Consider the following 25 data points (to save retyping, this list is also on the website as a text link)

8.26  6.33  10.4  5.27  5.35  5.61  6.12  6.19  5.2  7.01  8.74  7.78  7.02
6   6.5  5.8  5.12  7.41  6.52  6.21  12.28  5.6  5.38  6.6  8.74

Using this data

1: Using a randomization test, what is the \( p \) value for a test that the mean = 8?

Assuming the distribution is symmetric about the mean, we expected equal numbers above and below the mean, while we have 5 above the mean and 20 below. Recall that since we really don’t care which number is in excess, this is a two-sided test, and we need to compute the probability of seeing 5 (or fewer) above the mean and 5 (or fewer) below the mean. Under the null hypothesis, the expected number \( k \) is distributed as a binomial with \( n = 25 \) and \( p = 1/2 \). In R, \( \text{pbinom}(5, 25, 0.5) \) returns the probability that \( k \leq 5 \), or 0.002038658. Likewise, the chance of seeing 5 or fewer below the mean (i.e. \( k \geq 20 \)) is \( 1 - \text{pbinom}(20, 25, 0.5) \), or 0.0004552603. Hence, the probability of seeing something this extreme is 0.002493918, or 0.25%.

2: What is the jackknife estimate, its standard error, and the approximate confidence intervals for the estimated skew,

\[
\hat{S}_3 = \frac{1}{n} \sum_{i} (x_i - \bar{x})^3
\]

Using the R code from the R bootstrap notes, we first need to define the skew. Let \( x \) be the data vector, then define the skew with the function \( \text{SK} \),

```r
> SK <- function(x) sum((x-mean(x))^ 3) /length(x)
```

Here (from the notes) is the R code to bootstrap this data.

```r
jack <- numeric(length(x)-1)
pseudo <- numeric(length(x))
for (i in 1:length(x)) {
 for (j in 1:length(x))
 { if(j < i) jack[j] <- x[j] else if(j > i) jack[j-1] <- x[j]}
pseudo[i] <- length(x)*SK(x) -(length(x)-1)*SK(jack)
}
The jackknife estimator is the mean of the pseudovalues,

```r
> mean(pseudo)
[1] 8.735525
```

The standard error is the square root of the variance of the pseudovalues,

```r
> sqrt(var(pseudo)/length(x))
[1] 5.3032
```

The approximate 95% confidence interval is given by \( \text{mean(pseudo)} \pm t_{0.975,n-1} \sqrt{\text{var(pseudo)}/n} \)

Using R, the upper and lower limits become

```r
> mean(pseudo) + qt(0.975, length(x)-1)*sqrt(var(pseudo)/length(x))
[1] 19.68079
> mean(pseudo) - qt(0.975, length(x)-1)*sqrt(var(pseudo)/length(x))
[1] -2.209743
```
3: Using 1000 bootstrap samples, consider an estimate of the scaled kurtosis,

\[ \hat{K} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^4 - 3 \left( \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2 \right)^2 \]

Again, letting \( x \) be the data vector, then the estimated kurtosis is given by the function \( \text{kur} \).

> kur <- function(x) sum((x-mean(x))^4)/length(x) - 3*(sum((x-mean(x))^2)/length(x))^2

To generate 1000 bootstrap values the R code from the bootstrap notes is

> boot <- numeric(1000)
> for (i in 1:1000) boot[i] <- kur(sample(x,replace=T))

(a) Plot the distribution of bootstrap values

> hist(boot)
yields

The bootstrap estimator becomes

> mean(boot)
[1] 11.53792

(b) What is the estimated bias and the standard deviation for \( \hat{K} \)?

> bias <- mean(boot) - kur(x)
[1] -7.30799
> sqrt(var(boot))
[1] 13.27730

(c) Using the results for (a), what is an approximate 95% (normally-assumption) confidence interval for \( K \)?

Assuming normality, the approximate 95% confidence interval is given by

\[ \hat{K} \pm 1.96 \sqrt{\text{Var(bootstrap)}} \]

(or adjusting for the bias an lower and upper values of

> kur(x) - bias - 1.96*sqrt(var(boot))
[1] 0.1303970
> kur(x) - bias + 1.96*sqrt(var(boot))
[1] 52.1774

(d) Compute Efron’s 95% confidence limit for this data.

> quantile(boot,0.975)
[1] 36.45404

and

> quantile(boot,0.025)
[1] -8.53793

(e) Compute Hall’s 95% confidence limit for this data.

> 2*kur(x) - quantile(boot,0.025)
[1] 46.22974

and

> 2*kur(x) - quantile(boot,0.975)
[1] 1.237769

Fun R fact. If \( x \) is a vector of data, then the R command \( \text{sample}(x, \text{replace}=T) \) returns a sampling with replacement vector of the data in \( c \), i.e., a bootstrap sample.