Statistical Testing 2 - Empirical Distributions

Daniel Lawson University of Bristol

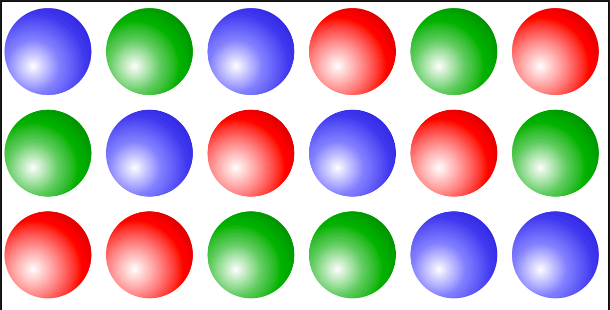
Lecture 02.2.2 (v1.0.1)

Resampling

- The main types of resampling tests include:
  - **jacknifing**, which is analysing subsets of data to estimate (variance of) parameter estimates
  - **bootstrapping**, which is resampling with replacement, to estimate (variance of) parameter estimates
  - **permutation**, which is resampling without replacement, to test a null hypothesis
  - **cross-validation**, which is analysing subsets of data to estimate out-of-sample prediction, for model performance
- Each of these methods can be applied to a wide variety of problems, and often requires thought to use appropriately.

Permutations

All permutations of three colors (each column is a permutation):



- Figure from Wikipedia[1]. There are in general $n!$ permutations.

[1] https://upload.wikimedia.org/wikipedia/commons/4/4c/Permutations_RGB.svg

Generating permutations

```
> 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
```

Use of permutations in testing

- Consider the following general class of problem:
  - H0: y is independent of x.
  - H1: y is dependent on x.
- x may be continuous, categorical, etc and y may depend on a number of other things.
- A **permutation test** will:
  - resample x, y pairs **under H0**,
  - Construct a test statistic T,
  - Test if T extreme in the real data, compared to the permutations?

Why permutations

- The main advantage is that the test is asymptotically correct and distribution free. We only (**!**) have to assume **exchangability**.
- Exchangability of what?
  - what would be **equal if the null hypothesis is true**, and
  - would be **differerent if the alternative hypothesis is true**?
- It is essential to **maintain any true correlation structure** when performing the test, otherwise the test is not correct.
- For example, if the indices were originally correlated, permutation will fail.
  - as from e.g. a time-series.

Some main types of test (1)

x1	x2	x3	y1	y2
4	12	-3	2	-24

- Permutation of **indices**:

x2	y1	x3	y2	x1
4	12	-3	2	-24

Some main types of test (1)

x1	x2	x3	y1	y2
4	12	-3	2	-24

- Permutation of **indices**:

x2	y1	x3	y2	x1
4	12	-3	2	-24

- Permutation of **signs**, retaining magnitudes:

x1	x2	x3	y1	y2
4	-12	3	-2	24

Some main types of test (2)

x1	x2	x3	y1	y2
4	12	-3	2	-24

- Permutation of **group** labels:

x1	y1	y2	x2	x3
4	12	-3	2	-24

Some main types of test (2)

x1	x2	x3	y1	y2
4	12	-3	2	-24

- Permutation of **group** labels:

x1	y1	y2	x2	x3
4	12	-3	2	-24

- Permutation **within group** labels:

x1	x2	x3	y1	y2
12	-3	4	-24	2

Monte-Carlo testing

- There are in general $n!$ permutations. This is typically too many for $n > 20$.
- We instead choose N **random permutations** from all the possible ones.
- Monte-Carlo testing is an important subject in its own right.
- Its often possible to place guarantees on the p-value from very few samples.

Monte-Carlo test

- To conduct a Monte-Carlo test, we construct N random datasets and add our real dataset.
- We then ask, is the **real dataset an outlier** with respect to the random datasets?
- Specifically, the p-value for a test T applied to X (where large values are considered strange) is:

$$\frac{\mathrm{Rank}(T(X); T(\{x_i\}))}{N+1}$$

- where Rank simply counts the number of cases as large or larger.

Heuristics for how many permutations to use

- The **smallest possible p-value** with N permuations is $1/(N+1)$. So 999 permutations gives a minimum of 0.001.
- The **variance** around a chosen threshold, say $p = 0.05$, is determined by the sampling distribution of the Binomial:

$$\mathrm{sd}(p) = \mathrm{sd}\left(\mathrm{Bin}(N, p)\right) = \sqrt{\frac{p(1-p)}{n}}$$

  - p is of course the true unknown probability, not the observed one.
  - But variance is an increasing function of p (for $p < 0.5$)
- A heuristic rule is: to be 95% confident that $p \leq t$ we need the empirical p-value to be less that $t - 1.96\mathrm{sd}(p = t)$
- For $N = 999$ and $t = 0.05$, $\mathrm{sd}(p = t) = 0.0135$ and therefore $p < 0.036$
- A similar calculation shows $N = 999$ wouldn't be enough to be sure we were less than 0.005.
- This is conservative… only if the distribution is Normal….(**!**) **Plot the distribution of T!**

Permutation example: TCP vs UDP size

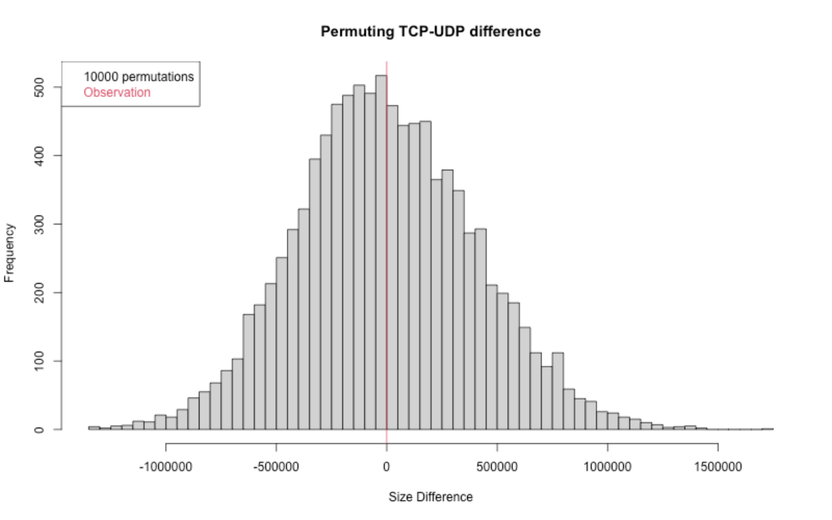
```
tcpudp=c(tcpsize,udpsize)
n1=length(tcpsize)
n2=length(udpsize)
myteststatistic=function(x,n1,n2){
    mean(x[1:n1]) - mean(x[n1+(1:n2)])}
```

Permutation example: TCP vs UDP size

```
tcpudp=c(tcpsize,udpsize)
n1=length(tcpsize)
n2=length(udpsize)
myteststatistic=function(x,n1,n2){
    mean(x[1:n1]) - mean(x[n1+(1:n2)])}

tobs=myteststatistic(tcpudp,n1,n2)
trep=sapply(1:10000,function(i){
    xrep=sample(tcpudp)
    myteststatistic(xrep,n1,n2)
})
mean(tobs<=trep)
# 0
```

Permutation example: TCP vs UDP size



Permutation example: FTP vs UDP size

- T-test suggests that FTP and UDP are different sizes

```
muudp=mean(log(udpsize))
t.test(log(ftpsize),mu=muudp)$p.value
## 0.003375621
```

Permutation example: FTP vs UDP size

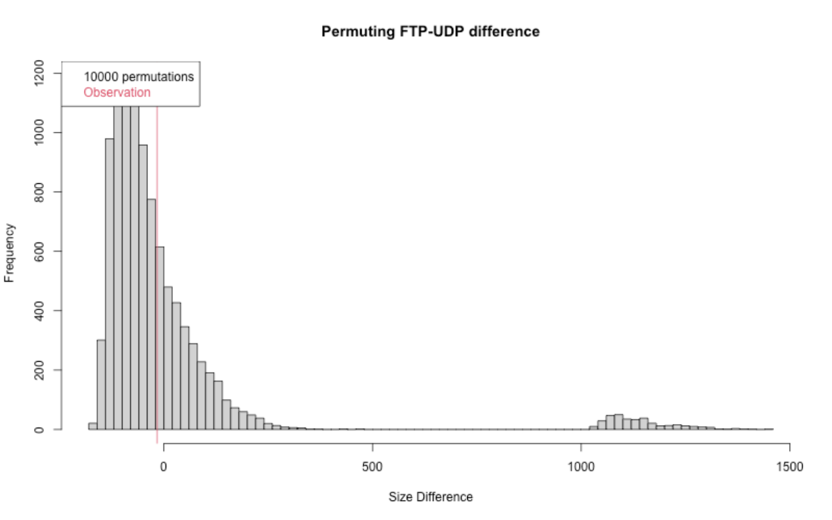
```
ftpudp=c(ftpsize,udpsize)
n1=length(ftpsize)
n2=length(udpsize)
```

Permutation example: FTP vs UDP size

```
ftpudp=c(ftpsize,udpsize)
n1=length(ftpsize)
n2=length(udpsize)

ftpudpobs=myteststatistic(ftpudp,n1,n2)
ftpudprep=sapply(1:10000,function(i){
    xrep=sample(ftpudp)
    myteststatistic(xrep,n1,n2)
})
mean(ftpudpobs<=ftpudprep)
## 0.3315
```

Permutation example: FTP vs UDP size



Permutation testing summary

- **Distributional assumptions** are often invalid (regular tests)
- **Exchangability assumptions** are often plausible (permutation tests)
- It is possible to get misleading inference if the assumptions of a test don't hold
- Permutation tests are really important for generating **plausible null hypotheses**, especially in cyber security

Reflection

- When are sampling approaches to testing appropriate?
- What do they test?
- What are the main ways to implement them?
- What problems can resampling tests solve? Where are they still difficult to apply?

Signposting

- Further reading:
  - Cosma Shalizi's Modern Regression Lectures (Lectures 26,28)
  - Cross Validation and Bootstrap Aggregating on Wikipedia
  - Chapters 18.7 of The Elements of Statistical Learning: Data Mining, Inference, and Prediction (Friedman, Hastie and Tibshirani).
  - Chapter 4 of Statistical Data Analysis by Glen Cowan Chapter 18
- Next up: Model selection