Permutation Tests

Nathaniel E. Helwig

Assistant Professor of Psychology and Statistics
University of Minnesota (Twin Cities)

Updated 04-Jan-2017

Copyright

Outline of Notes

1) Introduction to Permutations

- What is a permutation?
- Permutations in R
- Inference via permutations

2) One-Sample Permutations

- Overview
- Monte Carlo procedure
- Examples

3) Two-Sample Permutations

- Overview
- Monte Carlo procedure
- Examples

4) Correlation Permutations

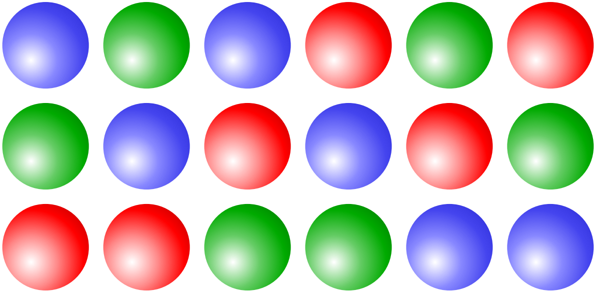
- Overview
- Monte Carlo procedure
- Examples

Introduction to Permutations

Permutation Defined

The word permutation refers to the arrangement of a set of objects into some specified order.

Each column is one possible permutation of the three colors:



From `https://upload.wikimedia.org/wikipedia/commons/4/4c/Permutations_RGB.svg`

Permuting a Data Vector

Given a data vector of length $n = 3$, there are 6 possible permutations:

- $\mathbf{x}_{(1)} = (x_1, x_2, x_3)$
- $\mathbf{x}_{(2)} = (x_1, x_3, x_2)$
- $\mathbf{x}_{(3)} = (x_2, x_1, x_3)$
- $\mathbf{x}_{(4)} = (x_2, x_3, x_1)$
- $\mathbf{x}_{(5)} = (x_3, x_1, x_2)$
- $\mathbf{x}_{(6)} = (x_3, x_2, x_1)$

More generally, there are $n!$ permutations for a vector of length n.

Generate All Possible Permutations

```
permutations <- function(n){
  if(n==1){
    return(matrix(1))
  } else {
    sp <- permutations(n-1)
    p <- nrow(sp)
    A <- matrix(nrow=n*p,ncol=n)
    for(i in 1:n){
      A[(i-1)*p+1:p,] <- cbind(i,sp+(sp>=i))
    }
    return(A)
  }
}
```

From `http://stackoverflow.com/questions/11095992/generating-all-distinct-permutations-of-a-list-in-r`

All Possible Permutations Examples

```
> permutations(2)
     [,1] [,2]
[1,]    1    2
[2,]    2    1
> permutations(3)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    1    3    2
[3,]    2    1    3
[4,]    2    3    1
[5,]    3    1    2
[6,]    3    2    1
```

Generate a Random Permutation

```
> set.seed(1)
> n = 5
> x = seq(0,20,length=n)
> x
[1]  0  5 10 15 20
> x[sample.int(n)]
[1]  5 20 15 10  0
> x[sample.int(n)]
[1] 20 15  5 10  0
```

Note that the `sample.int` function returns a random permutation of the integers 1 to `n`, where `n` is the user-specified input.

Why are Permutations Useful for Statistics?

Classic statistical paradigm is:

- collect some data
- form null hypothesis H_0
- design test statistic
- derive sampling distribution of test statistic under H_0

In many cases, the null hypothesis is the nil hypothesis, i.e., no effect.

Under the nil hypothesis, all possible outcomes (permutations) are equally likely, so permutations relate to sampling distributions.

Achieved Significance Level

Suppose we have some test statistic $\hat{\theta} = s(\mathbf{x})$, and suppose that larger values of $\hat{\theta}$ provide more evidence against H_0.

Given $\hat{\theta}$, the achieved significance level (ASL) of our test is

$$\mathrm{ASL} = P(\hat{\theta}^* \geq \hat{\theta} \mid H_0 \text{ true })$$

which is the probability of observing a test statistic as or more extreme than $\hat{\theta}$ under the assumption that H_0 is true.

- Can you think of another name for ASL?

One-Sample Permutation Tests

One-Sample (or Paired Sample) Problem

For the one-sample location problem, we have *n* observations

- $X_1, \ldots, X_n \overset{\text{iid}}{\sim} F$ if one-sample situation
- $Z_1, \ldots, Z_n \overset{\text{iid}}{\sim} F$ with $Z_j = X_j - Y_j$ if paired-sample situation

We want to make inferences about location of the data

- Let F denote the population distribution
- Let θ denote the median of F
- Null hypothesis is $H_0 : \theta = \theta_0$
- Three possible alternatives: $H_1 : \theta < \theta_0$, $H_1 : \theta > \theta_0$, $H_1 : \theta \neq \theta_0$,

Permutation Vector and Lemma (1-Sample)

Let $\mathbf{g} = (g_1, g_2, \ldots, g_n)$ denote the permutation vector denoting which observations are above θ_0 ($g_i = 1$) and which are below θ_0 ($g_i = -1$).

- There are 2^n different possible $\mathbf{g}$ vectors (each g_i can be 1 or -1)
- If $H_0 : \theta = \theta_0$ is true, then $P(X < \theta_0) = 0.5$ by definition

Permutation Lemma:
Under $H_0 : \theta = \theta_0$, the vector $\mathbf{g}$ has probability $1/2^n$ of equaling each of the 2^n different possible outcomes

Permutation Achieved Significance Level (1-Sample)

The permutation ASL is the permutation probability that $\hat{\theta}^*$ exceeds $\hat{\theta}$:

$$\mathrm{ASL}_{\mathrm{perm}} = \#\{|\hat{\theta}_b^*| \geq |\hat{\theta}|\}/2^n$$

where $\{\hat{\theta}_b^*\}_{b=1}^{2^n}$ is the set of all possible test statistics under H_0.

Note that the above is for the two-sided alternative $H_0 : \theta \neq \theta_0$

- For $H_0 : \theta < \theta_0$, we have $\mathrm{ASL}_{\mathrm{perm}} = \#\{\hat{\theta}_b^* \leq \hat{\theta}\}/2^n$
- For $H_0 : \theta > \theta_0$, we have $\mathrm{ASL}_{\mathrm{perm}} = \#\{\hat{\theta}_b^* \geq \hat{\theta}\}/2^n$

Problem: when 2^n is large, forming $\hat{\theta}_b^*$ for all 2^n possible **g** vectors is computationally expensive.

- Solution: use a Monte Carlo approach!

One-Sample Permutation Test (Monte Carlo)

Procedure for approximating $\mathrm{ASL}_{\mathrm{perm}}$ using Monte Carlo approach:

1. Randomly sample B permutation vectors $\mathbf{g}_1^*, \ldots, \mathbf{g}_B^*$
2. Evaluate the permutation replication $\hat{\theta}_b^* = s(\mathbf{g}_b^*, \mathbf{x})$ where $\mathbf{x} = (x_1, \ldots, x_n)$ is the observed vector of data
3. Approximate $\mathrm{ASL}_{\mathrm{perm}}$ using

$$\widehat{\mathrm{ASL}}_{\mathrm{perm}} = \#\{|\hat{\theta}_b^*| \geq |\hat{\theta}|\}/B$$

This assumes that the statistic $\hat{\theta} = s(\mathbf{g}, \mathbf{x})$ is designed such that larger absolute values provide more evidence against H_0.

Some Possible Statistics

We want to design some statistic $\hat{\theta}$ such that larger absolute values provide more evidence against H_0.

If we assume that F is symmetric around θ_0, then...

- θ_0 is both the median and mean of F under H_0
- Statistic 1: $\hat{\theta} = n^{-1} \sum_{i=1}^{n} |x_i - \theta_0| g_i = \bar{x}$
- Statistic 2: $\hat{\theta} = \sum_{i=1}^{n} R_i 1_{\{g_i=1\}} - \frac{n(n+1)}{4}$ where $R_i = \text{rank}(|x_i - \theta_0|)$

If we drop the symmetry assumption θ_0, then...

- Statistic 3: $\hat{\theta} = \sum_{i=1}^{n} 1_{\{g_i=1\}} - \frac{n}{2}$

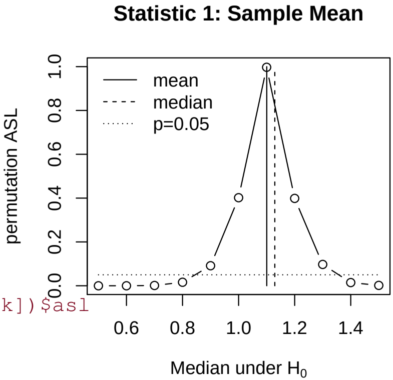
One-Sample Permutation Test: R Function

An R function for performing one-sample permutation tests:

```
perm1samp <- function(x,myfun=mean,mu=0,nsamp=10000,
                      alternative=c("two.sided","less","greater")){
  x = x - mu
  n = length(x)
  theta.hat = myfun(x)
  gmat = replicate(nsamp,sample(x=c(1,-1),size=n,replace=TRUE))
  theta.mc = apply(gmat*abs(x),2,myfun)
  if(alternative[1]=="less"){
    aslperm = sum(theta.mc <= theta.hat) / nsamp
  } else if(alternative[1]=="greater"){
    aslperm = sum(theta.mc >= theta.hat) / nsamp
  } else{
    aslperm = sum(abs(theta.mc) >= abs(theta.hat)) / nsamp
  }
  list(theta.hat=theta.hat,theta.mc=theta.mc,asl=aslperm)
}
```

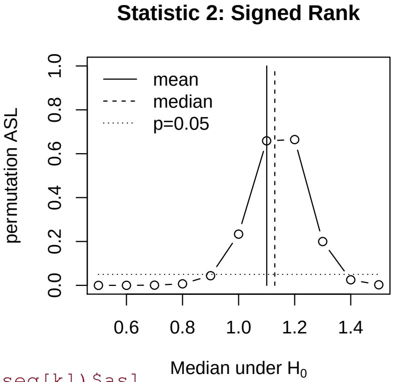
Example using Statistic 1 (sample mean)

```
> set.seed(1)
> n = 50
> x = rnorm(n,mean=1)
> mean(x)
[1] 1.100448
> se = (sd(x)/sqrt(n))
> cv = qt(.975,df=n-1)
> c(mean(x)-cv*se, mean(x)+cv*se)
[1] 0.8641687 1.3367278
> mseq = seq(0.5,1.5,by=0.1)
> pvals = rep(0,length(mseq))
> for(k in 1:length(mseq)){
+     pvals[k] = perm1samp(x,mu=mseq[k])$asl
+ }
```



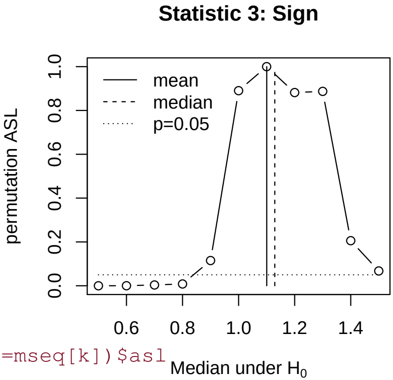
Example using Statistic 2 (signed rank)

```
> set.seed(1)
> n = 50
> x = rnorm(n,mean=1)
> mean(x)
[1] 1.100448
> median(x)
[1] 1.129104
> myfun <- function(x) {
+   n = length(x)
+   rx = rank(abs(x))
+   sum(rx[x>0]) - n*(n+1)/4
+ }
> mseq = seq(0.5,1.5,by=0.1)
> pvals = rep(0,length(mseq))
> for(k in 1:length(mseq)){
+   pvals[k] = perm1samp(x,myfun,mu=mseq[k])$asl
+ }
```



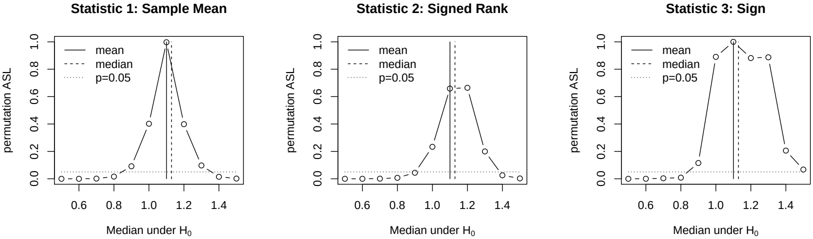
Example using Statistic 3 (sign)

```
> set.seed(1)
> n = 50
> x = rnorm(n,mean=1)
> mean(x)
[1] 1.100448
> median(x)
[1] 1.129104
> myfun <- function(x) {
+     n = length(x)
+     sum(x>0) - n/2
+ }
> mseq = seq(0.5,1.5,by=0.1)
> pvals = rep(0,length(mseq))
> for(k in 1:length(mseq)){
+     pvals[k] = perm1samp(x,myfun,mu=mseq[k])$asl
+ }
```



Comparing the Statistics

Note that as our test statistic uses less information, it becomes more robust (good thing) at the cost of losing power (bad thing):



Two-Sample Permutation Tests

Two-Sample Problem

For the two-sample location problem, we have $N = m + n$ observations

- $X_1, \ldots, X_m$ are iid random sample from population 1
- $Y_1, \ldots, Y_n$ are iid random sample from population 2

We want to make inferences about difference in distributions

- Let F_1 and F_2 denote distributions of populations 1 and 2
- Null hypothesis is same distribution $\Leftrightarrow H_0 : F_1(z) = F_2(z)$ for all z
- Alternative hypothesis is different distribution $\Leftrightarrow H_1 : F_1(z) \neq F_2(z)$ for some z

Permutation Vector and Lemma (2-Sample)

Let $\mathbf{g} = (g_1, g_2, \ldots, g_N)$ denote the permutation vector denoting which observation belongs to which group.

- Note that $\mathbf{g}$ contains m X-group labels and n Y-group labels
- g_i denotes group membership of z_i, where z_i is i-th observation for combined sample of N observations
- There are $\binom{N}{n}$ different possible $\mathbf{g}$ vectors

Permutation Lemma:
Under $H_0 : F_1(z) = F_2(z)\ \forall z$, the vector $\mathbf{g}$ has probability $1/\binom{N}{n} = \frac{m!n!}{N!}$ of equaling each of the $\binom{N}{n} = \frac{N!}{m!n!}$ different possible outcomes

Permutation Achieved Significance Level (2-Sample)

The permutation ASL is the permutation probability that $\hat{\theta}^*$ exceeds $\hat{\theta}$:

$$\mathrm{ASL}_{\mathrm{perm}} = \#\{|\hat{\theta}^*_b| \geq |\hat{\theta}|\} / \binom{N}{n}$$

where $\{\hat{\theta}^*_b\}_{b=1}^{\binom{N}{n}}$ is the set of all possible test statistics under H_0.

Note that the above is for the two-sided alternative $H_0 : \theta \neq \theta_0$

- For $H_0 : \theta < \theta_0$, we have $\mathrm{ASL}_{\mathrm{perm}} = \#\{\hat{\theta}^*_b \leq \hat{\theta}\} / \binom{N}{n}$
- For $H_0 : \theta > \theta_0$, we have $\mathrm{ASL}_{\mathrm{perm}} = \#\{\hat{\theta}^*_b \geq \hat{\theta}\} / \binom{N}{n}$

Problem: when $\binom{N}{n}$ is large, forming $\hat{\theta}^*_b$ for all $\binom{N}{n}$ possible **g** vectors is computationally expensive.

- Solution: use a Monte Carlo approach!

Two-Sample Permutation Test (Monte Carlo)

Procedure for approximating $\mathrm{ASL}_{\mathrm{perm}}$ using Monte Carlo approach:

1. Randomly sample B permutation vectors $\mathbf{g}_1^*, \ldots, \mathbf{g}_B^*$
2. Evaluate the permutation replication $\hat{\theta}_b^* = s(\mathbf{g}_b^*, \mathbf{z})$ where $\mathbf{z} = (z_1, \ldots, z_N)$ is the observed vector of combined data
3. Approximate $\mathrm{ASL}_{\mathrm{perm}}$ using

$$\widehat{\mathrm{ASL}}_{\mathrm{perm}} = \#\{|\hat{\theta}_b^*| \geq |\hat{\theta}|\}/B$$

This assumes that the statistic $\hat{\theta} = s(\mathbf{g}, \mathbf{z})$ is designed such that larger absolute values provide more evidence against H_0.

- Statistic 1: $\hat{\theta} = \bar{x} - \bar{y}$
- Statistic 2: $\hat{\theta} = \sum_{i=1}^{N} R_i 1_{\{g_i = 1\}} - \frac{m(N+1)}{2}$ where $R_i = \mathrm{rank}(|z_i - \theta_0|)$
- Statistic 3: $\hat{\theta} = \log(\hat{\sigma}_x^2 / \hat{\sigma}_y^2)$

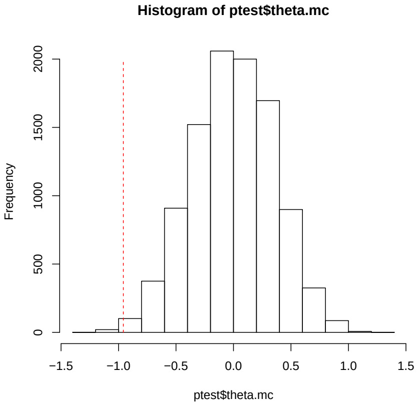
Two-Sample Permutation Test: R Function

An R function for performing two-sample permutation tests:

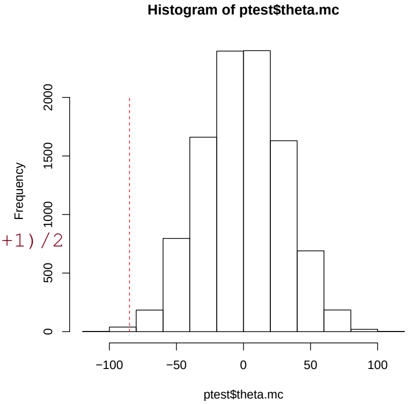
```
meandif <- function(x,y) mean(x) - mean(y)
perm2samp <- function(x,y,myfun=meandif,nsamp=10000,
                      alternative=c("two.sided","less","greater")){
  theta.hat = myfun(x,y)
  m = length(x)
  n = length(y)
  N = m + n
  z = c(x,y)
  gmat = replicate(nsamp,sample.int(N,m))
  theta.mc = apply(gmat,2,function(g,z){myfun(z[g],z[-g])},z=z)
  if(alternative[1]=="less"){
    aslperm = sum(theta.mc <= theta.hat) / nsamp
  } else if(alternative[1]=="greater"){
    aslperm = sum(theta.mc >= theta.hat) / nsamp
  } else{
    aslperm = sum(abs(theta.mc) >= abs(theta.hat)) / nsamp
  }
  list(theta.hat=theta.hat,theta.mc=theta.mc,asl=aslperm)
}
```

Example using Statistic 1 (mean difference)

```
> set.seed(1)
> x = rnorm(15)
> y = rnorm(20,mean=1)
> choose(35,15)
[1] 3247943160
> myfun=function(x,y) mean(x)-mean(y)
> myfun(x,y)
[1] -0.9578472
> mean(x) - mean(y)
[1] -0.9578472
> ptest = tsperm(x,y,myfun)
> ptest$theta.hat
[1] -0.9578472
> ptest$asl
[1] 0.0042
> hist(ptest$theta.mc)
> lines(rep(ptest$theta.hat,2),c(0,2000),col="red",lty=2)
```



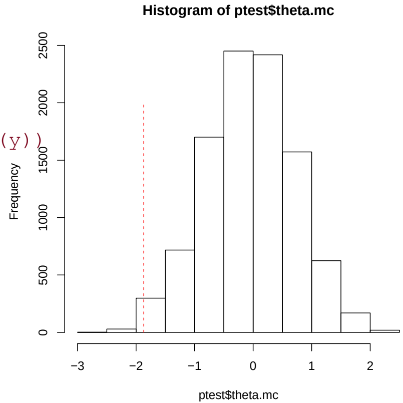
Example using Statistic 2 (rank sum)



```
> set.seed(1)
> x = rnorm(15)
> y = rnorm(20,mean=1)
> choose(35,15)
[1] 3247943160
> myfun = function(x,y){
+     m = length(x)
+     n = length(y)
+     rx = rank(c(x,y))
+     sum(rx[seq(along=x)]) - m*(m+n+1)/2
+ }
> myfun(x,y)
[1] -85
> ptest = perm2samp(x,y,myfun)
> ptest$theta.hat
[1] -85
> ptest$asl
[1] 0.0039
> hist(ptest$theta.mc)
> lines(rep(ptest$theta.hat,2),c(0,2000),col="red",lty=2)
```

Example using Statistic 3 (log variance ratio)

```
> set.seed(1)
> x = rnorm(15)
> y = rnorm(20,sd=3)
> choose(35,15)
[1] 3247943160
> myfun=function(x,y) log(var(x)/var(y))
> myfun(x,y)
[1] -1.867756
> log(var(x)/var(y))
[1] -1.867756
> ptest = tsperm(x,y,myfun)
> ptest$theta.hat
[1] -1.867756
> ptest$asl
[1] 0.01
> hist(ptest$theta.mc)
> lines(rep(ptest$theta.hat,2),c(0,2000),col="red",lty=2)
```



Correlation Permutation Tests

Association/Correlation Problem

Suppose we have paired data $(X_i, Y_i) \overset{\text{iid}}{\sim} F$ for $i = 1, \ldots, n$, where F is some bivariate distribution.

Question: are X and Y statistically associated with one another?

- X and Y are independent if and only if $F_{XY}(x, y) = F_X(x)F_Y(y)$
- If X and Y are correlated/associated, they are dependent
- Null hypothesis is $H_0 : \rho = 0$ where $\rho = \text{cor}(X, Y)$
- Different definitions of ρ measure different types of association

How can we use a permutation test to answer this question?

Permutation Vector and Lemma (Correlation)

Let $\mathbf{g} = (g_1, g_2, \ldots, g_n)$ denote the permutation vector which contains the integers $\{1, \ldots, n\}$ in some order.

- There are $n!$ different possible $\mathbf{g}$ vectors (orderings of y_i)
- If $H_0 : \rho = 0$ is true, then reordering of y_i doesn't affect correlation

Permutation Lemma:
Under $H_0 : \rho = 0$, the vector $\mathbf{g}$ has probability $1/n!$ of equaling each of the $n!$ different possible outcomes

Permutation Achieved Significance Level (Correlation)

The permutation ASL is the permutation probability that $\hat{\rho}^*$ exceeds $\hat{\rho}$:

$$\mathrm{ASL}_{\mathrm{perm}} = \#\{|\hat{\rho}_b^*| \geq |\hat{\rho}|\}/n!$$

where $\{\hat{\rho}_b^*\}_{b=1}^{n!}$ is the set of all possible test statistics under H_0.

Note that the above is for the two-sided alternative $H_0 : \rho \neq 0$

- For $H_0 : \rho < 0$, we have $\mathrm{ASL}_{\mathrm{perm}} = \#\{\hat{\rho}_b^* \leq \hat{\rho}\}/n!$
- For $H_0 : \rho > 0$, we have $\mathrm{ASL}_{\mathrm{perm}} = \#\{\hat{\rho}_b^* \geq \hat{\rho}\}/n!$

Problem: when $n!$ is large, forming $\hat{\theta}_b^*$ for all $n!$ possible **g** vectors is computationally expensive.

- Solution: use a Monte Carlo approach!

Correlation Permutation Test (Monte Carlo)

Procedure for approximating $\mathrm{ASL}_{\mathrm{perm}}$ using Monte Carlo approach:

1. Randomly sample B permutation vectors $\mathbf{g}_1^*, \ldots, \mathbf{g}_B^*$
2. Evaluate the permutation replication $\hat{\rho}_b^* = \mathrm{cor}(\mathbf{x}, \mathbf{y}_b)$ where $\mathbf{x}$ is the observed vector and $\mathbf{y}_b$ is b-th permuted copy of $\mathbf{y}$
3. Approximate $\mathrm{ASL}_{\mathrm{perm}}$ using

$$\widehat{\mathrm{ASL}}_{\mathrm{perm}} = \#\{|\hat{\rho}_b^*| \geq |\hat{\rho}|\}/B$$

This assumes that the correlation statistic $\hat{\rho} = \mathrm{cor}(\mathbf{x}, \mathbf{y})$ is designed such that larger absolute values provide more evidence against H_0.

- Could use any reasonable correlation measure
- Popular choices include Pearson, Spearman, and Kendall

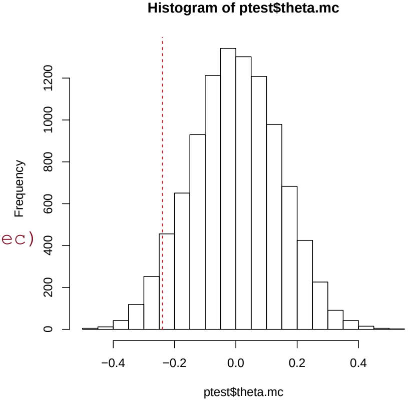
Correlation Permutation Test: R Function

An R function for performing correlation permutation tests:

```
permcor <- function(x,y,method="pearson",nsamp=10000,
                    alternative=c("two.sided","less","greater")){
  n = length(x)
  if(n!=length(y)) stop("lengths of x and y must match")
  theta.hat = cor(x,y,method=method)
  gmat = replicate(nsamp,sample.int(n))
  theta.mc = apply(gmat,2,function(g)cor(x,y[g],method=method))
  if(alternative[1]=="less"){
    aslperm = sum(theta.mc <= theta.hat) / nsamp
  } else if(alternative[1]=="greater"){
    aslperm = sum(theta.mc >= theta.hat) / nsamp
  } else{
    aslperm = sum(abs(theta.mc) >= abs(theta.hat)) / nsamp
  }
  list(theta.hat=theta.hat,theta.mc=theta.mc,asl=aslperm)
}
```

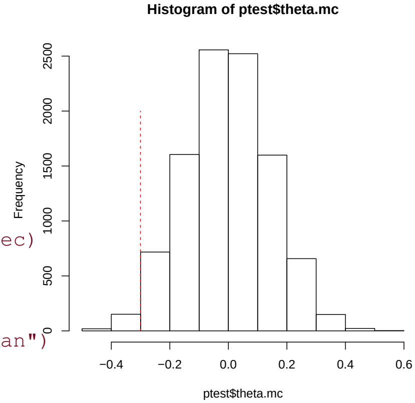
Example using Statistic 1 (Pearson)

```
> set.seed(1)
> n = 50
> x = rnorm(n)
> y = rnorm(n)
> rho = -0.2
> Amat = matrix(c(1,rho,rho,1),2,2)
> Aeig = eigen(Amat,symmetric=TRUE)
> evec = Aeig$vec
> evalsqrt = diag(Aeig$val^0.5)
> Asqrt = evec %*% evalsqrt %*% t(evec)
> z = cbind(x,y)%*%Asqrt
> x = z[,1]
> y = z[,2]
> ptest = permcor(x,y)
> ptest$asl
[1] 0.0966
> hist(ptest$theta.mc)
> lines(rep(ptest$theta.hat,2),c(0,2000),col="red",lty=2)
```



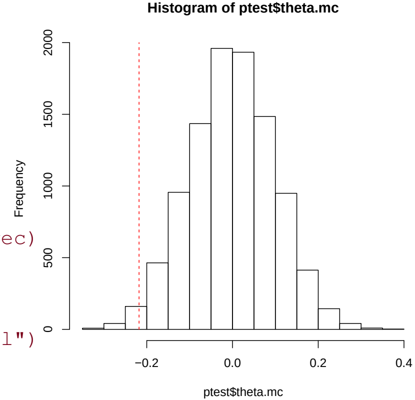
Example using Statistic 2 (Spearman)

```
> set.seed(1)
> n = 50
> x = rnorm(n)
> y = rnorm(n)
> rho = -0.2
> Amat = matrix(c(1,rho,rho,1),2,2)
> Aeig = eigen(Amat,symmetric=TRUE)
> evec = Aeig$vec
> evalsqrt = diag(Aeig$val^0.5)
> Asqrt = evec %*% evalsqrt %*% t(evec)
> z = cbind(x,y)%*%Asqrt
> x = z[,1]
> y = z[,2]
> ptest = permcor(x,y,method="spearman")
> ptest$asl
[1] 0.0338
> hist(ptest$theta.mc)
> lines(rep(ptest$theta.hat,2),c(0,2000),col="red",lty=2)
```



Example using Statistic 3 (Kendall)

```
> set.seed(1)
> n = 50
> x = rnorm(n)
> y = rnorm(n)
> rho = -0.2
> Amat = matrix(c(1,rho,rho,1),2,2)
> Aeig = eigen(Amat,symmetric=TRUE)
> evec = Aeig$vec
> evalsqrt = diag(Aeig$val^0.5)
> Asqrt = evec %*% evalsqrt %*% t(evec)
> z = cbind(x,y)%*%Asqrt
> x = z[,1]
> y = z[,2]
> ptest = permcor(x,y,method="kendall")
> ptest$asl
[1] 0.0247
> hist(ptest$theta.mc)
> lines(rep(ptest$theta.hat,2),c(0,2000),col="red",lty=2)
```


Histogram of ptest$theta.mc
Frequency
ptest$theta.mc