**TP = true positive**

**FN = false negative**

**FP = false positive**

**TN= true negative**

**satirtop1 = TP + FN**

**satirtop2 = FP + TN**

**sutuntop1 = TP + FP**

**sutuntop2 = FN + TN**

**toplam = satirtop1 + satirtop2**

**## Duyarlilik**

**sensitivity\_estimate = TP / satirtop1**

**sensitivity\_se = round((sqrt(sensitivity\_estimate \* (1 - sensitivity\_estimate)/satirtop1)), 4)**

**sensitivity\_low = 1 - Rbeta.inv(((1 + CI/100)/2), (satirtop1+1-TP), TP)**

**sensitivity\_upp = Rbeta.inv(((1 + CI/100)/2), (TP+1), (satirtop1-TP))**

**sensitivity\_randomtest = sutuntop1 / toplam**

**qualityindex\_sensitivity = (sensitivity\_estimate - sensitivity\_randomtest) / (1 - sensitivity\_randomtest)**

**## Secicilik**

**specificity = TN / satirtop2**

**specificity\_se = sqrt(specificity \* (1 - specificity) / satirtop2)**

**specificity\_low = 1 - Rbeta.inv((1 + CI/100)/2, (satirtop2+1-TN),TN)**

**specificity\_upp = Rbeta.inv((1 + CI/100)/2, (TN+1), (satirtop2-TN))**

**specificity\_randomtest = (1 - sensitivity\_randomtest)**

**qualityindex\_specificity = (specificity - 1 + sensitivity\_randomtest) / sensitivity\_randomtest**

**## Gain in Certainty**

**gain\_in\_certainty = sensitivity\_estimate + specificity**

**## Etkililik, yeterlilik ayni zamanda overall accuracy (toplam-genel dogruluk orani)**

**efficiency = (TP + TN) / toplam**

**efficiency\_se = sqrt(efficiency \* (1 - efficiency) / toplam)**

**efficiency\_low = 1-Rbeta.inv((1 + CI/100)/2, (toplam+1-TP-TN), (TP+TN))**

**efficiency\_upp = Rbeta.inv((1 + CI/100)/2, (TP+TN+1), (toplam-TP-TN))**

**efficiency\_randomtest = (satirtop1 / toplam) \* sensitivity\_randomtest + (1 - (satirtop1 / toplam)) \* (1 - sensitivity\_randomtest)**

**mis\_efficiency = 1 - efficiency**

**## Quality indeks**

**quality\_index = (efficiency - efficiency\_randomtest) / (1 - efficiency\_randomtest)**

**quality\_index\_se = sqrt((efficiency\*(1-efficiency)/(1-efficiency\_randomtest)^2+2\*(1-efficiency)\*(2\*efficiency\*efficiency\_randomtest-((TP\*((satirtop1 / toplam)+sensitivity\_randomtest)+TN\*(2-(satirtop1 / toplam)-sensitivity\_randomtest))/toplam))/(1-efficiency\_randomtest)^3+((1-efficiency)^2)\*((TP\*((satirtop1 / toplam)+sensitivity\_randomtest)^2+FN\*(-(satirtop1 / toplam)+1+sensitivity\_randomtest)^2+FP\*(1+(satirtop1 / toplam)-sensitivity\_randomtest)^2+TN\*(2-(satirtop1 / toplam)-sensitivity\_randomtest)^2)/toplam-4\*efficiency\_randomtest^2)/(1-efficiency\_randomtest)^4)/toplam)**

**quality\_index\_low = quality\_index + qnorm((1 - CI/100)/2)\*quality\_index\_se**

**quality\_index\_upp = quality\_index - qnorm((1 - CI/100)/2)\*quality\_index\_se**

**## Youden indeksi**

**youdens\_index = sensitivity\_estimate + specificity - 1**

**youdens\_index\_se = sqrt(sensitivity\_se^2+specificity\_se^2)**

**youdens\_index\_low = youdens\_index + qnorm((1 - CI/100)/2)\*youdens\_index\_se**

**youdens\_index\_upp = youdens\_index - qnorm((1 - CI/100)/2)\*youdens\_index\_se**

**number\_needed\_to\_diagnose = 1 / youdens\_index**

**number\_needed\_to\_misdiagnose = 1 / ( 1 - efficiency )**

**## Pozitif kestirim degeri**

**predictivevalue\_positivetest = TP / sutuntop1**

**predictivevalue\_positivetest\_se = sqrt(predictivevalue\_positivetest\*(1-predictivevalue\_positivetest)/sutuntop1)**

**predictivevalue\_positivetest\_low = 1 - Rbeta.inv((1 + CI/100)/2, (sutuntop1+1-TP),TP)**

**predictivevalue\_positivetest\_upp = Rbeta.inv((1 + CI/100)/2, (TP+1),(sutuntop1-TP))**

**predvalue\_positiverandomtest = (satirtop1 / toplam)**

**predvalue\_positiverandomtest\_mar\_ol = predvalue\_positiverandomtest\*sensitivity\_estimate+(1-predvalue\_positiverandomtest)\*(1-specificity)**

**## Negatif kestirim degeri**

**predictivevalue\_negativetest = TN / sutuntop2**

**predictivevalue\_negativetest\_se = sqrt(predictivevalue\_negativetest\*(1-predictivevalue\_negativetest)/sutuntop2)**

**predictivevalue\_negativetest\_low = 1 - Rbeta.inv((1 + CI/100)/2, (sutuntop2+1-TN),TN)**

**predictivevalue\_negativetest\_upp = Rbeta.inv((1 + CI/100)/2, (TN+1), (sutuntop2-TN))**

**predvalue\_negativerandomtest = 1 - (satirtop1 / toplam)**

**predvalue\_negativerandomtest\_mar\_ol = 1 - predvalue\_positiverandomtest\_mar\_ol**

**## Predictive Summary Index (PSI)**

**predictive\_summary\_index = predictivevalue\_positivetest + predictivevalue\_negativetest - 1**

**## Yanlis pozitif orani**

**false\_positiverate = FP / satirtop2**

**false\_positiverate\_se = sqrt(false\_positiverate\*(1-false\_positiverate)/satirtop2)**

**false\_positiverate\_low = 1 - Rbeta.inv((1 + CI/100)/2, (satirtop2+1-FP),FP)**

**false\_positiverate\_upp = Rbeta.inv((1 + CI/100)/2, (FP+1), (satirtop2-FP))**

**## Yansis negatif orani**

**false\_negativerate = FN / satirtop1**

**false\_negativerate\_se = sqrt(false\_negativerate\*(1-false\_negativerate)/satirtop1)**

**false\_negativerate\_low = 1 - Rbeta.inv((1 + CI/100)/2, (satirtop1+1-FN),FN)**

**false\_negativerate\_upp = Rbeta.inv((1 + CI/100)/2, (FN+1), (satirtop1-FN))**

**## Dogru pozitif orani**

**true\_positiverate = TP / (satirtop1)**

**## Yanlis siniflandirma orani**

**misclassification\_rate = (FN + FP) / toplam**

**misclassification\_rate\_se = sqrt(misclassification\_rate\*(1-misclassification\_rate)/toplam)**

**misclassification\_rate\_low = 1 - Rbeta.inv((1 + CI/100)/2, (toplam+1-FP-FN), (FP+FN))**

**misclassification\_rate\_upp = Rbeta.inv((1 + CI/100)/2, (FN+FN+1), (toplam-FP-FN))**

**## Prevelans**

**prevalence = (satirtop1 / toplam)**

**prevalence\_se = sqrt((satirtop1 / toplam)\*(1-(satirtop1 / toplam))/toplam)**

**prevalence\_low = 1 - Rbeta.inv((1 + CI/100)/2, (toplam+1-satirtop1), satirtop1)**

**prevalence\_upp = Rbeta.inv((1 + CI/100)/2, (satirtop1+1), (toplam-satirtop1))**

**##caret paketinden olculer**

**detection\_rate = TP/toplam**

**detection\_prevalence = (TP+FP)/toplam**

**balanced\_accuracy = (sensitivity\_estimate + specificity)/2**

**lift = (TP/(TP+FN))/((TP+FP)/toplam) ## https://gerardnico.com/wiki/data\_mining/error\_rate**

**precision = TP/(TP+FP)**

**recall = TP/(TP+FN)**

**## http://michael.hahsler.net/research/recomm\_lnai2002/lnai2002.pdf**

**conviction = (1-((TP+FP)/toplam))/(1-(TP/(TP+FN)))**

**matthews\_correlation\_coefficient = (TP \* TN + FN \* FP) / ((TP+FP)\*(FN+TN)\*(FP+TN)\*(TP+FN))^(1/2)**

**## Goreli Risk Orani (Relative risk)**

**## http://www.openepi.com/PDFDocs/TwobyTwoDoc.pdf**

**# https://www2.ccrb.cuhk.edu.hk/stat/confidence%20interval/CI%20for%20ratio.htm**

**relative\_risk = (TP/sutuntop1)/(FN/sutuntop2)**

**relative\_risk\_se = sqrt((1/TP)+(1/FN)-(1/(TP+FP))-(1/(FN+TN)))**

**relative\_risk\_low = exp(log(relative\_risk)+qnorm((1-CI/100)/2)\*relative\_risk\_se)**

**relative\_risk\_upp = exp(log(relative\_risk)-qnorm((1-CI/100)/2)\*relative\_risk\_se)**

**difference\_in\_proportion = (TP/sutuntop1) - (FN/sutuntop2)**

**## Leverage**

**## https://www3.nd.edu/~tweninge/pubs/AHBCW\_ANNIE08.pdf**

**leverage = (TP/sutuntop1)-((TP+FP)/toplam)**

**## http://www.ebm.med.ualberta.ca/TherapyCalc.html**

**control\_event\_rate = FN/sutuntop2**

**experimental\_event\_rate = TP/sutuntop1**

**## Absolute risk reduction**

**absolute\_risk\_reduction = -difference\_in\_proportion**

**absolute\_risk\_reduction\_se = sqrt((TP/sutuntop1)\*(1-(TP/sutuntop1))/sutuntop1+(FN/sutuntop2)\*(1-(FN/sutuntop2))/sutuntop2)**

**absolute\_risk\_reduction\_low = absolute\_risk\_reduction - qnorm((1-CI/100)/2)\*absolute\_risk\_reduction\_se**

**absolute\_risk\_reduction\_upp = absolute\_risk\_reduction + qnorm((1-CI/100)/2)\*absolute\_risk\_reduction\_se**

**## Relative risk reduction**

**relative\_risk\_reduction = absolute\_risk\_reduction / (FN/sutuntop2)**

**relative\_risk\_reduction\_low = 1 - exp(log(relative\_risk)-qnorm((1-CI/100)/2)\*relative\_risk\_se)**

**relative\_risk\_reduction\_upp = 1 - exp(log(relative\_risk)+qnorm((1-CI/100)/2)\*relative\_risk\_se)**

**##Number needed to treat**

**number\_needed\_to\_treat = 1 / abs(difference\_in\_proportion)**

**number\_needed\_to\_treat\_low = 1 /absolute\_risk\_reduction\_upp**

**number\_needed\_to\_treat\_upp = 1 /absolute\_risk\_reduction\_low**

**## Test duzeyi**

**test\_level = sensitivity\_randomtest**

**test\_level\_se = sqrt(test\_level\*(1-test\_level)/toplam)**

**test\_level\_low = 1 - Rbeta.inv((1 + CI/100)/2, (toplam+1-sutuntop1),sutuntop1)**

**test\_level\_upp = Rbeta.inv((1 + CI/100)/2, (sutuntop1+1), (toplam-sutuntop1))**

**## Pre-test odds**

**pretest\_odds = prevalence / (1 - prevalence)**

**## Pozitif Olabilirlik Orani**

**likelihoodratio\_positivetest = sensitivity\_estimate/(1-specificity)**

**likelihoodratio\_positivetest\_se = exp(sqrt((1-sensitivity\_estimate)/(sensitivity\_estimate\*satirtop1) +specificity/((1-specificity)\*satirtop2)))**

**likelihoodratio\_positivetest\_low = exp(log(likelihoodratio\_positivetest) + qnorm((1 - CI/100)/2)\*log(likelihoodratio\_positivetest\_se))**

**likelihoodratio\_positivetest\_upp = exp(log(likelihoodratio\_positivetest) - qnorm((1 - CI/100)/2)\*log(likelihoodratio\_positivetest\_se))**

**## post-test odds**

**posttest\_odds = pretest\_odds \* likelihoodratio\_positivetest**

**bayes\_ppv = (sensitivity\_estimate \* prevalence)/((sensitivity\_estimate \* prevalence) + (1 - specificity) \* (1 - prevalence))**

**## Post-test probability (positive test result)**

**posttest\_probability = (pretest\_odds \* likelihoodratio\_positivetest) / (1 + (pretest\_odds \* likelihoodratio\_positivetest))**

**## Negatif Olabilirlik Orani**

**likelihoodratio\_negativetest = (1-sensitivity\_estimate)/specificity**

**likelihoodratio\_negativetest\_se = exp(sqrt(sensitivity\_estimate/((1-sensitivity\_estimate)\*satirtop1) +(1-specificity)/(specificity\*satirtop2)))**

**likelihoodratio\_negativetest\_low = exp(log(likelihoodratio\_negativetest) + qnorm((1-CI/100)/2)\*log(likelihoodratio\_negativetest\_se))**

**likelihoodratio\_negativetest\_upp = exp(log(likelihoodratio\_negativetest) - qnorm((1-CI/100)/2)\*log(likelihoodratio\_negativetest\_se))**

**bayes\_npv = (sensitivity\_estimate \* (1 - prevalence))/((sensitivity\_estimate \* (1 - prevalence)) + (1 - specificity) \* prevalence)**

**## Post-test probability (negative test result)**

**posttest\_probability\_neg = (pretest\_odds \* likelihoodratio\_negativetest) / (1 + (pretest\_odds \* likelihoodratio\_negativetest))**

**## Ters negatif olabilirlik orani**

**inverse\_likelihoodratio\_negativetest = specificity/(1-sensitivity\_estimate)**

**inverse\_likelihoodratio\_negativetest\_se = exp(sqrt(sensitivity\_estimate/((1-sensitivity\_estimate)\*satirtop1) +(1-specificity)/(specificity\*satirtop2)))**

**inverse\_likelihoodratio\_negativetest\_low = exp(log(inverse\_likelihoodratio\_negativetest) + qnorm((1-CI/100)/2)\*log(inverse\_likelihoodratio\_negativetest\_se))**

**inverse\_likelihoodratio\_negativetest\_upp = exp(log(inverse\_likelihoodratio\_negativetest) - qnorm((1-CI/100)/2)\*log(inverse\_likelihoodratio\_negativetest\_se))**

**## Diagnostic Odds Ratio (Tani Odds Orani)**

**odds\_ratio = (TP/FN)/(FP/TN)**

**odds\_ratio\_se = exp(sqrt(1/TP+1/FP+1/TN+1/FN))**

**odds\_ratio\_low = exp(log(odds\_ratio) + qnorm((1-CI/100)/2)\*log(odds\_ratio\_se))**

**odds\_ratio\_upp = exp(log(odds\_ratio) - qnorm((1-CI/100)/2)\*log(odds\_ratio\_se))**

**odds\_ratio\_Haldane = ((TP+0.5)/(FN+0.5))/((FP+0.5)/(TN+0.5))**

**odds\_ratio\_Haldane\_se = exp(sqrt(1/(TP+0.5)+1/(FP+0.5)+1/(TN+0.5)+1/(FN+0.5)))**

**odds\_ratio\_Haldane\_low = exp(log(odds\_ratio\_Haldane) + qnorm((1-CI/100)/2)\*log(odds\_ratio\_Haldane\_se))**

**odds\_ratio\_Haldane\_upp = exp(log(odds\_ratio\_Haldane) - qnorm((1-CI/100)/2)\*log(odds\_ratio\_Haldane\_se))**

**error\_odds\_ratio = (sensitivity\_estimate/(1-sensitivity\_estimate))/(specificity/(1-specificity))**

**## Rate ratio**

**## http://omarkasule-05.tripod.com/id52.html**

**## http://sphweb.bumc.bu.edu/otlt/mph-modules/ep/ep713\_randomerror/ep713\_randomerror4.html**

**rate\_ratio = (TP/sutuntop1)/(TP/sutuntop2)**

**rate\_ratio\_low = exp(log(relative\_risk)+qnorm((1-CI/100)/2)\*sqrt(1/TP+1/FP))**

**rate\_ratio\_upp = exp(log(relative\_risk)-qnorm((1-CI/100)/2)\*sqrt(1/TP+1/FP))**

**## Risk Difference**

**## http://www.openepi.com/PDFDocs/TwobyTwoDoc.pdf**

**risk\_difference = (TP/sutuntop1) - (FN/sutuntop2)**

**risk\_difference\_se = (((TP/sutuntop1)\*(1-(TP/sutuntop1)))/sutuntop1)+(((FN/sutuntop2)\*(1-(FN/sutuntop2)))/sutuntop2)**

**risk\_difference\_low = risk\_difference + qnorm((1-CI/100)/2)\*sqrt(risk\_difference\_se)**

**risk\_difference\_upp = risk\_difference - qnorm((1-CI/100)/2)\*sqrt(risk\_difference\_se)**

**## Nitelenebilen risk**

**## https://www2.ccrb.cuhk.edu.hk/stat/confidence%20interval/CI%20for%20relative%20risk.htm**

**attributable\_risk = (TP/sutuntop1)-(FN/sutuntop2)**

**attributable\_risk\_se = sqrt((satirtop1/toplam)\*(1-(satirtop1/toplam))\*(1/sutuntop1+1/sutuntop2))**

**attributable\_risk\_low = attributable\_risk + qnorm((1-CI/100)/2)\*attributable\_risk\_se**

**attributable\_risk\_upp = attributable\_risk - qnorm((1-CI/100)/2)\*attributable\_risk\_se**

**## Attributable risk percent**

**## https://www2.ccrb.cuhk.edu.hk/stat/confidence%20interval/CI%20for%20relative%20risk.htm**

**attributable\_risk\_percent = ((relative\_risk-1)/relative\_risk)\*100**

**attributable\_risk\_percent\_low = attributable\_risk\_percent - (qnorm((1-CI/100)/2)\*attributable\_risk\_se)/attributable\_risk**

**attributable\_risk\_percent\_upp = attributable\_risk\_percent + (qnorm((1-CI/100)/2)\*attributable\_risk\_se)/attributable\_risk**

**## Population Attributable Risk**

**## https://www2.ccrb.cuhk.edu.hk/stat/confidence%20interval/CI%20for%20relative%20risk.htm**

**population\_attributable\_risk = (satirtop1/toplam)-(FN/sutuntop2)**

**## Population Attributable Risk Percent**

**population\_attributable\_risk\_percent = (population\_attributable\_risk / (satirtop1/toplam))\*100**

**## Kohen's Kappa**

**cohens\_kappa = (efficiency-efficiency\_randomtest)/(1-efficiency\_randomtest)**

**cohens\_kappa\_se = quality\_index\_se**

**cohens\_kappa\_low = cohens\_kappa + qnorm((1-CI/100)/2)\*cohens\_kappa\_se**

**cohens\_kappa\_upp = cohens\_kappa - qnorm((1-CI/100)/2)\*cohens\_kappa\_se**

**## Gozlenen Karar**

**observed\_agreement = efficiency**

**observed\_agreement\_se = efficiency\_se**

**observed\_agreement\_low = efficiency\_low**

**observed\_agreement\_upp = efficiency\_upp**

**## Risk karari,expected agreement**

**chance\_agreement = efficiency\_randomtest**

**## Pozitif karar**

**positive\_agreement = 2\*TP /(satirtop1+sutuntop1)**

**positive\_agreement\_se = sqrt(4\*TP\*(FP+FN)\*(TP+FP+FN))/(sutuntop1+satirtop1)^2**

**positive\_agreement\_low = positive\_agreement + qnorm((1-CI/100)/2)\*positive\_agreement\_se**

**positive\_agreement\_upp = positive\_agreement - qnorm((1-CI/100)/2)\*positive\_agreement\_se**

**## Negatif karar**

**negative\_agreement = 2\*TN/(satirtop2+sutuntop2)**

**negative\_agreement\_se = sqrt(4\*TN\*(FP+FN)\*(TN+FP+FN))/(sutuntop2+satirtop2)^2**

**negative\_agreement\_low = negative\_agreement + qnorm((1-CI/100)/2)\*negative\_agreement\_se**

**negative\_agreement\_upp = negative\_agreement - qnorm((1-CI/100)/2)\*negative\_agreement\_se**

**## Rand index**

**## http://www.cs.pomona.edu/classes/cs158/resources/158-10(EM).pdf**

**rand\_index = (TP+TN)/(TP+FN+FP+TN)**

**## e-measure**

**## http://michael.hahsler.net/research/recomm\_lnai2002/lnai2002.pdf**

**e\_measure = 1/((((1-CI)/100)\*(1-precision))+((1-((1-CI)/100))\*(1/recall)))**

**## discriminant power**

**## https://eva.fing.edu.uy/pluginfile.php/69453/mod\_resource/content/1/7633-10048-1-PB.pdf**

**discriminant\_power = (sqrt(3)/pi)\*(log (sensitivity\_estimate / (1-sensitivity\_estimate)) + log (specificity / (1-specificity)))**

**## F1 Score**

**## http://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/**

**f1\_score = 2\*(((TP/(TP+FN))\*(TP/(TP+FP)))/((TP/(TP+FN))+(TP/(TP+FP))))**

**## Byrt yanlilik indeksi**

**byrt\_bias\_index = (FN-FP)/toplam**

**## Byrt asimetrik indeks prevelansi**

**byrt\_prevalence\_asymmetry\_index = (TN-TP)/toplam**

**## Yanlilik duzeltmeli kappa**

**bias\_adjusted\_kappa = (efficiency-0.25\*((satirtop1+sutuntop1)^2+(satirtop2+sutuntop2)^2)/toplam^2)/(1-0.25\*((satirtop1+sutuntop1)^2+(satirtop2+sutuntop2)^2)/toplam^2)**

**## Yanlilik duzeltmeli kappa prevelansi**

**prevalence\_bias\_adjusted\_kappa = cohens\_kappa+(1-cohens\_kappa)\*(byrt\_prevalence\_asymmetry\_index^2-byrt\_bias\_index^2)**

**## Dice'in indeksi (Czekanowski)**

**dice\_index = positive\_agreement**

**dice\_index\_se= positive\_agreement\_se**

**dice\_index\_low = positive\_agreement\_low**

**dice\_index\_upp = positive\_agreement\_upp**

**## Yule's Q**

**yules\_q = (odds\_ratio - 1) / (odds\_ratio + 1)**

**yules\_q\_se = 0.5\*(1-yules\_q^2)\*sqrt(1/TP+1/FP+1/TN+1/FN)**

**yules\_q\_low = yules\_q + qnorm((1-CI/100)/2)\*yules\_q\_se**

**yules\_q\_upp = yules\_q - qnorm((1-CI/100)/2)\*yules\_q\_se**

**equitable\_threatscore = (TP-(sutuntop1\*satirtop1/toplam))/(TP+FP+FN-(sutuntop1\*satirtop1/toplam))**

**## Phi**

**phi = (TP\*TN-FN\*FP) / sqrt(satirtop1 \* satirtop2 \* sutuntop1 \* sutuntop2)**

**phi\_se = sqrt((1-phi^2+(phi+0.5\*phi^3)\*((satirtop1-satirtop2)\*(sutuntop1-sutuntop2)/sqrt(satirtop1\*satirtop2\*sutuntop1\*sutuntop2))-0.75\*phi^2\*((satirtop1-satirtop2)^2/(satirtop1\*satirtop2)+(sutuntop1-sutuntop2)^2/(sutuntop1\*sutuntop2)))/toplam)**

**phi\_low = phi + qnorm((1-CI/100)/2)\*phi\_se**

**phi\_upp = phi - qnorm((1-CI/100)/2)\*phi\_se**

**## Cramer V katsayisi**

**cramer\_v = sqrt((toplam\*(TP\*TN-FN\*FP)^2/(satirtop1\*satirtop2\*sutuntop1\*sutuntop2))/toplam)**

**## Olaganlik katsayisi**

**contingency\_coefficient = sqrt((toplam\*(TP\*TN-FN\*FP)^2/(satirtop1\*satirtop2\*sutuntop1\*sutuntop2))/(((toplam\*(TP\*TN-FN\*FP)^2/(satirtop1\*satirtop2\*sutuntop1\*sutuntop2)))+toplam))**

**## Goodman and Kruskal Gamma**

**goodman\_and\_kruskal\_gamma = (TP\*TN-FN\*FP)/(TP\*TN+FN\*FP)**

**goodman\_and\_kruskal\_gamma\_se = 0.5\*(1-yules\_q^2)\*sqrt(1/TP+1/FP+1/TN+1/FN)**

**goodman\_and\_kruskal\_gamma\_low = yules\_q + qnorm((1-CI/100)/2)\*yules\_q\_se**

**goodman\_and\_kruskal\_gamma\_upp = yules\_q - qnorm((1-CI/100)/2)\*yules\_q\_se**

**## Kendall's tau a**

**kendalls\_tau\_a = (TP\*TN-FN\*FP)/(toplam\*(toplam-1)/2)**

**## Kendall's tau b**

**## https://v8doc.sas.com/sashtml/stat/chap28/sect20.htm**

**## http://slideplayer.com/slide/10766029/**

**kendalls\_tau\_b = (TP\*TN-FN\*FP)/sqrt((TP\*TN+FN\*FP+(TP\*FN+FP\*TN))\*(TP\*TN+FN\*FP+(TP\*FP+FN\*TN)))**

**## Kendall's tau c**

**kendalls\_tau\_c = ((2\*2)\*(TP\*TN-FN\*FP)) / (toplam^2 \* (2-1))**

**## Somers'd R|C**

**somers\_d = (TP\*TN-FN\*FP)/(TP\*TN+FN\*FP+(TP\*FN+FP\*TN))**

**## Somers'd C|R**

**somers\_d\_2 = (TP\*TN-FP\*FN)/(TP\*TN+FP\*FN+(TP\*FP+FN\*TN))**

**## Scoot'un karar indeksi**

**scotts\_agreement\_index = bias\_adjusted\_kappa**

**## Dort-duzeyli Korelasyon**

**tetrachoric\_correlation = cos(pi/(1+sqrt(TP\*TN/FN/FP)))**

**tetrachoric\_correlation\_se = 0.375\*(1-tetrachoric\_correlation^2)\*sqrt((1/(TP))+(1/(FP))+(1/(TN))+(1/(FN)))**

**tetrachoric\_correlation\_low = tetrachoric\_correlation + qnorm((1-CI/100)/2)\*tetrachoric\_correlation\_se**

**tetrachoric\_correlation\_upp = tetrachoric\_correlation - qnorm((1-CI/100)/2)\*tetrachoric\_correlation\_se**

**## Goodman kruskal tau katsayisi**

**goodman\_kruskal\_tau = (toplam\*((TP^2+FP^2)/sutuntop1+(FN^2+TN^2)/sutuntop2)-satirtop1^2-satirtop2^2)/(toplam^2-satirtop1^2-satirtop2^2)**

**goodman\_kruskal\_tau\_se = toplam\*sqrt(TP\*(2\*(toplam-((TP^2+FN^2)/satirtop1+(FP^2+TN^2)/satirtop2))\*(toplam-sutuntop1)+(toplam^2-sutuntop1^2-sutuntop2^2)\*(2\*TP/satirtop1-(TP^2+FN^2)/satirtop1^2-(toplam-((TP^2+FN^2)/satirtop1+(FP^2+TN^2)/satirtop2))/toplam-1))^2+FN\*(2\*(toplam-((TP^2+FN^2)/satirtop1+(FP^2+TN^2)/satirtop2))\*(toplam-sutuntop2)+(toplam^2-sutuntop1^2-sutuntop2^2)\*(2\*FN/satirtop1-(TP^2+FN^2)/satirtop1^2-(toplam-((TP^2+FN^2)/satirtop1+(FP^2+TN^2)/satirtop2))/toplam-1))^2+FP\*(2\*(toplam-((TP^2+FN^2)/satirtop1+(FP^2+TN^2)/satirtop2))\*(toplam-sutuntop1)+(toplam^2-sutuntop1^2-sutuntop2^2)\*(2\*FP/satirtop2-(FP^2+TN^2)/satirtop2^2-(toplam-((TP^2+FN^2)/satirtop1+(FP^2+TN^2)/satirtop2))/toplam-1))^2+TN\*(2\*(toplam-((TP^2+FN^2)/satirtop1+(FP^2+TN^2)/satirtop2))\*(toplam-sutuntop2)+(toplam^2-sutuntop1^2-sutuntop2^2)\*(2\*TN/satirtop2-(FP^2+TN^2)/satirtop2^2-(toplam-((TP^2+FN^2)/satirtop1+(FP^2+TN^2)/satirtop2))/toplam-1))^2)/(toplam^2-sutuntop1^2-sutuntop2^2)^2**

**goodman\_kruskal\_tau\_low = goodman\_kruskal\_tau + qnorm((1-CI/100)/2)\*goodman\_kruskal\_tau\_se**

**goodman\_kruskal\_tau\_upp = goodman\_kruskal\_tau - qnorm((1-CI/100)/2)\*goodman\_kruskal\_tau\_se**

**## Simetrik Lambda**

**lambda\_symmetric = ((max(TP,FN)+max(FP,TN)+max(TP,FP)+max(FN,TN)-max(satirtop1,satirtop2)-max(sutuntop1,sutuntop2))/(2\*toplam-max(satirtop1,satirtop2)-max(sutuntop1,sutuntop2)))**

**if (lambda\_symmetric==0 || lambda\_symmetric==1)**

**{lambda\_symmetric\_se = 0}**

**else**

**{**

**lambda\_symmetric\_se = sqrt((TP\*((if(TP>FN) 1 else 0) + (if(TP>FP) 1 else 0) + ((if(sutuntop1>sutuntop2) 1 else 0)+ (if(satirtop1>satirtop2) 1 else 0))\*(lambda\_symmetric-1))^2+FN\*( (if(TP>FN) 0 else 1)+ (if(FN>TN) 1 else 0) +( (if(sutuntop1>sutuntop2) 0 else 1) + (if(satirtop1>satirtop2) 1 else 0))\*(lambda\_symmetric-1))^2+FP\*( (if(FP>TN) 1 else 0) + (if(TP>FP) 0 else 1) +( (if(sutuntop1>sutuntop2) 1 else 0) + (if(satirtop1<satirtop2) 1 else 0))\*(lambda\_symmetric-1))^2+TN\*((if(FP>TN) 0 else 1)+ (if(FN>TN) 0 else 1) +((if(sutuntop1>sutuntop2) 0 else 1)+ (if(satirtop1<satirtop2) 1 else 0))\*(lambda\_symmetric-1))^2-4\*toplam\*lambda\_symmetric^2)/(2\*toplam-max(satirtop1,satirtop2)-max(sutuntop1,sutuntop2))^2)**

**}**

**lambda\_symmetric\_low = lambda\_symmetric + qnorm((1-CI/100)/2)\*lambda\_symmetric\_se**

**lambda\_symmetric\_upp = lambda\_symmetric - qnorm((1-CI/100)/2)\*lambda\_symmetric\_se**

**## Lambda Asymmetric R|C**

**lambda\_criterion = (max(TP,FP)+max(FN,TN)-max(satirtop1,satirtop2))/(toplam-max(satirtop1,satirtop2))**

**if (lambda\_criterion==0 || lambda\_criterion==1)**

**{lambda\_criterion\_se = 0}**

**else**

**{**

**lambda\_criterion\_se = sqrt((toplam-max(TP,FP)-max(FN,TN))\*(max(TP,FP)+max(FN,TN)+max(satirtop1,satirtop2)-2\*(if(satirtop1>satirtop2) max(TP,FN) else max(FP,TN)))/(toplam-max(satirtop1,satirtop2))^3)**

**}**

**lambda\_criterion\_low = lambda\_criterion + qnorm((1-CI/100)/2)\*lambda\_criterion\_se**

**lambda\_criterion\_upp = lambda\_criterion - qnorm((1-CI/100)/2)\*lambda\_criterion\_se**

**## Lambda Asymmetric C|R**

**lambda\_criterion\_2 = (max(TP,FN)+max(FP,TN)-max(sutuntop1,sutuntop2))/(toplam-max(sutuntop1,sutuntop2))**

**if (lambda\_criterion\_2==0 || lambda\_criterion\_2==1)**

**{lambda\_criterion\_se\_2 = 0}**

**else**

**{**

**lambda\_criterion\_se\_2 = sqrt((toplam-max(TP,FN)-max(FP,TN))\*(max(TP,FN)+max(FP,TN)+max(sutuntop1,sutuntop2)-2\*(if(sutuntop1>sutuntop2) max(TP,FP) else max(FN,TN)))/(toplam-max(sutuntop1,sutuntop2))^3)**

**}**

**lambda\_criterion\_low\_2 = lambda\_criterion\_2 + qnorm((1-CI/100)/2)\*lambda\_criterion\_se\_2**

**lambda\_criterion\_upp\_2 = lambda\_criterion\_2 - qnorm((1-CI/100)/2)\*lambda\_criterion\_se\_2**

**## belirsizlik katsayisi (coefficient of uncertainty) simetrik**

**uncertainty\_coefficient\_symmetric = 2\*(satirtop1\*log(satirtop1/toplam)+satirtop2\*log(satirtop2/toplam)+sutuntop1\*log(sutuntop1/toplam)+sutuntop2\*log(sutuntop2/toplam)-(TP\*log(TP/toplam)+FN\*log(FN/toplam)+FP\*log(FP/toplam)+TN\*log(TN/toplam)))/(satirtop1\*log(satirtop1/toplam)+satirtop2\*log(satirtop2/toplam)+sutuntop1\*log(sutuntop1/toplam)+sutuntop2\*log(sutuntop2/toplam))**

**uncertainty\_coefficient\_symmetric\_se = 2\*sqrt(TP\*(-(TP\*log(TP/toplam)+FN\*log(FN/toplam)+FP\*log(FP/toplam)+TN\*log(TN/toplam))/toplam\*log(satirtop1\*sutuntop1/toplam^2)-(-(satirtop1\*log(satirtop1/toplam)+satirtop2\*log(satirtop2/toplam))/toplam+-(sutuntop1\*log(sutuntop1/toplam)+sutuntop2\*log(sutuntop2/toplam))/toplam)\*log(TP/toplam))^2+FN\*(-(TP\*log(TP/toplam)+FN\*log(FN/toplam)+FP\*log(FP/toplam)+TN\*log(TN/toplam))/toplam\*log(satirtop1\*sutuntop2/toplam^2)-(-(satirtop1\*log(satirtop1/toplam)+satirtop2\*log(satirtop2/toplam))/toplam+-(sutuntop1\*log(sutuntop1/toplam)+sutuntop2\*log(sutuntop2/toplam))/toplam)\*log(FN/toplam))^2+FP\*(-(TP\*log(TP/toplam)+FN\*log(FN/toplam)+FP\*log(FP/toplam)+TN\*log(TN/toplam))/toplam\*log(satirtop2\*sutuntop1/toplam^2)-(-(satirtop1\*log(satirtop1/toplam)+satirtop2\*log(satirtop2/toplam))/toplam+-(sutuntop1\*log(sutuntop1/toplam)+sutuntop2\*log(sutuntop2/toplam))/toplam)\*log(FP/toplam))^2+TN\*(-(TP\*log(TP/toplam)+FN\*log(FN/toplam)+FP\*log(FP/toplam)+TN\*log(TN/toplam))/toplam\*log(satirtop2\*sutuntop2/toplam^2)-(-(satirtop1\*log(satirtop1/toplam)+satirtop2\*log(satirtop2/toplam))/toplam+-(sutuntop1\*log(sutuntop1/toplam)+sutuntop2\*log(sutuntop2/toplam))/toplam)\*log(TN/toplam))^2)/toplam/(-(sutuntop1\*log(sutuntop1/toplam)+sutuntop2\*log(sutuntop2/toplam))/toplam+-(satirtop1\*log(satirtop1/toplam)+satirtop2\*log(satirtop2/toplam))/toplam)^2**

**uncertainty\_coefficient\_symmetric\_low = uncertainty\_coefficient\_symmetric + qnorm((1-CI/100)/2)\*uncertainty\_coefficient\_symmetric\_se**

**uncertainty\_coefficient\_symmetric\_upp = uncertainty\_coefficient\_symmetric - qnorm((1-CI/100)/2)\*uncertainty\_coefficient\_symmetric\_se**

**## (coefficient of uncertainty) R|C**

**uncertainty\_coeff\_crit = (satirtop1\*log(satirtop1/toplam)+satirtop2\*log(satirtop2/toplam)+sutuntop1\*log(sutuntop1/toplam)+sutuntop2\*log(sutuntop2/toplam)-(TP\*log(TP/toplam)+FN\*log(FN/toplam)+FP\*log(FP/toplam)+TN\*log(TN/toplam)))/(satirtop1\*log(satirtop1/toplam)+satirtop2\*log(satirtop2/toplam))**

**uncertainty\_coeff\_crit\_se = sqrt(TP\*(-(satirtop1\*log(satirtop1/toplam)+satirtop2\*log(satirtop2/toplam))/toplam\*log(TP/sutuntop1)+(-(sutuntop1\*log(sutuntop1/toplam)+sutuntop2\*log(sutuntop2/toplam))/toplam+(TP\*log(TP/toplam)+FN\*log(FN/toplam)+FP\*log(FP/toplam)+TN\*log(TN/toplam))/toplam)\*log(satirtop1/toplam))^2+FN\*(-(satirtop1\*log(satirtop1/toplam)+satirtop2\*log(satirtop2/toplam))/toplam\*log(FN/sutuntop2)+(-(sutuntop1\*log(sutuntop1/toplam)+sutuntop2\*log(sutuntop2/toplam))/toplam+(TP\*log(TP/toplam)+FN\*log(FN/toplam)+FP\*log(FP/toplam)+TN\*log(TN/toplam))/toplam)\*log(satirtop1/toplam))^2+FP\*(-(satirtop1\*log(satirtop1/toplam)+satirtop2\*log(satirtop2/toplam))/toplam\*log(FP/sutuntop1)+(-(sutuntop1\*log(sutuntop1/toplam)+sutuntop2\*log(sutuntop2/toplam))/toplam+(TP\*log(TP/toplam)+FN\*log(FN/toplam)+FP\*log(FP/toplam)+TN\*log(TN/toplam))/toplam)\*log(satirtop2/toplam))^2+TN\*(-(satirtop1\*log(satirtop1/toplam)+satirtop2\*log(satirtop2/toplam))/toplam\*log(TN/sutuntop2)+(-(sutuntop1\*log(sutuntop1/toplam)+sutuntop2\*log(sutuntop2/toplam))/toplam+(TP\*log(TP/toplam)+FN\*log(FN/toplam)+FP\*log(FP/toplam)+TN\*log(TN/toplam))/toplam)\*log(satirtop2/toplam))^2)/toplam/((satirtop1\*log(satirtop1/toplam)+satirtop2\*log(satirtop2/toplam))/toplam)^2**

**uncertainty\_coeff\_crit\_low = uncertainty\_coeff\_crit + qnorm((1-CI/100)/2)\*uncertainty\_coeff\_crit\_se**

**uncertainty\_coeff\_crit\_upp = uncertainty\_coeff\_crit - qnorm((1-CI/100)/2)\*uncertainty\_coeff\_crit\_se**

**## (coefficient of uncertainty) C|R**

**uncertainty\_coeff\_crit\_2 = (sutuntop1\*log(sutuntop1/toplam)+sutuntop2\*log(sutuntop2/toplam)+satirtop1\*log(satirtop1/toplam)+satirtop2\*log(satirtop2/toplam)-(TP\*log(TP/toplam)+FP\*log(FP/toplam)+FN\*log(FN/toplam)+TN\*log(TN/toplam)))/(sutuntop1\*log(sutuntop1/toplam)+sutuntop2\*log(sutuntop2/toplam))**

**uncertainty\_coeff\_crit\_se\_2 = sqrt(TP\*(-(sutuntop1\*log(sutuntop1/toplam)+sutuntop2\*log(sutuntop2/toplam))/toplam\*log(TP/satirtop1)+(-(satirtop1\*log(satirtop1/toplam)+satirtop2\*log(satirtop2/toplam))/toplam+(TP\*log(TP/toplam)+FP\*log(FP/toplam)+FN\*log(FN/toplam)+TN\*log(TN/toplam))/toplam)\*log(sutuntop1/toplam))^2+FP\*(-(sutuntop1\*log(sutuntop1/toplam)+sutuntop2\*log(sutuntop2/toplam))/toplam\*log(FP/satirtop2)+(-(satirtop1\*log(satirtop1/toplam)+satirtop2\*log(satirtop2/toplam))/toplam+(TP\*log(TP/toplam)+FP\*log(FP/toplam)+FN\*log(FN/toplam)+TN\*log(TN/toplam))/toplam)\*log(sutuntop1/toplam))^2+FN\*(-(sutuntop1\*log(sutuntop1/toplam)+sutuntop2\*log(sutuntop2/toplam))/toplam\*log(FN/satirtop1)+(-(satirtop1\*log(satirtop1/toplam)+satirtop2\*log(satirtop2/toplam))/toplam+(TP\*log(TP/toplam)+FP\*log(FP/toplam)+FN\*log(FN/toplam)+TN\*log(TN/toplam))/toplam)\*log(sutuntop2/toplam))^2+TN\*(-(sutuntop1\*log(sutuntop1/toplam)+sutuntop2\*log(sutuntop2/toplam))/toplam\*log(TN/satirtop2)+(-(satirtop1\*log(satirtop1/toplam)+satirtop2\*log(satirtop2/toplam))/toplam+(TP\*log(TP/toplam)+FP\*log(FP/toplam)+FN\*log(FN/toplam)+TN\*log(TN/toplam))/toplam)\*log(sutuntop2/toplam))^2)/toplam/((sutuntop1\*log(sutuntop1/toplam)+sutuntop2\*log(sutuntop2/toplam))/toplam)^2**

**uncertainty\_coeff\_crit\_low\_2 = uncertainty\_coeff\_crit\_2 + qnorm((1-CI/100)/2)\*uncertainty\_coeff\_crit\_se\_2**

**uncertainty\_coeff\_crit\_upp\_2 = uncertainty\_coeff\_crit\_2 - qnorm((1-CI/100)/2)\*uncertainty\_coeff\_crit\_se\_2**

**## Pearson ki-kare**

**pearson\_chi\_squ = toplam\*(TP\*TN-FN\*FP)^2/(satirtop1\*satirtop2\*sutuntop1\*sutuntop2)**

**pearson\_chi\_squ\_p = dchi(pearson\_chi\_squ, df=1)**

**##**

**with\_yate\_cor\_for\_pearson = toplam\*(abs(TP\*TN-FN\*FP)-0.5\*toplam)^2/(satirtop1\*satirtop2\*sutuntop1\*sutuntop2)**

**with\_yate\_cor\_p\_for\_pearson = dchi(with\_yate\_cor\_for\_pearson, df=1)**

**## Mantel Haenszel chi-square**

**## http://www.openepi.com/PDFDocs/TwobyTwoDoc.pdf**

**mantel\_haenszel = ((toplam-1)\*(TP\*TN-FN\*FP)^2)/(satirtop1\*satirtop2\*sutuntop1\*sutuntop2)**

**mantel\_haenszel\_p = dchi(mantel\_haenszel, df=1)**

**## Olasilik orani**

**likelih\_ratio = 2\*(TP\*log(TP/(sutuntop1\*satirtop1/toplam))+FN\*log(FN/(sutuntop2\*satirtop1/toplam))+FP\*log(FP/(sutuntop1\*satirtop2/toplam))+TN\*log(TN/(sutuntop2\*satirtop2/toplam)))**

**likelih\_ratio\_p =dchi(likelih\_ratio, df=1)**

**## Fisher'in exact testi**

**##fisher\_exact\_test = (TP-(satirtop1\*sutuntop1/toplam))^2/(satirtop1\*sutuntop1/toplam) + (FN-(satirtop1\*sutuntop2/toplam))^2/(satirtop1\*sutuntop2/toplam) + (FP-(satirtop2\*sutuntop1/toplam))^2/(satirtop2\*sutuntop1/toplam) + (TN-(satirtop2\*sutuntop2/toplam))^2/(satirtop2\*sutuntop2/toplam)**

**fisher <- c(TP, FN, FP, TN)**

**tab <- t(matrix(fisher, nrow=2,ncol=2))**

**fisher\_exact\_test = fisher.test(tab)$p.value**

**minimum\_exp\_fre = min((sutuntop1\*satirtop1/toplam):(sutuntop2\*satirtop2/toplam))**

**cells\_exp\_fre\_5 = (if((sutuntop1\*satirtop1/toplam)<5) 1 else 0)+ (if((sutuntop1\*satirtop2/toplam)<5) 1 else 0)+ (if((sutuntop2\*satirtop1/toplam) <5) 1 else 0)+ (if((sutuntop2\*satirtop2/toplam) <5) 1 else 0)**

**cells\_exp\_fre\_1 = (if((sutuntop1\*satirtop1/toplam) <1) 1 else 0)+ (if((sutuntop1\*satirtop2/toplam) <1) 1 else 0)+ (if((sutuntop2\*satirtop1/toplam) <1) 1 else 0)+ (if((sutuntop2\*satirtop2/toplam) <1) 1 else 0)**

**## Mc Nemar testi**

**mcNemar\_test = (FP-FN)^2/(FP+FN)**

**mcNemar\_test\_p = dchi(mcNemar\_test, df=1)**

**with\_Yate\_cor\_for\_mcnamer = (abs(FP-FN)-1)^2/(FP+FN)**

**with\_Yate\_cor\_p\_for\_mcnamer = dchi(with\_Yate\_cor\_for\_mcnamer, df=1)**

**## Belirsizlik (Entropi)**

**forbes\_NMI = sensitivity\_estimate\*predvalue\_positiverandomtest\*log((sensitivity\_estimate/predvalue\_positiverandomtest\_mar\_ol),base=2)+**

**(1-specificity)\*(1-predvalue\_positiverandomtest)\*log(((1-specificity)/predvalue\_positiverandomtest\_mar\_ol),base=2)+**

**(1-sensitivity\_estimate)\*predvalue\_positiverandomtest\*log(((1-sensitivity\_estimate)/(1-predvalue\_positiverandomtest\_mar\_ol)),base=2)+**

**specificity\*(1-predvalue\_positiverandomtest)\*log((specificity/(1-predvalue\_positiverandomtest\_mar\_ol)),base=2)**

**## satir icin entropy (test icin)**

**entropy\_hr = - ((sutuntop1/toplam) \* log(sutuntop1/toplam, base=2) + (sutuntop2/toplam) \* log(sutuntop2/toplam, base=2))**

**## sutun icin entropy (hastalik icin)**

**entropy\_hc = - ((satirtop1/toplam)\*log(satirtop1/toplam, base=2) + (satirtop2/toplam)\*log(satirtop2/toplam, base=2))**

**## birlesik entropi (joint entropy)**

**entropy\_hrc = - ( (TP/toplam)\*log(TP/toplam, base=2) + (FP/toplam)\*log(FP/toplam, base=2) + (FN/toplam)\*log(FN/toplam, base=2) + (TN/toplam)\*log(TN/toplam, base=2))**

**## bilgi icerigi (mutual information)**

**information\_r\_c = entropy\_hr + entropy\_hc - entropy\_hrc**

**## kosullu entropi (conditional entropy)**

**a = entropy\_hrc - entropy\_hr**

**c = entropy\_hrc - entropy\_hc**

**sim\_r\_c = information\_r\_c / (a + information\_r\_c + c)**

**dis\_r\_c = (a + c) / (a + information\_r\_c + c)**

**## goreli entropi (relative entropy, kullback-leibler uzakligi)**

**## pozitif test sonucu icin goreli entropi, Relative Improvement Over Chance (RIOC)**

**positive\_relative\_entropy = (TP/sutuntop1)\*log(((TP/sutuntop1)/ prevalence), base=2)+(FP/sutuntop1)\*log(((FP/sutuntop1)/(1- prevalence)), base=2)**

**## negatif test sonucu icin goreli entropi**

**negative\_relative\_entropy = (FN/sutuntop2)\*log(((FN/sutuntop2)/ prevalence), base=2)+(TN/sutuntop2)\*log(((TN/sutuntop2)/(1- prevalence)), base=2)**