Final project

Ya-Chen Lin (Lisa)

Task 01

```r
#combine 5 files and add position
k <- read.csv('proj_k15.csv')
qb <- read.csv('proj_qb15.csv')
rb <- read.csv('proj_rb15.csv')
te <- read.csv('proj_te15.csv')
wr <- read.csv('proj_wr15.csv')
cols <- unique(c(names(k), names(qb), names(rb), names(te), names(wr)))
k[,'pos'] <- 'k'
qb[,'pos'] <- 'qb'
rb[,'pos'] <- 'rb'
te[,'pos'] <- 'te'
wr[,'pos'] <- 'wr'
cols <- c(cols, 'pos')
k[,setdiff(cols, names(k))] <- 0
qb[,setdiff(cols, names(qb))] <- 0
rb[,setdiff(cols, names(rb))] <- 0
te[,setdiff(cols, names(te))] <- 0
wr[,setdiff(cols, names(wr))] <- 0
x <- rbind(k[,cols], qb[,cols], rb[,cols], te[,cols], wr[,cols])
newx <- x
#multiply numeric columns with percentage.
for(i in 1:nrow(x)){
  for(k in 3:18){
    if (newx[i,"Team"] == "CLE" | newx[i,"Team"] =="NO" | newx[i,"Team"] =="NYG" | newx[i,"Team"] == "P
      newx[i,k] <- newx[i,k] * 10/16
    }else{
      newx[i,k] <- newx[i,k] * 9/16
    }
  }
}
#sort and order data by 'fpts'
attach(newx)
dataorder <- newx[order(-fpts),]
#Subset the data by keeping the top 20 kickers, top 20 quarterbacks,
#top 40 running backs, top 60 wide recievers, and top 20 tight ends.
#Thus the projection data should only have 160 rows. (final - prodata)
orderk <- subset(dataorder, pos == "k")
orderqb <- subset(dataorder, pos == "qb")
orderrb <- subset(dataorder, pos == "rb")
orderte <- subset(dataorder, pos == "te")
orderwr <- subset(dataorder, pos == "wr")
prodata <- rbind(orderk[1:20,], orderqb[1:20,], orderrb[1:40,], orderwr[1:60,], orderte[1:20,])
#Read in the observed data
obs <- read.csv("nfl_current15.csv")
pros <- prodata[,-6]
obss <- obs[,c(1,2,17,16,18,12,11,13,14,15,5,6,7,4,8,9,10,3)]
```

```
newmat <- matrix(NA, nrow=320, ncol=ncol(pros))
for(i in 1:160) {
  n <- which(obss$Name == as.character(pros[i,1]))
  for(k in 1:ncol(pros)){
    newmat[(2*i)-1,k] <- as.character(pros[i,k])
    if(length(n) == 0){
      newmat[2*i,k] <- 0
    }else{
      newmat[2*i,k] <- as.character(obss[n,k])
    }
  }
}
colnames(newmat) <- colnames(pros)
newmatt <- as.data.frame(newmat)
finamat <- matrix(NA, ncol=16, nrow=160)
for(i in 1:160){
  finamat[i,16] <- as.character(newmatt[(2*i)-1,18])
  for(k in 1:15){
    finamat[i,k] <- as.numeric(newmatt[2*i,k+2]) - as.numeric(newmatt[(2*i)-1,k+2])
  }
}
colnames(finamat) <- c("field goals", "field goals attempted", "extra points ",
                       "passing attempts","passing completions","passing yards",
                       "passing touchdowns","passing interceptions","rushing attempts",
                       "rushing yards","rushing touchdowns","fumbles","receiving attempts",
                       "receiving yards","receiving touchdowns","pos")
difmat <- as.data.frame((finamat))
kf <- subset(difmat, pos=="k", select= -pos)
qbf <- subset(difmat, pos=="qb", select= -pos)
wrf <- subset(difmat, pos=="wr", select= -pos)
tef <- subset(difmat, pos=="te", select= -pos)
rbf <- subset(difmat, pos=="rb", select= -pos)
noise <- list(kicker=kf,quarterback=qbf,wide_receiver=wrf,tight_endse=tef,running_backs=rbf)
#final list with 15 columns of interest
```

Task 02

```
x <- newx#to compile. x is the projection data.
league <- function(stats=x,nTeams=10,cap=200,posReq=pos,points=pnts){
  setup <- list(stats, nTeams=nTeams, cap=cap, posReq=posReq, points=points)
  class(setup) <- "league"
  return(setup)
}
#calculate points
calcpoints <- function(x){
  a <- data.frame(x[1])
  pts <- data.frame(x$points)
  a[,'p_fg'] <- a[,'fg']*pnts$fg
  a[,'p_xpt'] <- a[,'xpt']*pnts$xpt
  a[,'p_pass_yds'] <- a[,'pass_yds']*pnts$pass_yds
  a[,'p_pass_tds'] <- a[,'pass_tds']*pnts$pass_tds
  a[,'p_pass_ints'] <- a[,'pass_ints']*pnts$pass_ints
  a[,'p_rush_yds'] <- a[,'rush_yds']*pnts$rush_yds
  a[,'p_rush_tds'] <- a[,'rush_tds']*pnts$rush_tds
  a[,'p_fumbles'] <- a[,'fumbles']*pnts$fumbles
  a[,'p_rec_yds'] <- a[,'rec_yds']*pnts$rec_yds
  a[,'p_rec_tds'] <- a[,'rec_tds']*pnts$rec_tds
  a[,'points'] <- rowSums(a[,grep("^p_", names(a))])
  return(a[,c("PlayerName", "points","pos")])
}
#build values
buildValues <- function(obj){
  posReq <- unlist(obj$posReq)
  nTeams <- unlist(obj$nTeams)
  cap <- unlist(obj$cap)
  x <- calcpoints(obj)
  x2 <- x[order(x[,'points'], decreasing=TRUE),]
  k.ix <- which(x2[,'pos']=='k')
  qb.ix <- which(x2[,'pos']=='qb')
  rb.ix <- which(x2[,'pos']=='rb')
  te.ix <- which(x2[,'pos']=='te')
  wr.ix <- which(x2[,'pos']=='wr')
  if(posReq['k'] == 0) {
    x2[k.ix, 'marg'] <- 0
  }else{
    x2[k.ix, 'marg'] <- x2[k.ix,'points'] - x2[k.ix[nTeams*posReq['k']],'points']
  }
  if(posReq['qb'] == 0) {
    x2[qb.ix, 'marg'] <- 0
  }else{
    x2[qb.ix, 'marg'] <- x2[qb.ix,'points'] - x2[qb.ix[nTeams*posReq['qb']],'points']
  }
  if(posReq['rb'] == 0){
    x2[rb.ix, 'marg'] <- 0
  }else{
    x2[rb.ix, 'marg'] <- x2[rb.ix,'points'] - x2[rb.ix[nTeams*posReq['rb']],'points']
  }
  if(posReq['te'] == 0){
    x2[te.ix, 'marg'] <- 0
  }else{
    x2[te.ix, 'marg'] <- x2[te.ix,'points'] - x2[te.ix[nTeams*posReq['te']],'points']
  }
  if(posReq['wr'] == 0){
    x2[wr.ix, 'marg'] <- 0
  }else{
    x2[wr.ix, 'marg'] <- x2[wr.ix,'points'] - x2[wr.ix[nTeams*posReq['wr']],'points']
  }
  x3 <- x2[x2[,'marg'] >= 0,]
  x3 <- x3[order(x3[,'marg'], decreasing=TRUE),]
  x3[,'value'] <- x3[,'marg']*(nTeams*cap-nrow(x3))/sum(x3[,'marg']) + 1
  for ( i in 1:length(posReq)){
    if (posReq[i] == 0){
      x3 <- x3[!x3[,'pos'] == names(posReq[i]),]
    }
  }
  x4 <- x2[x2[,'marg'] < 0,]
  x4[,'value'] <- 0
  x5 <- rbind(x3, x4)
  x6 <- x5[,c('PlayerName','value','pos')]
  return(x6)
}
#method for print
print.league <- function(obj){
  b <- buildValues(obj)
  greaterzero <- subset(b, value > 0)
  return(data.frame(greaterzero))
}

plot.league <- function(obj){
  a <- print(obj)
  plot(1:nrow(a), a[,"value"], xlab="Ranking", ylab="Dollar Value", main="Scatterplot of Dollar Value ba
       cex.lab=1.5,pch="O", cex.main=1.7, cex.axis=1.2)
}
#boxplot
boxplot.league <- function(obj){
  a <- print(obj)
  boxplot(a[,'value'] ~ a[,'pos'], xlab="Position", ylab='Dollar Value', cex.lab=1.5,
          main="Boxplot of Player's position and Dollar Value", cex.main=1.7, cex.axis=1.2)
}
#histogram
hist.league <- function(obj){
  a <- print(obj)
  hist(a[,'value'], xlab='Dollar Value', main="Player's Dollar Value Distribution",
       cex.lab=1.5, cex.main=1.7, cex.axis=1.2)
}
```

Task 03

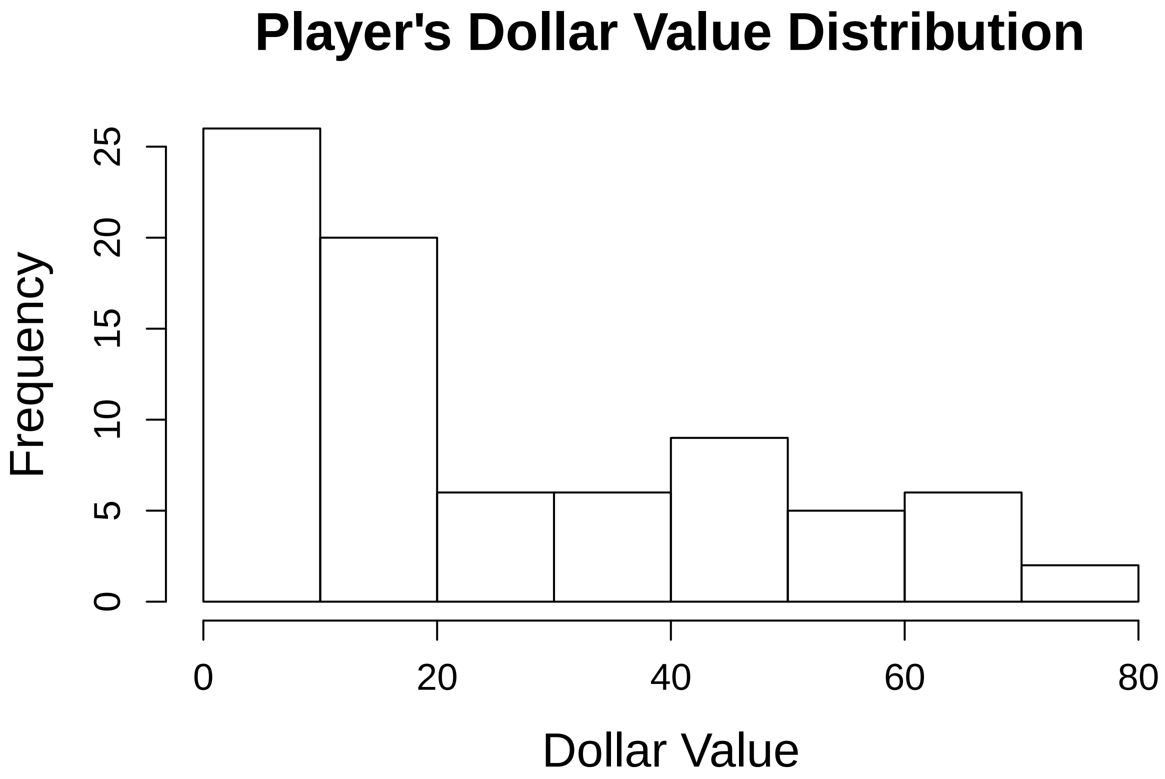
```
valueall <- function(obj,residuals, prodata){
  ks <- sample(1:nrow(residuals$kicker),1,replace=TRUE)
  qbs <- sample(1:nrow(residuals$quarterback),1,replace=TRUE)
  wrs <- sample(1:nrow(residuals$wide_receiver),1,replace=TRUE)
  tes <- sample(1:nrow(residuals$tight_endse),1,replace=TRUE)
  rbs <- sample(1:nrow(residuals$running_backs),1,replace=TRUE)
  for(i in 1:nrow(prodata)){
    for(k in 1:15){
      if(prodata[i,"pos"] == "k"){
        prodata[i, k+1] <-prodata[i, k+1]+ as.numeric(levels(residuals$kicker[ks,k]))[residuals$kicker[
      }else if(prodata[i,"pos"] == "qb"){
        prodata[i, k+1] <- prodata[i, k+1]+ as.numeric(levels(residuals$quarterback[qbs,k]))[residuals$
      }else if(prodata[i,"pos"] == "wr"){
        prodata[i, k+1] <- prodata[i, k+1]+ as.numeric(levels(residuals$wide_receiver[wrs,k]))[residual
      }else if(prodata[i,"pos"] == "te"){
        prodata[i, k+1] <- prodata[i, k+1]+ as.numeric(levels(residuals$tight_endse[tes,k]))[residuals$
      }else if(prodata[i,"pos"] == "rb"){
```

```
      prodata[i, k+1] <- prodata[i, k+1]+ as.numeric(levels(residuals$running_backs[rbs,k]))[residuals
    }
    if(prodata[i,k+1] < 0 ){
      prodata[i,k+1] <- 0
    }
  }

}
a <- prodata
pts <- data.frame(obj$points)
a[,'p_fg'] <- a[,'fg']*pnts$fg
a[,'p_xpt'] <- a[,'xpt']*pnts$xpt
a[,'p_pass_yds'] <- a[,'pass_yds']*pnts$pass_yds
a[,'p_pass_tds'] <- a[,'pass_tds']*pnts$pass_tds
a[,'p_pass_ints'] <- a[,'pass_ints']*pnts$pass_ints
a[,'p_rush_yds'] <- a[,'rush_yds']*pnts$rush_yds
a[,'p_rush_tds'] <- a[,'rush_tds']*pnts$rush_tds
a[,'p_fumbles'] <- a[,'fumbles']*pnts$fumbles
a[,'p_rec_yds'] <- a[,'rec_yds']*pnts$rec_yds
a[,'p_rec_tds'] <- a[,'rec_tds']*pnts$rec_tds
a[,'points'] <- rowSums(a[,grep("^p_", names(a))])
x <- a[,c("points","pos")]
posReq <- unlist(obj$posReq)
nTeams <- unlist(obj$nTeams)
cap <- unlist(obj$cap)
x2 <- x[order(x[,'points'], decreasing=TRUE),]
k.ix <- which(x2[,'pos']=='k')
qb.ix <- which(x2[,'pos']=='qb')
rb.ix <- which(x2[,'pos']=='rb')
te.ix <- which(x2[,'pos']=='te')
wr.ix <- which(x2[,'pos']=='wr')
k.x <- which(x[,'pos']=='k')
qb.x <- which(x[,'pos']=='qb')
rb.x <- which(x[,'pos']=='rb')
te.x <- which(x[,'pos']=='te')
wr.x <- which(x[,'pos']=='wr')
if(posReq['k'] == 0) {
  x[k.x, 'marg'] <- 0
}else{
  x[k.x, 'marg'] <- x[k.x,'points'] - x2[k.ix[nTeams*posReq['k']],'points']
}
if(posReq['qb'] == 0) {
  x[qb.x, 'marg'] <- 0
}else{
  x[qb.x, 'marg'] <- x2[qb.ix,'points'] - x2[qb.ix[nTeams*posReq['qb']],'points']
}
if(posReq['rb'] == 0){
  x[rb.x, 'marg'] <- 0
}else{
  x[rb.x, 'marg'] <- x[rb.x,'points'] - x2[rb.ix[nTeams*posReq['rb']],'points']
}
if(posReq['te'] == 0){
  x[te.x, 'marg'] <- 0
  }else{
    x[te.x, 'marg'] <- x[te.x,'points'] - x2[te.ix[nTeams*posReq['te']],'points']
  }
  if(posReq['wr'] == 0){
    x[wr.x, 'marg'] <- 0
  }else{
    x[wr.x, 'marg'] <- x[wr.x,'points'] - x2[wr.ix[nTeams*posReq['wr']],'points']
  }
  x3 <- x[x[,'marg'] >= 0,]

  x[,'value'] <- x[,'marg']*(nTeams*cap-nrow(x3))/sum(x3[,'marg']) + 1
  for ( i in 1:length(posReq)){
    if (posReq[i] == 0){
      x <- x[!x[,'pos'] == names(posReq[i]),]
    }
  }
  x[which(x[,'value'] < 0), 'value'] <- 0
  return(x[,c("value")])
}
#addNoise
addNoise <- function(obj,residuals,simulation,seed){
  set.seed(seed)
  prodata <- data.frame(obj[1])[,-c(2,6)]
  result <- replicate(simulation, valueall(obj,residuals,prodata))
  final <- list(obj,tp=obj$posReq, team=obj$nTeams, s=result,p=prodata[,"pos"], n=prodata[,"PlayerName"]
  class(final) <- "league"
  return(final)
}
#quantile
quantile.league <- function(obj,probs=c(0.25,0.5,0.75)){
  want <- unlist(obj$s)
  name <- unlist(obj$n)
  pos <- unlist(obj$p)
  np <- length(probs)
  mat <- matrix(NA, nrow=nrow(want), ncol=(np+1))

  for(i in 1:nrow(want)){
    mat[i,(np+1)] <- pos[i]
    for(k in 1:np){
        class(want[i,]) <- "league"
      mat[i,k] <- quantile(as.numeric(want[i,]), probs[k])

      }
  }
  attr(mat, "prob") <- probs
  return(mat)
}
#conf.
conf.interval <- function(obj,probs=c(0.25,0.5,0.75)){
  position <- obj$p
  numpo <- obj$tp
  nteam <- obj$team
  results <- quantile(obj,probs)
  prob <- attr(results, "prob")
  np <- length(prob)
  d <- ncol(results)
  data <- results[,-c(d)]
  result <- matrix(NA, nrow=nrow(data), ncol=ncol(data))
  for(i in 1:nrow(data)){
    for(k in 1:ncol(data)){
      result[i,k] <- as.numeric(data[i,k])
    }
  }

  kr <- result[which(position == "k"),]
  qbr <- result[which(position == "qb"),]
  wrr <- result[which(position == "wr"),]
  ter <- result[which(position == "te"),]
  rbr <- result[which(position == "rb"),]

  kr <- kr[order(kr[,np],decreasing=TRUE),]
  qbr <- qbr[order(qbr[,np],decreasing=TRUE),]
  wrr <- wrr[order(wrr[,np],decreasing=TRUE),]
  ter <- ter[order(ter[,np],decreasing=TRUE),]
  rbr <- rbr[order(rbr[,np],decreasing=TRUE),]

  fkr <- kr[1:(numpo$k*nteam),]
  colnames(fkr) <- prob
  fqbr <- qbr[1:(numpo$qb*nteam),]
  colnames(fqbr) <- prob
  fwrr <- wrr[1:(numpo$wr*nteam),]
  colnames(fwrr) <- prob
  fter <- ter[1:(numpo$te*nteam),]
  colnames(fter) <- prob
  frbr <- rbr[1:(numpo$rb*nteam),]
  colnames(frbr) <- prob
  newlist <- list(k=fkr,qb=fqbr,wr=fwrr, te=fter,rb=frbr)
  class(newlist) <- "league.conf.interval"
  return(newlist)
}
#plot method
plot.league.conf.interval <- function(obj,position){
  num <- ncol(obj[[position]])
  plot(unlist(obj[[position]][,num]),type='l',lty= num, ylab="Dollar Value", xlab="Ranking",
       main=paste("Dollar Value interval based on position", position))
  for(i in 1:(num-1)){
    lines(unlist(obj[[position]][,i]),lty=i)
  }

  legend('topright', legend=colnames(obj[[position]]),lty=1:num)

}
```

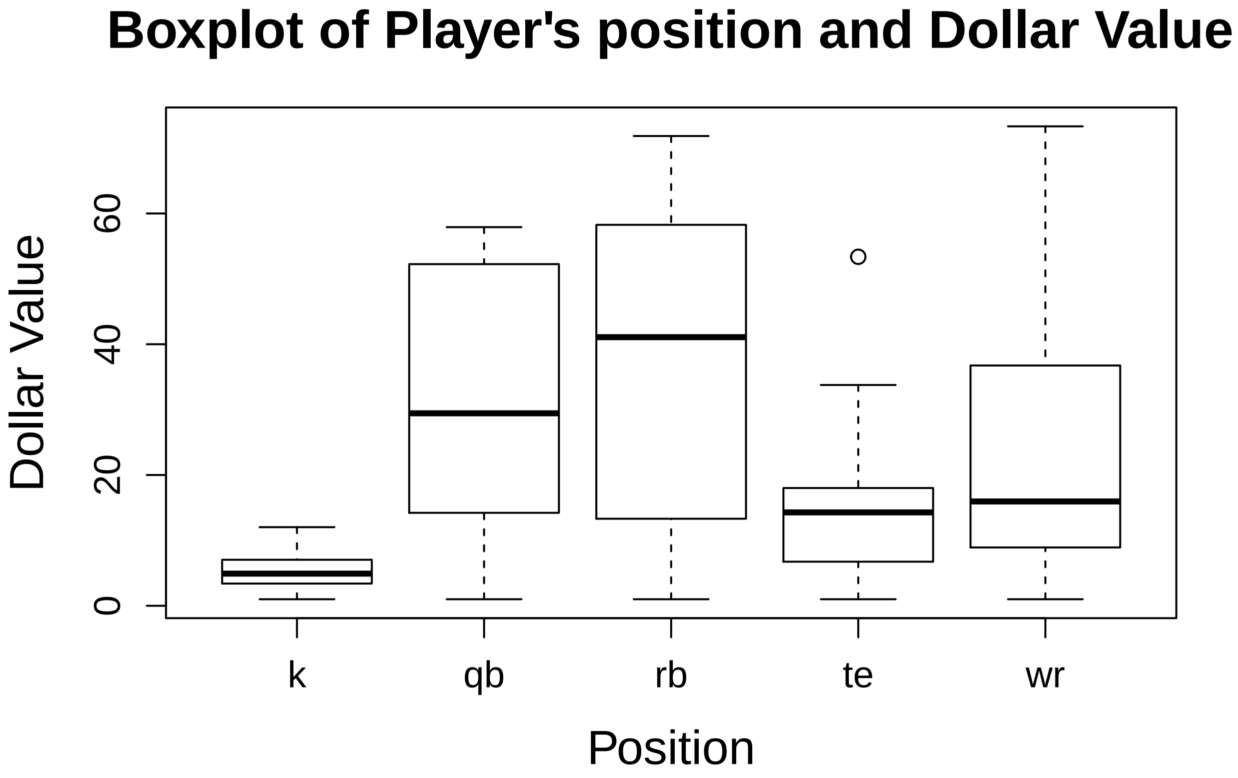
Test Q2

```
pos <- list(qb=1, rb=2, wr=3, te=1, k=1)
pnts <- list(fg=4, xpt=1, pass_yds=1/25, pass_tds=4, pass_ints=-2,
             rush_yds=1/10, rush_tds=6, fumbles=-2, rec_yds=1/20, rec_tds=6)
l <- league(stats=x, nTeams=10, cap=200, posReq=pos, points=pnts)
l
```

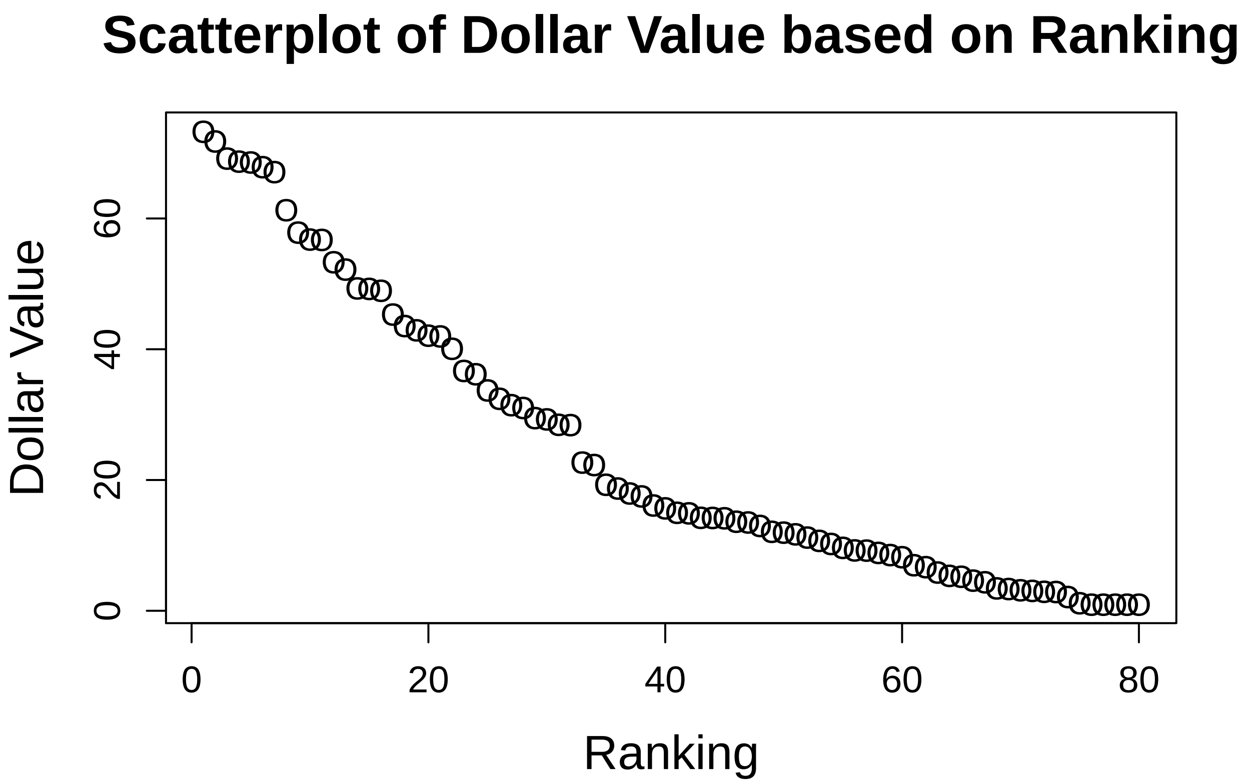
```
#I couldn't get this to print out. I created a print method for league
#class so now if I do print(l) it will shows the print method instead of
#just printing l but everything else works fine.
hist(l)
```



```
boxplot(l)
```

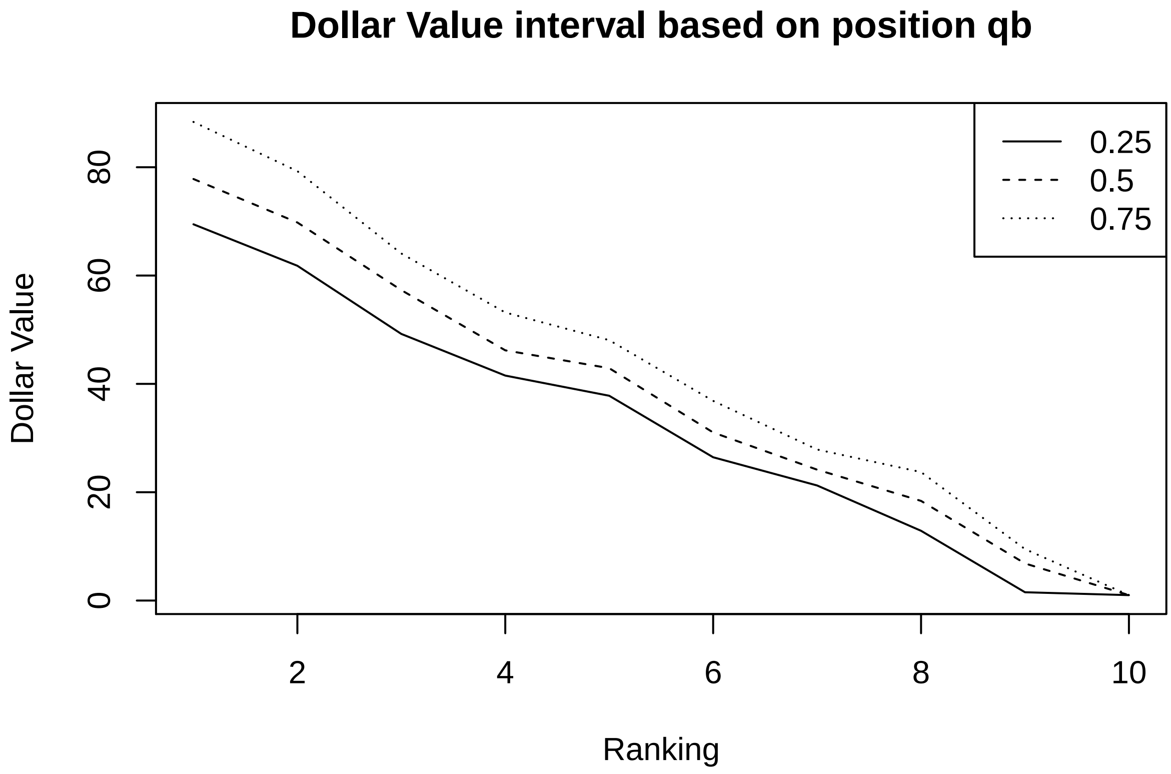

Boxplot of Player's position and Dollar Value
Dollar Value
0
20
40
60
k
qb
rb
te
wr
Position

```r
plot(l)
```

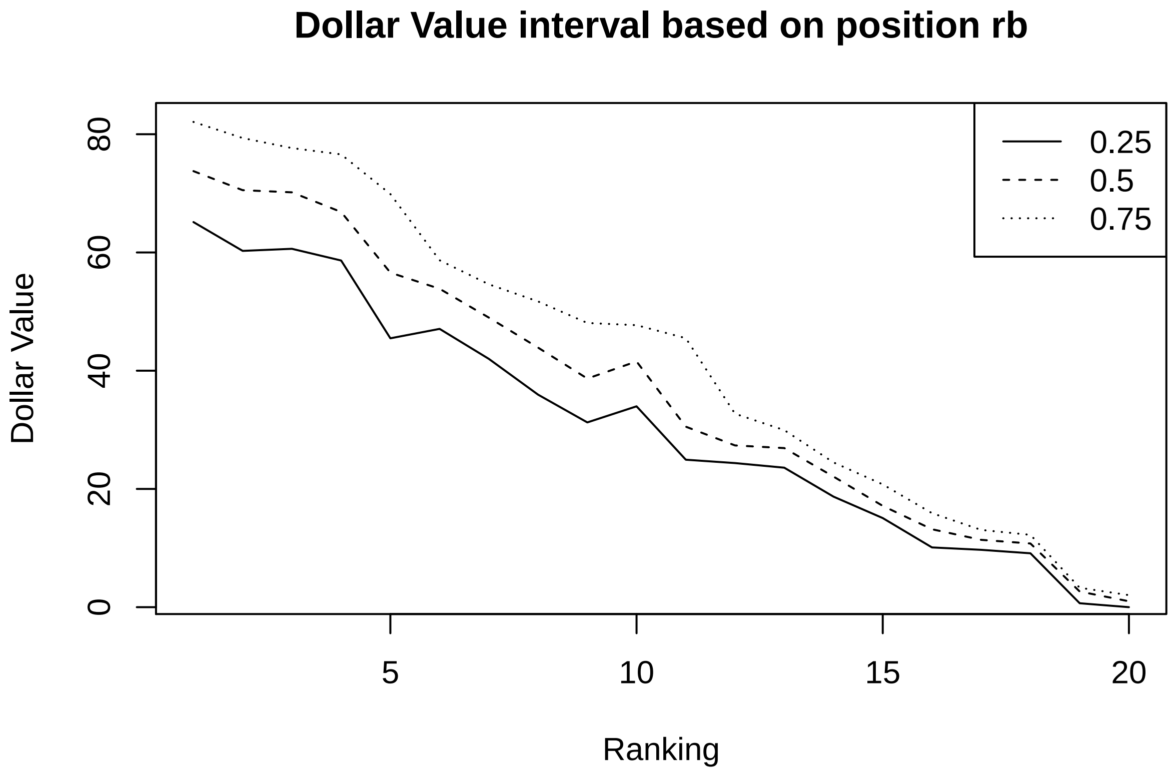


Test Q3

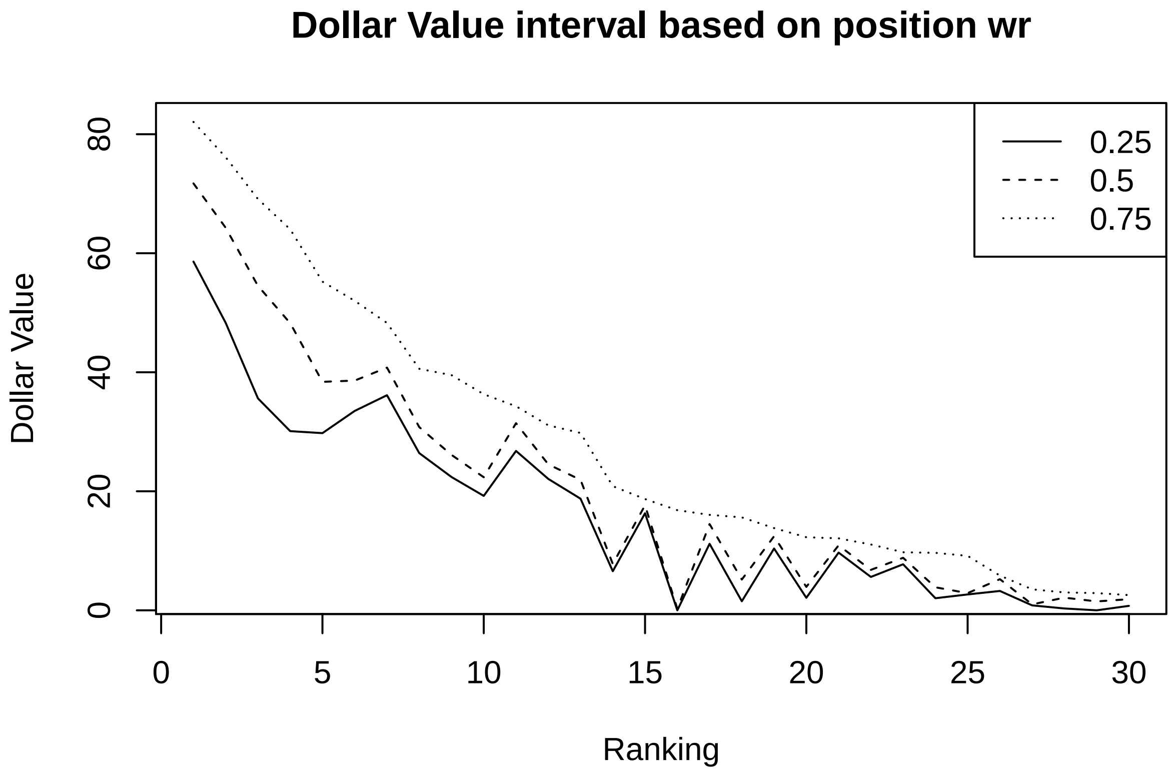
```
#hint, running 100 simulations roughly takes about 3 minutes.
l1 <- addNoise(l, noise, 500 ,seed=5)
ci <- conf.interval(l1)
plot(ci, 'qb')
```


Dollar Value interval based on position qb
Dollar Value
0
20
40
60
80
2
4
6
8
10
Ranking
0.25
0.5
0.75

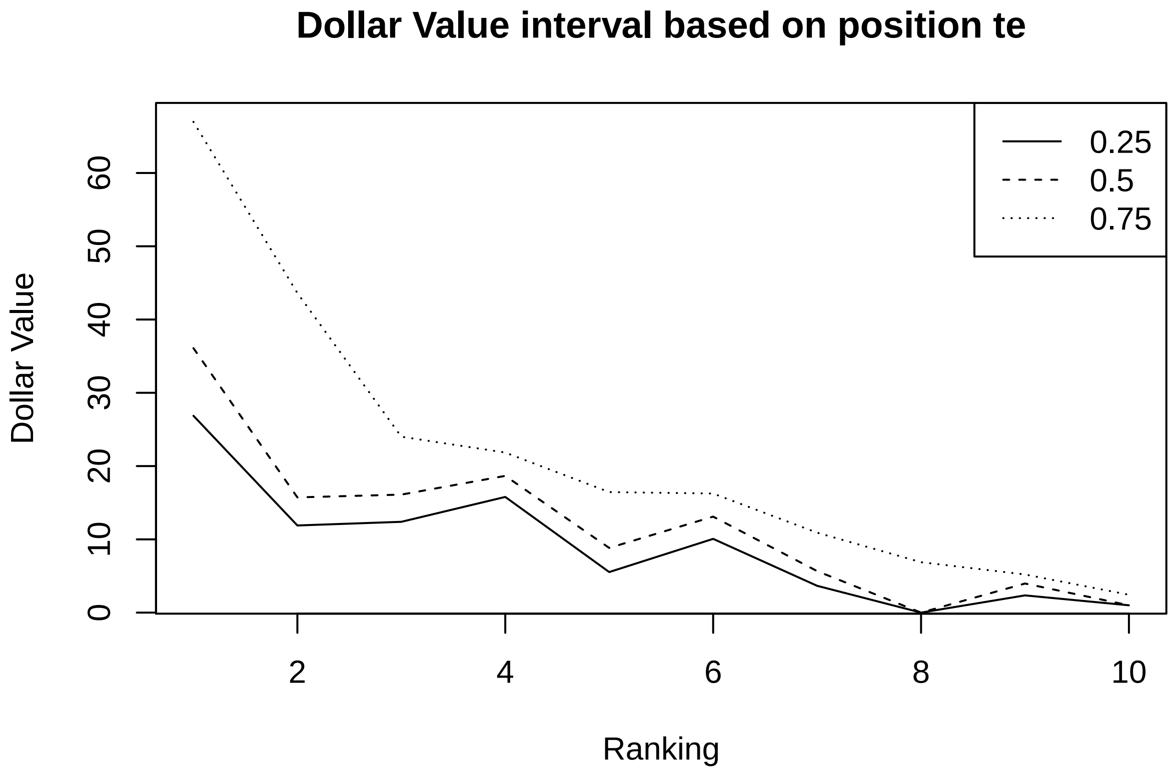
```r
plot(ci, 'rb')
```


Dollar Value interval based on position rb
Dollar Value
0
20
40
60
80
Ranking
5
10
15
20
0.25
0.5
0.75

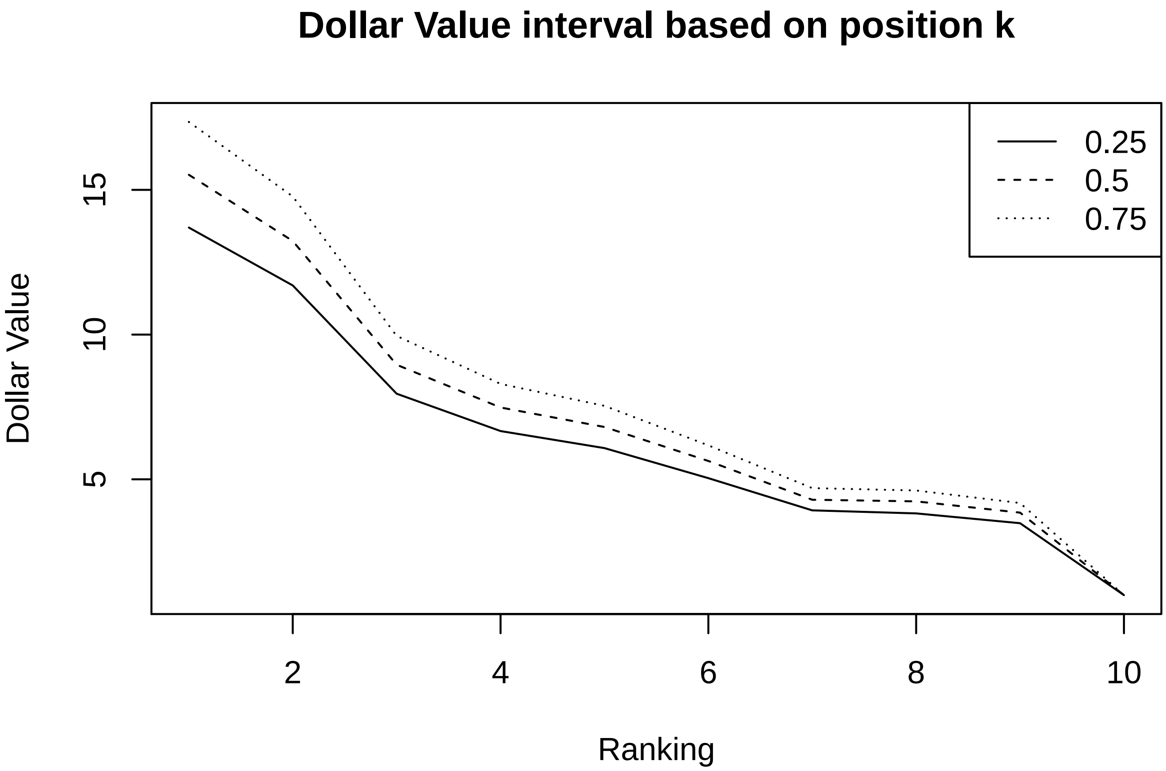
```
plot(ci, 'wr')
```


Dollar Value interval based on position wr
Dollar Value
Ranking
0
20
40
60
80
0
5
10
15
20
25
30
0.25
0.5
0.75

```r
plot(ci, 'te')
```


Dollar Value interval based on position te
Dollar Value
0
10
20
30
40
50
60
2
4
6
8
10
Ranking
0.25
0.5
0.75

```
plot(ci, 'k')
```


Dollar Value interval based on position k
Dollar Value
5
10
15
2
4
6
8
10
Ranking
0.25
0.5
0.75