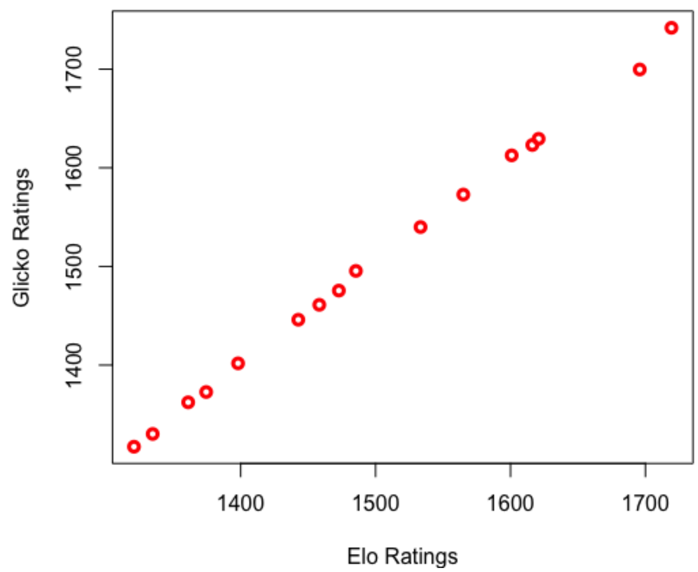
The optimal ω value is 42.

(b) Refit the Glicko system using a rating of 1500 and an initial standard deviation of 350 for all teams in 2009. Also use the optimized ω for running the Glicko system. Show the ratings of the teams, sorted from best to worst, in year 2022, along with the number of games, wins, draws and losses (again, these are part of the output of the fitted rating system output). Who are the top 3 teams in 2022? How do these teams compare to the ones you obtained in the Elo model? Plot the Glicko ratings (y-axis) against the Elo ratings (x-axis).

Solution:

```
> ratings.glicko.best <- glicko(rbind(df.train, df.val),
+                         init=c(1500, 350), cval=best.omega,
+                         rdmax=5000, history = TRUE, sort=FALSE)
> oo.glicko <- order(ratings.glicko.best$ratings$Rating, decreasing=T)
> ratings.glicko.best$ratings[oo.glicko,
+                        !names(ratings.glicko.best$ratings) %in% c("Lag")]
                     Player   Rating Deviation Games Win Draw Loss
12         Penrith Panthers 1741.925  52.32024   332 190    2  140
7           Melbourne Storm 1699.652  54.80781   332 235    2   95
15          Sydney Roosters 1629.388  51.20487   332 190    1  141
13    South Sydney Rabbitohs 1623.118  51.55021   332 191    1  140
11          Parramatta Eels 1612.608  50.33840   332 155    2  175
4          Cronulla Sharks 1572.866  49.69230   332 162    2  168
2         Canberra Raiders 1539.927  49.83728   332 160    1  171
10 North Queensland Cowboys 1495.481  50.46830   332 161    1  170
6          Manly Sea Eagles 1475.532  49.71778   332 177    1  154
1         Brisbane Broncos 1461.107  50.60821   332 174    2  156
14       St George Dragons 1445.998  49.94844   332 161    1  170
```

```
9         Newcastle Knights 1401.773  49.78066   332 129    4  199
8      New Zealand Warriors 1372.722  50.16755   332 138    3  191
5         Gold Coast Titans 1362.280  50.17143   332 131    1  200
16             Wests Tigers 1330.195  49.98680   332 138    0  194
3       Canterbury Bulldogs 1317.278  52.35714   332 152    0  180
>
> Teams.ratings = data.frame("Elo"=ratings.elo.best$ratings[,2],
+                  "Glicko.Est"=ratings.glicko.best$ratings[,2],
+                  "Glicko.Dev"=ratings.glicko.best$ratings[,3])
> Teams = ratings.elo.best$ratings[,1]
> row.names(Teams.ratings) <- Teams
> plot(Teams.ratings$Elo, Teams.ratings$Glicko.Est,
+      col="red", lwd=3,
+      xlab="Elo Ratings", ylab="Glicko Ratings")
```



The top 3 teams in 2022 are Penrith Panthers, Melbourne Storm and Sydney Roosters, which are the same as the ones obtained in the Elo model. The Glicko ratings and Elo ratings are pretty consistent, almost following a straight line.

(c) Calculate the probability that, in 2022, the Manly Sea Eagles would defeat the Cronulla Sharks, based on the Glicko ratings and standard deviations. How does this compare to the probability estimate from the Elo model?

Solution:

```
> prob.glicko = predict.fn.glicko("Manly Sea Eagles", "Cronulla Sharks",
+                      data.frame("est"=Teams.ratings[,2],
+                                 "stderr"=Teams.ratings[,3],
+                                 row.names = row.names(Teams.ratings)))
> prob.elo = 1/(1+10^(-(Teams.ratings["Manly Sea Eagles",1]-
+                       Teams.ratings["Cronulla Sharks",1])/400))
```

```
> prob.glicko
[1] 0.366594
> prob.elo
[1] 0.3702973
```

The probability estimates from the Glicko and the Elo models are quite close. The slight difference is due to the fact that Elo doesn't take the standard deviation into account.

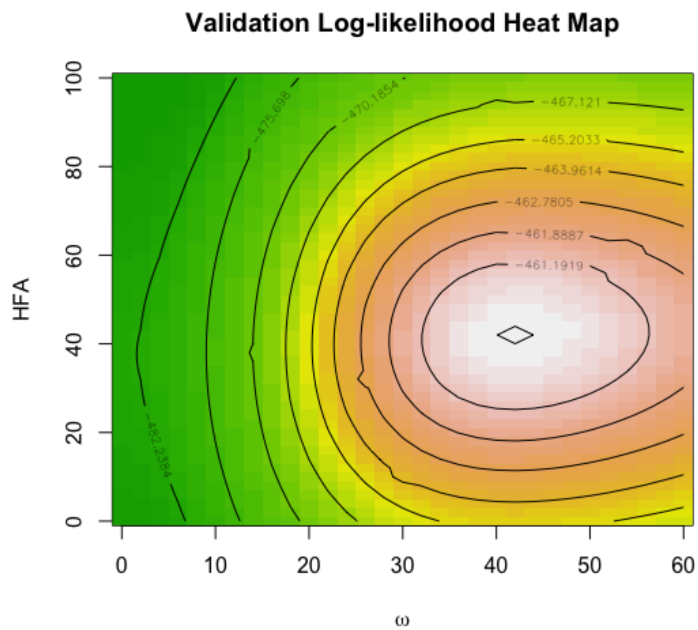
(d) As with the Elo system, the Glicko system can be extended to include a home-field advantage, which adds a value `gamma` to the rating difference between the home and away ratings to account for the advantage of playing at home. Also just like the `elo` function, the `glicko` function has a argument "`gamma`" that sets the value of HFA. Using the same division in part (a) of seasons into training and validation periods, optimize <u>both</u> ω and `gamma` by selecting a range of candidate ω (the ones you used in part (a)) and a candidate range of `gamma` values (try similar values to what you used for the Elo system), and optimize the predictive log-likelihood over all combinations of ω and `gamma`. To compute the predicted probability of a game outcome accounting for the home field advantage, add `gamma` to the rating of the home team in the expression for the probability. Assume that the home-field advantage is measured precisely enough that it does not impact the standard deviation of the rating difference between two teams.

   i. What are the optimized choices of ω and `gamma`? Create a heat map that displays the validation log-likelihood as a function of ω and `gamma` over the values in your candidate sets.

   <u>Solution:</u>

```
> # Calculates winning probability incorporating HFA
> predict.fn.glicko.hfa <- function(t1, t2, hfa, ratings){
+   rd = ratings[t1, "stderr"]^2+ratings[t2, "stderr"]^2
+   g = 1/sqrt(1+0.00001007252*rd)
+   E = 1/(1+10^(-g * (ratings[t1,"est"]-ratings[t2,"est"]+hfa)/400))
+   return (E)
+ }
>
> log.likelihood.glicko2 = matrix(0,length(ls.omega),length(ls.gamma))
>
> for (j in 1:length(ls.omega)){
+   for (i in 1:length(ls.gamma)) {
+     ratings.glicko = glicko(df.train, init = c(1500, 350),
+                        gamma = ls.gamma[i], cval=ls.omega[j],
+                        rdmax=15000, history = FALSE, sort=FALSE)
+     ll.tmp = 0
+     for (val.year in 2019:2022){
+       pred = predict.fn.glicko.hfa(
+              df.val[df.val$season==val.year,]$home,
+              df.val[df.val$season==val.year,]$visitor,
+              ls.gamma[i],
+              data.frame("est"=ratings.glicko$ratings[,2],
```

```
+              "stderr"=sqrt(ratings.glicko$ratings[,3]^2
+                           +ls.omega[j]^2),
+              row.names = ratings.glicko$ratings[,1]))
+       # Log-likelihood
+       ll.tmp <- ll.tmp+
+         sum(log(pred^df.val[df.val$season==val.year,]$outcome
+         *(1-pred)^(1-df.val[df.val$season==val.year,]$outcome)))
+       # Update the model
+       ratings.glicko <- glicko(df.val[df.val$season==val.year,],
+                        status = ratings.glicko$ratings,
+                        cval=ls.omega[j],
+                        gamma=ls.gamma[i], sort=FALSE)
+     }
+     log.likelihood.glicko2[j,i] = ll.tmp
+   }
+ }
>
> best.ind = which(log.likelihood.glicko2
>      == max(log.likelihood.glicko2), arr.ind = T)
> best.omega = ls.omega[best.ind[1]]
> best.gamma = ls.gamma[best.ind[2]]
> best.omega
[1] 42
> best.gamma
[1] 42
>
> image(ls.omega, ls.gamma,
+       log.likelihood.glicko2,
+       col=terrain.colors(50),
+       breaks=quantile(log.likelihood.glicko2, (0:50)/50),
+       xlab=expression(omega),
+       ylab="HFA",
+       main="Validation Log-likelihood Heat Map")
> contour(ls.omega, ls.gamma, log.likelihood.glicko2, add=T,
+         levels=quantile(log.likelihood.glicko2, (0:10)/10))
```



The best combination of ω and `gamma` is 42 and 42.

ii. Suppose a friend, who is noticing the parallel structure between problem 1 and problem 2 on this homework, expects the instructor to pose the following problem:

> For the optimized value of `gamma`, what is the estimated probability that one team defeats another on its home field, assuming both teams have the same Glicko rating?

Why is this question not answerable as asked?

Solution:

Because to estimate the probability that one team defeats another using the Glicko system, we also need information about the standard deviations. Only knowing they have equal ratings is not sufficient!

3. At the end of an NRL season, the top 8 teams (according to their total number of wins plus half the number of ties) compete in a "finals series", which is basically a post-season playoff tournament. Denote these eight teams as 1 (highest number of wins) to 8 (lowest number of wins). The structure of the tournament is as follows (see https://en.wikipedia.org/wiki/NRL_finals_system for more details):

- In round 1:
  - team 1 hosts team 4.
  - team 5 hosts team 8. The loser is eliminated.
  - team 6 hosts team 7. The loser is eliminated.
  - team 2 hosts team 3.
- In round 2:
  - the loser of 1 vs 4 hosts the winner of 5 vs 8.
  - the loser of 2 vs 3 hosts the winner of 6 vs 7.
- In round 3:

  - the winner of 1 vs 4 hosts the winner of the 2nd round 2 matchup (so one of teams 2, 3, 6 or 7)
  - the winner of 2 vs 3 hosts the winner of the 1st round 2 matchup (so one of teams 1, 4, 5 or 8)
- In round 4 (Grand final):
  - Winners of the round 3 play on a neutral site.

The winner of round 4 is the NRL champion.

(a) The top 8 teams at the end of the 2022 season were

Place	Team	Wins
1	Penrith Panthers	20
2	Cronulla Sharks	18
3	North Queensland Cowboys	17
4	Parramatta Eels	16
5	Melbourne Storm	15
6	Sydney Roosters	15
7	South Sydney Rabbitohs	14
8	Canberra Raiders	14

The wins column lists the number of wins out of 24 in the regular season.
Create a table that lists out the teams, the number of wins (out of 24 games), the Glicko rating, and the Glicko standard deviation (the last two from question 2(d)). Are all of the Glicko ratings in the same order as the number of wins in the 2022 season for these 8 teams?

Solution:

```
> # Refit Glicko with best omega and gamma from 2(d)
> ratings.glicko.best <- glicko(rbind(df.train, df.val),
+                               init=c(1500, 350), cval=best.omega,
+                               gamma=best.gamma, rdmax=5000,
+                               history = TRUE, sort=FALSE)
> Teams.ratings = data.frame("Glicko.Est"=ratings.glicko.best$ratings[,2],
+                            "Glicko.Dev"=ratings.glicko.best$ratings[,3])
> Teams = ratings.elo.best$ratings[,1]
> row.names(Teams.ratings) <- Teams
>
> Teams.name = c("Penrith Panthers", "Cronulla Sharks",
+                "North Queensland Cowboys", "Parramatta Eels",
+                "Melbourne Storm", "Sydney Roosters",
+                "South Sydney Rabbitohs", "Canberra Raiders")
> Teams.wins = c(20, 18, 17, 16, 15, 15, 14, 14)
> df = data.frame(name=Teams.name, wins=Teams.wins,
+           Glicko.Est=Teams.ratings[Teams.name,]$Glicko.Est,
+           Glicko.Dev=Teams.ratings[Teams.name,]$Glicko.Dev)
> df
```

```
                          name wins Glicko.Est Glicko.Dev
1          Penrith Panthers   20   1743.133   52.44229
2           Cronulla Sharks   18   1572.906   49.78332
3 North Queensland Cowboys   17   1494.940   50.70183
4           Parramatta Eels   16   1613.727   50.55030
5           Melbourne Storm   15   1702.521   54.99644
6           Sydney Roosters   15   1630.663   51.40341
7   South Sydney Rabbitohs   14   1624.640   51.75104
8          Canberra Raiders   14   1540.117   49.99397
```

No, the Glicko ratings and the number of wins are not exactly in the same order. For example, the Cronulla Sharks and the North Queensland Cowboys have lower Glicko ratings but higher number of wins among the 8 teams.

(b) Labeling the top 8 teams 1 through 8, create an 8×8 matrix in R called `P` where element `P[i,j]` is the probability team `i` defeats team `j` at team `i`'s home field, based on the Glicko ratings and standard deviations from question 2(d). Also create an 8×8 matrix in R called `P0` where element `P0[i,j]` is the probability team `i` defeats team `j` at a neutral field, based on the Glicko ratings and standard deviations from question 2(d), acknowledging that the HFA is 0 when playing on a neutral site. Display both matrices.

Solution:

```
> P = P0 = matrix(NA, 8, 8)
> for (i in 1:8) {
+   for (j in 1:8) {
+     P[i,j] = predict.fn.glicko.hfa(
+       df[i,]$name, df[j,]$name,
+       best.gamma,
+       data.frame("est"=df$Glicko.Est,
+                  "stderr"=df$Glicko.Dev,
+                  row.names = df$name))
+
+     P0[i,j] = predict.fn.glicko.hfa(
+       df[i,]$name, df[j,]$name,
+       0, # set HFA param to 0
+       data.frame("est"=df$Glicko.Est,
+                  "stderr"=df$Glicko.Dev,
+                  row.names = df$name))
+   }
+ }
> round(P,4)
       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]
[1,] 0.5586 0.7669 0.8358 0.7234 0.6136 0.7039 0.7109 0.7981
[2,] 0.3275 0.5587 0.6623 0.5017 0.3797 0.4779 0.4863 0.6035
[3,] 0.2393 0.4497 0.5587 0.3939 0.2835 0.3715 0.3794 0.4955
[4,] 0.3798 0.6142 0.7115 0.5587 0.4348 0.5351 0.5435 0.6568
[5,] 0.5019 0.7234 0.8018 0.6754 0.5584 0.6542 0.6617 0.7586
[6,] 0.4025 0.6364 0.7305 0.5819 0.4583 0.5586 0.5669 0.6779
```

```
[7,] 0.3944 0.6286 0.7238 0.5737 0.4500 0.5503 0.5586 0.6704
[8,] 0.2884 0.5129 0.6200 0.4557 0.3375 0.4323 0.4406 0.5587
> round(P0,4)
       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]
[1,] 0.5000 0.7221 0.8009 0.6739 0.5566 0.6526 0.6602 0.7575
[2,] 0.2779 0.5000 0.6077 0.4429 0.3260 0.4196 0.4279 0.5459
[3,] 0.1991 0.3923 0.5000 0.3392 0.2382 0.3183 0.3257 0.4369
[4,] 0.3261 0.5571 0.6608 0.5000 0.3782 0.4763 0.4847 0.6019
[5,] 0.4434 0.6740 0.7618 0.6218 0.5000 0.5992 0.6073 0.7130
[6,] 0.3474 0.5804 0.6817 0.5237 0.4008 0.5000 0.5084 0.6244
[7,] 0.3398 0.5721 0.6743 0.5153 0.3927 0.4916 0.5000 0.6164
[8,] 0.2425 0.4541 0.5631 0.3981 0.2870 0.3756 0.3836 0.5000
```

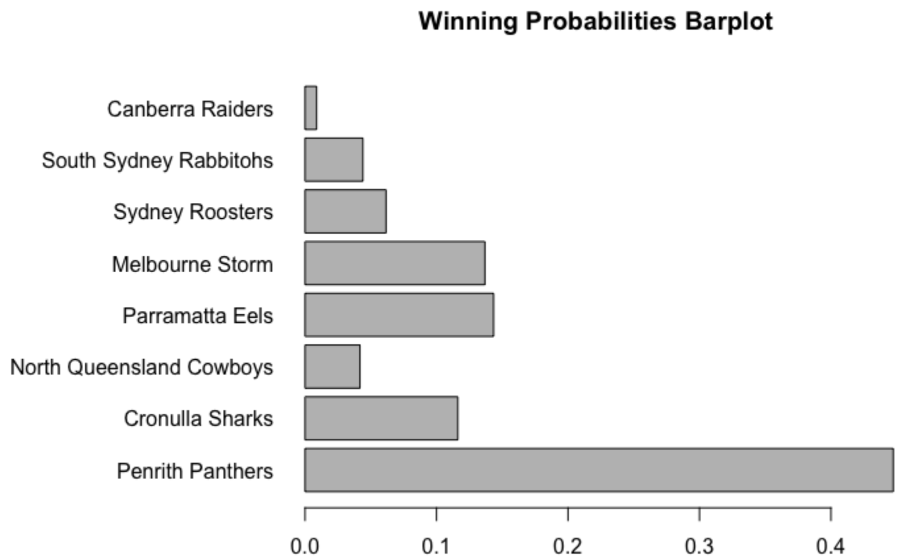
(c) Using the matrices `P` and `P0` you computed in part (b), simulate 2022 NRL finals series via Monte Carlo simulation, following the tournament design at the start of the problem. For the finals series, game outcomes are decisive; there are no ties. You will likely need to use the `rbinom` function to simulate game outcomes. Carry out the tournament simulation 5000 times. For each simulated tournament, save the team who wins the championship.

Based on simulation results, what is the estimated probability each team would win the championship? Create a barplot of the probabilities for each team. The teams that played in the Grand final were the Panthers and the Eels, with the Panthers winning. What were the probabilities of these two teams winning the championship?

Solution:

```
> N = 5000
> winners = numeric(N)
> set.seed(143)
> for (i in 1:N) {
+   # round 1
+   res1_1 = rbinom(1, 1, P[1,4])
+   winner1_1 = ifelse(res1_1==1, 1, 4)
+   loser1_1 = ifelse(res1_1==1, 4, 1)
+
+   res1_2 = rbinom(1, 1, P[5,8])
+   winner1_2 = ifelse(res1_2==1, 5, 8)
+
+   res1_3 = rbinom(1, 1, P[6,7])
+   winner1_3 = ifelse(res1_3==1, 6, 7)
+
+   res1_4 = rbinom(1, 1, P[2,3])
+   winner1_4 = ifelse(res1_4==1, 2, 3)
+   loser1_4 = ifelse(res1_4==1, 3, 2)
+
+   # round 2
+   res2_1 = rbinom(1, 1, P[loser1_1, winner1_2])
+   winner2_1 = ifelse(res2_1==1, loser1_1, winner1_2)
```

```
+
+   res2_2 = rbinom(1, 1, P[loser1_4, winner1_3])
+   winner2_2 = ifelse(res2_2==1, loser1_4, winner1_3)
+
+   # round 3
+   res3_1 = rbinom(1, 1, P[winner1_1, winner2_2])
+   winner3_1 = ifelse(res3_1==1, winner1_1, winner2_2)
+
+   res3_2 = rbinom(1, 1, P[winner1_4, winner2_1])
+   winner3_2 = ifelse(res3_2==1, winner1_4, winner2_1)
+
+   # round 4
+   res4 = rbinom(1, 1, P0[winner3_1, winner3_2])
+   winners[i] = ifelse(res4==1, winner3_1, winner3_2)
+ }
>
> probs = table(winners)/N
> rownames(probs) = df$name
> probs
winners
        Penrith Panthers          Cronulla Sharks
                  0.4474                   0.1162
North Queensland Cowboys          Parramatta Eels
                  0.0418                   0.1434
         Melbourne Storm          Sydney Roosters
                  0.1368                   0.0616
  South Sydney Rabbitohs         Canberra Raiders
                  0.0440                   0.0088
>
> par(mar=c(4,12,4,4)) # Increase margin size
> barplot(probs,
+         main="Winning Probabilities Barplot",
+         horiz=T, las=1)
```



The probabilities of the Panthers and the Eels winning the championship are 0.4474 and 0.1434, respectively.

Notice that the Sharks and the Cowboys, ranked as the top 2 and 3 teams, have relatively low estimated probabilities of winning the championship, consistent with their low Glicko ratings observed in (a).