Statistics with R for Biologists

James H. Bullard
Kasper Daniel Hansen
Margaret Taub

Berkeley, California
July 7-11, 2008

Statistics and Simulation

1 Averages

2 Correlation and outliers

3 Some fun

Interlude: generating random numbers

- We want to do statistics!
- We need to be able to generate random numbers according to distributions, compute probabilities, quantiles, densities.

d{distribution}	density
p{distribution}	probability
r{distribution}	random samples
q{distribution}	quantiles

In addition to these functions we have one of the most important functions, sample, which draws from a multinomial distribution with or without replacement.

```
> x <- rchisq(100, df = 10)
> pchisq(x, df = 1)
> dchisq(x, df = 10)
> qchisq(seq(0, 1, length = 10),
+       df = 1)
```

Working with random numbers

Random number generation is hard, but luckily R has a lot of functionality, like rbinom, rnorm, rgamma, rexp, rpoisson, rt, rf, rchisq and mvrnorm from MASS and numerous other packages.
For debugging purposes set.seed can be very convenient.
When you generate a random variable a random seed is stored in the global environment (see `ls(all = TRUE)`). This seed should never be set directly and it *will be saved* with your workspace!
There are many misuses of set.seed around – running multiple MCMC chains, generating random variables on different nodes of a cluster, setting it inside a function. . .
Don't let set.seed become a habit!

What is the mean of a probability distribution?

The concept of averaging is *essential* for modern science: *repeating* the same experiment multiple times and then averaging the measurements improves the accuracy of the experiment.

This observation is connected to the frequentist interpretation of probability: that the probability of rolling 1 on a die is $1/6$ means that the proportion of dice showing 1 amongst many die rolls is equal to $1/6$.

These observations are ingrained in our modern thinking, but they have not always been obvious.

Simulation

Example

Simulate 1000 genotypes from a single locus in Hardy-Weinberg equilibrium. Look at the distribution using hist. What does the `prob` argument do? (Hint: $P(AA) = p^2, P(Aa) = 2p(1-p), P(aa) = (1-p)^2$.)

Continuous distributions are often described through their *density* which one can think of as a smoothed histogram.

```
> xx <- rgamma(1000, shape = 1)
> hist(xx)
```

Interlude: plotting functions

If `f` is a function, `plot(f)` plots the function on $[0,1]$ per default. For overlay, there is an `add` argument.

`density(xx)` estimates the density function for observations `xx` and returns a *function*.

Example

Plot the estimated density of `xx` and overlay the true density on the plot. Vary the `bw` argument to `density`. Overlay the density on the histogram.

Expectation

Without further ado, let us simulate some numbers and look at the sample mean.

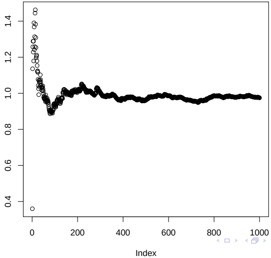
```
> xx <- rgamma(1000, shape = 1)
> mean(xx)
```

```
[1] 1.013627
```

Example

Let us see what happens as we get more and more observations. Plot the mean as a function of the number of observations (see next slide). Try setting the number of replicates really high like 10^8.

Running average



A mean is a mean

Expectation / mean

a theoretical property of a distribution.

Average / sample mean / empirical mean

the average of some observations, a random quantity.

The law of large numbers states that the average is close to the expectation – or that the average is a good *estimate* of the mean.

Counter example

Most distributions have an expectation – but not all!
A counterexample is the Cauchy-distribution (also known as the t-distribution with 1 df). It is not artificial: if X and Y are two independent standard normal distributions, X/Y is Cauchy-distributed.

Example

Simulate 1000 Cauchy-distributed random variables. How does the running mean behave? What about if you generate $1,000,000$?

Variances

In the same way most distributions have an expectation, most distributions have a variance.
Usually the variance is on a strange quadratic scale (use the standard deviation instead: $\mathrm{sd} = \sqrt{\mathrm{Var}}$), but it is useful for theoretical computations.

The variance of the sample mean

Statistics with R for Biologists

Averages

Correlation and outliers

Some fun

We saw that every time we repeated the simulation, we got a new sample mean, which was usually "close" to the expectation. We can say something about the (theoretical) variance of the sample mean (which is different from the variance of X_1):

$$\mathrm{Var}\left(\frac{X_1 + \cdots + X_n}{n}\right) = \frac{1}{n}\mathrm{Var}\left(X_1\right)$$

The variance of the sample mean decreases as the number of observations increases. This is essentially a proof of the law of large numbers.
Can we say something more precise about the fluctuations of the sample mean?
By fluctuations we mean, "what happens when we replicate the sample mean many times".

Replicating the mean many times

Statistics with R for Biologists

Averages

Correlation and outliers

Some fun

First we replicate the sample mean taken over 100 observations, 1000 times:

```
> ss <- rowMeans(matrix(rgamma(100 *
+     1000, shape = 1), ncol = 100))
> length(ss)

[1] 1000
```

ss contains 1000 realizations of the sample mean of 100 observations.
In order to see something we need to scale properly, making the variables have mean zero and variance 1 (in this case the mean and the variance are both 1).
And then we need to "zoom" using a factor $\sqrt{n}$:

```
> scaleSS <- sqrt(100) * (ss - 1)/sqrt(1)
```

CLT

Statistics with R for Biologists

Averages

Correlation and outliers

Some fun

We can very precisely quantify fluctuations on this scale – scaleSS is normally distributed.
This is the Central Limit Theorem.
It states that the sample mean – normalized to have zero mean and variance 1 – is normally distributed when scaled by the square root of the number of observations.

Example

Look at the distribution of scaleSS

A better way

Statistics with R for Biologists

Averages

Correlation and outliers

Some fun

Histograms and density plots are not great for comparing distributions.
A better way is to use QQ-plots (quantile-quantile).
A *q*-quantile of a distribution is the number so that *q* of the observations are below it. The median is the 50%-quantile.

Example

Do summary on scaleSS. Explain the output. Look at the quantile function and explain how quantile is related to sort.

QQ plot

The basic idea is simple: plot two “things” from the two distributions that should be the same.
For two sample vectors, this is essentially equivalent to plotting the two sorted vectors against each other.

```
> xx <- rgamma(100, shape = 1)
> yy <- rgamma(100, shape = 1)
> qqplot(xx, yy)
> plot(sort(xx), sort(yy))
```

(It gets more complicated if the two vectors do not have the same length.)

QQ plots 2

Against a theoretical distribution, we use the quantile function for that distribution:

```
> qqplot(xx, qgamma(ppoints(x), shape = 1))
```

Or we can just use a random sample:

```
> qqplot(xx, rgamma(10000, shape = 1))
```

For the normal distribution we have `qqnorm`.

QQ plots, why

The main advantage of QQ plots is that we have a good interpretation of what a deviation from a straight line means. If the observations come from the same distribution, but scale and location shifted, we still see a straight line. Heavy and lighter tails yield S-shapes.

Example

Try QQ plots of a Normal(a,b) vs. a Normal(0,1), a t-distribution vs. a Normal(0,1), Gamma(shape = 6) vs Gamma(shape = 1), Gamma(shape = 6, scale = 1) vs Gamma(shape = 6, shape = 5).

Interlude: cumulative distribution functions

Another way to characterize a distribution is through the cumulative distribution function: $F(x) = P(X \leq x)$. The quantile function is the inverse to the cdf.

Example

Try to plot the cdf for the Normal(0,2) distribution vs the Normal(0,1) distribution.

You can estimate an empirical distribution function using ecdf.

QQ plots, CLT

Example

Investigate whether `scaleSS` is normally distributed. What happens if the sample means were calculated using 1000, 50, or 10 observations?
What happens if you don't normalize to have zero mean and variance 1 (but still scale with $\sqrt{n}$)? What if you don't scale at all?
How do you assess whether the straight line is "good"?

Dependence

Independence is crucial as a statistical concept.
Dependence (in a statistical) sense is very hard to describe.
From a statistical perspective, we really need independence after any systematic effect has been removed.
Correlation is often used to measure dependence.

An example: linear regression

```
> xx <- 1:10
> yy <- xx + rnorm(10)
> plot(xx, yy)
> cor(xx, yy)
> cor(xx, yy - xx)
```

What is important is that the *residuals* `yy-xx` look independent.

Correlation

Correlation between two (paired) variables is a number in $[-1, 1]$, with a 1 being strong positive relationship, -1 being a strong negative relationship. If the two variables are independent, they will have a correlation of zero.
Correlation is often described as a measure of the degree of *linear* relationship between the two variables.

$$\mathrm{cor}(X, aX + b) = \mathrm{sign}(a)$$

It is affected by nonlinear transformations (but not linear).

Example

Plot and discuss the following examples and whether correlation really captures the dependency. For the last one, compute the correlation before and after a log transformation

```
> yy <- rnorm(100)
> xx <- rnorm(100)
> yy <- c(rnorm(100), 5:10)
> xx <- c(rnorm(100), 5:10)
> yy <- c(rnorm(100), rnorm(100,
+     mean = 5))
> xx <- c(rnorm(100), rnorm(100,
+     mean = 5))
> yy <- c(rnorm(100), 5:10) + 3
> xx <- c(rnorm(100), 5:10) + 3
> xx <- c(rnorm(100, mean = 5), 8:10)
> yy <- xx + rnorm(103, sd = 0.5)
```

Correlation and outliers

Correlation is (together with many other measures) susceptible to outliers.

```
> cor(c(rnorm(1000), 50), c(rnorm(1000),
+     50))
[1] 0.7131679
```

This can go in both directions.
What is an outlier? Two perspectives:

- Outliers are “strange” data points, perhaps the result of bad measurements or bad record keeping.
- Outliers are just extreme data.

In the first case one might want to identify and deal with them – perhaps discard them. But watch out for discarding important data.

R^2 – explained variation

A measure often misused is R^2, which is described as the explained variation – “The proportion of variance explained by the model”.
For a linear regression (model) R^2 makes sense. For most other models it is very hard to define and use.
For a linear regression R^2 is simply the square of the correlation coefficient.

Misuse of R^2

If the data follow a linear regression model, R^2 does indeed say something valuable. However, R^2 does not necessarily tell you how good a fit a linear regression is to your data.
There is a famous illustration of this, namely Anscombe’s example:

```
> example(anscombe)
```

- Heteroscedasticity,
- non-linearity,
- outliers

all cause problems.

Simulating molecular evolution

We would like to simulate two DNA sequences from (say) chimpanzee and human using the Jukes-Cantor model of evolution.
Construct a function with two inputs t (time since divergence) and N (number of bases). Simulate the ancestral sequences by a simple iid. model. The function should return the evolved sequence and perhaps the ancestral sequence as well.

Simulating molecular evolution

In the Jukes-Cantor model, the probability of going from say A to C in time t is given by

$$P(C \mid A) = \exp(tQ), Q = \begin{pmatrix} -.75 & .25 & .25 & .25 \\ .25 & -.75 & .25 & .25 \\ .25 & .25 & -.75 & .25 \\ .25 & .25 & .25 & -.75 \end{pmatrix}$$

In order to compute $\exp(tQ)$ we will use the fact that if Q is diagonalized as $Q = UDU^{-1}$, with D being a diagonal matrix and U being an orthogonal matrix,

$$e^{Qt} = Ue^{Dt}U^{-1}$$

(This is called the eigen decomposition.)

Simulating molecular evolution

- You need this conditional distribution in order to generate the new evolved sequence. A convenient way to represent such a distribution is though a matrix.
- There are a lot of technical details here - we will ignore them. Our goal is to simulate data. Useful R functions are sample and eigen.