HOMEWORK 8 SOLUTIONS

Alex Chin

1. Fitting a Bradley-Terry model.

(a) First read in the data.

```
table = read.csv('NBA_record.csv')
teams = read.csv('teams.txt', header=FALSE, as.is=TRUE)
num_games = nrow(table)
num_teams = nrow(teams)
```

The Bradley-Terry log-likelihood, as defined in Lecture 24, is

$$\sum_{m=1}^{n} Y_m(\alpha + \beta_{i_m} - \beta_{j_m}) - \log(1 + e^{\alpha+\beta_{i_m}-\beta_{j_m}}).$$

Let's write a function to compute this.

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

We can train the model using the `optim` function.

```
theta0 = rep(0, num_teams)
result = optim(theta0, loglik,
  Home=table$Home, Away=table$Away, Y=table$Y,
  method='BFGS', control=list('fnscale'=-1))
```

Now we print the 8 teams with the highest Bradley-Terry scores:

```
coefs = c(0, result$par[-1])
ranking = order(coefs, decreasing=TRUE)
data.frame(team=teams[ranking[1:8],],
           score=coefs[ranking[1:8]])
```

Stats 200: Autumn 2016.

```
##                        team      score
## 1 10 Golden State Warriors  1.90274553
## 2     27 San Antonio Spurs  1.29320880
## 3    6 Cleveland Cavaliers  0.48246253
## 4       28 Toronto Raptors  0.42940629
## 5 21 Oklahoma City Thunder  0.42045913
## 6  13 Los Angeles Clippers  0.28898908
## 7          1 Atlanta Hawks  0.00000000
## 8            16 Miami Heat -0.01215086
```

The intercept term indicates the home court advantage:

```
result$par[1]

## [1] 0.4626864
```

So the home team has a $\hat{\alpha} = 0.4626864$ greater log-odds of winning than the away team. This means, for example, that we can predict the home team has a $e^{-\hat{\alpha}} = 0.62959$ chance of winning if the two teams are evenly matched.

(b) Let's define the log-likelihood function without an intercept and run the optimization.

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

theta0 = rep(0, num_teams - 1)
result_noalpha = optim(theta0, loglik_noalpha,
  Home=table$Home, Away=table$Away, Y=table$Y,
  method='BFGS', control=list('fnscale'=-1))
```

We can take a look at the optimal log-likelihood values:

```
print(result$value)

## [1] -680.2417

print(result_noalpha$value)

## [1] -705.08
```

Now we can perform a generalized likelihood ratio test, using a χ^2 cutoff with 1 degree of freedom.

```
statistic = -2 * (result_noalpha$value - result$value)
p_value = 1 - pchisq(statistic, df=1)
print(statistic)

## [1] 49.67658

print(p_value)

## [1] 1.812994e-12
```

The p-value is extremely small, so we can safely reject the null hypothesis that there is no home court advantage.

(c) Logistic regression models the log-odds of the probability of winning as linear in the covariates,

$$\log \frac{p_m}{1-p_m} = \alpha + \beta_1 x_{m,1} + \cdots + \beta_{30}, x_{m,30},$$

Taking the specified design matrix reduces the above expression to

$$\log \frac{p_m}{1-p_m} = \alpha + \beta_{i_m} - \beta_{j_m}$$

where i_m and j_m are the indices for the home and away teams that played in game m, respectively. From here we can see that $p_m = p_{i_m j_m}$, the Bradley-Terry specification of the probability of team i_m beating team j_m. Above, the interpretation of the first estimated coefficient is the home court advantage and the interpretation of the tenth coefficient is the log-odds probability of team 10 beating team 1 at a neutral venue.

(d) First we build the model matrix.

```
X = matrix(0, nrow=num_games, ncol=num_teams)
for (m in 1:num_games) {
  X[m, 1] = 1
  home = table$Home[m]; if (home != 1) X[m, home] = 1
  away = table$Away[m]; if (away != 1) X[m, away] = -1
}
```

We run the logistic regression, and verify that the coefficients obtained from `optim` and `glm.fit` match.

```r
model = glm.fit(X, table$Y, family=binomial())
data.frame(label=c("intercept", teams[-1,1]),
  optim=result$par,
  glm=model$coefficients)
```

```
##                        label       optim         glm
## 1                  intercept  0.46268643  0.46258355
## 2           2 Boston Celtics -0.04631213 -0.04648549
## 3            3 Brooklyn Nets -1.61273464 -1.61284430
## 4        4 Charlotte Hornets -0.04467431 -0.04484980
## 5            5 Chicago Bulls -0.34269585 -0.34288850
## 6      6 Cleveland Cavaliers  0.48246253  0.48230358
## 7         7 Dallas Mavericks -0.30639779 -0.30652209
## 8           8 Denver Nuggets -0.84037107 -0.84055558
## 9          9 Detroit Pistons -0.25148051 -0.25161339
## 10  10 Golden State Warriors  1.90274553  1.90258509
## 11        11 Houston Rockets -0.38169132 -0.38186156
## 12         12 Indiana Pacers -0.17399666 -0.17414817
## 13   13 Los Angeles Clippers  0.28898908  0.28888477
## 14     14 Los Angeles Lakers -1.87604922 -1.87617349
## 15      15 Memphis Grizzlies -0.34574217 -0.34587860
## 16            16 Miami Heat -0.01215086 -0.01229379
## 17        17 Milwaukee Bucks -0.82875230 -0.82893701
## 18 18 Minnesota Timberwolves -1.08557680 -1.08567514
## 19   19 New Orleans Pelicans -1.02941242 -1.02954690
## 20        20 New York Knicks -0.92056670 -0.92063507
## 21  21 Oklahoma City Thunder  0.42045913  0.42029832
## 22          22 Orlando Magic -0.75226660 -0.75242422
## 23     23 Philadelphia 76ers -2.55271651 -2.55279786
## 24           24 Phoenix Suns -1.47403237 -1.47425350
## 25 25 Portland Trail Blazers -0.21387094 -0.21399684
## 26       26 Sacramento Kings -0.85175169 -0.85185252
## 27      27 San Antonio Spurs  1.29320880  1.29305390
## 28        28 Toronto Raptors  0.42940629  0.42930221
## 29              29 Utah Jazz -0.45062893 -0.45075953
## 30     30 Washington Wizards -0.39769361 -0.39782372
```

2. A heteroscedastic linear model.

(a) The log-likelihood is

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

so that the MLE $\hat{\beta}$ minimizes the weighted sum of squares

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

Taking derivatives and solving for zero gives the explicit solution

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

(b) The estimator has mean

$$\mathbf{E}[\hat{\beta}] = \frac{1}{\sum_{i=1}^{n} x_i^2/\sigma_i^2} \sum_{i=1}^{n} \frac{x_i}{\sigma_i^2}\mathbf{E}[Y_i] = \frac{1}{\sum_{i=1}^{n} x_i^2/\sigma_i^2} \sum_{i=1}^{n} \frac{x_i}{\sigma_i^2}\beta x_i = \beta,$$

so it is unbiased.

The variance is

$$\mathbf{V}[\hat{\beta}] = \frac{\sum_{i=1}^{n} x_i^2 \mathbf{V}[Y_i]/\sigma_i^4}{\left(\sum_{i=1}^{n} x_i^2/\sigma_i^2\right)^2} = \frac{\sum_{i=1}^{n} x_i^2 \sigma_i^2/\sigma_i^4}{\left(\sum_{i=1}^{n} x_i^2/\sigma_i^2\right)^2} = \frac{1}{\sum_{i=1}^{n} x_i^2/\sigma_i^2},$$

using $\mathbf{V}[Y_i] = \sigma_i^2$.

(c) Taking derivatives, we have

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

Therefore, $I_{\mathbf{Y}}(\beta) = -\mathbf{E}_\beta[\ell''(\beta)] = \sum_{i=1}^{n} x_i^2/\sigma_i^2$. From part (b), this is exactly $1/\mathbf{V}[\hat{\beta}]$.

(d) Taking the derivative of $\sum_i (Y_i - \beta x_i)^2$ and solving gives

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

and its expectation is

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

(e) The variance of $\tilde{\beta}$ is given by

$$\mathbf{V}[\tilde{\beta}] = \frac{\sum_{i=1}^{n} x_i^2 \mathbf{V}[Y_i]}{(\sum_{i=1}^{n} x_i^2)^2} = \frac{\sum_{i=1}^{n} x_i^2 \sigma_i^2}{(\sum_{i=1}^{n} x_i^2)^2}.$$

If $\sigma_i^2 = \sigma_0^2$, then this reduces to

$$\mathbf{V}[\tilde{\beta}] = \frac{\sigma_0^2}{\sum_{i=1}^n x_i^2}.$$

In our case the model matrix X is a single vector $(x_1, \ldots, x_n)^T$, so that $(X^T X)^{-1} = 1/\sum_{i=1}^n x_i^2$. Hence the variance formula above is consistent with the general formula $\sigma_0^2(X^T X)^{-1}$.

(f) Applying the Cauchy-Schwarz inequality with $a_i = |x_i \sigma_i|$ and $b_i = |x_i/\sigma_i|$, we obtain the inequality

$$\left(\sum_{i=1}^n x_i^2 \sigma_i^2\right)\left(\sum_{i=1}^n \frac{x_i^2}{\sigma_i^2}\right) \geq \left(\sum_{i=1}^n x_i^2\right)^2,$$

and hence, rearranging terms,

$$\mathbf{V}[\tilde{\beta}] = \frac{\sum_{i=1}^n x_i^2 \sigma_i^2}{(\sum_{i=1}^n x_i^2)^2} \geq \frac{1}{\sum_{i=1}^n x_i^2/\sigma_i^2} = \mathbf{V}[\hat{\beta}].$$

The Cramèr-Rao lower bound states that any unbiased estimator has variance no smaller than the inverse Fisher information. Since the variance of $\hat{\beta}$ attains the lower bound by part (c), and $\tilde{\beta}$ is unbiased, the above inequality is expected.

(g) Plugging in $(Y_i - \tilde{\beta} x_i)^2$ for σ_i^2 in the variance expression in part (e) gives the plugin standard error estimate

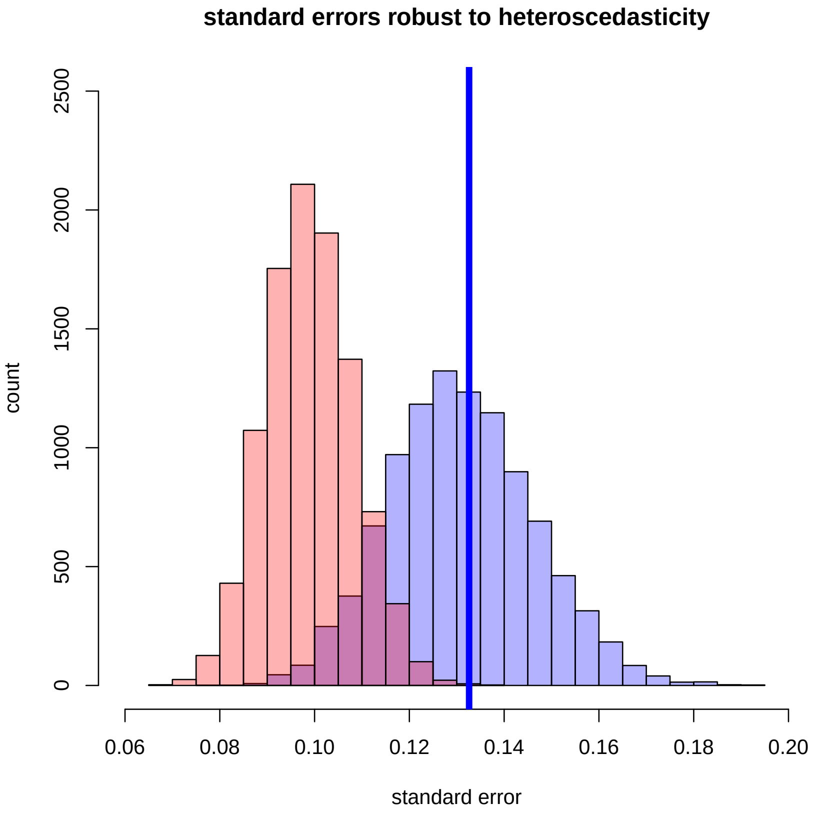
$$\widehat{\text{se}}(\tilde{\beta}) = \sqrt{\frac{\sum_{i=1}^n x_i^2 (Y_i - \tilde{\beta} x_i)^2}{(\sum_{i=1}^n x_i^2)^2}}.$$

(h) We run the simulation.

```
set.seed(2016)
x = (1:100) / 100
sigma = (1:100) / 100
B = 10000

estimates = numeric(B)
default_se = numeric(B)
robust_se = numeric(B)

get_robust_se = 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)
  estimates[i] = summary(model)[["coefficients"]][["x","Estimate"]]
  default_se[i] = summary(model)[["coefficients"]][["x","Std. Error"]]
  robust_se[i] = get_robust_se(x, Y, estimates[i])
}
```

```
true_se = sd(estimates)
print(true_se)

## [1] 0.1326056

hist(robust_se, col=rgb(0, 0, 1, 0.3), breaks=20,
  xlim=c(0.06, 0.2), ylim=c(0, 2500),
  xlab="standard error", ylab="count",
  main="standard errors robust to heteroscedasticity")
hist(default_se, col=rgb(1, 0, 0, 0.3), breaks=20, add=TRUE)
abline(v=true_se, col="blue", lwd=5)
```



Above, the default standard error estimates from `lm` are printed in **red** and the robust plugin standard error estimates from part (g) are printed in **blue**, with the solid **blue** line indicating the true empirical standard error. We see that the robust estimates are more accurate. Notably, the default estimates are too small, which would produce too narrow confidence intervals and too small p-values if we were to use them for our coefficient estimates.