18.S096 Problem Set 5 Spring 2018
Due Date: 4/3/2018
Where: On Stellar, prior to 11:59pm

Collaboration on homework is encouraged, but you will benefit from independent effort to solve the problems before discussing them with other people. **You must write your solution in your own words. List all your collaborators.**

Problem 1. Generating Pseudo-Random Laplace Random Variables

Let $X_1, X_2, \ldots X_n$ be a random sample from the $Laplace(\mu, b)$ distribution with density:

$$f(x \mid \mu, b) = \frac{1}{2b} e^{-\frac{|x-\mu|}{b}}, \; -\infty < x < \infty.$$

with two parameters:

μ (location)
b (scale)

1(a) Suppose two random variables X and U are independent with

$$\begin{aligned} X &\sim Exponential(rate = 1), \\ U: &\; P(U = +1) = P(U = -1) = 1/2 \end{aligned}$$

For constants

$$a : -\infty < a < +\infty, \qquad b : 0 < b + \infty,$$

define the random variable

$$Y = a + b * U * X$$

Prove that $Y \sim Laplace(a, b)$

Solution:

We compute the Moment Generating Function of Y and show that it equals the MGF of the $Laplace(a, b)$ distribution.

First we compute the MGF X:

$$\begin{aligned} M_X(t) &= E[e^{tX}] = \int_0^\infty e^{tx} f(x) dx \\ &= \int_0^\infty e^{tx} e^{-x} dx = \int_0^\infty e^{-(1-t)x dx} \\ &= (1-t)^{-1} \\ &\quad \text{(for } t < 1\text{)} \end{aligned}$$

$$
\begin{aligned}
M_Y(t) &= E[e^{tY}] = E[e^{t(a+bUX)]} \\
&= e^{ta} \times E[e^{tbUX}] \\
&= e^{ta} \times E_U[E_{X|U}(e^{tbUX})] \\
&= e^{ta} \times \Big[P(U=+1) \times E_{X|U=+1}(e^{tbUX}) + P(U=-1) \times E_{X|U=-1}(e^{tbUX})\Big] \\
&= e^{ta} \times \Big[.5E_X(e^{+tbX}) + .5E_X(e^{-tbX})\Big] \quad \text{(by independence)} \\
&= e^{ta} \times \Big[.5M_X(tb) + .5M_X(-tb)\Big] \\
&= e^{ta} \times \tfrac{1}{2}\Big[\tfrac{1}{1-bt} + \tfrac{1}{1+bt}\Big] \\
&= \tfrac{e^{ta}}{1-b^2t^2}
\end{aligned}
$$

This is the MGF of the $Laplace(a, b)$ distribution (see previoius problem set). By the uniqueness of MGFs $Y \sim Lapace(a, b)$.

1(b) Write an R functon $rlaplace(n, location, scale)$ to generate n pseudo-random $Laplace(a, b)$ random variates with arguments:

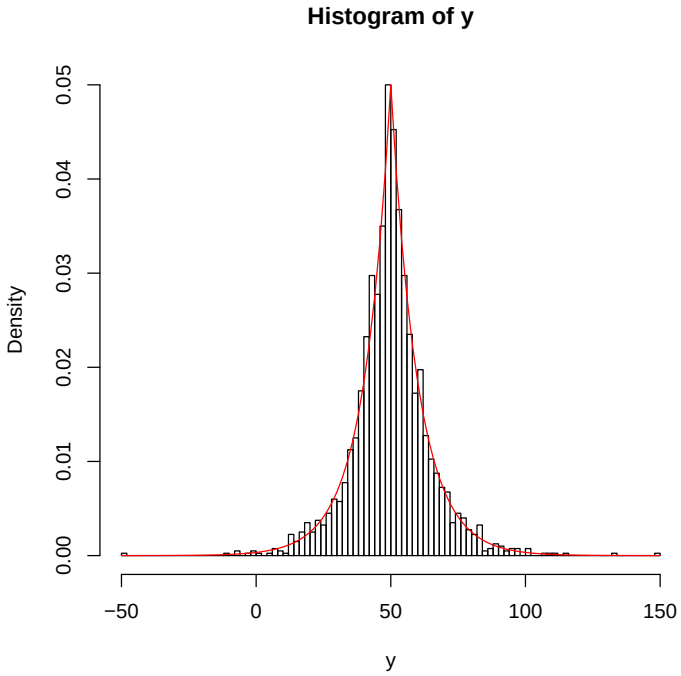
n (sample size), $location = a$ and $scale = b$.

```
> rlaplace<-function(n,location=0,scale=1){
+ n0=ifelse(length(n)>1, length(n), n)
+ x0=rexp(n0,rate=1)
+ u0=2*(rbinom(n0,size=1,prob=.5)-.5)
+ y0=location + scale*u0*x0
+ return(y0)
+ }
```

1(c) Test your function in (a) by generating a sample (y) of size $n = 2000$ from the $Laplace(location = 50, scale = 10)$

- Display the histogram of y and superimpose the *true* density function.
- Compute the sample mean and variance; compare these to the theoretical expectation/mean and variance of the *true* distribution.
- Compute the sample median and mean absolute deviation from the median; compare these to the theoretical values.

```
> y=rlaplace(2000,location=50,scale=10)
> hist(y,nclass=100,freq=FALSE)
> # From previous problem set:
> dlaplace<-function(x, location=50, scale=10){
+   dx=(0.5/scale)* exp(-abs(x-location)/scale)
+ }
> curve(dlaplace, add=TRUE, col="red")
>
```




```
> print(mean(y))

[1] 50.06388

> #   This value corresponds to a true mean of 50 (location).
>
> print(var(y))

[1] 205.4808

> #   The variance of a Laplace(a,b) distribution is
> #     2*b*b which in this case is 200=2*10*10
> #   The sample variance uses the denominator (n-1)
> #   which gives an unbiased estimate of the true variance
>
> print(median(y))

[1] 49.86969

> #   This value is the mle of the location parameter (50)
>
> print(mean(abs(y-median(y))))

[1] 9.997109
```

```
> #   This value is the mle of the scale parameter (10)

$location
[1] 0.0003495251

$scale
[1] 0.006028806
```

Generalized Likelihood Ratio Tests
(Background for Problem 2)

Suppose $\vec{y} = (y_1, \ldots, y_{n0})$ is an i.i.d. sample from a $Laplace(\theta)$ distribution with $\theta = (a, b)$, where a is the location parameter and b is the scale parameter.

- Consider testing the null hypothesis:

  $$H_0 : \theta = \theta_0 = (a_0, b_0) \in R \times R_+$$

  versus the alternative hypothesis:

  $$H_1 : \theta \neq \theta_0$$

  where $a_0 \in R$ is the location parameter and $b_0 \in R_+$ is the scale parameter under H_0.

- The null and alternative hypotheses are "nested":

  $$\Theta_0 = \{\theta,\ H_0\ is\ True\},$$

  $$\Theta_1 = \{\theta,\ H_1\ is\ True\},$$

  and $\Theta_0 \subseteq closure(\Theta_1)$.

- The generalized likelihood ratio test statistic of H_0 versus H_1 is given by:

  $$LRStat = 2 \times [\ell(\hat{\theta}) - \ell(\theta_0)]$$

  where $\hat{\theta}$ is the MLE of $\theta \in \Theta_1$ for the sample $\vec{y}$ and θ_0 is the value of θ if H_0 is true, and $\ell(\cdot)$ is the log-likelihood of the data:

  $$\ell(\theta) = log(p(\vec{y} \mid \theta)) = \sum_{i=1}^{n} log(p(y_i \mid \theta))$$

  (Note: the same logic applies for corresponding hypotheses about a distribution family with density/pmf function $p(y_i \mid \theta)$ indexed by the parameter θ. For example the distribution family could be Normal/Gaussian with unknown mean/sd or it could be gamma with unknown shape/scale.)

- The asymptotic distribution theory of generalized likelihood ratio tests states that if the nested null hypothesis H_0 is true, then the approximate distribution of $LRStat$ is a Chi-squared random variable with degrees of freedom d equal to 2 : the dimension of

  $$\Theta_1 = \{\theta,\ H_1\ is\ True\}$$

  minus the dimension of

  $$\Theta_0 = \{\theta,\ H_0\ is\ True\}.$$

- Conducting the generalized likelihood ratio test of H_0 versus H_1 consists of the following steps:

  - Compute the MLE $\hat{\theta} = \hat{\theta}(\vec{y})$.
  - Compute the test statistic

    $$\hat{T} = LRStat = 2 \times [\ell(\hat{\theta}) - \ell(\theta_0)]$$

– Compute the p-value of the test statistic:

$$\begin{aligned} p\text{-}value &= P(LRStat \geq \hat{T} \mid H_0) \\ &\approx 1 - pchisq(LRStat, df = 2) \end{aligned}$$

The second line corresponds to using the asymptotic distribution of the test statistic where $pchisq()$ is the cdf of the Chi-squared distribution with degrees of freedom $df = 2$.

The goodness-of-fit of the asymptotic distribution can be evaluated by constructing a Monte Carlo estimate of the true distribution of the test statistic assuming the null hypothesis is true.

– Decide to reject H_0 if the p-value is sufficiently small (close to 0).

(If the asymptotic distribution is a poor fit, compute p-values using the Monte Carlo fit to the true distribution.)

Problem 2: Monte Carlo Study of Likelihood Ratio Test

Conduct a Monte Carlo study of the distribution of the Likelihood Ratio Test statistic and its approximate p-value for the Laplace Distribution Null and Alternative. For the Null Hypothesis assume that the location and scale parameters of the Laplace distribution are the same as the Laplace MLEs for the S&P 500 Returns data:

```
> # install.packages("zoo")
> library(zoo)
> SP500<-read.zoo(file="SP500.csv")
> y<-diff(log(SP500))
> # with y equal to the daily log returns of S&P 500 index
> n0=length(y)        #  sample size
> location.mle=median(y)
> scale.mle=mean(abs(y-location.mle))
> cat("\n ",n0, location.mle,scale.mle,"\n")

  439 0.0003495251 0.006028806
```

2(a) For each of $ntrials = 2000$,

- Generate $\vec{y}^*$, an i.i.d. Laplace sample, with
  $n0 = 439$ observations, and
  $\theta_0 = (a_0, b_0)$ with
  $a_0 = 0.00035$
  $b_0 = 0.006029$.
  Use the R function $rlaplace()$ created in problem 1.
- Compute $\hat{\theta}^*$, the MLE of θ given $\vec{y}^*$
- Compute the Generalized likelihood ratio test statistic
  $LRStat = 2[\ell(\hat{\theta}^*) - \ell(\theta_0)]$
  Note: For each trial, store the statistic $LRStat$ as well as $\ell(\hat{\theta}^*)$ and $\ell(\theta_0)$.

2(b) Plot a histogram of the $ntrials = 2000$ values of the test statistic.

Superpose the density of a Chi-squared distribution with 2 degrees of freedom. Is the density a good fit to the distribution?

2(c) Compute the approximate p-values of the test statistic $LRStat$: (using the R function $pchisq()$ and plot their histogram.

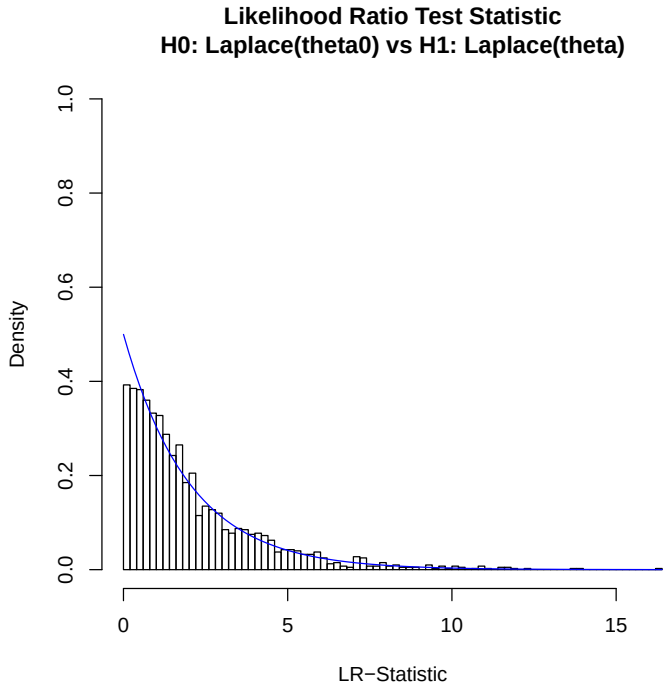
Superpose the density of the null distribution (a uniform(0,1) distribution) for the p-values. Is this density a good fit to the distribution?

2(a)

```
> library("MASS")
> ntrials=2000
> n0=length(y)
> location0=param.mle$location
> scale0=param.mle$scale
> #rlaplace
> mat.loglikes.laplace<-matrix(NA,nrow=ntrials, ncol=4)
> set.seed(1)
> for (itrial in 1:ntrials){
+   #itrial=1
+   y0=rlaplace(n0, location=location0, scale=scale0)
+   y0.location.mle=median(y0)
+   y0.scale.mle=mean(abs(y0 - y0.location.mle))
+   y0.laplace.loglik=sum(log(dlaplace(y0,
+     location=y0.location.mle, scale=y0.scale.mle)))
+
+     mat.loglikes.laplace[itrial,1]<-y0.laplace.loglik
+ # Null log likelihood
+     mat.loglikes.laplace[itrial,2]<-sum(log(
+     dlaplace(y0,location=location0,scale=scale0)))
+ }

> # The likelihood ratio test statistic is
> # twice the difference in the log-likelihood:
> LRStat=2*(mat.loglikes.laplace[,1]-mat.loglikes.laplace[,2])
> summary(LRStat)

     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
 0.001394  0.653800  1.444000  2.063000  2.824000 16.250000

> # Note: LRStat should be non-negative.
> #     A previous r script used fitdistr()
> #     compute mles and loglikelihood
> #     Negative values of LRStat resulted due to
> #     numerical error in maximizing the log-likelihood #
```

2(b)

```
> hist(LRStat,  nclass=100,
+      freq=FALSE,
+      xlab="LR-Statistic", ylim=c(0,1.),
+      main=paste(
+         c("Likelihood Ratio Test Statistic\n",
+           "H0: Laplace(theta0) vs H1: Laplace(theta)"),
+         collapse="")
+ )
> dchisq.2=function(x){dchisq(x,df=2)}
> curve(dchisq.2,add=TRUE,col='blue', from=.001)
```

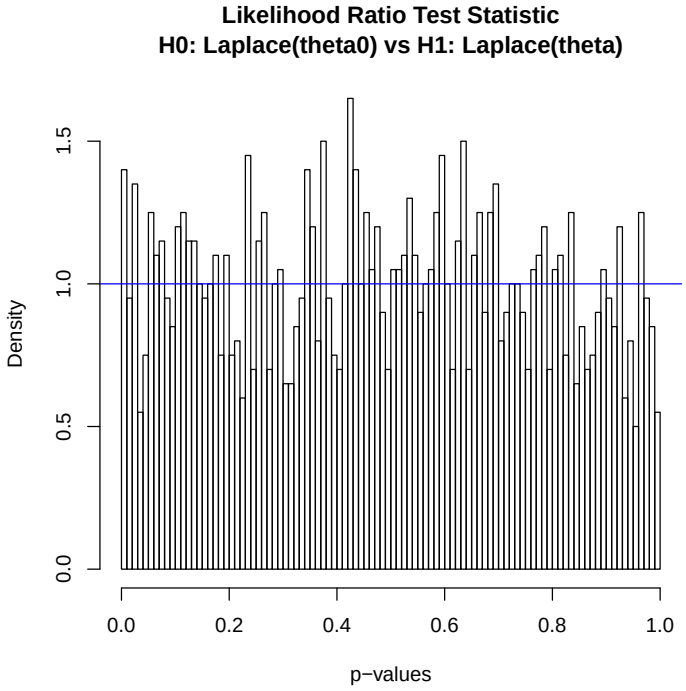


Note how closely the distribution of the LR-statistic follows the Chi-Square distribution with 2 degrees of freedom (which is the $Exponential(1)$).

2(c)

```
>   hist(1-pchisq(LRStat, df=2), nclass=100,freq=FALSE,
+      xlab="p-values",
+      main=paste(
+         c("Likelihood Ratio Test Statistic\n",
+           "H0: Laplace(theta0) vs H1: Laplace(theta)"),
+         collapse="")
```

```
+ )
> abline(h=1.,col='blue')
>
```



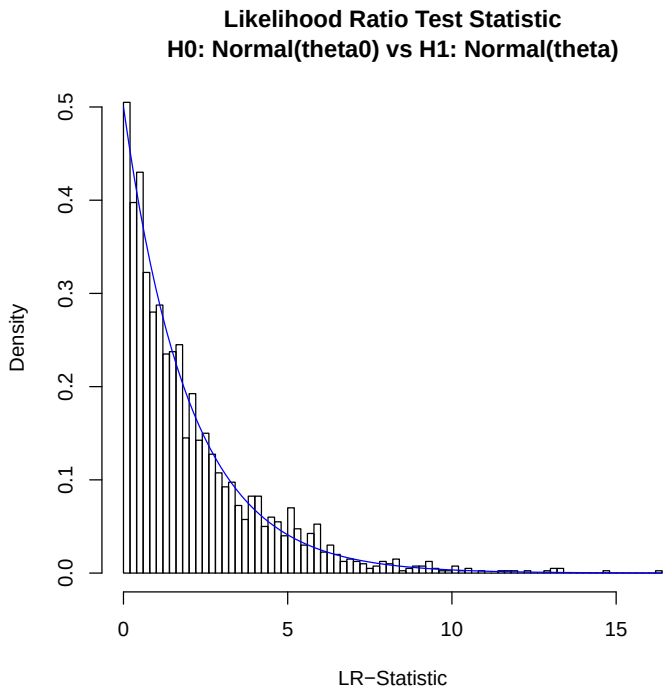
Note how closely the distribution of the approximate p-values follows the Uniform(0,1) distribution.

Problem 3. Monte Carlo Study of Likelihood Ratio Test (Part II).

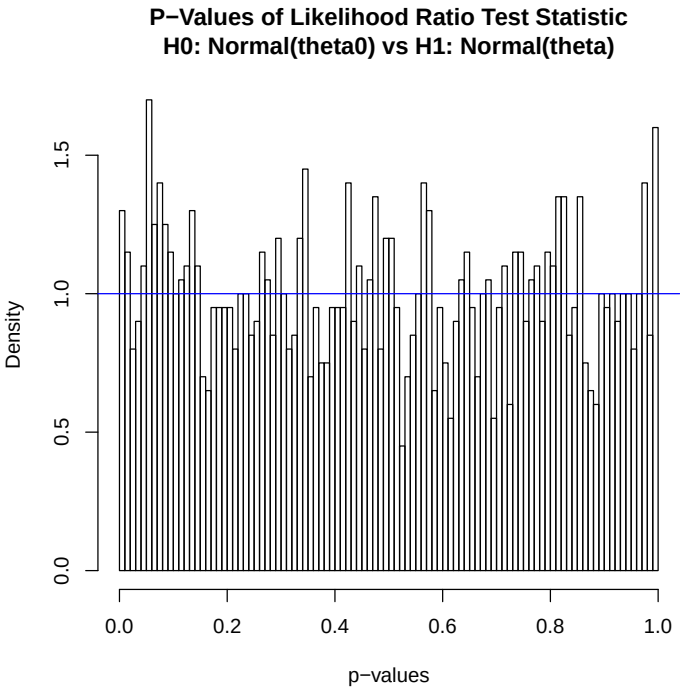
Repeat the Monte Carlo simulation of problem 2 except assume the family of distributions is Normal rather than Laplace. For the Null Hypothesis assume that the mean and standard deviation are the same as the Normal MLEs for the S&P 500 Returns data:

```
> # install.packages("zoo")
> library(zoo)
> SP500<-read.zoo(file="SP500.csv")
> y<-diff(log(SP500))
> # with y equal to the daily log returns of S&P 500 index
> n0=length(y)        #  sample size
> mean.mle=mean(y)
> sd.mle=sqrt((n0-1)*var(y)/n0)
> cat("\n ",n0, mean.mle,sd.mle,"\n")

  439 4.361316e-05 0.00840262

> library("MASS")
> set.seed(1)
> ntrials=2000
> n0=length(y)
> mean.mle=mean(y)
> sd.mle=sqrt((n0-1)*var(y)/n0)
> mat.loglikes.normal<-matrix(NA,nrow=ntrials, ncol=2)
> for (itrial in 1:ntrials){
+   #itrial=1
+   y0=rnorm(n0, mean=mean.mle,sd=sd.mle)
+   y0.mean.mle=mean(y0)
+   y0.var.mle=mean((y0-y0.mean.mle)^2)
+   y0.normal.loglik=sum(log(dnorm(y0,mean=y0.mean.mle,
+                                  sd=sqrt(y0.var.mle))))
+
+   mat.loglikes.normal[itrial,1]<-y0.normal.loglik
+   mat.loglikes.normal[itrial,2]<-sum(log(
+     dnorm(y0,mean=mean.mle,sd=sd.mle)))
+ }

> # The likelihood ratio test statistic is
> # twice the difference in the log-likelihood:
> LRStat=2*(mat.loglikes.normal[,1]-mat.loglikes.normal[,2])
> summary(LRStat)

     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
 0.000179  0.558900  1.437000  2.090000  2.888000 16.350000

> # Note: LRStat should be non-negative.
>
```

```
> hist(LRStat,  nclass=100,
+      freq=FALSE,
+      xlab="LR-Statistic",
+      main=paste(
+         c("Likelihood Ratio Test Statistic\n",
+           "H0: Normal(theta0) vs H1: Normal(theta)"),
+         collapse="")
+ )
> dchisq.2=function(x){dchisq(x,df=2)}
> curve(dchisq.2,add=TRUE,col='blue', from=.001)
>
>
```




```
>   hist(1-pchisq(LRStat, df=2), nclass=100,freq=FALSE,
+      xlab="p-values",
+      main=paste(
+         c("P-Values of Likelihood Ratio Test Statistic\n",
+           "H0: Normal(theta0) vs H1: Normal(theta)"),
+         collapse="")
+ )
> abline(h=1.,col='blue')
```



Note that the sampling distribution of LR statistic follows the Chi-square (df=2) distribution very closely. When the null hypothesis is true, computing p-values using this distribution are good approximations to the true p-values.

Problem 4. Monte Carlo Study of Generalized Likelihood Ratio Tests of non-nested Hypotheses.

4(a) Extend the Monte-Carlo Study of Problem 2 which assumes that the sample data follow a Laplace distribution.

For each trial of the Monte Carlo study, in addition to fitting the Laplace distribution by MLE,

- Fit the Normal distribution by maximum likelihood to the same data: $\hat{\phi} = (\hat{\mu}, \hat{\sigma})$
- Compute: $\ell(\hat{\phi})$ (the Normal log-likelihood)
- Compute : $LRStat^* = 2 \times [\ell(\hat{\phi}) - \ell(\hat{\theta})]$
  where the log likelihoods are computed according to the Laplace distribution for $\hat{\theta}$ (for the null hypothesis) and according to the Gaussian/normal distribution for $\hat{\phi}$ (for the alternative hypothesis).

4(b) Plot the histogram of the Monte Carlo values of $LRStat^*$

Note: Since the two hypotheses are not nested, the distribution of the test statistic is generally unknown. The Monte Carlo study provides an approximation to the distribution of the test statistic given the null hypothesis that the sample is from a Laplace distribution (with unknown location and scale). Strong evidence against the null hypothesis occurs when the test statistic $LRStat^*$ is extremely high relative to this Monte Carlo distribution.

4(c) For the S&P 500 log daily returns, compute $LRStat^*$.

How extreme is this statistic relative to the Monte Carlo distribution of $LRStat^*$ if the Laplace-Distribution null hypothesis is true, i.e., compute an approximate p-value for the statistic.

4(d) Extend the Monte-Carlo Study of Problem 3 which assumes that the sample data follow a Normal distribution the same way as parts (a), (b), and (c) with the roles of Normal vs Laplace distributions reversed. (I.e., construct a Monte Carlo distribution of the likelihood ratio test of Laplace (H_1) versus Normal (H_0) distributions when the Normal distribution (H_0) is true. Compute an approximate p-value for the test of the null hypothesis that the data are a realization of a Normal sample).

First, for part (a), where null distribution is Laplace:

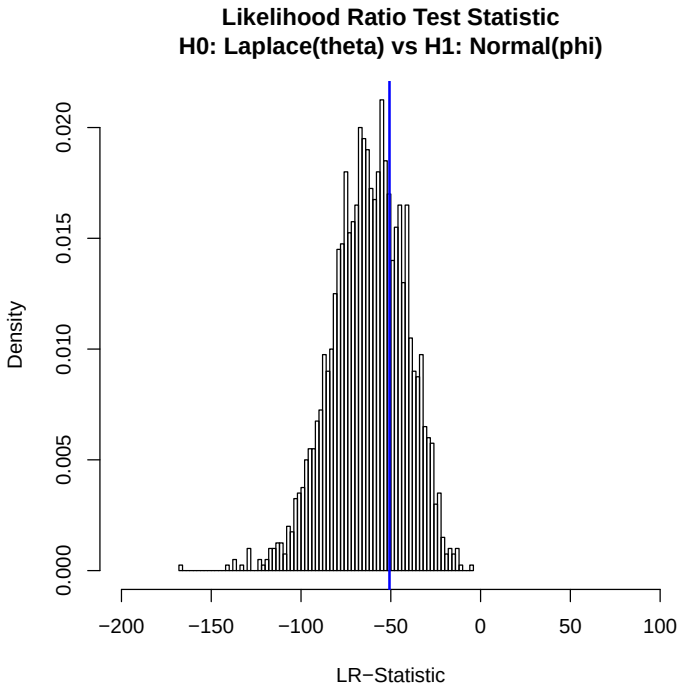
```
> ntrials=2000
> n0=length(y)
> location0=location.mle
> scale0=scale.mle
> mat.loglikes.laplace<-matrix(NA,nrow=ntrials, ncol=4)
```

```
> set.seed(1)
> for (itrial in 1:ntrials){
+   #itrial=1
+   y0=rlaplace(n0, location=location0, scale=scale0)
+
+   # loglik of ml fit to laplace
+   y0.location.mle=median(y0)
+   y0.scale.mle=mean(abs(y0 - y0.location.mle))
+   y0.laplace.loglik=sum(log(dlaplace(y0,
+     location=y0.location.mle, scale=y0.scale.mle)))
+
+     mat.loglikes.laplace[itrial,1]<-y0.laplace.loglik
+ # Null log likelihood
+     mat.loglikes.laplace[itrial,2]<-sum(log(
+     dlaplace(y0,location=location0,scale=scale0)))
+ # Normal log likelihood:
+
+   y0.mean.mle=mean(y0)
+   y0.var.mle=mean((y0-y0.mean.mle)^2)
+   y0.normal.loglik=sum(log(dnorm(y0,mean=y0.mean.mle,
+                                  sd=sqrt(y0.var.mle))))
+
+   mat.loglikes.laplace[itrial,3]<-y0.normal.loglik
+
+     }
```

4 (b) and (c)

```
> ##  Extending the study to LRStatStar
> # The likelihood ratio test statistic is
> # twice the difference in the log-likelihood:
> LRStatStar=(-2)*(mat.loglikes.laplace[,1]-mat.loglikes.laplace[,3])
> summary(LRStatStar)

    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-166.400  -75.750  -61.680  -62.350  -47.520   -4.273

> # Note: For nested hypotheses LRStat should be non-negative.
> #     but for non-nested the statistic can be negative
> hist(LRStatStar,  nclass=100, xlim=c(-200,100),
+      freq=FALSE,
+      xlab="LR-Statistic",      main=paste(
+         c("Likelihood Ratio Test Statistic\n",
+           "H0: Laplace(theta) vs H1: Normal(phi)"),
+         collapse="")
+ )
> #dchisq.2=function(x){dchisq(x,df=2)}
```

```
> #curve(dchisq.2,add=TRUE,col='blue', from=.001)
>
>
> # Realized value of LRStatstar for S&P 500 returns y
>
> #   Compute log likelihoods for Laplace and Normal:
> #     Laplace log likelihood of sample
>   y.laplace.loglik=sum(log(dlaplace(y,
+     location=location.mle, scale=scale.mle)))
> #     Normal log likelihood of sample
>   y.mean.mle=mean(y)
>   y.var.mle=mean((y-y.mean.mle)^2)
>   y.normal.loglik=sum(log(dnorm(y,mean=y.mean.mle,
+                                 sd=sqrt(y.var.mle))))
> y.LRStatStar<-(2)*(y.normal.loglik - y.laplace.loglik)
> abline(v=y.LRStatStar,col='blue',lwd=2)
> print(y.LRStatStar)

[1] -50.73609

> # Approximate p-value computed from simulation
> pval.LaplaceVsNormal=mean(LRStatStar >=y.LRStatStar)
> print(pval.LaplaceVsNormal)

[1] 0.303

>
```

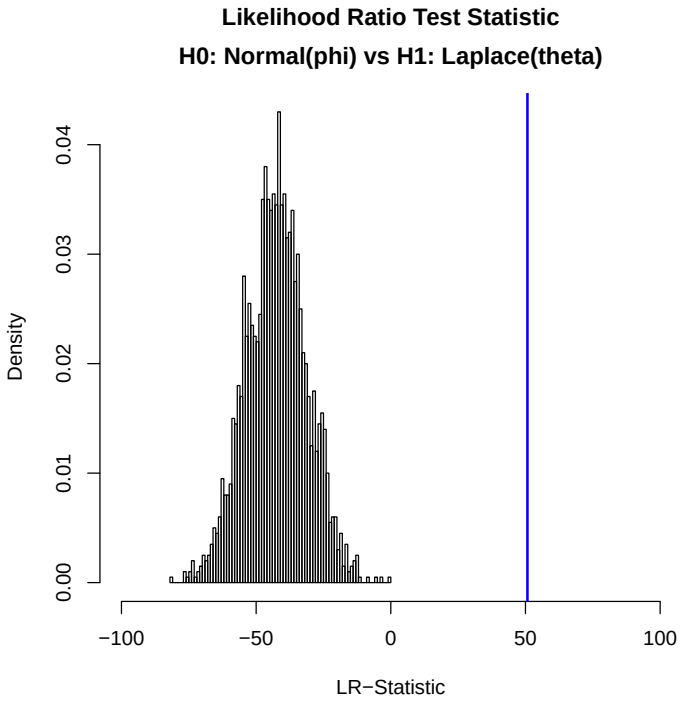


Note that the realized value of the test statistic is not extreme under Null hypothesis that the data are a sample from the Laplace distribution. The approximae p-value is 0.303.

4 (d)

```
> library("MASS")
> set.seed(1)
> ntrials=2000
> n0=length(y)
> mean.mle=mean(y)
> sd.mle=sqrt((n0-1)*var(y)/n0)
> mat.loglikes.normal<-matrix(NA,nrow=ntrials, ncol=3)
> for (itrial in 1:ntrials){
+   #itrial=1
+   y0=rnorm(n0, mean=mean.mle,sd=sd.mle)
+   y0.mean.mle=mean(y0)
+   y0.var.mle=mean((y0-y0.mean.mle)^2)
+   y0.normal.loglik=sum(log(dnorm(y0,mean=y0.mean.mle,
+                                  sd=sqrt(y0.var.mle))))
+
+   mat.loglikes.normal[itrial,1]<-y0.normal.loglik
+   mat.loglikes.normal[itrial,2]<-sum(log(
+     dnorm(y0,mean=mean.mle,sd=sd.mle)))
```

```
+
+   y0.location.mle=median(y0)
+   y0.scale.mle=mean(abs(y0 - y0.location.mle))
+   y0.laplace.loglik=sum(log(dlaplace(y0,
+     location=y0.location.mle, scale=y0.scale.mle)))
+   mat.loglikes.normal[itrial,3]<-y0.laplace.loglik
+ }

> ##  Extending the study to LRStatStar
> # The likelihood ratio test statistic is
> # twice the difference in the log-likelihood:
> LRStatStar.b=(-2)*(mat.loglikes.normal[,1]-mat.loglikes.normal[,3])
> summary(LRStatStar.b)

    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-81.2700 -50.1600 -42.3900 -42.4900 -35.0300  -0.3456

> # Note: For non-nested hypotheses, LRStat can be negative
> hist(LRStatStar.b,  nclass=100,
+      freq=FALSE,xlim=c(-100,100),
+      xlab="LR-Statistic",      main=paste(
+         c("Likelihood Ratio Test Statistic\n",
+           "H0: Normal(phi) vs H1: Laplace(theta)")))
> y.LRStatStar.b<-(-2)*(y.normal.loglik - y.laplace.loglik)
> abline(v=y.LRStatStar.b,col='blue',lwd=2)
> print(y.LRStatStar.b)

[1] 50.73609

> # Approximate p-value computed from simulation
> pval.NormalVsLaplace=mean(LRStatStar.b >=y.LRStatStar.b)
> print(pval.NormalVsLaplace)

[1] 0
```



Note that the realized value of the test statistic is extreme, giving significantly strong evidence against the Null hypothesis that the data are a sample from the Normal distribution. The approximate p-value is 0.

Problem 5. Permutation Tests

Consider gene expression data on leukemia patients discussed in Efron and Hastie (2016) *Computer Age Statistical Inference* (https://web.stanford.edu/ hastie/CASI/). The data consist of:

- 72 patients
  - 47 with Acute Lymphoblastic Leukemia (ALL)
  - 25 with Acute Myeloid Leukemia (AML)
- 7128 genes

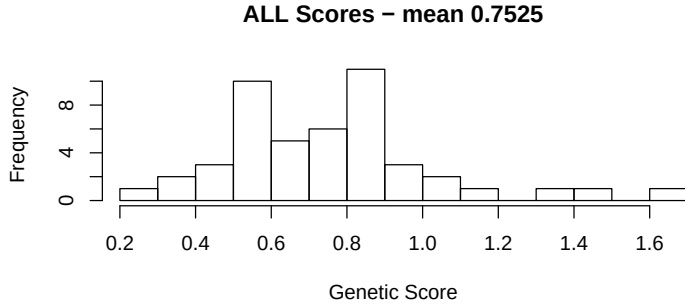
The following code reads the data into R and defines relevant variables corresponding to gene 136 and conducts a two-sample t-test comparing the mean gene expression in the two groups(assuming equal within-group variances).

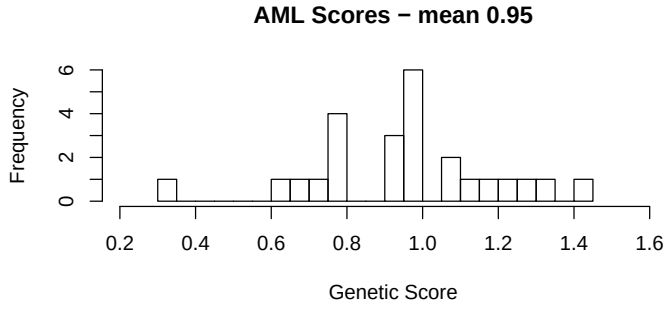
```
> leukemia_big <- read.csv(file="EfronData/leukemia_big.csv")
> dim(leukemia_big)

[1] 7128   72

> names(leukemia_big)

 [1] "ALL"    "ALL.1"  "ALL.2"  "ALL.3"  "ALL.4"  "ALL.5"  "ALL.6"  "ALL.7"
 [9] "ALL.8"  "ALL.9"  "ALL.10" "ALL.11" "ALL.12" "ALL.13" "ALL.14" "ALL.15"
[17] "ALL.16" "ALL.17" "ALL.18" "ALL.19" "AML"    "AML.1"  "AML.2"  "AML.3"
[25] "AML.4"  "AML.5"  "AML.6"  "AML.7"  "AML.8"  "AML.9"  "AML.10" "AML.11"
[33] "AML.12" "AML.13" "ALL.20" "ALL.21" "ALL.22" "ALL.23" "ALL.24" "ALL.25"
[41] "ALL.26" "ALL.27" "ALL.28" "ALL.29" "ALL.30" "ALL.31" "ALL.32" "ALL.33"
[49] "ALL.34" "ALL.35" "ALL.36" "ALL.37" "ALL.38" "ALL.39" "ALL.40" "ALL.41"
[57] "ALL.42" "ALL.43" "ALL.44" "ALL.45" "ALL.46" "AML.14" "AML.15" "AML.16"
[65] "AML.17" "AML.18" "AML.19" "AML.20" "AML.21" "AML.22" "AML.23" "AML.24"

> leukemiaType<-as.factor(substring(names(leukemia_big),
+                                   first=1,last=3))
> genej<-as.numeric(t(leukemia_big[j<-136,]))
> genej.byType<-split(genej,leukemiaType)
```

5(a) Compare the genetic scores on gene 136 for the two groups by plotting histograms for each group ALL and AML, one above the other (make the x axis of each histogram have the same range of values to facilitate the comparison).

```
> par(mfcol=c(2,1))
> xlim00=c(min(genej), max(genej))
> hist(genej.byType[[1]], xlab="Genetic Score",xlim=xlim00,
+      main=paste(c(names(genej.byType)[1]," Scores - mean ",
+       as.character(round(mean(genej.byType[[1]]),digits=4))),
```

```
+       collapse=""),nclass=20)
> hist(genej.byType[[2]], xlab="Genetic Score",xlim=xlim00,
+      main=paste(c(names(genej.byType)[2]," Scores - mean ",
+       as.character(round(mean(genej.byType[[2]]),digits=4))),
+       collapse=""),nclass=20)
>
```





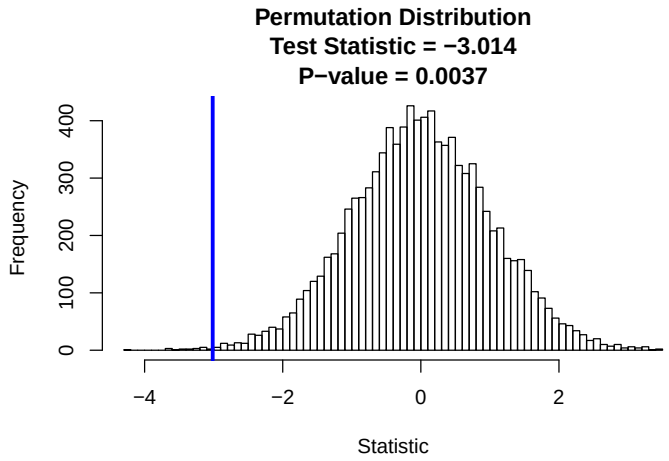
5(b) The following R command conducts the two-sample t-test:

```
> genej.ttest<-t.test(genej.byType[[1]], genej.byType[[2]],
+                     var.equal = TRUE)
> genej.ttest

        Two Sample t-test

data:  genej.byType[[1]] and genej.byType[[2]]
t = -3.014, df = 70, p-value = 0.003589
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.32817995 -0.06680742
sample estimates:
mean of x mean of y
0.7524794 0.9499731
```

```
> names(genej.ttest)

[1] "statistic"   "parameter"   "p.value"     "conf.int"    "estimate"
[6] "null.value"  "alternative" "method"      "data.name"
```

Approximate the permutation distribution of the two-sample t-test statistic by writing an R script to conduct a Monte Carlo study with $10,000$ Monte-Carlo replicates of the permutation distribution.

```
> x=genej.byType[[1]] ; y=genej.byType[[2]]
> leukemiaType0=as.numeric(leukemiaType)
> library(perm) # necessary for function: calcPvalsMC()
> mytwosample.exact.mc<-function (scores, group,
+                                 alternative = "two.sided",
+                                 nmc = 10^4 -1,
+                                 seed = 1234321, digits = 12,
+                                 p.conf.level = 0.99, setSEED = TRUE){
+     t0.test<-t.test(scores~group,var.equal=TRUE)
+     t0 <-t0.test$statistic
+     N <- nmc
+     if (setSEED)
+         set.seed(seed)
+     ti <- rep(NA, N)
+     for (i in 1:N) {
+       ti.test<-t.test(scores~sample(group), var.equal=TRUE)
+         ti[i] <- ti.test$statistic
+         }
+     out <- calcPvalsMC(ti, t0, digits, alternative, FALSE, p.conf.level)
+     result<-list(t0=t0, ti=ti, out=out)
+     result
+   }

> par(mfcol=c(1,1))
> genej.permTest<-mytwosample.exact.mc(genej,leukemiaType0)
> genej.pvalue<-mean(abs(genej.permTest$ti) >=abs(genej.permTest$t0))
> hist(genej.permTest$ti,nclass=100,xlab="Statistic",
+      main=paste(c("Permutation Distribution\n",
+                   "Test Statistic = ", as.character(round(genej.permTest$t0,digits=4)
+                   "P-value = ",as.character(round(genej.pvalue,digits=4))), collapse=
> abline(v=genej.permTest$t0,col='blue',lwd=3)
> save(file="prob5.RData",list=c("genej.permTest","genej.pvalue"))
```



5(c) Use the R library "coin" and the R function *independent_test*() to test whether there are significant differences between the ALL and AML groups for gene 136.

```
>   library(coin)
> independence_test(genej ~ leukemiaType)

        Asymptotic General Independence Test

data:  genej by leukemiaType (ALL, AML)
Z = -2.8558, p-value = 0.004293
alternative hypothesis: two.sided
```

5(d) Use the R library "perm" and the R function $permTS()$ with $method =$ "*exact.mc*" to conduct the the two-sample permutation test for gene 136.

```
>   library(perm)
>   permTS(genej ~ leukemiaType, method="exact.mc")

        Exact Permutation Test Estimated by Monte Carlo

data:  genej by leukemiaType
p-value = 0.002
alternative hypothesis: true mean leukemiaType=ALL - mean leukemiaType=AML is not equal
sample estimates:
mean leukemiaType=ALL - mean leukemiaType=AML
                                    -0.1974937

p-value estimated from 999 Monte Carlo replications
99 percent confidence interval on p-value:
 0.00000000 0.01057916
```

5(e) Comment on the consistency of results from parts (b), (c), and (d), explaining any differences in the definitions of the test statistics.

The permutation tests in the *coin* and *perm* packages are based on rank-sum test statistics as opposed to the two-sample t test statistic. While different, the results are consistent giving very similar p-values.

5(f) Compute the two-sample t-test statistics for all 7128 genes in the Leukemia data set.

- Display a histogram of all the test statistic (using a density scale, i.e., "freq=FALSE").
- Draw a vertical line $abline(v =)$ at the t statistic value for Gene 136.
- Superpose on the histogram the density of the null distribution for the two-sample t statistic.
- Suppose π_0 is the prior probability of a gene having no difference between the two subject groups. What values of π_0 are consistent with the data?

Leukemia Data: 7128 t-tests

```
> mat.leukemia.ttests<-matrix(NA,nrow=nrow(leukemia_big),ncol=2)
> leukemiaType<-as.factor(substring(names(leukemia_big),
+                                   first=1,last=3))
> for (j in 1:nrow(leukemia_big)){
+   genej<-as.numeric(t(leukemia_big[j,]))
+   genej.byType<-split(genej,leukemiaType)
+   genej.ttest<-t.test(genej.byType[[1]], genej.byType[[2]],
+                       var.equal = TRUE)
+ mat.leukemia.ttests[j,1]<-genej.ttest$statistic
+ mat.leukemia.ttests[j,2]<-genej.ttest$p.value  }
> sum(mat.leukemia.ttests[,2] <= mat.leukemia.ttests[136,2])

[1] 981

>

> par(mfcol=c(1,1))
> hist(mat.leukemia.ttests[,1], nclass = 100,
+      freq=FALSE,
+      main="Two-Sample t Statistics\n(For each of 7128 genes)",xlab="t Statistic", yli
> abline(v=mat.leukemia.ttests[136,1], col='blue', lwd=3)
> legendtext0=paste(c("Gene 136: t=",as.character(round(mat.leukemia.ttests[136,1], dig
> tdensity.df70<-function(x){dt(x,df=70)}
> curve(tdensity.df70,add=TRUE,col='blue')
> legend(x=-14,y=.25, legend=legendtext0, col='blue', lwd=3,cex=0.6)
> # To determine what values of pi0 are consistent with the
```

```
> # data, note that the marginal distribution of the
> # t statistic has density equal to
> #     pi0 x null density plus (1-pi0) x alternate density.
> # Since the density of the histogram peaks at 0.25, it
> # follows that values of pi0 are less than or equal to
> #
> pi0=.25/dt(0,df=70)
> pi0

[1] 0.6288991

> # We add a curve corresponding to the largest value of pi0
> # which does not over-estimate the density at $t=0.$
>
> pi0tdensity.df70<-function(x){pi0*dt(x,df=70)}
> curve(pi0tdensity.df70,add=TRUE,col='red')
>
```