InterRaterAgreementReproducibility

Purpose

This calculates kappa between an expert and a non-expert for rating availability of code and data for the 28 papers published in BioNLP 2016. I've done two versions: with different numbers of categories for the two raters ("Yes" and "No" for the expert, versus "Yes", "No", and "Maybe" for the non-expert), and with identical numbers of categories for the two raters. In the later case, I added the "Maybe" category for the expert, and left those cells empty. When I calculated Kappa with different numbers of categories, the results made no sense at all, so I'm reporting only the numbers for the same number of categories (i.e., where I added an empty column for "Maybe" for the expert).

Data

See the file XX at XX.github.xx.

```
code.different.categories <- as.data.frame(rbind(c(9, 1), c(0, 13), c(3, 2)))
code.same.categories <-      as.data.frame(rbind(c(9, 1, 0), c(0, 13, 0), c(3, 2, 0)))

data.different.categories <- as.data.frame(rbind(c(15, 3), c(1, 7), c(1, 1)))
data.same.categories <- as.data.frame(rbind(c(15, 3, 0), c(1, 7, 0), c(1, 1, 0)))
```

```
# Cohen's kappa for two raters
cohen.kappa(code.different.categories)
```

```
## Warning in cohen.kappa1(x, w = w, n.obs = n.obs, alpha = alpha): upper or
## lower confidence interval exceed abs(1) and set to +/- 1.
```

```
## 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     0.00  0.00
## weighted kappa     -1    -0.98 -0.56
##
##   Number of subjects = 3
```

```
cohen.kappa(code.same.categories)
```

```
## 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  0.411     0.63  0.85
## weighted kappa   -0.043     0.34  0.72
##
##   Number of subjects = 28
```

```r
cohen.kappa(data.different.categories)
```

```
## 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 -0.12     0.14  0.40
## weighted kappa   -0.69     0.00  0.69
##
##  Number of subjects = 3
```

```r
cohen.kappa(data.same.categories)
```

```
## 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  0.29     0.57  0.85
## weighted kappa    0.13     0.49  0.85
##
##  Number of subjects = 28
```

The 5-annotator study of a stratified sample of natural language processing papers

```r
all.annotators.data <- read.table("/Users/kev/Documents/InterRaterAgreementReproducibility/Reproducibil
                                  header=T, sep=",")
```

what happens with missing values??

‘‘{r calculate.kappa.stratified.sample}

```r
# need a function to calculate confusion matrix between two annotators where the number of papers annot
#my.table <- as.table(rbind(c(0, 1, 1), c(1, 1, 1)))
#confusionMatrix(my.table)

annotators <- c("ann_kev", "ann_jin", "ann_aur", "ann_pra", "ann_neg")
#pairs.of.annotators <- pairs(annotators)
pairs.of.annotators <- combn(annotators, 2)
for(i in 1:ncol(pairs.of.annotators)) {
  #print(all.annotators.data$pairs.of.annotators[1,i])
  #print(all.annotators.data$pairs.of.annotators[2,i])
  annotator.1 <- pairs.of.annotators[1,i]
  annotator.2 <- pairs.of.annotators[2,i]
  #print(all.annotators.data$ann_kev)
```

```r
  print(annotator.1)
  #print(all.annotators.data[,annotator.1])
  print(annotator.2)
  #print(all.annotators.data[,annotator.2])
  my.confusion.matrix <- confusionMatrix(all.annotators.data[,annotator.1], all.annotators.data[,annota
  #print(summary(my.confusion.matrix))
  print(my.confusion.matrix)
}
```

```
## [1] "ann_kev"
## [1] "ann_jin"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 7 0
##          1 1 1
##
##                Accuracy : 0.8889
##                  95% CI : (0.5175, 0.9972)
##     No Information Rate : 0.8889
##     P-Value [Acc > NIR] : 0.7362
##
##                   Kappa : 0.6087
##  Mcnemar's Test P-Value : 1.0000
##
##             Sensitivity : 0.8750
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.5000
##              Prevalence : 0.8889
##          Detection Rate : 0.7778
##    Detection Prevalence : 0.7778
##       Balanced Accuracy : 0.9375
##
##        'Positive' Class : 0
##
## [1] "ann_kev"
## [1] "ann_aur"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 10  2
##          1  0  4
##
##                Accuracy : 0.875
##                  95% CI : (0.6165, 0.9845)
##     No Information Rate : 0.625
##     P-Value [Acc > NIR] : 0.02917
##
##                   Kappa : 0.7143
##  Mcnemar's Test P-Value : 0.47950
```

3

```
## 
##               Sensitivity : 1.0000
##               Specificity : 0.6667
##            Pos Pred Value : 0.8333
##            Neg Pred Value : 1.0000
##                Prevalence : 0.6250
##            Detection Rate : 0.6250
##      Detection Prevalence : 0.7500
##         Balanced Accuracy : 0.8333
## 
##          'Positive' Class : 0
## 
## [1] "ann_kev"
## [1] "ann_pra"
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction 0 1
##          0 7 5
##          1 0 4
## 
##                  Accuracy : 0.6875
##                    95% CI : (0.4134, 0.8898)
##       No Information Rate : 0.5625
##       P-Value [Acc > NIR] : 0.22690
## 
##                     Kappa : 0.4118
##    Mcnemar's Test P-Value : 0.07364
## 
##               Sensitivity : 1.0000
##               Specificity : 0.4444
##            Pos Pred Value : 0.5833
##            Neg Pred Value : 1.0000
##                Prevalence : 0.4375
##            Detection Rate : 0.4375
##      Detection Prevalence : 0.7500
##         Balanced Accuracy : 0.7222
## 
##          'Positive' Class : 0
## 
## [1] "ann_kev"
## [1] "ann_neg"
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction  0  1
##          0 11  0
##          1  1  2
## 
##                  Accuracy : 0.9286
##                    95% CI : (0.6613, 0.9982)
##       No Information Rate : 0.8571
##       P-Value [Acc > NIR] : 0.3851
## 
```

```
##                   Kappa : 0.7586
##  Mcnemar's Test P-Value : 1.0000
##
##             Sensitivity : 0.9167
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.6667
##              Prevalence : 0.8571
##          Detection Rate : 0.7857
##    Detection Prevalence : 0.7857
##       Balanced Accuracy : 0.9583
##
##        'Positive' Class : 0
##
## [1] "ann_jin"
## [1] "ann_aur"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 5 3
##          1 0 1
##
##                Accuracy : 0.6667
##                  95% CI : (0.2993, 0.9251)
##     No Information Rate : 0.5556
##     P-Value [Acc > NIR] : 0.3743
##
##                   Kappa : 0.2703
##  Mcnemar's Test P-Value : 0.2482
##
##             Sensitivity : 1.0000
##             Specificity : 0.2500
##          Pos Pred Value : 0.6250
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5556
##          Detection Rate : 0.5556
##    Detection Prevalence : 0.8889
##       Balanced Accuracy : 0.6250
##
##        'Positive' Class : 0
##
## [1] "ann_jin"
## [1] "ann_pra"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 3 5
##          1 0 1
##
##                Accuracy : 0.4444
##                  95% CI : (0.137, 0.788)
##     No Information Rate : 0.6667
```

```
##       P-Value [Acc > NIR] : 0.95758
##
##                    Kappa : 0.1176
##   Mcnemar's Test P-Value : 0.07364
##
##              Sensitivity : 1.0000
##              Specificity : 0.1667
##           Pos Pred Value : 0.3750
##           Neg Pred Value : 1.0000
##               Prevalence : 0.3333
##           Detection Rate : 0.3333
##     Detection Prevalence : 0.8889
##        Balanced Accuracy : 0.5833
##
##         'Positive' Class : 0
##
## [1] "ann_jin"
## [1] "ann_neg"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 7 1
##          1 0 1
##
##                 Accuracy : 0.8889
##                   95% CI : (0.5175, 0.9972)
##      No Information Rate : 0.7778
##      P-Value [Acc > NIR] : 0.372
##
##                    Kappa : 0.6087
##   Mcnemar's Test P-Value : 1.000
##
##              Sensitivity : 1.0000
##              Specificity : 0.5000
##           Pos Pred Value : 0.8750
##           Neg Pred Value : 1.0000
##               Prevalence : 0.7778
##           Detection Rate : 0.7778
##     Detection Prevalence : 0.8889
##        Balanced Accuracy : 0.7500
##
##         'Positive' Class : 0
##
## [1] "ann_aur"
## [1] "ann_pra"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 9 8
##          1 1 7
##
##                 Accuracy : 0.64
```

```
##                  95% CI : (0.4252, 0.8203)
##     No Information Rate : 0.6
##     P-Value [Acc > NIR] : 0.4246
##
##                   Kappa : 0.3284
##  Mcnemar's Test P-Value : 0.0455
##
##             Sensitivity : 0.9000
##             Specificity : 0.4667
##          Pos Pred Value : 0.5294
##          Neg Pred Value : 0.8750
##              Prevalence : 0.4000
##          Detection Rate : 0.3600
##    Detection Prevalence : 0.6800
##       Balanced Accuracy : 0.6833
##
##        'Positive' Class : 0
##
## [1] "ann_aur"
## [1] "ann_neg"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 9 0
##          1 3 2
##
##                Accuracy : 0.7857
##                  95% CI : (0.492, 0.9534)
##     No Information Rate : 0.8571
##     P-Value [Acc > NIR] : 0.8719
##
##                   Kappa : 0.4615
##  Mcnemar's Test P-Value : 0.2482
##
##             Sensitivity : 0.7500
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.4000
##              Prevalence : 0.8571
##          Detection Rate : 0.6429
##    Detection Prevalence : 0.6429
##       Balanced Accuracy : 0.8750
##
##        'Positive' Class : 0
##
## [1] "ann_pra"
## [1] "ann_neg"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 6 0
##          1 6 2
```

```
##
##                Accuracy : 0.5714
##                  95% CI : (0.2886, 0.8234)
##     No Information Rate : 0.8571
##     P-Value [Acc > NIR] : 0.99835
##
##                   Kappa : 0.2222
##  Mcnemar's Test P-Value : 0.04123
##
##             Sensitivity : 0.5000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.2500
##              Prevalence : 0.8571
##          Detection Rate : 0.4286
##    Detection Prevalence : 0.4286
##       Balanced Accuracy : 0.7500
##
##          'Positive' Class : 0
##
```

```r
# No NAs: works fine
#library(caret)
#junk.1 <- c(0, 1, 1)
#junk.2 <- c(1, 0, 1)
#confusionMatrix(junk.1, junk.2)
```

```r
# now let's trim these down to only the papers for which the two
# annotators both have an annotation.
only.papers.annotated.by.both <- function(annotator.1, annotator.2) {
  # validate your input a bit--both vectors should be the same length
  if (length(annotator.1) != length(annotator.2)) {
    print("XXXX Annotation vector lengths different... XXXX")
  }

  # OK, the data's somewhat validated, so go through both vectors
  # and keep only the papers that were annotated by both of them.
  # Note that you have to consider the possibility where the annotated
  # papers are not continuous, so you can't just stop as soon as you
  # hit an NA.

  # initialize a couple things you'll need
  annotator.1.filtered <- c()
  annotator.2.filtered <- c()

  for(i in 1:length(annotator.1)) {
    if (is.na(annotator.1[i]) | is.na(annotator.2[i])) {
      #print("Beurk...")
      # ...and do nothing else
    } else {
      #print("Youpie")
      annotator.1.filtered <- c(annotator.1.filtered, annotator.1[i])
      annotator.2.filtered <- c(annotator.2.filtered, annotator.2[i])
    }
```

```r
  } # close for-loop through input vectors
  #return(annotator.1, annotator.2)
  return.value <- as.data.frame(cbind(annotator.1.filtered, annotator.2.filtered))
  return(return.value)
}
```

Now we'll take out the missing values.

```r
# need a function to calculate confusion matrix between two annotators where the number of papers annot
#my.table <- as.table(rbind(c(0, 1, 1), c(1, 1, 1)))
#confusionMatrix(my.table)

annotators <- c("ann_kev", "ann_jin", "ann_aur", "ann_pra", "ann_neg")
#pairs.of.annotators <- pairs(annotators)
pairs.of.annotators <- combn(annotators, 2)

pairwise.kappas <- c()
pairwise.accuracies <- c()

for(i in 1:ncol(pairs.of.annotators)) {
  #print(all.annotators.data$pairs.of.annotators[1,i])
  #print(all.annotators.data$pairs.of.annotators[2,i])
  annotator.1 <- pairs.of.annotators[1,i]
  annotator.2 <- pairs.of.annotators[2,i]
  #print(all.annotators.data$ann_kev)
  print(annotator.1)
  #print(all.annotators.data[,annotator.1])
  print(annotator.2)
  #print(all.annotators.data[,annotator.2])

  #all.annotators.data[,annotator.1] <- only.papers.annotated.by.both(all.annotators.data[,annotator.1]

  #summary(only.papers.annotated.by.both)
  #my.confusion.matrix <- confusionMatrix(all.annotators.data[,1], all.annotators.data[,2])

  just.papers.annotated.by.both <- only.papers.annotated.by.both(all.annotators.data[,annotator.1], all

  my.confusion.matrix <- confusionMatrix(just.papers.annotated.by.both[,1], just.papers.annotated.by.bo
  print(my.confusion.matrix)

  #print(attributes(my.confusion.matrix))


  pairwise.kappas <- c(pairwise.kappas, my.confusion.matrix$overall["Kappa"])
  #print("Kappa: ")
  #print(my.kappa)
  pairwise.accuracies <- c(pairwise.accuracies, my.confusion.matrix$overall["Accuracy"])
  #print("Accuracy: ")
  #print(my.accuracy)
}
```

```
## [1] "ann_kev"
## [1] "ann_jin"
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 7 0
##          1 1 1
##
##                Accuracy : 0.8889
##                  95% CI : (0.5175, 0.9972)
##     No Information Rate : 0.8889
##     P-Value [Acc > NIR] : 0.7362
##
##                   Kappa : 0.6087
##  Mcnemar's Test P-Value : 1.0000
##
##             Sensitivity : 0.8750
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.5000
##              Prevalence : 0.8889
##          Detection Rate : 0.7778
##    Detection Prevalence : 0.7778
##       Balanced Accuracy : 0.9375
##
##        'Positive' Class : 0
##
## [1] "ann_kev"
## [1] "ann_aur"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 10  2
##          1  0  4
##
##                Accuracy : 0.875
##                  95% CI : (0.6165, 0.9845)
##     No Information Rate : 0.625
##     P-Value [Acc > NIR] : 0.02917
##
##                   Kappa : 0.7143
##  Mcnemar's Test P-Value : 0.47950
##
##             Sensitivity : 1.0000
##             Specificity : 0.6667
##          Pos Pred Value : 0.8333
##          Neg Pred Value : 1.0000
##              Prevalence : 0.6250
##          Detection Rate : 0.6250
##    Detection Prevalence : 0.7500
##       Balanced Accuracy : 0.8333
##
##        'Positive' Class : 0
##
```

```
## [1] "ann_kev"
## [1] "ann_pra"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 7 5
##          1 0 4
##
##                Accuracy : 0.6875
##                  95% CI : (0.4134, 0.8898)
##     No Information Rate : 0.5625
##     P-Value [Acc > NIR] : 0.22690
##
##                   Kappa : 0.4118
##  Mcnemar's Test P-Value : 0.07364
##
##             Sensitivity : 1.0000
##             Specificity : 0.4444
##          Pos Pred Value : 0.5833
##          Neg Pred Value : 1.0000
##              Prevalence : 0.4375
##          Detection Rate : 0.4375
##    Detection Prevalence : 0.7500
##       Balanced Accuracy : 0.7222
##
##        'Positive' Class : 0
##
## [1] "ann_kev"
## [1] "ann_neg"
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  0  1
##          0 11  0
##          1  1  2
##
##                Accuracy : 0.9286
##                  95% CI : (0.6613, 0.9982)
##     No Information Rate : 0.8571
##     P-Value [Acc > NIR] : 0.3851
##
##                   Kappa : 0.7586
##  Mcnemar's Test P-Value : 1.0000
##
##             Sensitivity : 0.9167
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.6667
##              Prevalence : 0.8571
##          Detection Rate : 0.7857
##    Detection Prevalence : 0.7857
##       Balanced Accuracy : 0.9583
##
```

```
##          'Positive' Class : 0
##
## [1] "ann_jin"
## [1] "ann_aur"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 5 3
##          1 0 1
##
##               Accuracy : 0.6667
##                 95% CI : (0.2993, 0.9251)
##     No Information Rate : 0.5556
##     P-Value [Acc > NIR] : 0.3743
##
##                  Kappa : 0.2703
##  Mcnemar's Test P-Value : 0.2482
##
##            Sensitivity : 1.0000
##            Specificity : 0.2500
##         Pos Pred Value : 0.6250
##         Neg Pred Value : 1.0000
##             Prevalence : 0.5556
##         Detection Rate : 0.5556
##   Detection Prevalence : 0.8889
##      Balanced Accuracy : 0.6250
##
##          'Positive' Class : 0
##
## [1] "ann_jin"
## [1] "ann_pra"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 3 5
##          1 0 1
##
##               Accuracy : 0.4444
##                 95% CI : (0.137, 0.788)
##     No Information Rate : 0.6667
##     P-Value [Acc > NIR] : 0.95758
##
##                  Kappa : 0.1176
##  Mcnemar's Test P-Value : 0.07364
##
##            Sensitivity : 1.0000
##            Specificity : 0.1667
##         Pos Pred Value : 0.3750
##         Neg Pred Value : 1.0000
##             Prevalence : 0.3333
##         Detection Rate : 0.3333
##   Detection Prevalence : 0.8889
```

```
##       Balanced Accuracy : 0.5833
##
##          'Positive' Class : 0
##
## [1] "ann_jin"
## [1] "ann_neg"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 7 1
##          1 0 1
##
##                Accuracy : 0.8889
##                  95% CI : (0.5175, 0.9972)
##     No Information Rate : 0.7778
##     P-Value [Acc > NIR] : 0.372
##
##                   Kappa : 0.6087
##  Mcnemar's Test P-Value : 1.000
##
##             Sensitivity : 1.0000
##             Specificity : 0.5000
##          Pos Pred Value : 0.8750
##          Neg Pred Value : 1.0000
##              Prevalence : 0.7778
##          Detection Rate : 0.7778
##    Detection Prevalence : 0.8889
##       Balanced Accuracy : 0.7500
##
##          'Positive' Class : 0
##
## [1] "ann_aur"
## [1] "ann_pra"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 9 8
##          1 1 7
##
##                Accuracy : 0.64
##                  95% CI : (0.4252, 0.8203)
##     No Information Rate : 0.6
##     P-Value [Acc > NIR] : 0.4246
##
##                   Kappa : 0.3284
##  Mcnemar's Test P-Value : 0.0455
##
##             Sensitivity : 0.9000
##             Specificity : 0.4667
##          Pos Pred Value : 0.5294
##          Neg Pred Value : 0.8750
##              Prevalence : 0.4000
```

```
##         Detection Rate : 0.3600
##   Detection Prevalence : 0.6800
##      Balanced Accuracy : 0.6833
##
##       'Positive' Class : 0
##
## [1] "ann_aur"
## [1] "ann_neg"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##         0 9 0
##         1 3 2
##
##               Accuracy : 0.7857
##                 95% CI : (0.492, 0.9534)
##    No Information Rate : 0.8571
##    P-Value [Acc > NIR] : 0.8719
##
##                  Kappa : 0.4615
##  Mcnemar's Test P-Value : 0.2482
##
##            Sensitivity : 0.7500
##            Specificity : 1.0000
##         Pos Pred Value : 1.0000
##         Neg Pred Value : 0.4000
##             Prevalence : 0.8571
##         Detection Rate : 0.6429
##   Detection Prevalence : 0.6429
##      Balanced Accuracy : 0.8750
##
##       'Positive' Class : 0
##
## [1] "ann_pra"
## [1] "ann_neg"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##         0 6 0
##         1 6 2
##
##               Accuracy : 0.5714
##                 95% CI : (0.2886, 0.8234)
##    No Information Rate : 0.8571
##    P-Value [Acc > NIR] : 0.99835
##
##                  Kappa : 0.2222
##  Mcnemar's Test P-Value : 0.04123
##
##            Sensitivity : 0.5000
##            Specificity : 1.0000
##         Pos Pred Value : 1.0000
```

```
##           Neg Pred Value : 0.2500
##                Prevalence : 0.8571
##            Detection Rate : 0.4286
##      Detection Prevalence : 0.4286
##         Balanced Accuracy : 0.7500
##
##          'Positive' Class : 0
##
```

```
# No NAs: works fine
#library(caret)
#junk.1 <- c(0, 1, 1)
#junk.2 <- c(1, 0, 1)
#confusionMatrix(junk.1, junk.2)
```

Now let's plot kappa and accuracies, just so that we can get a sense of how much variability there is in them. Accuracy is percent agreement, in some terminologies.

```
mean.kappa <- mean(pairwise.kappas)
mean.kappa <- round(mean.kappa, digits=2)
standard.deviation.kappa <- sd(pairwise.kappas)
standard.deviation.kappa <- round(standard.deviation.kappa, digits=2)
mean.accuracy <- mean(pairwise.accuracies)
mean.accuracy <- round(mean.accuracy, digits=2)
standard.deviation.accuracy <- sd(pairwise.accuracies)
standard.deviation.accuracy <- round(standard.deviation.accuracy, digits=2)
#table.main.title <- paste("Kappa and percent agreement for all pairs of annotators\n Kappa mean ", mea
table.main.title <- paste("Kappa and percent agreement for all pairs of annotators")
plot(pairwise.accuracies, pairwise.kappas,
     main=table.main.title,
     xlim=c(0, 1.0),
     xlab="Percent agreement",
     ylim=c(0, 1.0),
     ylab="Kappa"
     )
```

Kappa and percent agreement for all pairs of annotators
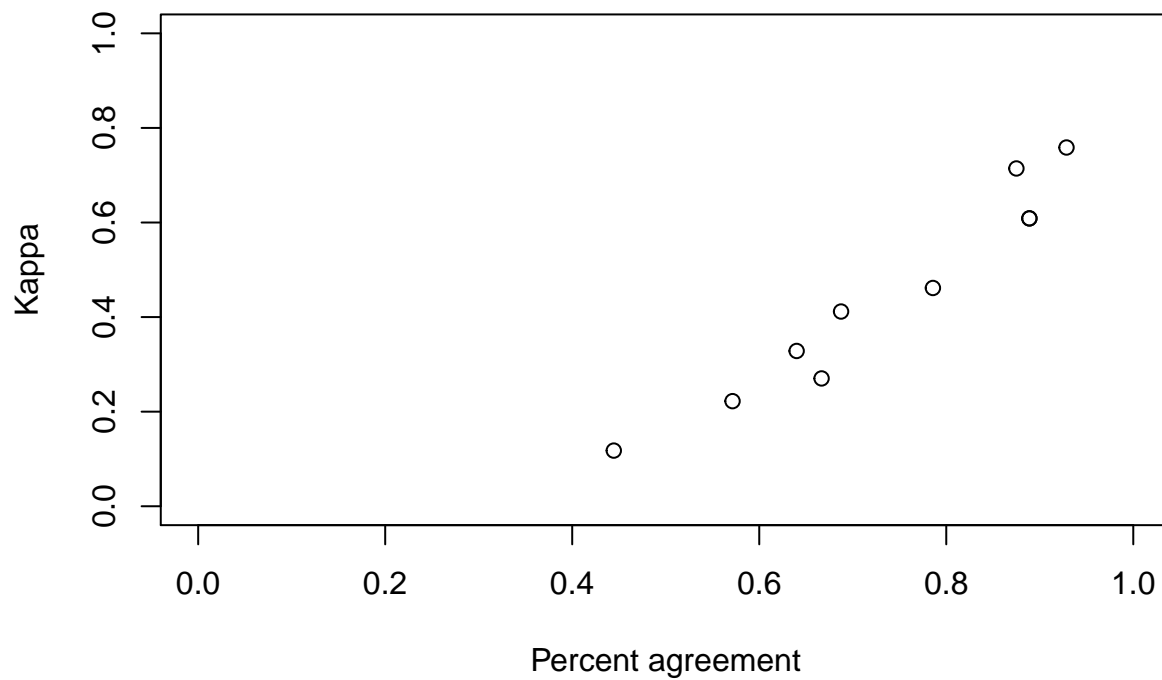

```r
print("Pairwise kappa mean:")
```

```
## [1] "Pairwise kappa mean:"
```

```r
print(mean.kappa)
```

```
## [1] 0.45
```

```r
print("Pairwise kappa standard deviation:")
```

```
## [1] "Pairwise kappa standard deviation:"
```

```r
print(standard.deviation.kappa)
```

```
## [1] 0.22
```

```r
print("Pairwise kappas:")
```

```
## [1] "Pairwise kappas:"
```

```r
sorted.pairwise.kappas <- sort(pairwise.kappas)
print(sorted.pairwise.kappas)
```

```
##     Kappa     Kappa     Kappa     Kappa     Kappa     Kappa     Kappa
## 0.1176471 0.2222222 0.2702703 0.3283582 0.4117647 0.4615385 0.6086957
##     Kappa     Kappa     Kappa
## 0.6086957 0.7142857 0.7586207
```

```r
print("Pairwise percent agreement mean:")
```

```
## [1] "Pairwise percent agreement mean:"
```

```r
print(mean.accuracy)
```

```
## [1] 0.74
```

```r
print("Pairwise percent agreement standard deviation:")
```

```
## [1] "Pairwise percent agreement standard deviation:"
```

```r
print(standard.deviation.accuracy)
```

```
## [1] 0.16
```

```r
print("Pairwise percent agreement:")
```

```
## [1] "Pairwise percent agreement:"
```

```r
sorted.pairwise.accuracies <- sort(pairwise.accuracies)
print(sorted.pairwise.accuracies)
```

```
##  Accuracy  Accuracy  Accuracy  Accuracy  Accuracy  Accuracy  Accuracy
## 0.4444444 0.5714286 0.6400000 0.6666667 0.6875000 0.7857143 0.8750000
##  Accuracy  Accuracy  Accuracy
## 0.8888889 0.8888889 0.9285714
```

Document session for reproducibility purposes.

```r
sessionInfo()
```

```
## R version 3.3.1 (2016-06-21)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: OS X 10.11.6 (El Capitan)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] caret_6.0-71   ggplot2_2.1.0   lattice_0.20-33 psych_1.6.12
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.6        nloptr_1.0.4       plyr_1.8.4
##  [4] class_7.3-14       iterators_1.0.8    tools_3.3.1
```

```
##  [7] digest_0.6.9       lme4_1.1-12      evaluate_0.10
## [10] gtable_0.2.0       nlme_3.1-128     mgcv_1.8-12
## [13] Matrix_1.2-6       foreach_1.4.3    yaml_2.1.14
## [16] parallel_3.3.1     SparseM_1.72     e1071_1.6-8
## [19] stringr_1.0.0      knitr_1.15.1     MatrixModels_0.4-1
## [22] stats4_3.3.1       rprojroot_1.1    grid_3.3.1
## [25] nnet_7.3-12        foreign_0.8-66   rmarkdown_1.2
## [28] minqa_1.2.4        reshape2_1.4.1   car_2.1-3
## [31] magrittr_1.5       backports_1.0.4  scales_0.4.0
## [34] codetools_0.2-14   htmltools_0.3.5  MASS_7.3-45
## [37] splines_3.3.1      pbkrtest_0.4-6   mnormt_1.5-5
## [40] colorspace_1.2-6   quantreg_5.29    stringi_1.1.1
## [43] munsell_0.4.3
```