S&DS 542 HW 11

Joanna Chen

1. Fitting Bradley-Terry.

The file NBA record.csv contains the results of all 1230 NBA games from the 2015–2016 regular season. The 30 teams are numbered from 1 to 30, according to the file teams.txt. Each row of NBA record.csv indicates the home team, away team, and outcome Y for one game, where Y = 1 if the home team won and Y = 0 otherwise.

(a) Fit the Bradley-Terry model with an intercept term α for the home-court advantage to this data, using maximum likelihood.

```
library(readr)
setwd("~/Desktop/HW 11")
table <- read_csv("NBA_record.csv")
```

```
## Parsed with column specification:
## cols(
##   Home = col_double(),
##   Away = col_double(),
##   Y = col_double()
## )
```

```
teams <- read_csv("teams.txt",col_names = F)
```

```
## Parsed with column specification:
## cols(
##   X1 = col_character()
## )
```

Let us first define a function *loglik*. Notes (23.2) shows that

$$\begin{aligned} l(\alpha, \beta_2, \ldots, \beta_k) &= \sum_{m=1}^{n} \left[Y_m \log\left(\frac{p_{i_m j_m}}{1 - p_{i_m j_m}} \right) + \log\left(1 - p_{i_m j_m}\right) \right] \\ &= \sum_{m=1}^{n} \left[Y_m \left(\alpha + \beta_{i_m} - \beta_{j_m}\right) - \log\left(1 + e^{\alpha + \beta_{i_m} - \beta_{j_m}}\right) \right]. \end{aligned}$$

So we can define as follows.

```
loglik = function(theta, Home, Away, Y) {
            alpha = theta[1]
            beta = c(0, theta[-1])
            return(sum(Y * (alpha + beta[Home] - beta[Away]) - log(1 + exp(alpha + beta[Home] - beta[Aw
          }
```

That returns the log-likelihood for the Bradley-Terry model given inputs

$\theta = (\alpha, \beta_2, ..., \beta_k)$ (constraining $\beta_1 = 0$), Home $= (i_1, \ldots, i_n)$, Away $= (j_1, \ldots, j_n)$, and $\mathrm{Y} = (Y_1, \ldots, Y_n)$

Then, we maximize the log-likelihood by

```
theta0 = rep(0, 30)  # initialization for theta (for example the all 0's vector)
result = optim(theta0,loglik,Home=table$Home,Away=table$Away,Y=table$Y,method='BFGS',control=list('fnsc
```

What are the 5 teams (in ranked order) with the highest Bradley-Terry scores?

```
coefs = order(c(0, result$par[2:30]), decreasing = T)
teams[coefs[1:5],]
```

```
## # A tibble: 5 x 1
##   X1
##   <chr>
## 1 10 Golden State Warriors
## 2 27 San Antonio Spurs
## 3 6 Cleveland Cavaliers
## 4 28 Toronto Raptors
## 5 21 Oklahoma City Thunder
```

What is the estimated increase in the log-odds of winning for playing at home versus away?

```
result$par[1]
```

```
## [1] 0.4626864
```

Playing at home has 46% log-odds of winning than playing away from home.

(b) Fit the Bradley-Terry model without the intercept term α. (You may define a new function loglik null = function(theta,Home,Away,Y) where now $\theta = (\alpha, \beta_2, ..., \beta_k)$, and use optim as before.)

```
# without intercept
loglik_b = function(theta, Home, Away, Y) {
            beta = c(0, theta)
            return(sum(Y * (beta[Home] - beta[Away]) - log(1 + exp(beta[Home] - beta[Away]))))
          }
theta0 = rep(0, 29)
result2 = optim(theta0, loglik_b,Home=table$Home,Away=table$Away,Y=table$Y,method='BFGS',control=list('
```

By comparing the log-likelihood value to that of part (a), compute the generalized likelihood ratio test statistic for testing the null hypothesis of no home-court advantage, $H_0 : \alpha = 0$. Report a p-value for this test, based on the theoretical χ^2_1 null distribution.

```
result$value # log-likelihood value of (a)
```

```
## [1] -680.2417
```

```
result2$value # log-likelihood value of (b)
```

```
## [1] -705.08
```

```
GLRT = 2 * (result$value - result2$value)
p_value = 1 - pchisq(GLRT, df=1)
print(c(paste("GLRT statistic = ",GLRT), paste("p-value = ", p_value)))
```

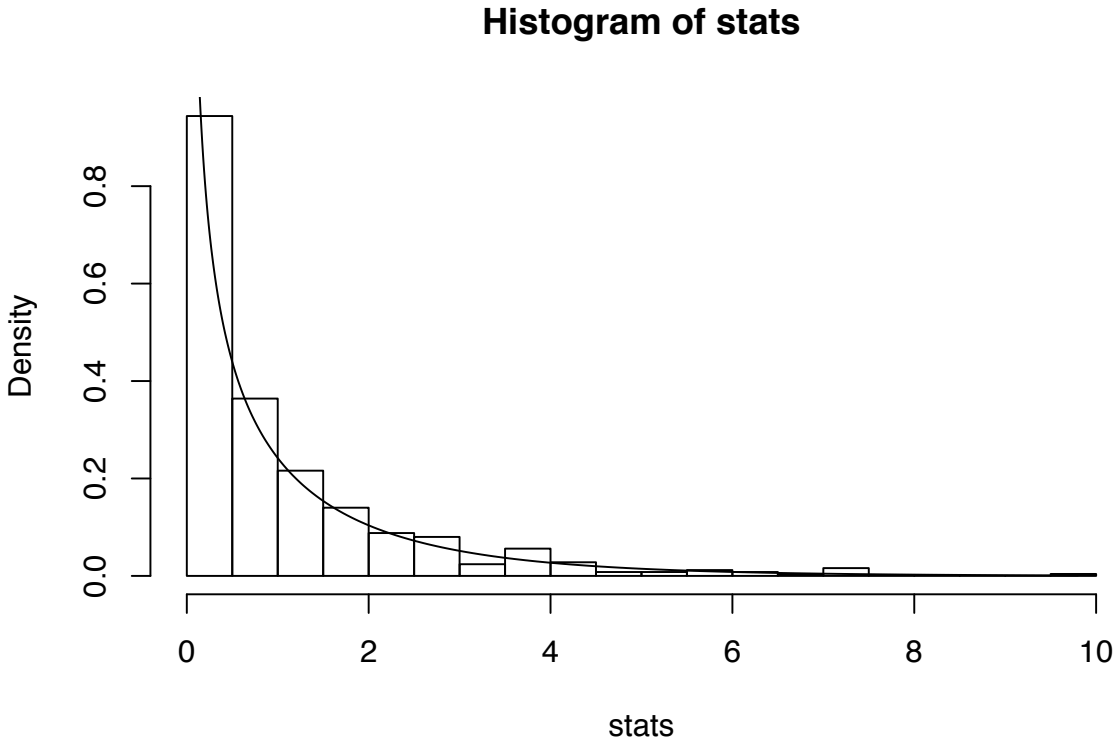
```
## [1] "GLRT statistic =  49.676582332994" "p-value =  1.81299419921288e-12"
```

Since our test statistic $49.68 > 3.84$ with p = 1.8e-12 < 0.05, we reject the null hypothesis and conclude that there exists home court advantage.

(c) Obtain a permutation null distribution for your test statistic in part (b) as follows: Independently for each game $m = 1, ..., n$, randomly replace (i_m, j_m, Y_m) by $(j_m, i_m, 1Y_m)$ with probability 1/2. Recompute your GLRT statistic on this permuted data, and repeat this B = 500 times. (This may take a few minutes to compute.)

Plot a histogram of your 500 test statistics, and overlay the theoretical χ^2_1 PDF on your histogram. Does your permutation null distribution seem to match the χ^2_1 distribution for this data? (You may refer to Problem 1(b) of Homework 9 for how to make this plot. The χ^2_1 PDF may be computed as dchisq(x.grid,df=1).)

```
B = 500
stats = numeric(B)
#do bootstrap
for (b in 1:B){
  n = 1230
  k = 30
  # print(b)
  Home = table$Home
  Away = table$Away
  Y = table$Y
  # do swapping
  swap = which(rbinom(n,1,0.5) == 1)
  # flip Y's
  Y[swap] = 1-Y[swap]
  # flip home and away
  temp = Home[swap]
  Home[swap] = Away[swap]
  Away[swap] = temp

  # solve for theta under full model
  theta0 = rep(0,k)
  result = optim(theta0, loglik,Home=Home,Away=Away,Y=Y,method='BFGS',control=list('fnscale'=-1))

  #solve for theta under submodel
  theta0 = rep(0,k-1)
  result_null = optim(theta0, loglik_b,Home=Home,Away=Away,Y=Y,method='BFGS',control=list('fnscale'=-1))

  #compute statistics
  stats[b] = -2 *(result_null$value - result$value)
}

x.grid = seq(0,max(stats),by = 0.01)
f.grid = dchisq(x.grid,df=1)
hist(stats,breaks = 20,freq = FALSE)
lines(x.grid,f.grid)
```



The permutation null distribution seem to match the χ^2_1 distribution for this data.

2. Heteroskedastic linear model. Consider independent observations

Consider independent observations

$$Y_i \sim \mathcal{N}\left(\beta x_i, \sigma_i^2\right)$$

for $i = 1, ..., n$, where $x_1, ..., x_n$ are deterministic and known values of a covariate.

(a)

Suppose that the error variances $\sigma_1^2, \ldots, \sigma_n^2$ are also known. Show that the MLE $\hat{\beta}$ for β minimizes the weighted least-squares criterion

$$\sum_{i=1}^{n} \frac{1}{\sigma_i^2} \left(Y_i - \beta x_i\right)^2 .$$

The probability density function of Y_i is

$$f\left(Y_i; \beta\right) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left[-\frac{\left(Y_i - \beta x_i\right)^2}{2\sigma_i^2}\right].$$

The likelihood function is

$$L(\beta) = \Pi_{i=1}^{n} \sigma_i^{-1} (2\pi)^{-n/2} \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(Y_i - \beta x_i\right)^2\right],$$

then the log-likelihood function is

$$\ell(\beta) = -\sum_{i=1}^{n} \log\sigma_i - \frac{n}{2} \log 2\pi - \sum_{i=1}^{n} \frac{1}{2\sigma_i^2} \left(Y_i - \beta x_i\right)^2$$

The MLE minimizes the $\ell(\beta)$. Since the n and σ are known in first two terms, in other words, the first two terms are constants, it minimizes the last term with a constant weight, for example,

$$\tilde{l} = \sum_{i=1}^{n} \frac{1}{\sigma_i^2} \left(Y_i - \beta x_i\right)^2 .$$

Derive a simple closed-form expression for $\hat{\beta}$

Take the derivative w.r.t β and set it to zero, we have

$$0 = \frac{\partial \tilde{l}}{\partial \beta} = -\sum_{i=1}^{n} \frac{1}{\sigma_i^2} 2\left(Y_i - \beta x_i\right) \cdot x_i = \sum_{i=1}^{n} \frac{1}{\sigma_i^2} 2\beta x_i^2 - \sum_{i=1}^{n} \frac{1}{\sigma_i^2} 2 Y_i x_i.$$

Therefore,

$$\hat{\beta} = \frac{\sum_{i=1}^{n} Y_i x_i / \sigma_i^2}{\sum_{i=1}^{n} x_i^2 / \sigma_i^2}.$$

Show that it is also unbiased for β.

$$E(\hat{\beta}) = E(\frac{\sum_{i=1}^{n} Y_i x_i / \sigma_i^2}{\sum_{i=1}^{n} x_i^2 / \sigma_i^2}) = \frac{\sum_{i=1}^{n} E(Y_i) x_i / \sigma_i^2}{\sum_{i=1}^{n} x_i^2 / \sigma_i^2} = \frac{\sum_{i=1}^{n} \beta x_i x_i / \sigma_i^2}{\sum_{i=1}^{n} x_i^2 / \sigma_i^2} = \beta.$$

Therefore, β is unbiased.

(b)

Suppose now that $\sigma_1^2, \ldots, \sigma_n^2$ are unknown. In this situation, we may wish to use the usual least-squares estimator $\hat{\beta}$, which minimize

$$\sum_{i=1}^{n} \left(Y_i - \beta x_i\right)^2$$

Derive a simple closed-form expression for $\tilde{\beta}$, and show that it is also unbiased for β.

Take the derivative w.r.t. β and set it to 0, we have

$$\sum_{i=1}^{n} 2\left(Y_i - \beta x_i\right) \cdot x_i = \sum_{i=1}^{n} 2 Y_i x_i - \sum_{i=1}^{n} 2\beta x_i^2 = 0.$$

Therefore,

$$\tilde{\beta} = \frac{\sum_{i=1}^{n} Y_i x_i}{\sum_{i=1}^{n} x_i^2}.$$

To show it's unbias,

$$E(\tilde{\beta}) = \frac{\sum_{i=1}^{n} E\left(Y_i\right) x_i}{\sum_{i=1}^{n} x_i^2} = \frac{\sum_{i=1}^{n} \beta x_i^2}{\sum_{i=1}^{n} x_i^2} = \beta$$

(c)

Derive a formula for the variance of $\tilde{\beta}$ in terms of $\sigma_1^2, \ldots, \sigma_n^2$ and $x1, ..., xn$. Estimating each σ_i^2 by the squared residual $\left(Y_i - \tilde{\beta} x_i\right)^2$, suggest a plug-in estimate for the standard error of $\tilde{\beta}$. This estimate is robust to possible differences in the variances $\sigma_1^2, \ldots, \sigma_n^2$.
The variance of $\tilde{\beta}$

$$Var(\tilde{\beta}) = \frac{\sum_{i=1}^{n} Var(Y_i) x_i^2}{\left(\sum_{i=1}^{n} x_i^2\right)^2} = \frac{\sum_{i=1}^{n} \sigma_i^2 x_i^2}{\left(\sum_{i=1}^{n} x_i^2\right)^2}$$

Estimating each σ_i^2 by the squared residual $\left(Y_i - \tilde{\beta}x_i\right)^2$, we have

$$\hat{\text{se}}(\tilde{\beta}) = \sqrt{\frac{\sum_{i=1}^n \left(Y_i - \tilde{\beta}x_i\right)^2 x_i^2}{\left(\sum_{i=1}^n x_i^2\right)^2}}$$

(d)

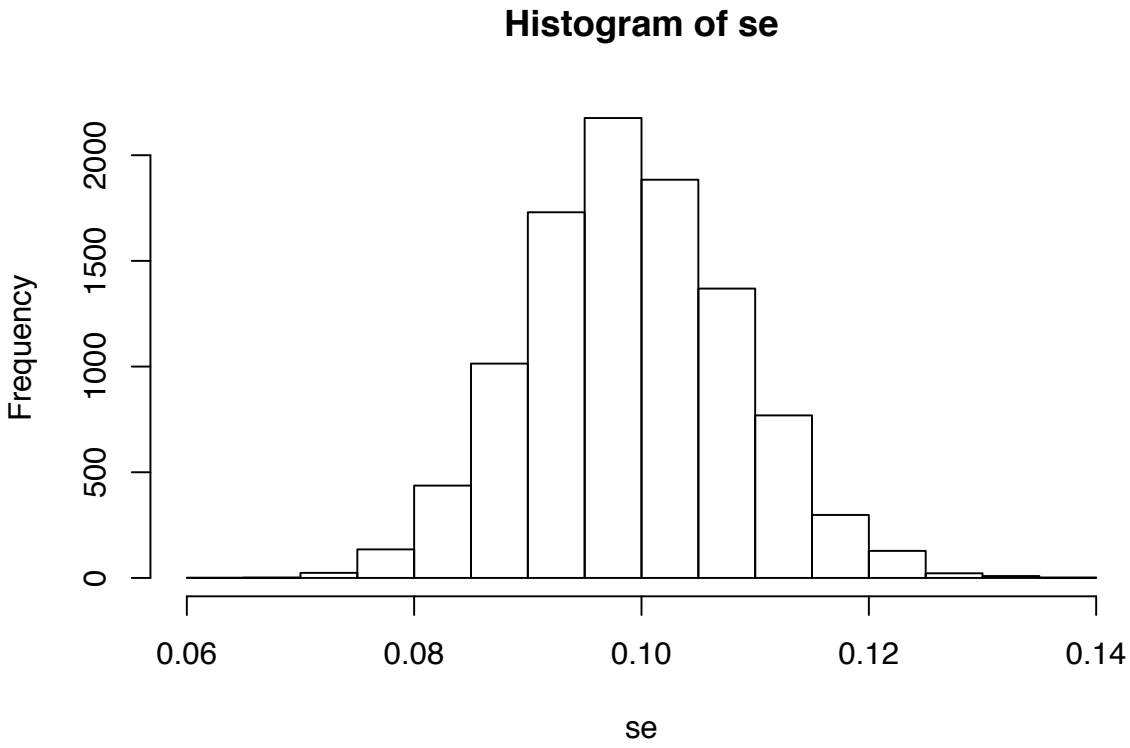
Perform a simulation that compares your standard error estimate in (c) to the usual standard error estimate in linear regression software, as follows: Let $\beta = 1, (x_1, x_2, \ldots, x_{100}) = (0.01, 0.02, \ldots, 1)$, and $\left(\sigma_1^2, \sigma_2^2, \ldots, \sigma_{100}^2\right) = \left(0.01^2, 0.02^2, \ldots, 1^2\right)$. In each of $B = 10000$ simulations, generate $Y_1, \ldots, Y_{100}$, and fit the linear regression model $Y = \beta x +$ error using any standard software to obtain the least-squares estimate $\tilde{\beta}$ and an estimated standard error for $\tilde{\beta}$ Compute also the estimated standard error using your method in part (c). Report the true (empirical) standard deviation of $\tilde{\beta}$ across the B simulations, and plot two histograms of the estimated standard errors using the two different methods. Summarize briefly your findings.

```
set.seed(123)
x = (1:100)/100
sigma = (1:100)/100
B = 10000

estimate = numeric(B) # least-squares estimate beta tilde
se = numeric(B) # estimated SE for beta tilde
se2 = numeric(B) # estimated SE using the method in part (c)

se2_fun <- function(x, Y, estimate) {
     sqrt(sum(x^2 * (Y - estimate * x)^2)) / sum(x^2)
}

for(i in 1:B){
  Y = x + rnorm(n = 100, mean = 0, sd = sigma)
  model = lm(Y ~ x + 0)
  estimate[i] = summary(model)[["coefficients"]][["x","Estimate"]]
  se[i] = summary(model)[["coefficients"]][["x","Std. Error"]]
  se2[i] = se2_fun(x, Y, estimate[i])
}
```

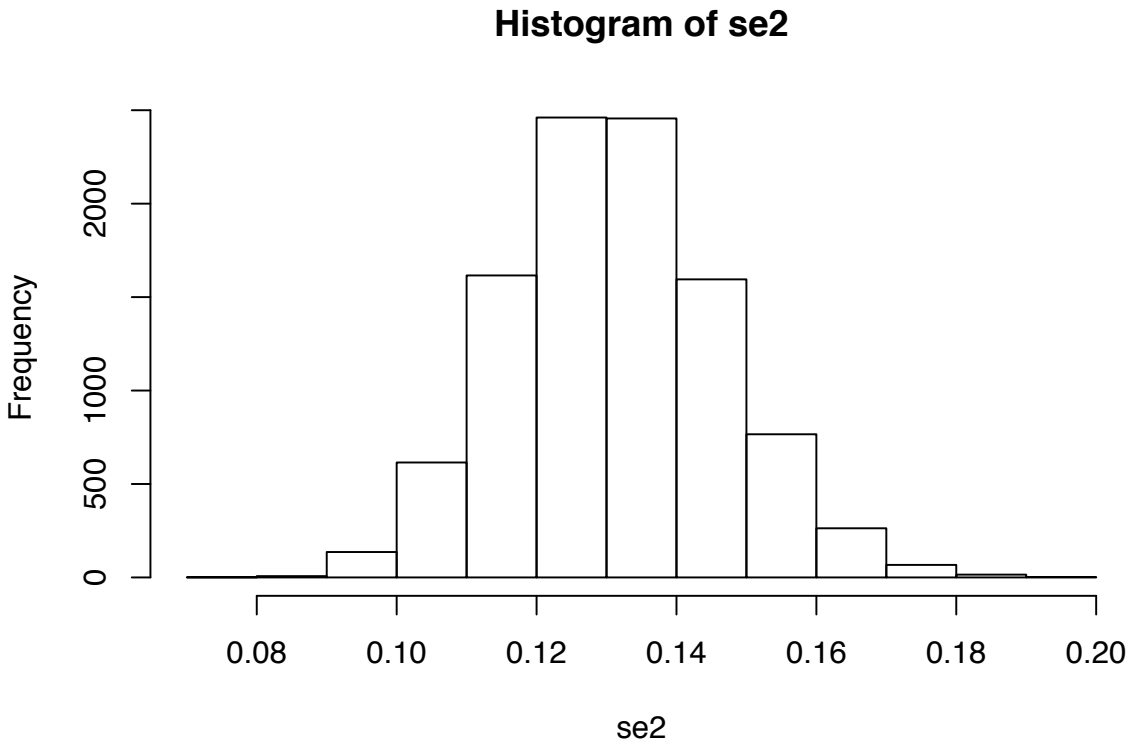
```
sd(estimate)
```

```
## [1] 0.1335631
```

```
hist(se)
```




```
hist(se2)
```



Summarize briefly your findings.

The true (empirical) standard deviation of $\tilde{\beta}$ across the 10000 simulations is 0.134. We ploted two histograms

of the estimated standard errors using the two different methods. From the plot, we can see that the histogram of se2 is around or centered at the true SD, but the histogram of se doesn't show that. Therefore we conclude that se2 using the method in part (c) is more accurate than se.