- Rewrite the bootstrap program you wrote for assignment 3
- Where you are taking the mean, put in a user provided statistical function
- Everyplace you take a standard deviation, use a different bootstrap program to calculate the standard error of your statistic instead.
- So inside the bootstrap loop in the main program, you call a different bootstrap program to estimate variance
- Apply both the original bootstrap confidence estimate of the mean, and your new bootstrap using the median as the estimate of location to the data set special.sample in xy.pck.
- Provide the 95% confidence intervals.

95% confidence interval with mean

```
> my.bootstrapci(special.sample,10000,0.05)
$bootstrap.confidence.interval
    97.5%      2.5%
0.5234152 5.1906752

$normal.confidence.interval
[1] 0.5553075 5.2093548
```

95% confidence interval with median

```
> my.newbootci(special.sample, median, 10000, 0.05)
$bootstrap.confidence.interval
   97.5%     2.5%
1.797046 5.443089

$normal.confidence.interval
[1] 1.600637 5.506167
```

Code (with comments):

```
#original code for mean as estimate
my.bootstrapci<- function(vec0,nboot=10000,alpha=0.1) {
  #stores the length, mean and standard deviation for the sample vector
  n0<-length(vec0)
  mean0<-mean(vec0)
  sd0<-sqrt(var(vec0))
  #bootstrap vector is initialized
  bootvec<-NULL
  #the following bootstrap resampling loop runs nboot times
  for( i in 1:nboot){
    #vector is resampled with replacement
    vecb<-sample(vec0,replace=T)
    #standard deviation is calulated for the resampled vector
```

```
  sdb<-sqrt(var(vecb))
  # the following while loop insures that for small n if the resampled vector has a sd of 0 it is
resampled
  #until the resampled vector no longer has a std of 0 wherein it exits the while loop
  while(sdb==0){
    #vector is resampled with replacement
    vecb<-sample(vec0,replace=T)
    #standard deviation is recalulated for the resampled vector
    sdb<-sqrt(var(vecb))
  }
  #the mean for the resampled vector (with sd not equal to 0 ) is calculated and stored
  meanb<-mean(vecb)
  #the boot strap vector is appended with resampled vector distribution
  bootvec<-c(bootvec,(meanb-mean0)/(sdb/sqrt(n0)))
 }
 #the lower and upper quantiles are found for the bootstrapped distribution
 lq<-quantile(bootvec,alpha/2)
 uq<-quantile(bootvec,1-alpha/2)
 #the pivotal bootstrap confidence interval bounds are calculated
 LB<-mean0-(sd0/sqrt(n0))*uq
 UB<-mean0-(sd0/sqrt(n0))*lq
 #the normal theory confidence interval bounds are calculated
 NLB<-mean0-(sd0/sqrt(n0))*qt(1-alpha/2,n0-1)
 NUB<-mean0+(sd0/sqrt(n0))*qt(1-alpha/2,n0-1)
 #outputs both confidence intervals
 list(bootstrap.confidence.interval=c(LB,UB),normal.confidence.interval=c(NLB,NUB))
}

#helper function for finding variance of a statistic thats an estimate of location or etc like median
my.bootstrap.exp <-
 function(vec0,statfunc,nboot=100)
 {
  #initializes bootvec
  bootvec<-NULL
  # bootstrap for loop
  for( i in 1:nboot){
    #vector is resampled with replacement
    vecb<-sample(vec0,replace=T)
    #statistic of vecb is calculated
    statb<-statfunc(vecb)
    #the stat is appended to boot vec
    bootvec<-c(bootvec,statb)
  }
  #list of the mean of all stat values in bootvec and their variance
```

```
  list(bootmean=mean(bootvec),bootvar=var(bootvec))
 }

#new function to find ci with different stat as estimates
my.newbootci<- function(vec0,statfunc,nboot=10000,alpha=0.1) {
 #stores the length, stat and se for the vector
 n0<-length(vec0)
 stat0<-statfunc(vec0)
 statboot<-my.bootstrap.exp(vec0,statfunc,100)
 sd0<-sqrt(statboot$bootvar)
 #bootstrap vector is initialized
 bootvec<-NULL
 #the following bootstrap resampling loop runs nboot times
 for( i in 1:nboot){
  if((i/100)==floor(i/100)){
   print(i)
   # prints the simulation number by 10s
  }
  #my.bootstrap.exp is called and does the following:
  #vector is resampled with replacement
  #the stat funtions is appeneded into a bootvec
  #variance and mean is calculated
  statboot<-my.bootstrap.exp(vec0,statfunc,100)
  #se is calulated for the resampled vector
  sdb<-sqrt(statboot$bootvar)
  # the following while loop insures that for small n if the resampled vector has a sd of 0 it is
resampled until the resampled vector no longer has a std of 0 wherein it exits the while loop
  while(sdb==0){
   statboot<-my.bootstrap.exp(vec0,statfunc,100)
   sdb<-sqrt(statboot$bootvar)
  }
   #resampling with replacement
   vecb<-sample(vec0,replace=T)
   #stat calculated of resampled vec
   statb<-statfunc(vecb)
  #the boot strap vector is appended with resampled vector distribution
  bootvec<-c(bootvec,(statb-stat0)/(sdb))
}
 #the lower and upper quantiles are found for the bootstrapped distribution
 lq<-quantile(bootvec,alpha/2)
 uq<-quantile(bootvec,1-alpha/2)
 #the pivotal bootstrap confidence interval bounds are calculated
 LB<-stat0-(sd0)*uq
 UB<-stat0-(sd0)*lq
```

```
  #the normal theory confidence interval bounds are calculated
  NLB<-stat0-(sd0)*qt(1-alpha/2,n0-1)
  NUB<-stat0+(sd0)*qt(1-alpha/2,n0-1)

  #outputs both confidence intervals
  list(bootstrap.confidence.interval=c(LB,UB),normal.confidence.interval=c(NLB,NUB))
}
```