1 Categorical data

1.1 Binomial confidence intervals

```
> library(DescTools)
> # For specificity of 98.99% (98/99)
> # So assume you have a study where out of the 99 negative individuals,
> # 98 got a negative test and 1 a positive test
> BinomCI(98, 99, conf.level = 0.95, method = c("wilson",
+                                               "wald",
+                                               "jeffreys",
+                                               "clopper-pearson"))
> my.object <- BinomCI(98, 99, conf.level = 0.95, method = c("wilson"))
> my.object
```

1.2 DOR diagnostic odds ratio

```
> # LR.p is LR + and LR.n is LR -
> my.DOR.func <- function(LR.p,LR.n){
+
+   DOR <- LR.p/LR.n
+   return(DOR)
+ }
> my.DOR.func(13,0.03)
```

```
[1] 433.3333
```

1.3 odds ratio

```
> # a,b,c,d are the entries in a 2X2 table
> my.OR.function <- function(a,b,c,d) {
+
+  OR <- (a*d)/(b*c)
+  return(OR)
+ }
> my.OR.function(4,3,5,7)
```

1.4 Se,Sp,PV,LR,AP,TP

```
> library(epiR)
> #create a table
> dat <- as.table(matrix(c(367, 55, 26, 678),
+                        nrow = 2, byrow = TRUE))
> colnames(dat) <- c("D+","D-")
> rownames(dat) <- c("T+","T-")

> dat
> epi.tests(dat, conf.level = 0.95)
> my.DOR.func(12.45,0.07)
> b <- epi.tests(dat, conf.level = 0.95)
> b$elements
```

2 kappa

2.1 kappa from a data set

```
> library(irr)
> data(anxiety)
> head(anxiety)
> kappa2(anxiety[,1:2])
```

2.2 kappa from two vectors

```
> library(psych)
> rater1 = c(1,2,3,4,5,6,7,8,9) # rater one's ratings
> rater2 = c(1,3,1,6,1,5,5,6,7) # rater one's ratings
> cohen.kappa(x=cbind(rater1,rater2))
```

2.3 kappa from a table

```
> mydat1 <- matrix(c(85,5,5,5),
+                 ncol=2,byrow=TRUE)
> cohen.kappa(mydat1)
> mydat2 <- matrix(c(45,5,5,45),
+                 ncol=2,byrow=TRUE)
> cohen.kappa(mydat2)
```

3 Continuous data

3.1 ROC

```
> ##################### to be used for example data set from aSAH in lecture
> library(pROC)
> #to check if data set loaded and some descriptive stats
> data(aSAH)
> head(aSAH)
> tail(aSAH)
> str(aSAH)
> summary(aSAH)
> boxplot(aSAH$s100b~aSAH$outcome,ylab="s100b", xlab="outcome")
> t.test(aSAH$s100b~aSAH$outcome,var.equal = FALSE)
> ?t.test
```

3.2 ROC and AUC

```
> #from website
>
> # to plot ROC
> plot.roc(aSAH$outcome, aSAH$s100b)
> roc_object_1 <- roc(aSAH$outcome,aSAH$s100b)    #define a roc object
> roc_object_1 <- roc(ex$Provirus,ex$p45)
> # to estimate full AUC:
> auc(roc_object_1)
> # to estimate partial AUC for se and sp, each between 0.9 and 1
> auc(roc_object_1, partial.auc=c(1, 0.90),
+     partial.auc.focus="se", partial.auc.correct=TRUE)
> auc(roc_object_1, partial.auc=c(1, 0.90),
+     partial.auc.focus="sp",partial.auc.correct=TRUE)
>
```

3.3 pAUC

```
> # plot ROC with pAUC for se
> roc1 <- roc(aSAH$outcome,   aSAH$s100b, percent=TRUE,
+ # arguments for auc
+ partial.auc=c(100, 90), partial.auc.correct=TRUE,
+ partial.auc.focus="sens",
+ # arguments for ci
+ ci=TRUE, boot.n=100, ci.alpha=0.9, stratified=FALSE,
+ # arguments for plot
+ plot=TRUE, auc.polygon=TRUE, max.auc.polygon=TRUE,
+ grid=TRUE, print.auc=TRUE, show.thres=TRUE)

> ##  plot ROC with pAUC for sp
> roc1 <- roc(aSAH$outcome,   aSAH$s100b, percent=TRUE,
+ # arguments for auc
+ partial.auc=c(100, 90), partial.auc.correct=TRUE,
+ # arguments for ci
+ ci=TRUE, boot.n=100, ci.alpha=0.9, stratified=FALSE,
+ # arguments for plot
+ plot=TRUE, auc.polygon=TRUE, max.auc.polygon=TRUE, grid=TRUE,
+ print.auc=TRUE, show.thres=TRUE)
>

> ##Plot beautiful ROC curve with pAUC (0.9,1) for se and sp
> plot.roc(aSAH$outcome, aSAH$s100b, # data
+ percent=TRUE, # show all values in percent
+ partial.auc=c(100, 90), partial.auc.correct=TRUE, # define a partial AUC (pAUC)
+ print.auc=TRUE, #display pAUC value on the plot with following options:
+ print.auc.pattern="Corrected pAUC (100-90%% SP):\n%.1f%%",
+ print.auc.col="#1c61b6",
+ auc.polygon=TRUE,
+ auc.polygon.col="#1c61b6", # show pAUC as a polygon
+ max.auc.polygon=TRUE,
+ max.auc.polygon.col="#1c61b622", # also show the 100% polygon
+ main="Partial AUC (pAUC)")
> plot.roc(aSAH$outcome, aSAH$s100b,
+ percent=TRUE, add=TRUE,
+ type="n", # add to plot, but don't re-add the ROC itself (useless)
+ partial.auc=c(100, 90), partial.auc.correct=TRUE,
+ partial.auc.focus="se", # focus pAUC on the sensitivity
+ print.auc=TRUE,
+ print.auc.pattern="Corrected pAUC (100-90%% SE):\n%.1f%%",
+ print.auc.col="#008600",
+ print.auc.y=40, # do not print auc over the previous one
+ auc.polygon=TRUE,
+ auc.polygon.col="#008600",
+ max.auc.polygon=TRUE,
+ max.auc.polygon.col="#00860022")
```

3.4 ROC with CI for se and sp

```
> roc_object_1 <- plot.roc(aSAH$outcome, aSAH$s100b,
+ main="Confidence intervals of specificity/sensitivity",
+ percent=TRUE,
+ ci=TRUE, of="se", # ci of sensitivity
+ specificities=seq(0, 100, 5), # on a select set of specificities
+ ci.type="shape", ci.col="#1c61b6AA") # plot the CI as a blue shape
> plot(ci.sp(roc_object_1, sensitivities=seq(0, 100, 5)), # ci of specificity
+ type="bars") # print this one as bars
```

3.5 compare se and sp

```
> #compare two ROC curves
> roc_object_1 <- plot.roc(aSAH$outcome, aSAH$s100b,
+ main="Statistical comparison", percent=TRUE, col="#1c61b6")
> roc_object_2 <- lines.roc(aSAH$outcome, aSAH$ndka,
+                           percent=TRUE, col="#008600")
> testobject <- roc.test(roc_object_1, roc_object_2)
> text(50, 50, labels=paste("p-value =", format.pval(testobject$p.value)),
+      adj=c(0, .5))
> legend("bottomright", legend=c("S100B", "NDKA"),
+        col=c("#1c61b6", "#008600"), lwd=2)
```

4 experimental and field data set on different serological assays for FeLV

```
> ## for experimental data
> library(pROC)
> #read in data
> ex<-read.csv("ROC_exp.csv",header=T, sep = ";") #adapt the path according to your local ma
> fd<-read.csv("ROC_field.csv",header=T)
```