Stat 205: Introduction to Nonparametric Statistics

Lecture 07: Other Non-Parametric Approaches

Instructor David Donoho; TA: Yu Wang

- Nonparametrics $\neq$ Rank Methods!
- Examples
  - Wald-Wolfowitz Two-Sample Runs Test
  - Friedman-Rafsky Multivariate Runs Test
  - Bootstrap

Analysis of Runs

- String of n H/T's
- Run = sequence of pure Heads or pure Tails
- Example
  - String `HHHTTTHHHTHHHTTTT`
  - Runs `HHH TTT HHH T HHH TTTT`
- Properties one can study:
  - Number of Runs
  - Distribution of Run Lengths
  - Longest Run

Permutation Distribution of Number of Runs

- String $S = (s_i)_{i=1}^n$ of n 1/0's
- n_1 1's, n_0 0's.
- $N(S) = \#Runs$
- Permuted String $S_\pi = (s_{\pi(i)})_{i=1}^n$.
- Permutation distribution

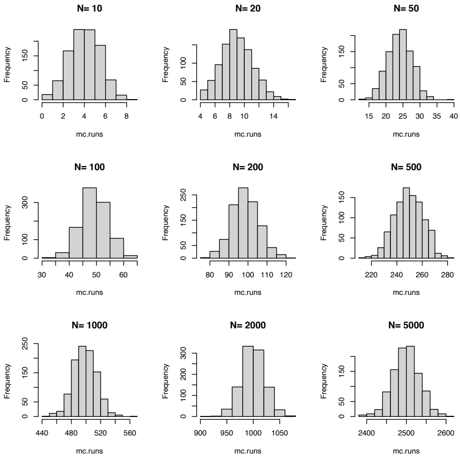
$$P_0(N = r) = \frac{\#\{\pi : N(S_\pi) = r\}}{n!}$$

  - r even

$$P_{0,n}(N = 2k) = \frac{2\binom{n_0-1}{k-1}\binom{n_1-1}{k-1}}{\binom{n_0+n_1}{n_1}}$$

  - r odd

$$P_{0,n}(N = 2k+1) = \frac{\binom{n_0-1}{k}\binom{n_1-1}{k-1} + \binom{n_0-1}{k-1}\binom{n_1-1}{k}}{\binom{n_0+n_1}{n_1}}$$


N= 10
N= 20
N= 50
N= 100
N= 200
N= 500
N= 1000
N= 2000
N= 5000
Frequency
mc.runs

R Code for number of runs

```
n.runs   <- function(x){ sum(diff(x) !=0) }
Y           <- rep(c(0,1),N/2);
mc.runs <- replicate(M, n.runs(sample(Y,replace=T)) )
hist(mc.runs, main=paste("N=",N))
```

Normal Approximation

- (approx) Z-score

$$Z = \frac{N - \mu_N}{\sigma_N}$$

$$\mu_N = \frac{2n_0 n_1}{n_0 + n_1} + 1$$

$$\sigma_N^2 = \frac{2n_0 n_1(2n_0 n_1 - n_0 - n_1)}{(n_0 + n_1)^2(n_0 + n_1 + 1)}$$

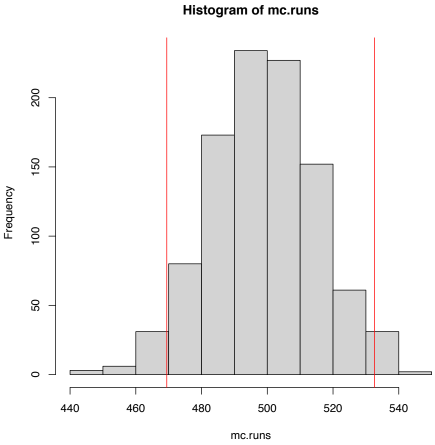
- Example: $n_0 = n_1$

$$\mu_N = \frac{n}{2} + 1$$

$$\sigma_N^2 = \frac{n(n/2 - 1)}{2(n + 1)}$$

- Asymptotically:

$$P_0\{N \in n/2 \pm (\sqrt{n} + 1)\} \geq .95$$



$$N = 1000, \ P_{0,n}\{N \notin n/2 \pm (\sqrt{n} + 1)\} \approx 0.053$$

R Code for evaluating number of runs

```
mu = N/2+1;
sigma2 = N *( N/2 - 1) /( 2 *(N+1))

lo = mu + 2*sqrt(sigma2)
hi = mu - 2*sqrt(sigma2)

abline(v=lo,col='red')
abline(v=hi,col='red')

alpha = sum(abs(mc.runs - mu)/sqrt(sigma2) > 2)/lengt
```

Prints '0.053'

Example: Longest Run

- $n = 200$, $n_0 = 100$, $n_1 = 100$.
- Most people initially think sequences should be roughly alternating: 0 1 0 1 0 1 0 1 0 1 . . . 0 1
- Long runs seem (from this viewpoint) suspicious
- Are they really?
- Mark Schilling, "The Longest Run of Heads", *College Mathematics Journal*

The two sequences shown below each purportedly represent the results of 200 tosses of a fair coin. One of these is an actual sequence obtained from coin tossing, while the other sequence is artificial. Can you decide, in sixty seconds or less, which of the sequences is more likely to have arisen from actual coin tossing and which one is the imposter?

Sequence #1

T HHHHT T T T HHHHT HHHHHHHHHT T THHT T HHHHHT T T T T HHT HHT HHH T
T T HT T HHHHT HT T T HT T T HHT T T T HHHHHHT T T HHT T HHHT HHHHHT T T T
T HT T T HHT T HT T HHT T T HHT T T HH THHT HHT T T T T HHT HHHHHHT HT HT T
HT HT T HHHT T HHT HT HHHHHHHHT THT T HHHT HHT T HT T T T T T HHHT HHH

Sequence #2

T HT HT T T HT T T T T HT HT T T HT T HHHT HHT HT HT HT T T T HHT T HHT T HHH T
HHHT T HHHT T T HHHT HHHHT T T HT HT HHHHT HT T T HHHT HHT HT T T HHT H
HHT HHHHT T HT HHT HHHT T T HT HHHT HHT T T HHHT T T T HHHT HT HHHHT H
T.T HHT T T T HT HT HT T HT HHT T HT T THT T T T HHHHT HT HHHT T HHHHHT HH

The above challenge is based on a classroom experiment originally performed by Révész [**14**]. The class is divided into two groups. In the first group, each student is instructed to toss a coin 200 times and record the resulting sequence of heads and tails. Each student in the second group is merely to write down a sequence of heads and tails that the student believes is a reasonable *simulation* of 200 tosses of a fair coin. Given the combined results of the two groups, Révész claims that the students can be classified back into their original groups with a surprising degree of accuracy by means of a very simple criterion: In students' simulated patterns, the longest run of consecutive heads or consecutive tails is almost invariably *too short* relative to that which tends to arise from actual coin tossing.

Some Math

- R_n longest run of pure heads, in string with $n_1/n = p$

$$E[R_n] \approx \log_{1/p}(nq) + \gamma/\log(1/p) - 1/2$$

$$Var[R_n] \approx \frac{\pi^2}{6}\log^2(1/p) + \frac{1}{12}$$

- In case $p = 1/2$:

$$E[R_n] \approx \log_2(n/2) - 2/3$$

- $E[R_{200}] \approx 7$

A very easy rule of thumb is that the longest head run for a fair coin is very likely to be within three either way from the integer nearest to $\log_2(n/2)$. Applying this rule for $n = 200$, we find that reasonable limits for R_{200} are 4 and 10. The actual probability that the longest head run is between these values turns out to be 95.3%, which slightly exceeds the lower bound of 94.5% guaranteed by Table 1. For R'_{200}, the longest run of heads *or* of tails in 200 tosses, simply add one to each of the limits.

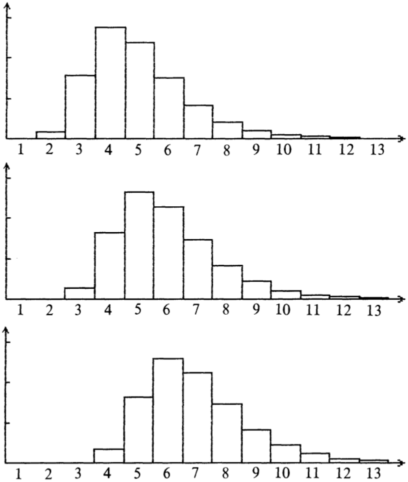


Figure 3
Distributions of R_n for (a) $n = 50$, (b) $n = 100$, (c) $n = 200$

Table 1
Prediction Interval Probabilities for R_n ($p = 1/2$)

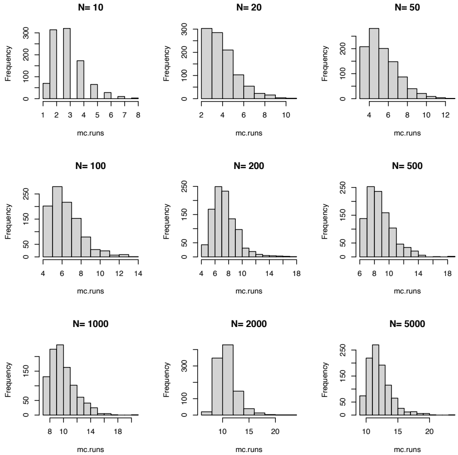
Width of interval	Minimum probability that R_n lies in the interval
1	23.6%
2	44.9%
3	62.3%
4	75.5%
5	84.6%
6	90.7%
7	94.5%
8	96.8%
9	98.2%
10	99.0%

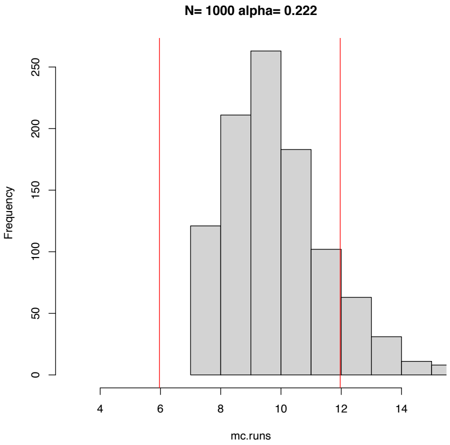
Table 2
Exact and Approximate Probabilities for R_{200} ($p = 1/2$)

x	$P(R_{200} = x)$ (Exact)	$P(R_{200} = x)$ (Approx.)
0–3	.001	.002
4	.033	.042
5	.165	.166
6	.257	.248
7	.224	.219
8	.146	.146
9	.083	.084
10	.044	.045
11	.023	.024
12	.011	.012
> 12	.012	.012

R Code for simulating longest runs

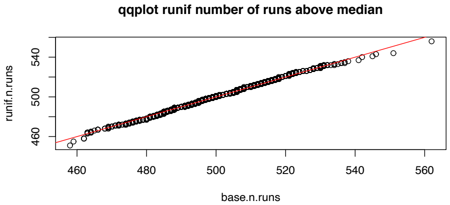
```
longest.run <- function(x){
I = diff(x) !=0 ;
M = max(diff(which(I)))
}
Y <- rep(c(0,1),N/2);
long.runs <- replicate(M, longest.run(sample(Y,replace=F)) )
```

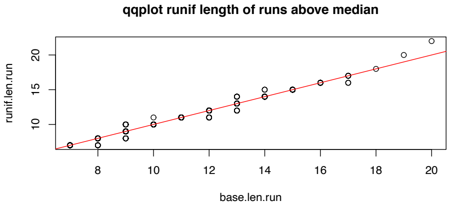

N= 10
N= 20
N= 50
N= 100
N= 200
N= 500
N= 1000
N= 2000
N= 5000
Frequency
mc.runs


N= 1000 alpha= 0.222
Frequency
0
50
100
150
200
250
4
6
8
10
12
14
mc.runs

Testing Random Number Generator

- Generate n 'uniform random' numbers $X_i \in [0, 1]$
- $s_i = 1\{X_i \geq \frac{1}{2}\}$
- Test for number of runs.
- Test for longest run.


qqplot runif number of runs above median
runif.n.runs
460
500
540
460
480
500
520
540
560
base.n.runs


qqplot runif length of runs above median
runif.len.run
10
15
20
8
10
12
14
16
18
20
base.len.run

$$N = 1000, \ M = 1000$$

R Code for MC Study of runif()

```
r.above.med <- function(N){
X <- runif(N);
m <- median(X);
X > m
}
#
runif.n.runs  <- replicate(M, n.runs(r.above.med(N)) )
base.n.runs  <- replicate(M, n.runs(sample(Y,replace=F)) )
qqplot(base.n.runs,runif.n.runs,main="qqplot runif number of runs above median")
abline(0,1,col="red")
#
runif.len.run <- replicate(M, longest.run(r.above.med(N)) )
base.len.run  <- replicate(M, longest.run(sample(Y,replace=F)) )
qqplot(base.len.run,runif.len.run,main="qqplot runif length of runs above median")
abline(0,1,col="red")
#
rank.test(base.n.runs,runif.n.runs)
rank.test(base.len.run,runif.len.run)
```

Output From MC Study of runif()

```
> rank.test(base.n.runs,runif.n.runs)
$Sphi
[1] 1.926924

$statistic
            [,1]
[1,] 0.08613157

$p.value
           [,1]
[1,] 0.9313618

attr(,"class")
[1] "rank.test"
> rank.test(base.len.run,runif.len.run)
$Sphi
[1] -2.136845

$statistic
             [,1]
[1,] -0.09551486

$p.value
           [,1]
[1,] 0.9239059

attr(,"class")
[1] "rank.test"
```

Goodness-of-Fit Testing, 1

We have

- Theory: F_0.
- Observations: $X_i \sim_{iid} F$.
- Hypotheses:

$$H_0 : F = F_0 \qquad H_A : F \neq F_0.$$

Goodness-of-Fit Testing, 2

Distribution-free Tests, 1

- Theory: F_0
- Observations: $X_i \sim_{iid} F$
- Would-be Uniform RV's

$$Y_i = F_0(X_i)$$

- Theorem: *if X_i iid F_0 continuous, then Y_i iid Unif$[0, 1]$*

Goodness-of-Fit Testing, 2

Distribution-free Tests, 2

- $F_n^Y(t) = \frac{1}{n}\sum_{i=1}^n 1\{Y_i \leq t\}$
- $U(t) = t$
- Discrepancy measure

$$D(\{X_i\}_{i=1}^n; F_0, \Delta) = \Delta(F_n^Y, U)$$

- Here Δ is a user-supplied measure of discrepancy; Examples:
  - Kolmogorov-Smirnov

$$\Delta_{KS}(F, G) = \sup_t |F(t) - G(t)|.$$

  - Anderson-Darling

$$\Delta_{AD}(F, G) = \int_0^1 |F(t) - G(t)|^2 dt.$$

Goodness-of-Fit Testing, 3

Distribution-free Tests, 3

- Theorem: *Under* H_0*, the probability distribution of* $D(\{X_i\}_{i=1}^n; F_0, \Delta)$ *is the same for all* F_0 *with a continuous CDF.*
- Additionally, for many discrepancies, when F_0 is *not* continuous, the RV $D(\{X_i\}_{i=1}^n; F_0, \Delta)$ is stochastically smaller.
- Consequently, define a critical value $\delta(\alpha) \equiv \delta(\alpha; \Delta, n)$

$$P\{D(\{V_i\}_{i=1}^n; U, \Delta) \geq \delta\} = \alpha,$$

where V_i $i = 1, \ldots, n$ is iid $U[0, 1]$, then, for *every* choice of a null hypothesis $H_0 : F = F_0$, we have

$$P_{H_0}\{D(\{X_i\}_{i=1}^n; F_0, \Delta) \geq \delta\} \leq \alpha.$$

- This means that we get a rigorously valid level-α test using

$$\text{Reject } H_0 \text{ if } D(\{X_i\}_{i=1}^n; F_0, \Delta) \geq \delta(\alpha, n)$$

Goodness-of-Fit Testing 4

Example: Kolmogorov-Smirnov

$$D_{KS}(\{X_i\}; F_0) = \max_i |F_0(X_i) - \frac{i}{n}|.$$

Example: Anderson-Darling

$$D_{AD}(\{X_i\}; F_0) = \sum_i |F_0(X_i) - \frac{i}{n}|^2.$$

Example: χ^2, where F_0 is discrete with possible values $\{y_k\}_{k=1}^K$.

$$D_{\chi^2} = n \cdot \sum_{k=1}^K \frac{(F_n(\{y_k\}) - F_0(\{y_k\}))^2}{F_0(\{y_k\})}$$

R Code:Theory of Number of Runs

```
p.perm.runs = function(k,n0,n1){
 n=n0+n1;
 d = choose(n,n1);
 k0 = k%/%2;
 if(k %% 2){
nkm1 = choose(n0-1,k0)  *choose(n1-1,k0-1);
nkm0 = choose(n0-1,k0-1)*choose(n1-1,k0);
pk = (nkm1+nkm0)/d;
} else {
    nkm1 = choose(n0-1,k0-1) *choose(n1-1,k0-1);
    pk = (2*nkm1)/d;
    }
    pk
}
#
p.runs <- function(n0,n1){
n <- n0+n1;
d <- rep(0,n);
m <- min(n0,n1);
for(k in 2:(2*m+1)){
d[k] <- p.perm.runs(k,n0,n1)
}
d
}
```

R code: testing agreement with Theory

```
M = 10000;
n = 20;
Y <- rep(c(0,1),n/2);
runif.n.runs <- replicate(M, n.runs(r.above.med(n)) )
base.n.runs  <- replicate(M, n.runs(sample(Y,replace=F)) )
#
counts.x<-table(base.n.runs)
labels.x<-as.numeric(names(counts.x))
d <- p.runs(10,10)
p.x <- d[labels.x]
chisq.test(counts.x,p=p.x,rescale=TRUE)
#
counts.y<-table(runif.n.runs)
labels.y<-as.numeric(names(counts.y))
d <- p.runs(10,10)
p.y <- d[labels.y]
chisq.test(counts.y,p=p.y,rescale=TRUE)
```

Output from R code

```
> M = 10000;
> n = 20;
> Y <- rep(c(0,1),n/2);
> runif.n.runs <- replicate(M, n.runs(r.above.med(n)) )
> base.n.runs  <- replicate(M, n.runs(sample(Y,replace=F)) )
> #
> counts.x<-table(base.n.runs)
> labels.x<-as.numeric(names(counts.x))
> d <- p.runs(10,10)
> p.x <- d[labels.x]
> chisq.test(counts.x,p=p.x,rescale=TRUE)

Chi-squared test for given probabilities

data:  counts.x
X-squared = 11.197, df = 15, p-value = 0.7385

> #
> counts.y<-table(runif.n.runs)
> labels.y<-as.numeric(names(counts.y))
> d <- p.runs(10,10)
> p.y <- d[labels.y]
> chisq.test(counts.y,p=p.y,rescale=TRUE)

Chi-squared test for given probabilities

data:  counts.y
X-squared = 10.925, df = 16, p-value = 0.8141
```

p-values (0.74,0.81) indicate test has passed!

Wald-Wolfowitz Two-Sample Runs test

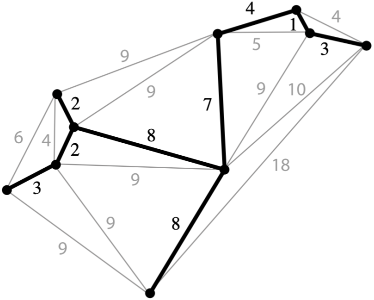
- Two Samples $X = (X_i)_{i=1}^{n_X}$; $Y = (Y_i)_{i=1}^{n_Y}$
- Combine and sort: `Z = sort(c(X,Y))`
- For simplicity, assume no collisions: $X \cap Y = \emptyset$.
- $(s_j)_{j=1}^{n}$ binary labels:

$$s_j = \begin{cases} 1 & z_j \in Y \\ 0 & z_j \in X \end{cases} .$$

- Supply (s_j) to R's `runs.test()`.
- Motivation: Under null, adjacent samples equally likely same/different origin
  Under alternative, runs signal observations from *same* origin

Graph-Based Two-Sample Tests

- The Friedman and Rafsky test is a generalization of Wald-Wolfowitz runs test to higher dimensions
- The difficulty is that we need to sort observations
- Friedman and Rafsky purpose to use minimal spanning trees as a multivariate generalization of the univariate sorted list

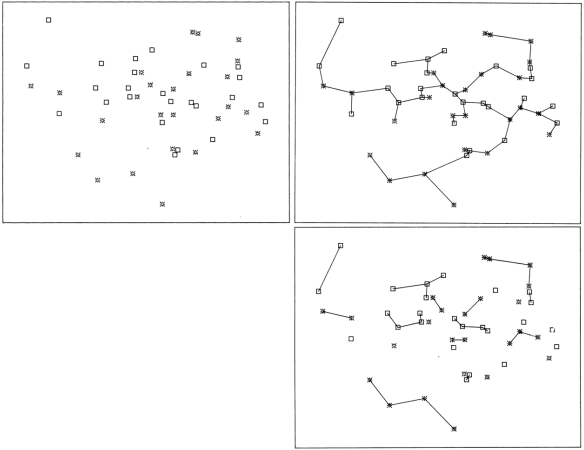


Christof Seiler, Stanford Stat 205 (2016)

Graph-Based Two-Sample Tests

- For univariate sample, the edges of the MST are defined by adjacent observations in the sorted list
- The Wald-Wolfowitz runs test can be described in this alternative way:
  1. Construct minimal spanning trees of pooled univariate observations
  2. Remove all edges for which the defining nodes originate from different samples
  3. Define the test statistics as the number of disjoint subtrees that result
- For multivariate samples, just construct minimal spanning tree in step 1 from multivariate observations

Graph-Based Two-Sample Tests



Source: Friedman and Rafsky (1979)

Graph-Based Two-Sample Tests

- Reject H_0 for small and large number of subtrees (runs)
- The null distribution of the test statistics can be computed using permutation tests
    - fix tree
    - permute labels
- Good power in finite samples for multivariate data (against general alternatives: location, spread, and shape)

Graph-Based Two-Sample Tests

- Has been applied to mapping cell populations in flow cytometry data (Hsiao et al. 2016)
  - two cell populations
  - d measurements on each cell
  - determine whether the expression of a cellular marker is statistically different
  - suggesting candidates for new cellular phenotypes
  - indicate splitting or merging of cell populations
- Recent development for very high-dimensional data sets (Chen and Friedman 2015)

Theory

Wald-Wolfowitz Two-Sample runs test is **consistent against all alternatives**.

- Suppose that critical value $\nu_\alpha(n_X, n_Y)$ for N_n is normalized

$$P_{0,n}\{N_n(s^{X,Y}) > \nu_\alpha(n_X, n_Y)\} = \alpha$$

- Suppose that $F_X \neq F_Y$, n_X,$n_Y \to \infty$, $n_X/(n_X + n_Y) \to \gamma \in (0,1)$:

$$\lim_{n\to\infty} P_{F_X,F_Y}\{N_n(s^{X,Y}) > \nu_\alpha(n_X, n_Y)\} = 1.$$