Epi Lab color code

Software/Packages/Add-ins required

Software/Packages/Add-ins recommended

Description text

R code to copy/paste into console

R code to copy/paste into console that needs adjustment to your personal workspace

Websites where you can download requirements

Lab #9 requirements

- R - http://cran.r-project.org/bin/windows/base/
- R Studio - www.rstudio.com/ide/download/desktop
- Internet connection
- R packages "irr' and "psych"

Inter-lab proficiency testing cont...

We continue with our short series on inter-lab proficiency testing and in this month's exercise we look at Kappa statistics which are often used to evaluate raters, which in our case are laboratories testing samples with a test under evaluation.

Please note that there is much debate about the usefulness of these statistics and certainly its still just a start to the analysis, but in the example we used with the ELISA testing where the outcome was a dichotomous "positive" or "negative" result we think its useful. A useful text on the subject can be found at: http://john-uebersax.com/stat/kappa.htm

Functions and Code covered - Lab 9

Cohen's kappa functions in the irr (**kappa2**) and the psych (**cohen.kappa**) packages with the latter giving confidence intervals.
kappam.fleiss function in the irr package for multiple rater evaluation

The code

#In Back Page Epi lab #8 we looked at the raw proportional analysis of 3 different labs testing an avian influenza ELISA kit. The following code is a repeat of that with a final output of the total raw agreement, the positive raw agreement and the negative raw agreement. Copy and paste the following GREEN text into your RStudio console and the result should be the same as that on the top of the following page and the same as what was seen at the end of lab #8.

```
rm(list = ls())
elisadata<-read.csv("http://www.jdata.co.za/backpagelabs/backpagelabs_jdg_agreementraw.csv")
rownames(elisadata)<- elisadata[,1]
elisadata<- elisadata[,-1]
elisadata[elisadata==''']<-NA
elisadata<-droplevels(elisadata)
lab1versuslab2_raw<-table(elisadata$lab1,elisadata$lab2,dnn=c(colnames(elisadata[1:2])))
l1l2raw<-(lab1versuslab2_raw[1,1]+lab1versuslab2_raw[2,2])/sum(lab1versuslab2_raw)
lab1versuslab3_raw<-table(elisadata$lab1,elisadata$lab3,dnn=c(colnames(elisadata[1]),colnames(elisadata[3])))
l1l3raw<-(lab1versuslab3_raw[1,1]+lab1versuslab3_raw[2,2])/sum(lab1versuslab3_raw)
lab2versuslab3_raw<-table(elisadata$lab2,elisadata$lab3,dnn=c(colnames(elisadata[2:3])))
l2l3raw<-(lab2versuslab3_raw[1,1]+lab2versuslab3_raw[2,2])/sum(lab2versuslab3_raw)
arrayRAW<-array(c(NA,l1l2raw,l1l3raw,l1l2raw,NA,l2l3raw,l1l3raw,l2l3raw,NA),c(3,3))
dimnames(arrayRAW)<-list(c("Lab1","Lab2","Lab3"),c("Lab1","Lab2","Lab3"))
lab1versuslab2_pos<-table(elisadata$lab1,elisadata$lab2,dnn=c(colnames(elisadata[1:2])))
l1l2pos<-lab1versuslab2_pos[2,2]/(lab1versuslab2_pos[2,2]+lab1versuslab2_pos[2,1]+lab1versuslab2_pos[1,2])
lab1versuslab3_pos<-table(elisadata$lab1,elisadata$lab3,dnn=c(colnames(elisadata[1]),colnames(elisadata[3])))
l1l3pos<-lab1versuslab3_pos[2,2]/(lab1versuslab3_pos[2,2]+lab1versuslab3_pos[2,1]+lab1versuslab3_pos[1,2])
lab2versuslab3_pos<-table(elisadata$lab2,elisadata$lab3,dnn=c(colnames(elisadata[2:3])))
l2l3pos<-lab2versuslab3_pos[2,2]/(lab2versuslab3_pos[2,2]+lab2versuslab3_pos[2,1]+lab2versuslab3_pos[1,2])
arraypos<-array(c(NA,l1l2pos,l1l3pos,l1l2pos,NA,l2l3pos,l1l3pos,l2l3pos,NA),c(3,3))
dimnames(arraypos)<-list(c("Lab1","Lab2","Lab3"),c("Lab1","Lab2","Lab3"))
lab1versuslab2_neg<-table(elisadata$lab1,elisadata$lab2,dnn=c(colnames(elisadata[1:2])))
l1l2neg<-lab1versuslab2_neg[1,1]/(lab1versuslab2_neg[1,1]+lab1versuslab2_neg[1,2]+lab1versuslab2_neg[2,1])
lab1versuslab3_neg<-table(elisadata$lab1,elisadata$lab3,dnn=c(colnames(elisadata[1]),colnames(elisadata[3])))
l1l3neg<-lab1versuslab3_neg[1,1]/(lab1versuslab3_neg[1,1]+lab1versuslab3_neg[1,2]+lab1versuslab3_neg[2,1])
lab2versuslab3_neg<-table(elisadata$lab2,elisadata$lab3,dnn=c(colnames(elisadata[2:3])))
l2l3neg<-lab2versuslab3_neg[1,1]/(lab2versuslab3_neg[1,1]+lab2versuslab3_neg[2,1])
arrayneg<-array(c(NA,l1l2neg,l1l3neg,l1l2neg,NA,l2l3neg,l1l3neg,l2l3neg,NA),c(3,3))
dimnames(arrayneg)<-list(c("Lab1","Lab2","Lab3"),c("Lab1","Lab2","Lab3"))
arrayRAW
arraypos
arrayneg
```

```
> arrayRAW
          Lab1      Lab2      Lab3
Lab1        NA 0.9191617 0.9807939
Lab2 0.9191617        NA 0.9261364
Lab3 0.9807939 0.9261364        NA
> arraypos
     Lab1      Lab2      Lab3
Lab1   NA 0.5500000 0.7000000
Lab2 0.55        NA 0.5272727
Lab3 0.70 0.5272727        NA
> arrayneg
          Lab1      Lab2      Lab3
Lab1        NA 0.9102990 0.9798928
Lab2 0.9102990        NA 0.9195046
Lab3 0.9798928 0.9195046        NA
```

#Based on this result it seemed as if Lab's1 and 3 have decent raw agreement compared to Lab's 1 and 2 and Lab's 2 and 3. We also clearly saw in this example how the negative agreement masked some poor positive agreement, which might have been missed if total raw agreement was the only proportional agreement evaluated.

#This month we go beyond the raw analysis and evaluate rater agreement using some statistical methods. There are a number of packages in R which may help, and in this lab we'll be using two of them. The first is `irr`, which from its description file is evaluates "`Various Coefficiants of Interrater Reliability and Agreement`", which sounds pretty much exactly what we are looking for!

```
install.packages("irr")
library(irr)
```

#Keep in mind we are dealing with dichotomous data in this example. The one important aspect of our data is that we have more than 2 raters (we have 3 i.e. the 3 labs) which influences the tests we can use to evaluate them all at once. To start we evaluate each lab to each other on a one-on-one basis, in a similar way to how we have done with the raw agreement. For that we can use the **Cohen's kappa**, which establishes whether agreement exceeds that expected under the null hypothesis of random ratings. Kappa statistics results generally fall between 0 (poor agreement) and 1 (perfect agreement)

A

```
# As mentioned in the preamble: kappa coefficients are not necessarily the ideal test to use, there are considerations to take - read these at http://john-uebersax.com/stat/kappa.htm
cohen1vs2irr<-kappa2(elisadata[,c(1,2)]);cohen1vs2irr
```

#note the "`[,c(1,2])]`" above indicates that you want to test the `elisadata` data set but that you want to use all available rows (blank space before the ",c") and that you only want to use the first 2 columns (which in this case happen to be Lab 1 and Lab2).
#the result returns a number of aspects, including the method used, the number of subjects evaluated, the number of raters, the name of the coefficient and its value, and the p value of the test.

B

```
# if you are looking for confidence intervals then the psych library may be more helpful.
install.packages("psych")
library(psych)
cohen1vs2irrpsych<-cohen.kappa(elisadata[,c(1,2)]);cohen1vs2irrpsych
```

#one really nice part of the psych package is that instead of piecemeal working out the kappa statistic for each lab combination (like we did in the raw agreement and what we would have done with the `irr` package) you can compare each lab to each other in one function

C

#Since we only have 3 columns in our data set the function is simply:

```
cohen.kappa(elisadata)
```

#again, when you have more columns in your dataset you'd need to be more specific - so the following function is identical

```
cohen.kappa(elisadata[,c(1,2,3)])
```

D

#for the detailed output use the print function with all=TRUE

```
print(cohen.kappa(elisadata), all=TRUE)
```

#so there you have a matrix of kappa statistics for each lab combination with a confidence interval of each statistic. Very similar to the raw agreement we see that Lab 1 and 3 have better agreement than lab 1 and 2, although here the worst agreement is between lab 2 and 3, which differs slightly from the raw agreement where the worst agreement was generally between lab 1 and 2.

#This leads us to a further question - can we evaluate the agreement between the labs as a whole (so not each combination)? Cohen's kappa is only useful for 2 raters, so how do we evaluate the entire dataset where a result has been given by each lab - and in this example possibly establish whether this ELISA could be used with confidence between all labs testing with it?

#we jump back to the `irr` package now and use the `kappam.fleiss` function. The requirement for the Fleiss' kappa test is that only binary (dichotomous) or nominal scales can be used, which fits into our dataset well.

E

```
elisadata.fleiss<-kappam.fleiss(elisadata);elisadata.fleiss
```

#again the result set will have multiple components listed with it

#these last 2 labs are just an introduction to rater agreement, I hope it gets you going when encountering this problem. My conclusions from the last 2 labs for the ELISA data would be

- raw agreement shows strong negative agreement but there may be issues with positive agreement.
- Cohen's kappa shows good agreement between lab1 and 3 compared to the other two combinations of lab 2 and 3 and lab 1 and 2. This may prompt an enquiry to lab 2 as to their ELISA technique/workflow and possibly try identify any issues they may be having to improve the system?
- the total agreement is probably good enough to warrant further work on evaluating the test

#a last note on the kappa statistics reported here- it seems as if some authors have put subjective values across the board to agreement - i.e. to say that if agreement is between 0.6 and 0.8 its substantial and if its 0.2 - 0.4 its fair agreement, but this can apparently be a dangerous path to follow since situations differ between raters and tests. Rather evaluate the kappa statistic on the data you are working with and if you compare it compare it with a very similar situation.

References

R Core Team (2014). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/

Matthias Gamer, Jim Lemon and Ian Fellows Puspendra Singh(2012). irr: Various Coefficients of Interrater Reliability and Agreement. R package version 0.84. http://CRAN.R-project.org/package=irr

Revelle, W. (2014) psych: Procedures for Personality and Psychological Research, Northwestern University, Evanston, Illinois, USA, http://CRAN.R-project.org/package=psych Version=1.4.4.

The result

```
Cohen's Kappa for 2 Raters (Weights: unweighted)

Subjects = 334
  Raters = 2
   Kappa = 0.663

       z = 12.2
 p-value = 0
```

A

```
Call: cohen.kappa1(x = x, w = w, n.obs = n.obs, alpha = alpha)

Cohen Kappa and Weighted Kappa correlation coefficients and confidence boundaries
                 lower estimate upper
unweighted kappa  0.55     0.66  0.78
weighted kappa    0.55     0.66  0.78

 Number of subjects = 334
```

B

```
Cohen Kappa (below the diagonal) and Weighted Kappa (above the diagonal)
For confidence intervals and detail print with all=TRUE
     lab1 lab2 lab3
lab1 1.00 0.66 0.81
lab2 0.66 1.00 0.65
lab3 0.81 0.65 1.00
```

C

```
$cohen.kappa
          lab1      lab2      lab3
lab1 1.0000000 0.6634823 0.8136246
lab2 0.6634823 1.0000000 0.6489721
lab3 0.8136246 0.6489721 1.0000000

$`lab1 lab2`
Call: cohen.kappa1(x = x1, w = w, n.obs = n.obs, alpha = alpha)

Cohen Kappa and Weighted Kappa correlation coefficients and confidence boundaries
                 lower estimate upper
unweighted kappa  0.55     0.66  0.78
weighted kappa    0.55     0.66  0.78

 Number of subjects = 334

$`lab1 lab3`
Call: cohen.kappa1(x = x1, w = w, n.obs = n.obs, alpha = alpha)

Cohen Kappa and Weighted Kappa correlation coefficients and confidence boundaries
                 lower estimate upper
unweighted kappa  0.72     0.81  0.91
weighted kappa    0.72     0.81  0.91

 Number of subjects = 781

$`lab2 lab3`
Call: cohen.kappa1(x = x1, w = w, n.obs = n.obs, alpha = alpha)

Cohen Kappa and Weighted Kappa correlation coefficients and confidence boundaries
                 lower estimate upper
unweighted kappa  0.52     0.65  0.77
weighted kappa    0.52     0.65  0.77

 Number of subjects = 352
```

D - Matrix of all results

D - Lab 1 vs Lab 2

D - Lab 1 vs Lab 3

D - Lab 2 vs Lab 3

D

```
Fleiss' Kappa for m Raters

Subjects = 322
  Raters = 3
   Kappa = 0.734

       z = 22.8
 p-value = 0
```

E