The interactive origin of iconiciy: Permutation tests

Load libraries

```
library(gplots)
library(lattice)
library(lme4)
```

Load data

```
finalLangs = read.csv("../data/finalLanguages/FinalLanguages.csv", stringsAsFactors = F)
# make a variable which stores condition, chain and generation
finalLangs$cond2 = paste(finalLangs$Cond,finalLangs$Chain, finalLangs$Gen)
```

Run permutation test

For each output language, take the difference in means in spikiness ratings between spiky and non-spiky meanings. Compare this to 1000 permutations of the numbers.

```
# Set the random seed for reproducibility
set.seed(1278)

# what factor should the data be split by?
split = finalLangs[finalLangs$cond2==unique(finalLangs$cond2)[1],]$Shape

# for each language (a single generation's output)
res = tapply(finalLangs$RatedSpikiness, finalLangs$cond2, function(X){
  # calculate the true difference
  trueDiff = -diff(tapply(X, split,mean))
  # permute the numbers and re-calculate difference
  permDiff = replicate(1000,
        {-diff(tapply(sample(X), split,mean)) })

  # work out p and z-socres
  p = 1- sum(trueDiff > permDiff ) / length(permDiff)
  z.score = (trueDiff - mean(permDiff)) / sd(permDiff)
  return(c(p,z.score))
})
```

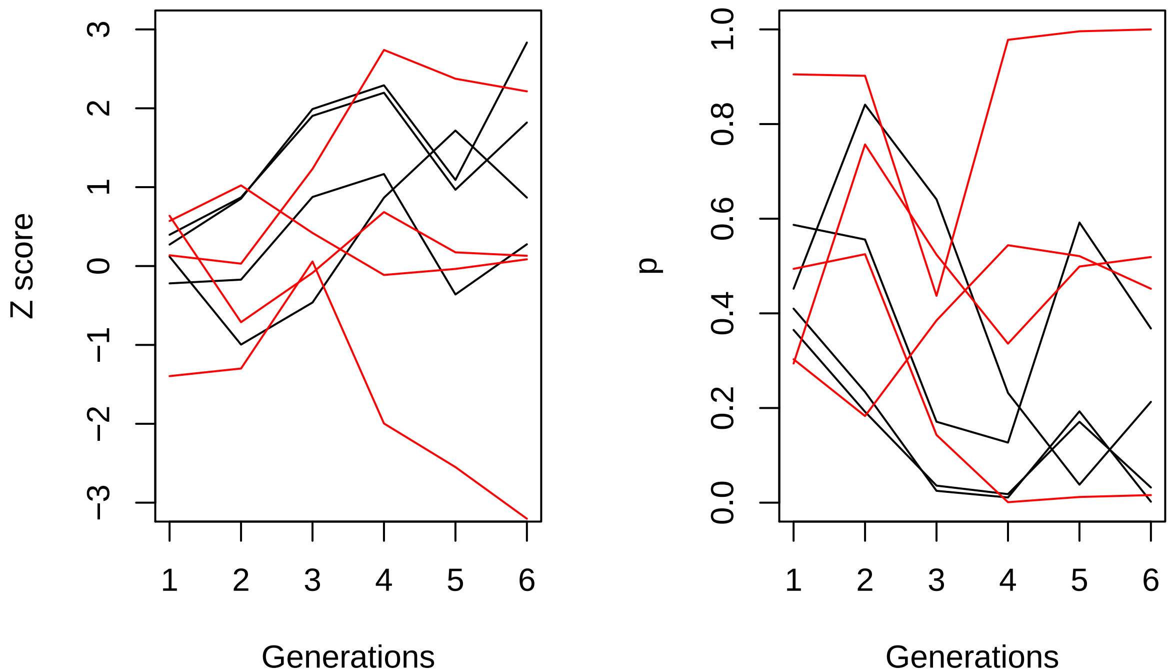
Recast results into data frame:

```
res2 = data.frame(
p = sapply(res,head,n=1),
z = sapply(res,tail,n=1),
t(sapply(names(res),function(X){
  strsplit(X," ")[[1]]
}))
```

```
, stringsAsFactors = F)
names(res2) = c("p",'z','condition','chain','gen')
```

Plot the results. Each line represents an independent chain. The red lines show the results for the Learning condition. The results suggest that there is no strong difference between the conditions. One of the learning chains decreases in iconicity, due to that chain focussing on specifying the colour rather than the shape.

```
par(mfrow=c(1,2))
plot(c(1,6), c(-3,3), type='n',ylab='Z score', xlab='Generations')
for(i in unique(res2$chain)){
  dx =res2[res2$chain==i,]
  lines(dx$gen,dx$z, col=c("black","red")[(dx$condition=="Learn")+1])
}

plot(c(1,6), c(0,1), type='n',ylab='p', xlab='Generations')
for(i in unique(res2$chain)){
  dx =res2[res2$chain==i,]
  lines(dx$gen,dx$p, col=c("black","red")[(dx$condition=="Learn")+1])
}
```




```
rownames(res2) = NULL
res2
```

```
##        p           z     condition chain gen
## 1  0.365  0.39651582 Communication     0   1
## 2  0.192  0.87023750 Communication     0   2
## 3  0.036  1.90398924 Communication     0   3
## 4  0.018  2.19686551 Communication     0   4
## 5  0.171  0.96732544 Communication     0   5
## 6  0.032  1.81968720 Communication     0   6
## 7  0.410  0.27163156 Communication     1   1
## 8  0.234  0.85438818 Communication     1   2
## 9  0.025  1.99017220 Communication     1   3
## 10 0.011  2.29125403 Communication     1   4
```

```
## 11 0.193  1.09194544 Communication     1   5
## 12 0.002  2.83378086 Communication     1   6
## 13 0.452  0.11931504 Communication     2   1
## 14 0.841 -0.99573110 Communication     2   2
## 15 0.641 -0.46244255 Communication     2   3
## 16 0.232  0.86643742 Communication     2   4
## 17 0.038  1.71762100 Communication     2   5
## 18 0.213  0.86711511 Communication     2   6
## 19 0.587 -0.21997100 Communication     3   1
## 20 0.556 -0.17252929 Communication     3   2
## 21 0.171  0.87557683 Communication     3   3
## 22 0.127  1.16617467 Communication     3   4
## 23 0.592 -0.36031071 Communication     3   5
## 24 0.368  0.27612849 Communication     3   6
## 25 0.303  0.57103499         Learn     4   1
## 26 0.183  1.02227196         Learn     4   2
## 27 0.385  0.41949120         Learn     4   3
## 28 0.544 -0.11348068         Learn     4   4
## 29 0.521 -0.03655111         Learn     4   5
## 30 0.452  0.08509545         Learn     4   6
## 31 0.494  0.13627918         Learn     5   1
## 32 0.525  0.02997840         Learn     5   2
## 33 0.143  1.23116378         Learn     5   3
## 34 0.001  2.73954890         Learn     5   4
## 35 0.012  2.37478332         Learn     5   5
## 36 0.016  2.21463208         Learn     5   6
## 37 0.905 -1.39552168         Learn     6   1
## 38 0.902 -1.29934990         Learn     6   2
## 39 0.437  0.05754949         Learn     6   3
## 40 0.978 -1.99616121         Learn     6   4
## 41 0.996 -2.54919070         Learn     6   5
## 42 1.000 -3.20395726         Learn     6   6
## 43 0.294  0.63720477         Learn     7   1
## 44 0.757 -0.71198396         Learn     7   2
## 45 0.524 -0.08605136         Learn     7   3
## 46 0.336  0.68413893         Learn     7   4
## 47 0.499  0.17370403         Learn     7   5
## 48 0.519  0.12991654         Learn     7   6
```

Looking only at red-coloured meanings

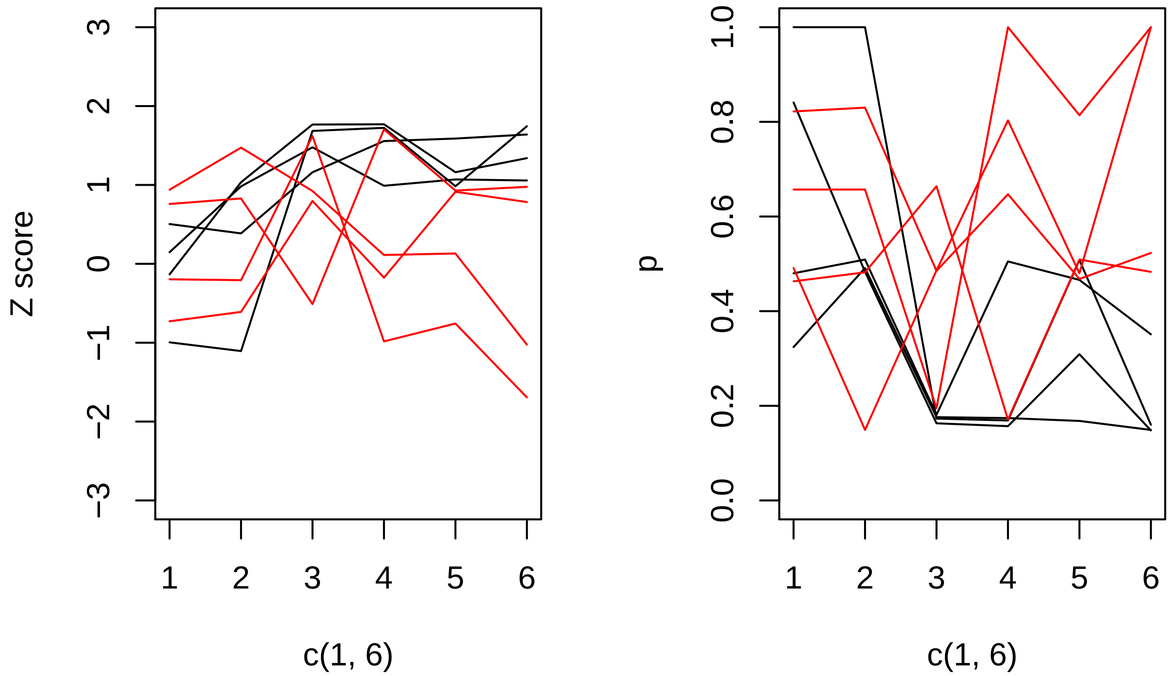
The mixed effects results suggested that there is a difference between conditions. The absence of an effect of condition on iconicity here is probably due to the fact that condition may interact with the other meaning dimension colour and border (e.g. iconicity may be stronger in words for red objects than green or blue objects).

Below we run the same analysis, but just for the colour 'Red'.

```
finalLangs2 = finalLangs[finalLangs$Colour=="Rojo",]


# Set the random seed for reproducibility
set.seed(1278)

# what factor should the data be split by?
split = finalLangs2[finalLangs2$cond2==unique(finalLangs2$cond2)[1],]$Shape

# for each language (a single generation's output)
resRed = tapply(finalLangs2$RatedSpikiness, finalLangs2$cond2, function(X){
  # calculate the true difference
  trueDiff = -diff(tapply(X, split,mean))
  # permute the numbers and re-calculate difference
  permDiff = replicate(1000,
        {-diff(tapply(sample(X), split,mean)) })

  # work out p and z-socres
  p = 1- sum(trueDiff > permDiff ) / length(permDiff)
  z.score = (trueDiff - mean(permDiff)) / sd(permDiff)
  return(c(p,z.score))
})
res2Red = data.frame(
p = sapply(resRed,head,n=1),
z = sapply(resRed,tail,n=1),
t(sapply(names(resRed),function(X){
  strsplit(X," ")[[1]]
}))
, stringsAsFactors = F)
names(res2Red) = c("p",'z','condition','chain','gen')
```

Plot the results. In this case, we do see a division between the two conditions by the last generation.

```
par(mfrow=c(1,2))
plot(c(1,6), c(-3,3), type='n',ylab='Z score')
for(i in unique(res2Red$chain)){
  dx =res2Red[res2Red$chain==i,]
  lines(dx$gen,dx$z, col=c("black","red")[(dx$condition=="Learn")+1])
}

plot(c(1,6), c(0,1), type='n',ylab='p')
for(i in unique(res2Red$chain)){
  dx =res2Red[res2Red$chain==i,]
  lines(dx$gen,dx$p, col=c("black","red")[(dx$condition=="Learn")+1])
}
```



Z score
3
2
1
0
−1
−2
−3
1
2
3
4
5
6
c(1, 6)
p
1.0
0.8
0.6
0.4
0.2
0.0
1
2
3
4
5
6
c(1, 6)