Question 1

(a)

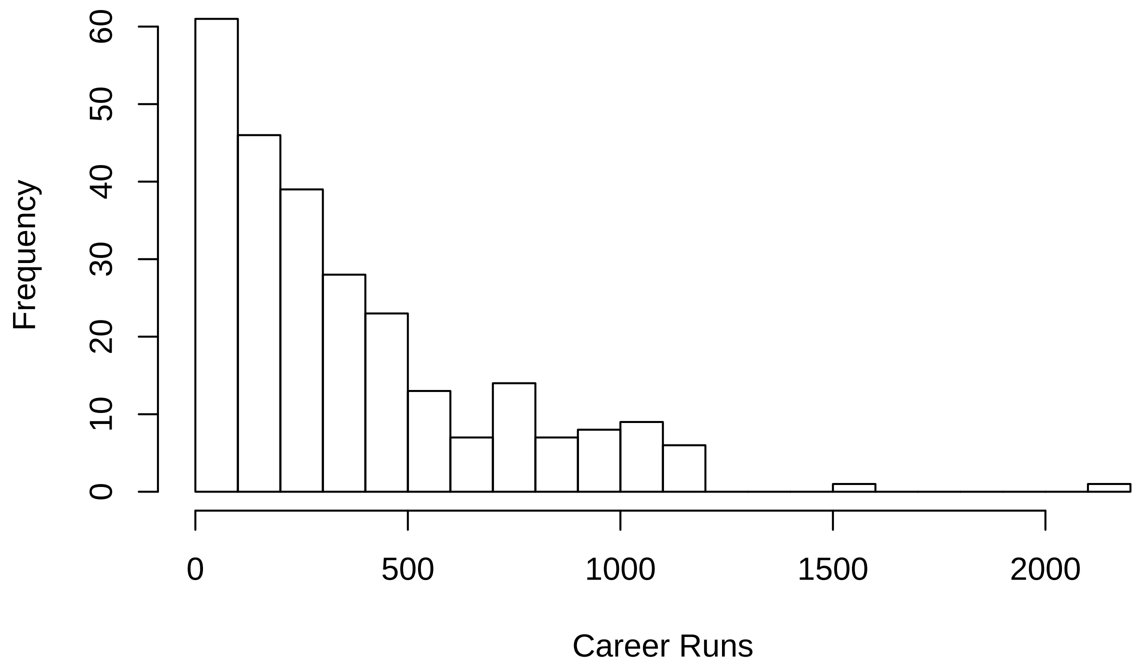
```
library(ISLR)
# We want the baseball CRuns data from this package
data("Hitters")
# IMPORTANT: We will work only with players having complete records:
C_Hitters <- na.omit(Hitters)
```

```
sdn <- function(z) {
  N = length(z)
  sd(z) * sqrt((N - 1)/N)
}
skew <- function(z) {
  3 * (mean(z) - median(z))/sdn(z)
}
```

```
powerfun <- function(x, alpha) {
  if (sum(x <= 0) > 1)
    stop("x must be positive")
  if (alpha == 0)
    log(x) else if (alpha > 0) {
    x^alpha
  } else -x^alpha
}
```

i.

```
hist( C_Hitters$CRuns, breaks="FD", xlab="Career Runs", main="" )
```



ii. The mean of Career Runs is:

```
mean(C_Hitters$CRuns)
```

```
## [1] 361.2205
```

The Pearson's second skewness coefficient is:

```
skew(C_Hitters$CRuns)
```

```
## [1] 1.009357
```

iii.

```
createSkewFunction <- function(y.pop) {
  skew2nd <- function(alpha) {
    skew( powerfun(x=y.pop, alpha) )
  }
}
```

The value of α which makes the skewness of the power transformed variable equal to zero is:

```r
CRunsskew = createSkewFunction(C_Hitters$CRuns)
alpha.star = uniroot( CRunsskew, interval=c(-2,2) )$root
alpha.star
```

```
## [1] 0.1964067
```

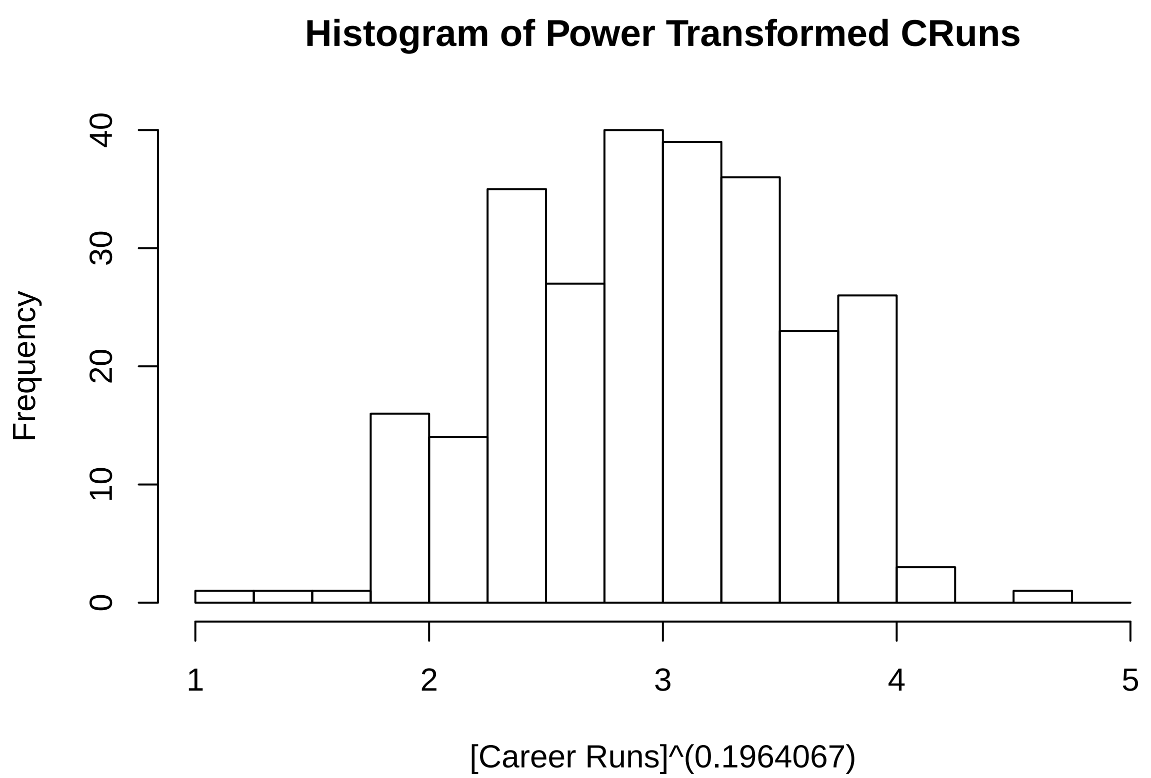
iv.

The skewness on the power-transformed CRuns variable is:

```r
skew(powerfun(C_Hitters$CRuns, 0.1964067))
```

```
## [1] 1.230268e-07
```

```r
hist( powerfun(C_Hitters$CRuns, 0.1964067), xlab="[Career Runs]^(0.1964067)",
      main="Histogram of Power Transformed CRuns", breaks=seq(1, 5, 0.25))
```



v.

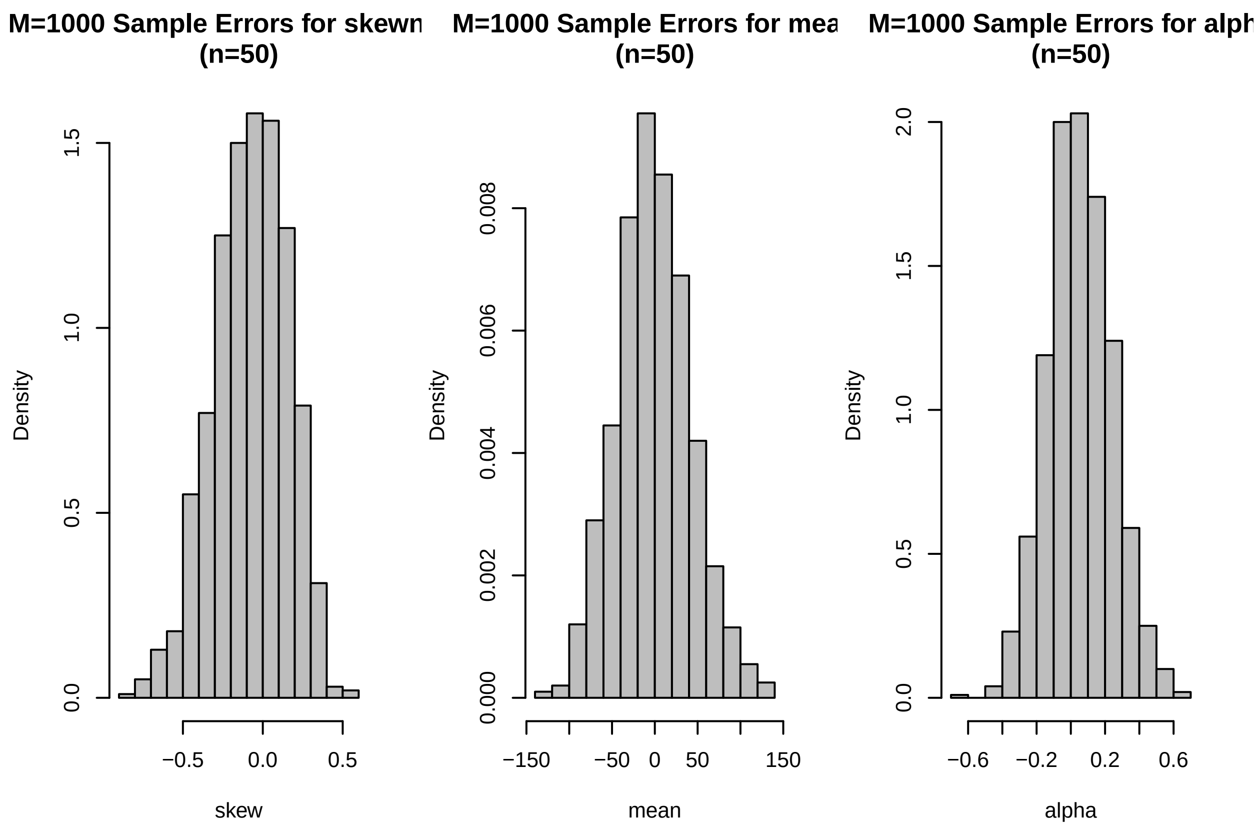
```r
attr3 <- function(y.pop) {
  fn.alpha <- uniroot(createSkewFunction(y.pop), interval=c(-2,2))$root
  fn.mean <- mean(y.pop)
```

```r
  fn.skew <- skew(y.pop)
  val <- cbind(fn.mean, fn.skew, fn.alpha)
  return(val)
}

attr3(C_Hitters$CRuns)
```

```
##       fn.mean  fn.skew  fn.alpha
## [1,] 361.2205 1.009357 0.1964067
```

(b)

```r
sim.attr3 <- function(pop=NULL, n=NULL, m=1000) {
  N = length(pop);
  set.seed(341)
  mean_f <- c();
  skew_f <- c();
  alpha_f <- c();
  for (i  in 1:m) {
    sam.values = pop[sample(N, n, replace=FALSE)]
    mean_f[i] = attr3(sam.values)[1]
    skew_f[i] = attr3(sam.values)[2]
    alpha_f[i] = attr3(sam.values)[3]
  }
  alpha.hat <- data.frame("mean" = mean_f, "skew" = skew_f, "alpha" = alpha_f)
  return(alpha.hat)
}

avesSamp <- sim.attr3(C_Hitters$CRuns, 50)

par(mfrow=c(1,3))
hist(avesSamp$skew - skew(C_Hitters$CRuns), prob=TRUE, xlab="skew", main="M=1000 Sample Errors for skew
hist(avesSamp$mean - mean(C_Hitters$CRuns), prob=TRUE, xlab="mean", main="M=1000 Sample Errors for mean
hist(avesSamp$alpha - alpha.star, prob=TRUE, xlab="alpha", main="M=1000 Sample Errors for alpha\n(n=50)
```



(c)

```
sam = c(220, 97, 256, 241, 137, 83, 140, 186, 34, 135, 50, 191, 213, 216,
58, 91, 244, 263, 240, 51, 258, 254, 224, 89, 62, 86, 247, 5, 166,
81, 61, 136, 217, 157, 207, 47, 124, 25, 260, 32, 160, 114, 246, 143,
57, 261, 70, 2, 110, 181)

# Values
samp <- c()
for (i in 1:50) {
  samp[i] = C_Hitters[sam[i],]$CRuns
}
```

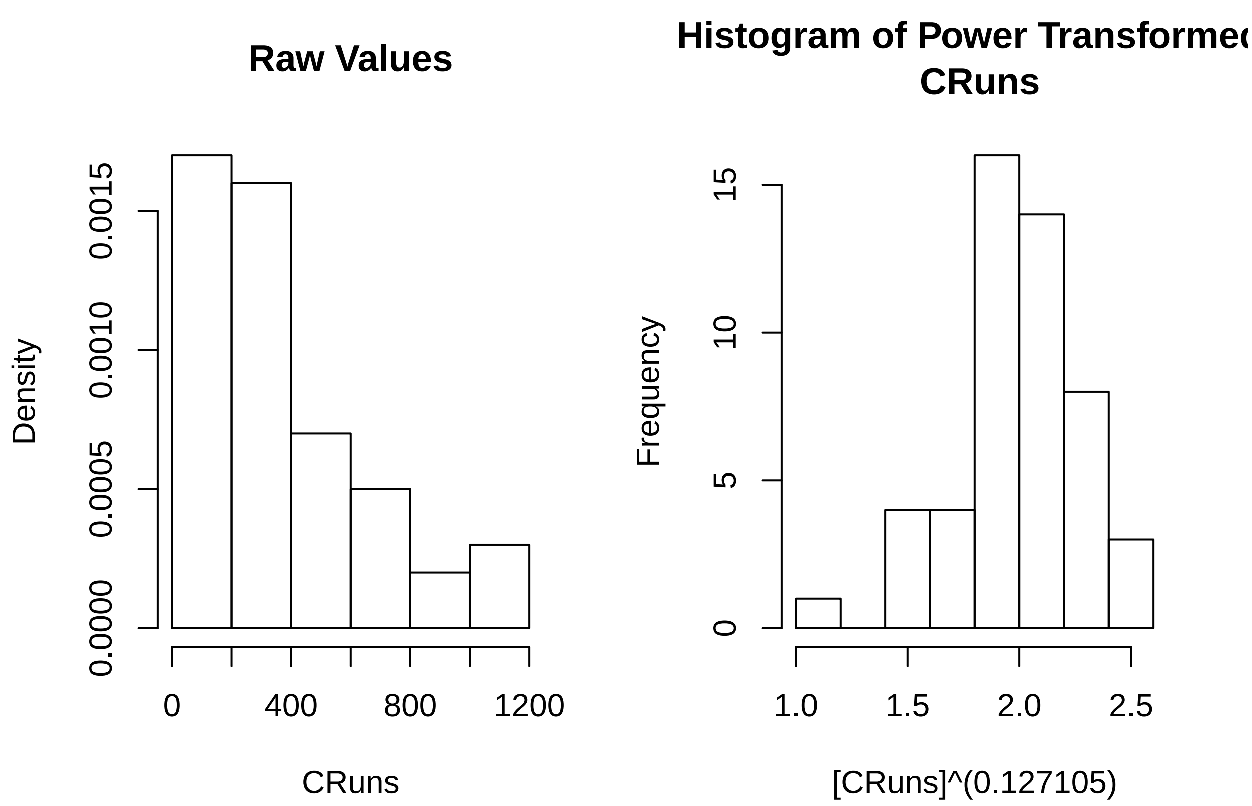
i. Three attributes of interest: mean, skewness and alpha -

```
attr3(samp)
```

```
##      fn.mean  fn.skew fn.alpha
## [1,]  357.42 1.164963 0.127105
```

ii.

```r
par(mfrow=c(1,2))
hist(samp, breaks="FD", prob=TRUE, xlab="CRuns", main="Raw Values")
hist(powerfun(samp, 0.127105), xlab="[CRuns]^(0.127105)",
main="Histogram of Power Transformed \n CRuns", breaks="FD")
```



iii.

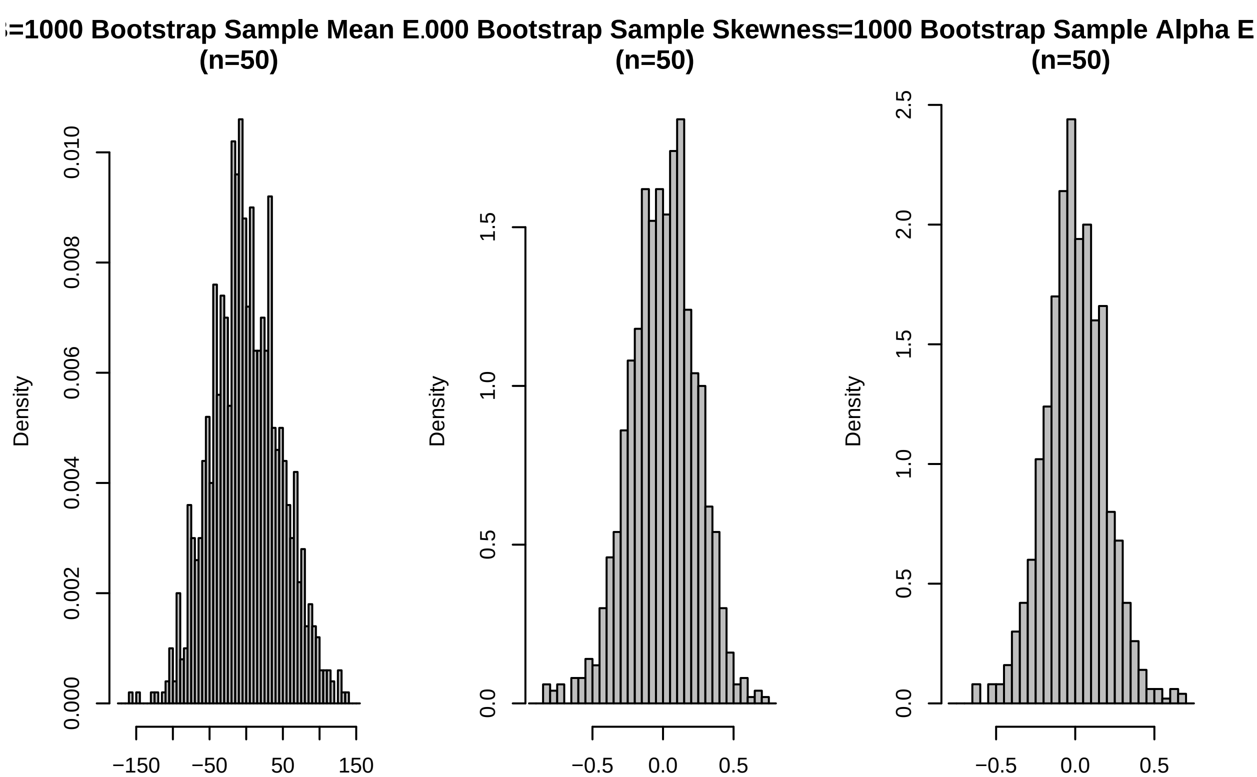
```r
popSize <- function(pop) {
  if (is.vector(pop))
  {if (is.logical(pop))
    sum(pop) else length(pop)}
  else nrow(pop)
}

getSample <- function(pop, size, replace=FALSE) {
  N <- popSize(pop)
  pop[sample(1:N, size, replace = replace)]
}

set.seed(341)
unitNum <- length(C_Hitters$CRuns)
n <- 50
Pstar <- getSample(1:unitNum, n, replace = FALSE)
B <- 1000
Sstar <- sapply(1:B, FUN = function(b) getSample(Pstar, n, replace = TRUE))
```

The mean, skewness and alpha value for each bootstrap sample

```
avesBootSampMean <- sapply(1:B, FUN = function(i) attr3(C_Hitters[Sstar[, i], "CRuns"])[1])
avesBootSampSkew <- sapply(1:B, FUN = function(i) attr3(C_Hitters[Sstar[, i], "CRuns"])[2])
avesBootSampAlpha <- sapply(1:B, FUN = function(i) attr3(C_Hitters[Sstar[, i], "CRuns"])[3])
```

Histograms of the bootstrap sample error for each attribute

```
par(mfrow=c(1,3))

range.avediffMean <- extendrange(c(avesSamp$mean - mean(avesSamp$mean), avesBootSampMean - mean(avesBoo
range.avediffSkew <- extendrange(c(avesSamp$skew - mean(avesSamp$skew), avesBootSampSkew - mean(avesBoo
range.avediffAlpha <- extendrange(c(avesSamp$alpha - mean(avesSamp$alpha), avesBootSampAlpha - mean(ave

hPopAvediffMean <- hist(range.avediffMean, breaks=50, plot = FALSE)
hPopAvediffSkew <- hist(range.avediffSkew, breaks=50, plot = FALSE)
hPopAvediffAlpha <- hist(range.avediffAlpha, breaks=50, plot = FALSE)

hist(avesBootSampMean - mean(avesBootSampMean), xlim = range.avediffMean, breaks = hPopAvediffMean$brea
     freq = FALSE, col = "grey", main="B=1000 Bootstrap Sample Mean Errors \n(n=50)", xlab = "")

hist(avesBootSampSkew - mean(avesBootSampSkew), xlim = range.avediffSkew, breaks = hPopAvediffSkew$brea
     freq = FALSE, col = "grey", main="B=1000 Bootstrap Sample Skewness Errors \n(n=50)", xlab = "")

hist(avesBootSampAlpha - mean(avesBootSampAlpha), xlim = range.avediffAlpha, breaks = hPopAvediffAlpha$
     freq = FALSE, col = "grey", main="B=1000 Bootstrap Sample Alpha Errors \n(n=50)", xlab = "")
```



iv.

The standard errors for each sample estimate

```
sdn <- function(y.pop) {
  N = length(y.pop)
  sqrt(var(y.pop) * (N-1)/(N))
}

standard.error <- cbind(mean = sdn(avesBootSampMean), skew = sdn(avesBootSampSkew), alpha = sdn(avesBoo
standard.error
```

```
##          mean      skew     alpha
## [1,] 46.33594 0.2344221 0.1902876
```

```
bootstrap_CI <- function(boot.estimates) {
  total.tab = matrix(0, 1, 2)
  rownames(total.tab) = c("Percentile Interval")
  colnames(total.tab) = c("Lower", "Upper")
  # percentile interval

  total.tab[1,] <- quantile(boot.estimates, probs=c(0.025, 0.975))
  total.tab
}
```

The 95% CI for mean of population

```
bootstrap_CI(avesBootSampMean)
```

```
##                       Lower   Upper
## Percentile Interval 280.174 461.368
```

The 95% CI for skewness of population

```
bootstrap_CI(avesBootSampSkew)
```

```
##                         Lower    Upper
## Percentile Interval 0.4202678 1.339707
```

The 95% CI for alpha of population

```
bootstrap_CI(avesBootSampAlpha)
```

```
##                          Lower     Upper
## Percentile Interval -0.1247403 0.6343719
```

(d)

An estimate of the coverage probability for mean estimates

```
n <- 50
CL.100.mean <- matrix(0,100,2)
for(x in 1:100) {
  Pstar <- getSample(1:unitNum, n, replace = FALSE)
  Sstar <- sapply(1:B, FUN = function(b) getSample(Pstar, n, replace = TRUE))
  avesBootSampMeanNew <- sapply(1:B, FUN = function(i) mean(C_Hitters[Sstar[, i], "CRuns"]))
  CL.100.mean[x,] = quantile(avesBootSampMeanNew, probs=c(0.025, 0.975))
}
```