MA 576 HW 7

Benjamin Draves

Exercise 3

Part A

First we build the proportional hazards model to estimate the hazard ratio. The confidence intervals for these values are given below.

```
#packages
library(ggplot2)
#read in data
bladder = read.table("~/Desktop/Courses/MA 576/data/bladder.txt", header = TRUE)
head(bladder)
```

```
##   time group uncensored number
## 1    0     1          0      1
## 2    1     1          0      2
## 3    4     1          0      1
## 4    7     1          0      1
## 5   10     1          0      1
## 6    6     1          1      1
```

```
#remove pathalogical case
bladder = bladder[-1,]

#format survival data
t = bladder$time

#fit proportional hazard model
m1 = glm(uncensored~ group + number, family = poisson, data = bladder,
         offset = log(time))

#hazard ratio
exp(coef(m1)[2])
```

```
##     group
## 0.5857315
```

```
exp(-coef(m1)[3])
```

```
##   number
## 1.685656
```

```
#Confidence Intervals
CI  = confint(m1)
```

```
## Waiting for profiling to be done...
```

```r
hazPCI =  as.numeric(CI[2,])
hazTCI = as.numeric(-CI[3,2:1])

hazPCI
```

```
## [1] -1.14099202  0.04396409
```

```r
hazTCI
```

```
## [1] -0.05670311  1.12825299
```

Part B

```r
#iteraction model
m2 = glm(uncensored~ group + number + group:number, family = poisson
         ,data = bladder,
         offset = log(time))
summary(m2)
```

```
##
## Call:
## glm(formula = uncensored ~ group + number + group:number, family = poisson,
##     data = bladder, offset = log(time))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2681  -1.1049   0.1769   1.3922   2.6084
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.2903     1.3474  -2.442   0.0146 *
## group          0.4260     0.9136   0.466   0.6410
## number         0.4387     0.9136   0.480   0.6311
## group:number  -0.7071     0.6452  -1.096   0.2731
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 161.88  on 84  degrees of freedom
## Residual deviance: 154.31  on 81  degrees of freedom
## AIC: 256.31
##
## Number of Fisher Scoring iterations: 7
```

```r
#not significant iteraction model
```

After fitting the iteraction term, we see that the interaction is not signficiant. That is we do not have evidence to suggest that the drug type along with the number of tumors removed has an effet on this proportional hazards model.

Part C

```
bladder2 = matrix(NA, nrow = sum(bladder$time), ncol = 4)
current = 1
for(i in 1:nrow(bladder)){
  n = bladder[i,"time"]
  tmp = matrix(NA, ncol = 4, nrow = n)
  tmp[,1] = 1:n
  tmp[,2] = rep(bladder[i,2], n)
  tmp[,3] = c(rep(0, n-1), bladder[i,3])
  tmp[,4] = rep(bladder[i,4], n)
  bladder2[current:(current + n-1),] = tmp
  current = current + n
}

colnames(bladder2) = colnames(bladder)
bladder2 = data.frame(bladder2)
bladder2[,2] = as.factor(bladder2[,2])
bladder2[,4] = as.factor(bladder2[,4])
bladder2$time2 = (bladder2$time)^2

#t^2 model
m3 = glm(uncensored~ group + number + time + time2, family = poisson
         ,data = bladder2)
summary(m3)
```

```
##
## Call:
## glm(formula = uncensored ~ group + number + time + time2, family = poisson,
##     data = bladder2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.4349  -0.2887  -0.2111  -0.1586   2.7389
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.2701654  0.3034380  -7.481 7.35e-14 ***
## group2      -0.4064664  0.3016949  -1.347   0.1779
## number2     -0.4020016  0.3016606  -1.333   0.1827
## time        -0.0893493  0.0398033  -2.245   0.0248 *
## time2        0.0009759  0.0009971   0.979   0.3277
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 328.91  on 1554  degrees of freedom
```

```
## Residual deviance: 307.20  on 1550  degrees of freedom
## AIC: 411.2
##
## Number of Fisher Scoring iterations: 6
```

```
#t model
m4 = glm(uncensored~ group + number + time2, family = poisson
         ,data = bladder2)
summary(m4)
```

```
##
## Call:
## glm(formula = uncensored ~ group + number + time2, family = poisson,
##     data = bladder2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.3602  -0.2868  -0.2344  -0.1756   2.8488
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.7342554  0.2350894 -11.631   <2e-16 ***
## group2      -0.4237538  0.3014179  -1.406   0.1598
## number2     -0.4355757  0.3013525  -1.445   0.1483
## time2       -0.0013113  0.0005028  -2.608   0.0091 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 328.91  on 1554  degrees of freedom
## Residual deviance: 311.71  on 1551  degrees of freedom
## AIC: 413.71
##
## Number of Fisher Scoring iterations: 7
```

```
anova(m3, m4, "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: uncensored ~ group + number + time + time2
## Model 2: uncensored ~ group + number + time2
##   Resid. Df Resid. Dev Df Deviance
## 1      1550     307.20
## 2      1551     311.71 -1  -4.5141
```

Here we use a general point process model to model the hazard of recurrence. We expand the dataset to contain binned times $1, 2, \ldots, t_i$ where t_i is the time of the ith censoring. We simply repeat covariate values along with a new censoring variable $(0, 0, \ldots, 1)$. After this model with time covariates (t, t^2), we see that that only significant random varible is given by the intercept and the

linear time variable. The baseline number of occurrences is given by $\exp(-2.2701654) = 0.1032951$. Moreover, we see a decrease in the expected number of occurences for those in the drug group compared to the placebo group. Similarly we expect the number of censorings to decrease,

Exerice 4

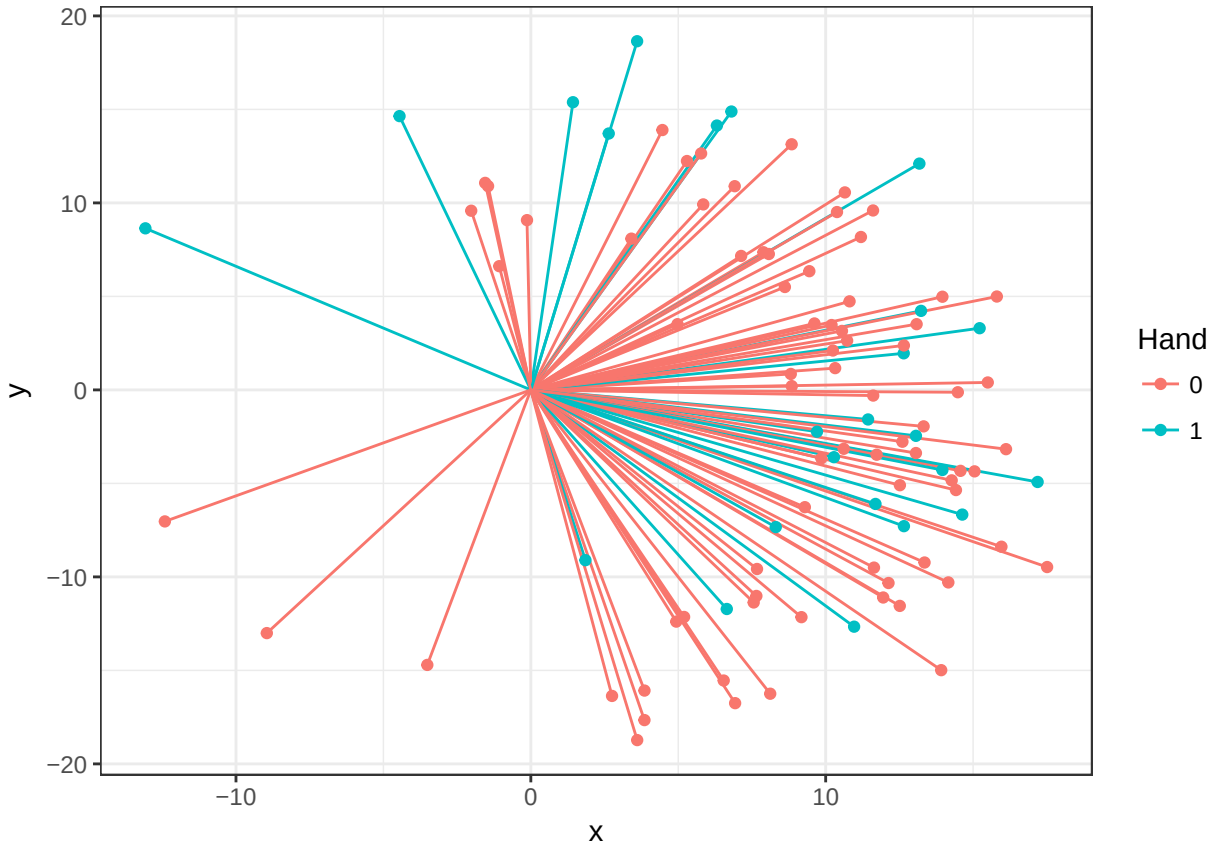
Part A

```
#read in data
baseball = as.data.frame(read.table("~/Desktop/Courses/MA 576/data/baseball.txt", header = TRU
head(baseball)
```

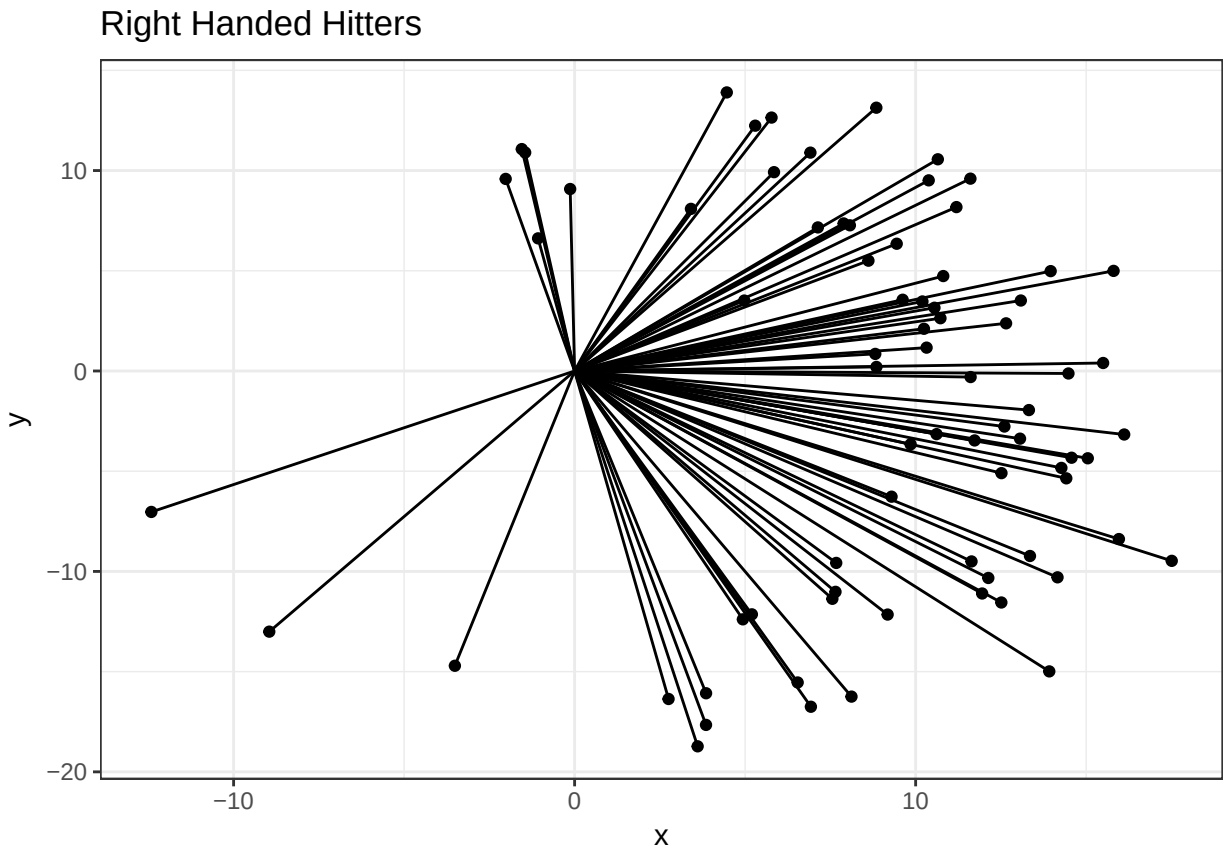
```
##      speed hand      dir
## 1 90.51042    0 5.927594
## 2 98.02963    0 5.799443
## 3 96.86197    0 5.110764
## 4 97.02960    0 5.537792
## 5 91.21170    0 5.689565
## 6 93.96255    1 1.380422
```

Below, we make some visuals describing the data presented here. Directly to the right corresponds to center field and the length of the vector corresponds to the pitch speed.

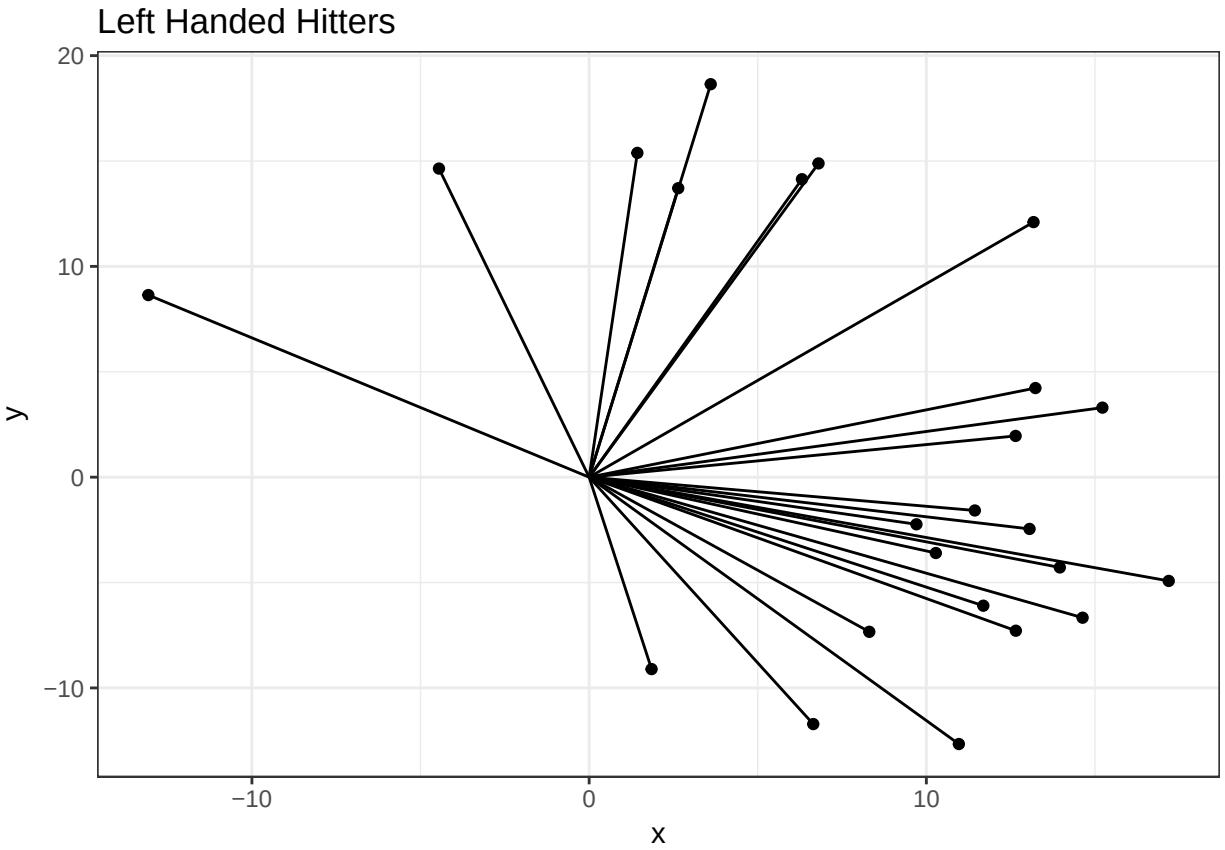
```
#visuals
baseball$x = (baseball$speed-80) * cos(baseball$dir)
baseball$y = (baseball$speed-80) * sin(baseball$dir)
baseball$Hand = as.factor(baseball$hand)
baseball$zero = 0

baseballR = baseball[which(baseball$hand == 0),]
baseballL = baseball[which(baseball$hand == 1),]

p1 = ggplot(aes(x, y, col = Hand),data = baseball) +
    geom_segment(aes(x = zero, y = zero, xend = x, yend = y )) +
    theme_bw()+
    geom_point()
p1
```




```
p2 = ggplot(aes(x, y),data = baseballR) +
    geom_segment(aes(x = zero, y = zero, xend = x, yend = y )) +
    theme_bw()+
    geom_point() + ggtitle("Right Handed Hitters")
p2
```




```
p3 = ggplot(aes(x, y),data = baseballL) +
    geom_segment(aes(x = zero, y = zero, xend = x, yend = y )) +
    theme_bw()+
    geom_point() + ggtitle("Left Handed Hitters")
p3
```



Now, to construct this model, we note that for right handers: for slower pitches we expect their swing to be “early" or $sin(dir) > 0$. Conversely, for faster pitches we expect the batter to be late corresponding to $sin(dir) < 0$. For this reason, for right handed hitters only, we ciuld build Gamma glm with $log(MPH) = \beta_0 + \beta_1 sin(dir)$. In this model we would expect $\beta_1 < 0$ which corresponds to the relation: as sin(dir) increases we expect the pitch speed to decrease.

For left handers we expect slower pitch with result inearly swings or $sin(dir) < 0$. Faster pitches corresponds to late swings which can be measured with $sin(dir) > 0$. Similarily for the right harnders, we could model left handed hitters by building a gamma glm with $log(MPH) = \beta_0 + \beta_2 sin(dir)$. Notice we can simply combine these models and we fit

$$\log(MPH) = \beta_0 + \beta_1 Hand * sin(dir) + \beta_2 (1 - Hand) * sin(dir)$$

```
#append some additional variables
baseball$liter = baseball$y * baseball$hand
baseball$riter = baseball$y * (1 - baseball$hand)

#fit gamma glm
m1 = glm(speed ~ riter +liter ,data = baseball, family = Gamma(link = "log"))
summary(m1)
```

```
##
## Call:
## glm(formula = speed ~ riter + liter, family = Gamma(link = "log"),
##     data = baseball)
```

```
##
## Deviance Residuals:
##       Min         1Q     Median         3Q        Max
## -0.074023  -0.020373  -0.000838   0.017069   0.053652
##
## Coefficients:
##               Estimate Std. Error  t value Pr(>|t|)
## (Intercept)  4.5373923  0.0026506 1711.812  < 2e-16 ***
## riter       -0.0020015  0.0003248   -6.163 1.63e-08 ***
## liter        0.0013880  0.0005460    2.542   0.0126 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.0006802432)
##
##     Null deviance: 0.096325  on 99  degrees of freedom
## Residual deviance: 0.066107  on 97  degrees of freedom
## AIC: 467.67
##
## Number of Fisher Scoring iterations: 3
```

```
exp(coef(m1))
```

```
## (Intercept)       riter       liter
##  93.4467977   0.9980005   1.0013890
```

The paramter estimates of this model can be interpreted as follows. $e^{\beta_0} = 93.4467977$: Expected speed of pitch when hit to dead center ($sin(0) = 0$). $e^{\beta_1} = 0.9980005$: modulation in expected speed for left handers as $\exp(\sin(\varphi_i))$ changes by a single unit. $e^{\beta_2} = 1.0013890$: modulation in expected speed for right handers as $\exp(\sin(\varphi_i))$ changes by a single unit.

Part B

```
#set up data frame
X = as.matrix(cbind(baseball[,c("speed","hand")]))

#initialize variables
beta = matrix(c(0, 0), nrow =2)
mu0 = 0
beta.diff = Inf
mu.diff = Inf

#iterate until covergence
while(mu.diff > .00000000000000001){
  beta.diff = Inf

  while(beta.diff > .0000000000000001){
    #set eta
    eta = as.vector(X %*% beta)
```

```
    #set linear space prediction
    z = eta + sin(baseball$dir - mu0 - 2*atan(eta)) * (1 + eta^2)/2

    #set G
    G = diag(c(2/(1 + eta^2)))

    #get beta
    Fisher = t(X) %*% G^2 %*% X
    beta.new = solve(Fisher) %*% t(X) %*% G^2 %*% z

    #set beta.diff
    beta.diff = sum((beta - beta.new)^2)

    #reset variables
    beta = beta.new
  }

  #update mu
  est = 2*atan(X%*%beta)
  mu.new = atan(sum(sin(baseball$dir - est))/sum(cos(baseball$dir - est)))

  #set mu diff
  mu.diff = (mu0 - mu.new)^2

  #set mu0
  mu0 = mu.new
}

#load in Ainv function
library("circular")
```

```
##
## Attaching package: 'circular'
```

```
## The following objects are masked from 'package:stats':
##
##     sd, var
```

```
est = 2*atan(X%*%beta)

#get R values
R = sqrt(sum(sin(baseball$dir - est))^2 + sum(cos(baseball$dir - est))^2)
Rbar = R /nrow(X)

#get estimate for kappa
Kappa = A1inv(Rbar)

#Covariance stuff
```

```
g = as.matrix(2/(1+eta^2), nrow = 100)
y = baseball$dir
CovB = 1/(Kappa * A1(Kappa))*solve(Fisher) * as.numeric((1 + 1/(nrow(X) - t(g)%*% X %*% solve(
VarMu = 1/(nrow(X) - as.numeric(t(g)%*% X %*% solve(Fisher)%*% t(X) %*% g))
VarK = 1/(nrow(X)*A1FirstDerivative(Kappa))

#build beta CI
SE = sqrt(diag(CovB))
t = qt(.975, nrow(X) - length(beta) - 2)

cbind(L = beta -SE * t,C = beta, R = beta + SE*t)
```

```
##                [,1]        [,2]        [,3]
## speed  -0.08163063 -0.01015326  0.06132411
## hand  -12.33410624  0.13380693 12.60172010
```

```
#build mu0 CI
SE = sqrt(VarMu)
cbind(L =  mu0 -SE * t, R = mu0 + SE*t)
```

```
##              L        R
## [1,] -61.75454 64.68806
```

```
#build Kappa CI
SE = sqrt(VarK)
cbind(L =  Kappa -SE * t, R = Kappa + SE*t)
```

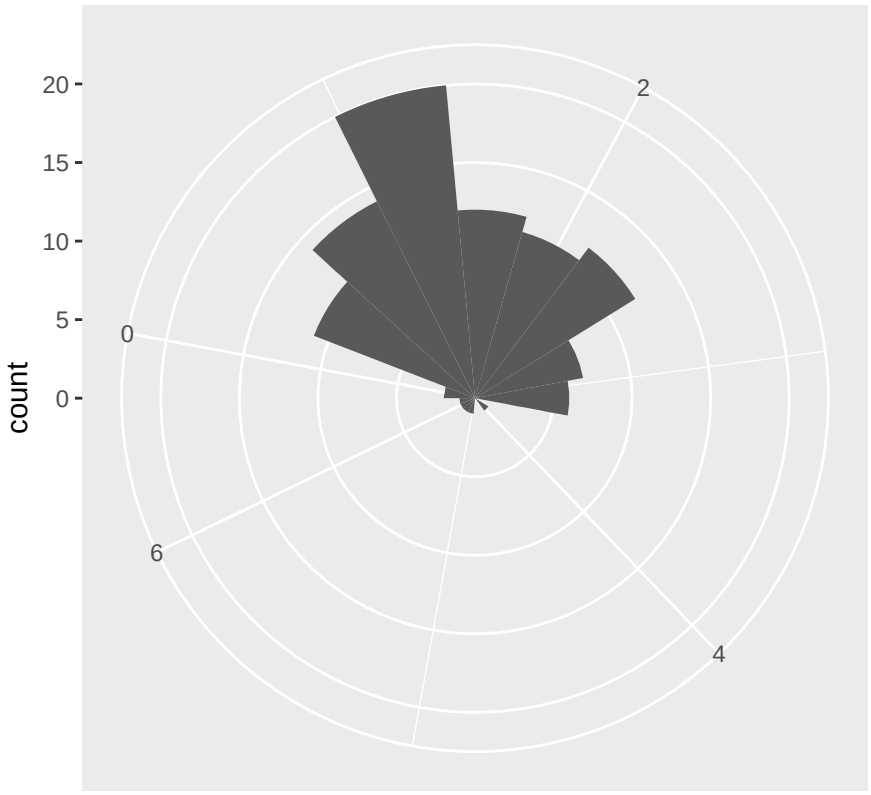
```
##             L       R
## [1,] 1.246502 2.10289
```

Here we implement the von Mises IRLS estimation procedure. We choose to use the arctangent-link function. From here, we estimate model parameters as well as corresponding variance estimates which can be intrepretted as follows. As we do not fit an intercept model, we see the the β coefficents correspond to changes in change in tangent - angle corresponding to changes in the covariates. Namely, for $\hat{\beta}_{Hand} = 0.13380693$, we expect for a baseline pitch speed for the left handed batters to hit the ball to left field more. This suggests that left handed hitters are behind the pitch more often than right handed hitters. Moreover, for every unit increase in speed the expected ball direction to decrease at a rate of -0.01015326 in terms of the tangent function. This suggests that as speed increases, hits tend to go to the opposite field for right handed hitters.

Confidence intervals for all estimated paramters are given above. We see that with the expception of κ, every interval contains zero suggesting that we do not have evidence to suggest that the parameters are not different from zero. The model from part a and the model here are similar in the sense they are modeling the same phenoma in two different ways. We know there is a connection between handedness, pitch speed, and hit direction. The first model predicts pitch speed from handedness and hit direction. Here we estimate the direction of hit direction from pitch speed and handedness and both models offer similar results with varying degree of confidence.

Part C

```
est = 2*atan(X %*% beta)
resid = data.frame(Residual = baseball$dir - est)%%(2*pi)
ggplot(data = resid, aes(Residual))+
  coord_polar(, start = -pi/2)+
  geom_histogram(binwidth  = pi/8)
```



Here we see the residuals for this model are roughlt normally distributed in the $(0,\pi)$ range centered around one. We prefer these were centered around zero and hence this plot suggests that this model misspecifies the hit direciton in some small way. Wiht most of the errors clustered in the same direction, however, we see that these residuals share a common symmetric (possible right skewed) distribution.