Task1Seminar4

Group A3

2025-12-5

Step 1: Data Preparation

We first load the data.

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.1     v stringr   1.5.1
## v ggplot2   4.0.0     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
library(irr)
```

```
## Warning: package 'irr' was built under R version 4.5.2

## Loading required package: lpSolve

## Warning: package 'lpSolve' was built under R version 4.5.2
```

```r
library(psych)
```

```
## Warning: package 'psych' was built under R version 4.5.2

##
## Attaching package: 'psych'
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```r
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 4.5.2
```

```
## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
##
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
library(lme4)
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```r
library(broom.mixed)
library(DescTools)
```

```
## Warning: package 'DescTools' was built under R version 4.5.2
```

```
##
## Attaching package: 'DescTools'
##
## The following objects are masked from 'package:psych':
##
##     AUC, ICC, SD
```

```r
library(knitr)
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following objects are masked from 'package:DescTools':
##
##     MAE, RMSE
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
df <- read_csv("data_t1.csv", col_types = cols())
```

```
## New names:
## * `` -> `...1`
```

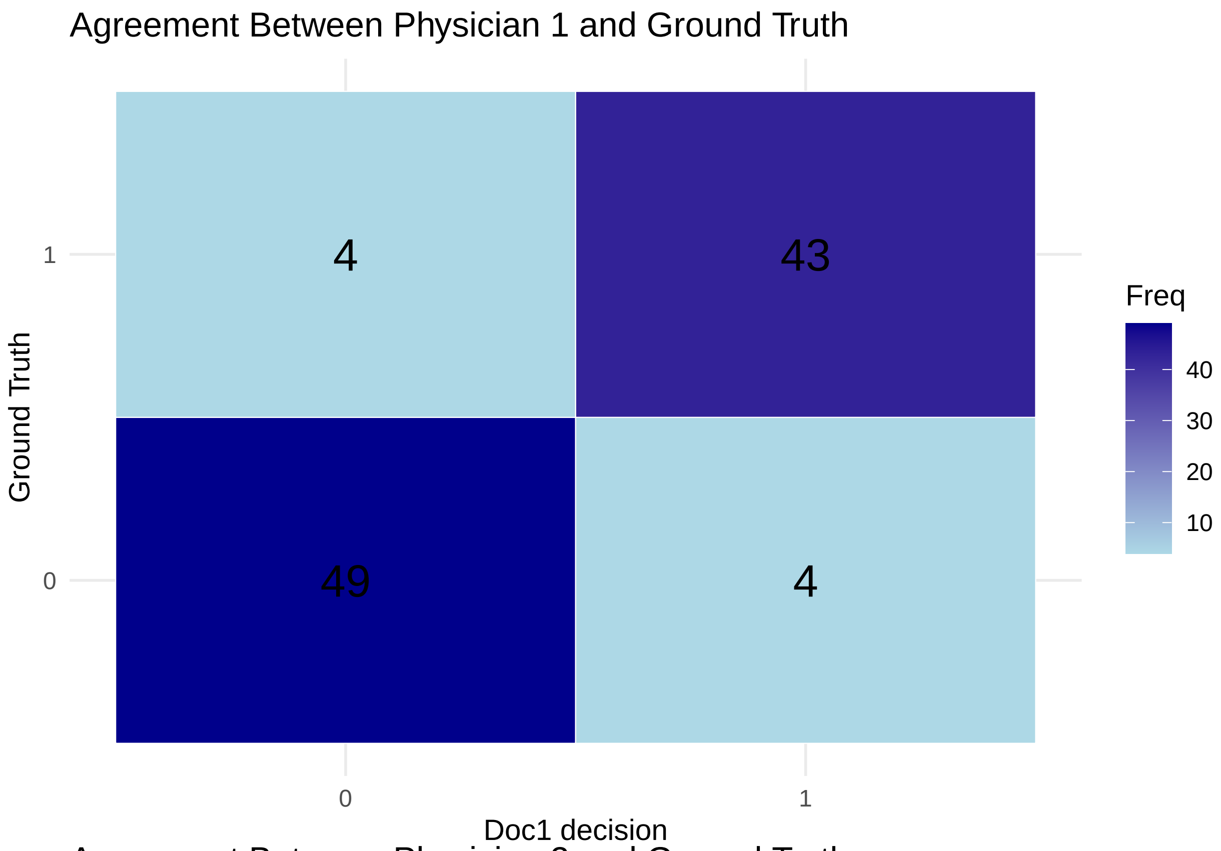
```
head(df)
```

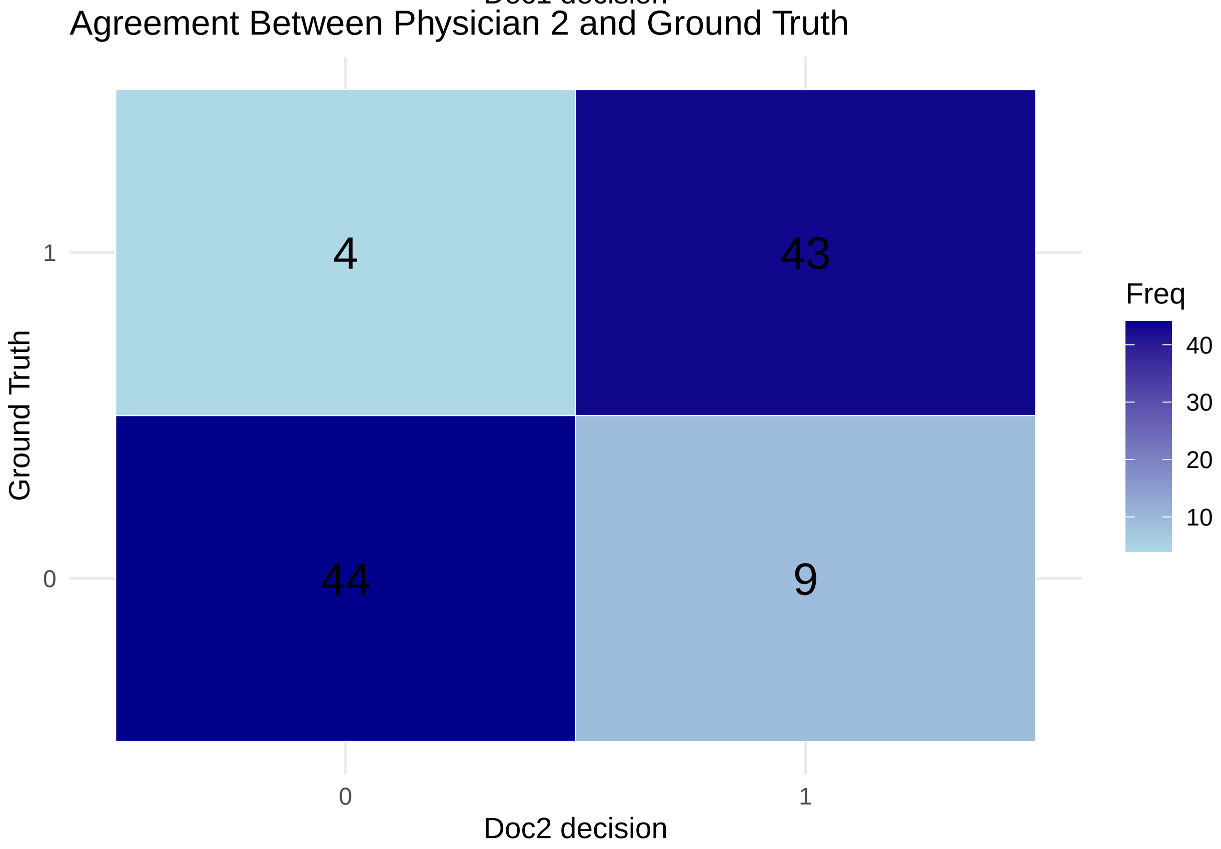
```
## # A tibble: 6 x 4
##    ...1 Patient_Diagnosis Rater1 Rater2
##   <dbl>             <dbl>  <dbl>  <dbl>
## 1     1                 0      0      0
## 2     2                 1      1      0
## 3     3                 0      0      0
## 4     4                 1      0      1
## 5     5                 1      1      1
## 6     6                 0      0      1
```

```
df <- df[ , -1]
df <- df %>%
  rename(truth = Patient_Diagnosis,
         doc1 = Rater1,
         doc2 = Rater2) %>%
  mutate(across(c(truth, doc1, doc2), ~as.integer(.)))  # ensure 0/1 ints
```

From the loaded data, there are 2 raters in this study. The study include 100 individuals; each individual is diagnosed by two doctors separately, and has a reference diagnostic result also.

Step 2: Doctors' Decision vs. Ground Truth





```
##      Doc1
## Truth  0  1
##     0 49  4
##     1  4 43
```

```
##      Doc2
## Truth  0  1
##     0 44  9
##     1  4 43
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 49  4
##          1  4 43
##
##                Accuracy : 0.92
##                  95% CI : (0.8484, 0.9648)
##     No Information Rate : 0.53
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8394
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9149
##             Specificity : 0.9245
##          Pos Pred Value : 0.9149
##          Neg Pred Value : 0.9245
##              Prevalence : 0.4700
##          Detection Rate : 0.4300
##    Detection Prevalence : 0.4700
##       Balanced Accuracy : 0.9197
##
##        'Positive' Class : 1
##
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 44  4
##          1  9 43
##
##                Accuracy : 0.87
##                  95% CI : (0.788, 0.9289)
##     No Information Rate : 0.53
##     P-Value [Acc > NIR] : 4.774e-13
##
##                   Kappa : 0.7406
##
##  Mcnemar's Test P-Value : 0.2673
```

```
##
##             Sensitivity : 0.9149
##             Specificity : 0.8302
##          Pos Pred Value : 0.8269
##          Neg Pred Value : 0.9167
##              Prevalence : 0.4700
##          Detection Rate : 0.4300
##    Detection Prevalence : 0.5200
##       Balanced Accuracy : 0.8725
##
##        'Positive' Class : 1
##
```

Doctor 1 has the accuracy of 92%, detects 91.5% of true cancer cases, correctly rules out cancer in 92.4% of non-cancer cases. Kappa value is 0.839, showing strong agreement.P-value for McNemar's test is 1, showing no significant difference between false positives and false negatives. Doc1 shows very strong diagnostic performance, does not show systematic bias toward over- or under-referral.

Doctor 2 has the accuracy of 87%, detects 91.5% of true cancer cases, correctly rules out cancer in 83.0% of non-cancer cases. Kappa value is 0.740, showing relatively strong agreement.P-value for McNemar's test is 0.267, showing no significant asymmetry in errors, but doc2 tends toward more false positives.

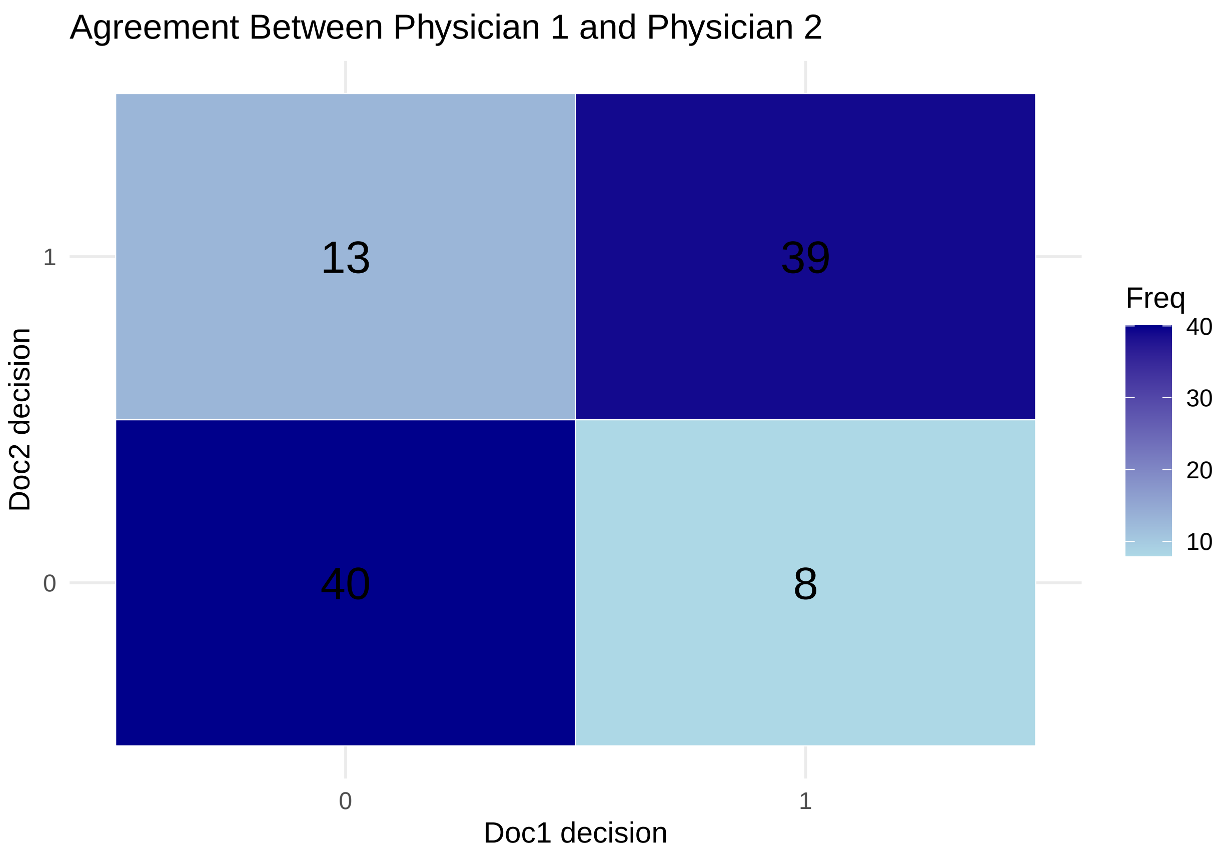
From the accuracy, specificity, PPV and NPV, diagnostic decision of doctor 1 is considered more precise than doctor 2. Doctor 2 tend to diagnose some healthy individuals as diseased.

Step 3: Comparing Between Doctors

We also did agreement tests for the two doctors.

```
##     Doc2
## Doc1  0  1
##    0 40 13
##    1  8 39
```

```
##     Doc2
## Doc1    0    1
##    0 0.40 0.13
##    1 0.08 0.39
```



```
##  Cohen's Kappa for 2 Raters (Weights: unweighted)
##
##  Subjects = 100
##    Raters = 2
##     Kappa = 0.581
##
##         z = 5.84
##   p-value = 5.25e-09
```

```
##
##  McNemar's Chi-squared test with continuity correction
##
## data:  tab_docs
## McNemar's chi-squared = 0.7619, df = 1, p-value = 0.3827
```

Out of 100 cases, two doctors agreed on most cases (79%). The Kappa value is 0.581, indicating moderate agreement. McNemar's test shows a p-value of 0.3827, neither doctor is systematically over- or under-referring compared to the other.

```r
df$patient_id <- 1:nrow(df)
df_long <- df %>%
  pivot_longer(cols = c(doc1, doc2),
               names_to = "rater",
               values_to = "decision")

df_long$decision <- as.integer(df_long$decision)
```

```r
df_long$truth <- as.integer(df_long$truth)

model <- glmer(
  decision ~ truth + (1 | rater) + (1 | patient_id),
  data = df_long,
  family = binomial
)
```

```
## boundary (singular) fit: see help('isSingular')
```

```r
summary(model)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: decision ~ truth + (1 | rater) + (1 | patient_id)
##    Data: df_long
## 
##       AIC       BIC    logLik -2*log(L)  df.resid 
##     141.6     154.8     -66.8     133.6       196 
## 
## Scaled residuals: 
##     Min      1Q  Median      3Q     Max 
## -3.2787 -0.3739 -0.3739  0.3050  2.6747 
## 
## Random effects:
##  Groups     Name        Variance  Std.Dev. 
##  patient_id (Intercept) 7.272e-16 2.697e-08
##  rater      (Intercept) 0.000e+00 0.000e+00
## Number of obs: 200, groups:  patient_id, 100; rater, 2
## 
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  -1.9677     0.2961  -6.645 3.03e-11 ***
## truth         4.3426     0.4736   9.169  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Correlation of Fixed Effects:
##       (Intr)
## truth -0.625
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```r
library(DHARMa)
```

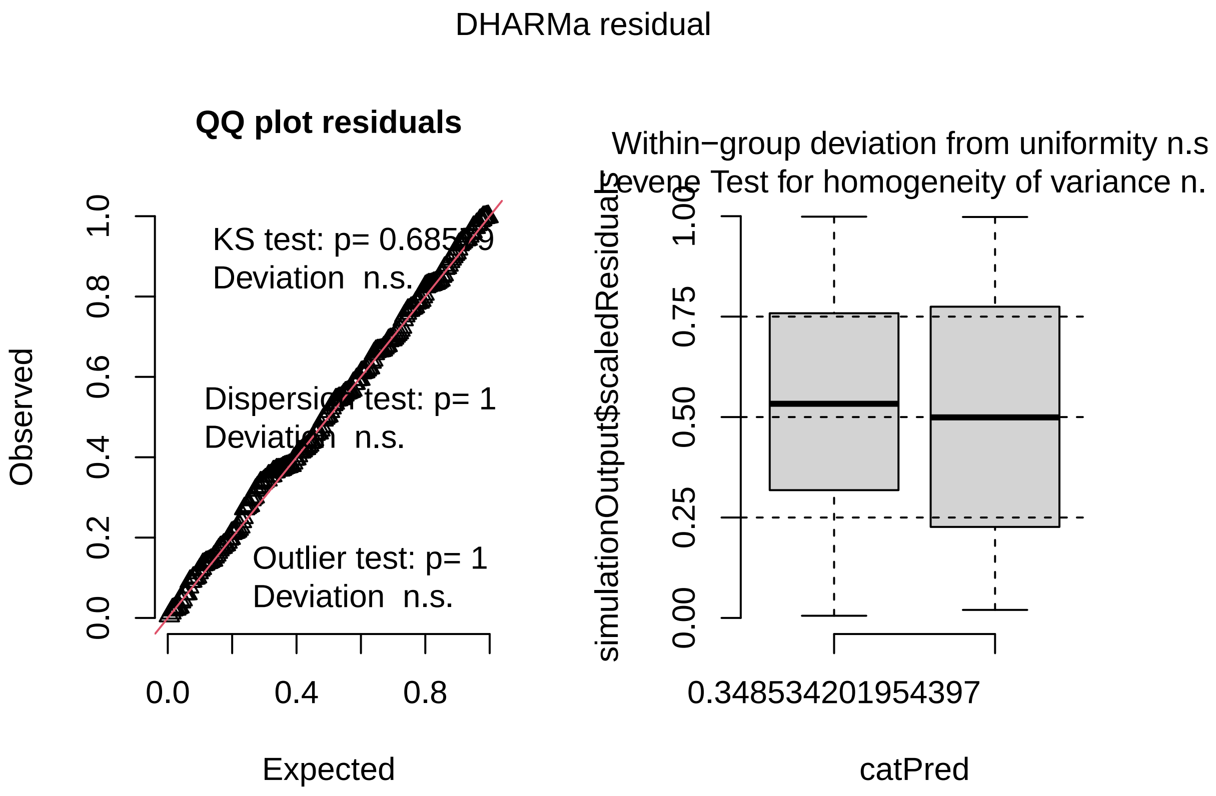
```
## This is DHARMa 0.4.7. For overview type '?DHARMa'. For recent changes, type news(package = 'DHARMa')
```

```r
library(jtools)
```

```
## Warning: package 'jtools' was built under R version 4.5.2
```

```
##
## Attaching package: 'jtools'

## The following object is masked from 'package:DescTools':
##
##     %nin%
```

```
m1_res <- simulateResiduals(model)
plot(m1_res)
```



From the mixed effect model, we can se that random effect is insignificant in this case, showing patient individuals and doctors' decisions behave almost identically in this dataset. Decisions are almost entirely explained by the truth, not by which doctor or which patient.

Residuals are mostly small meaning model fits well.

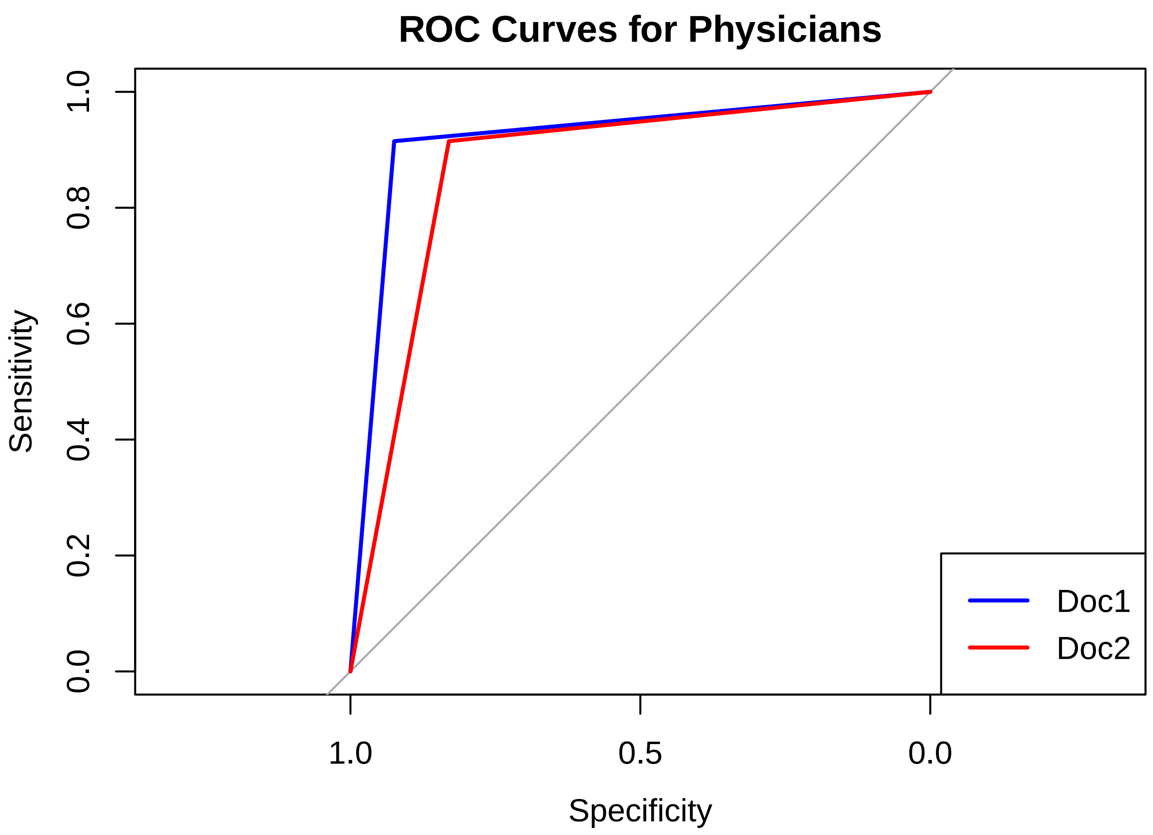
```
roc1 <- roc(df$truth, df$doc1)
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```

```
roc2 <- roc(df$truth, df$doc2)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```r
plot(roc1, col="blue", main="ROC Curves for Physicians")
lines(roc2, col="red")
legend("bottomright", legend=c("Doc1","Doc2"), col=c("blue","red"), lwd=2)
```



```r
roc.test(roc1, roc2)
```

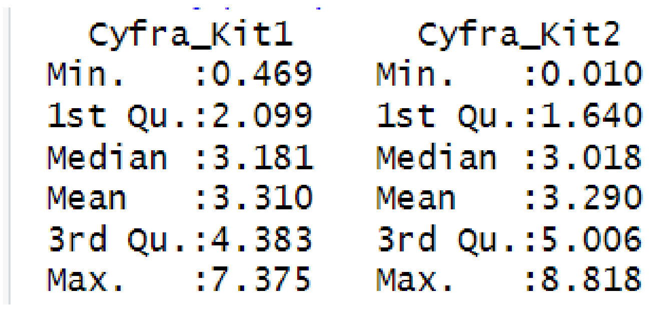
```
##
##  DeLong's test for two correlated ROC curves
##
## data:  roc1 and roc2
## Z = 1.0389, p-value = 0.2989
## alternative hypothesis: true difference in AUC is not equal to 0
## 95 percent confidence interval:
##  -0.04182046  0.13616008
## sample estimates:
## AUC of roc1 AUC of roc2
##   0.9197110   0.8725411
```

AUC of both doctor is high; and there is no statistically significant difference in AUC between Doc1 and Doc2.

Task 2:

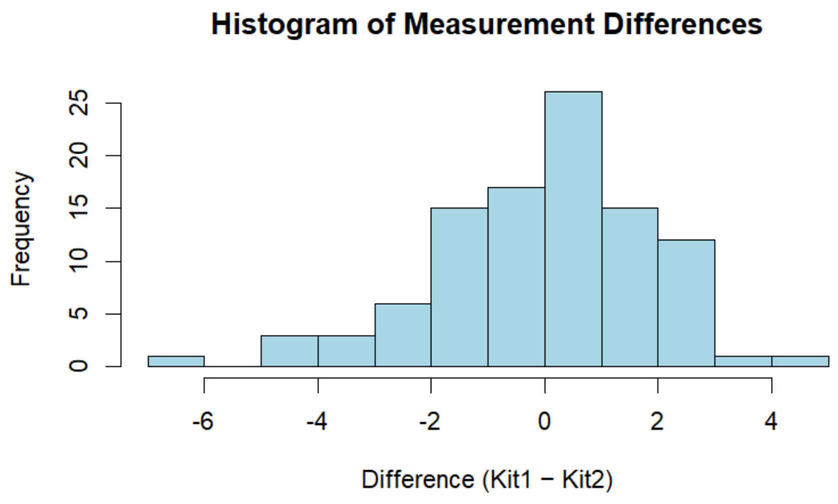
Explore methods for analysing the agreement and compare these.

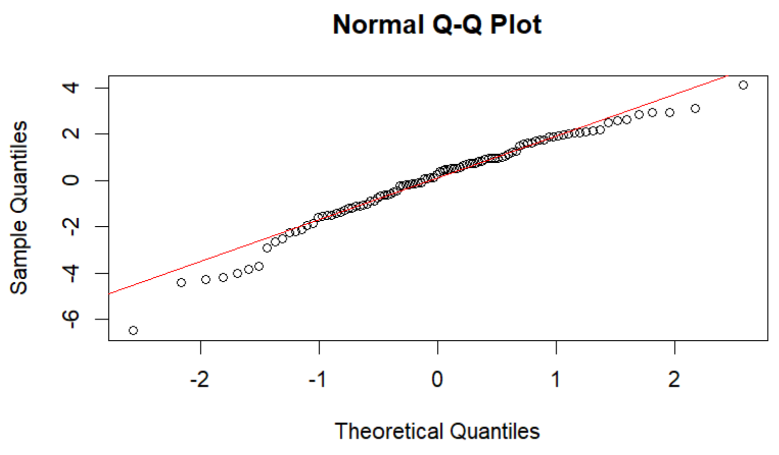
We start by exploring the data set. Which includes two different measurement kits.



```
  Cyfra_Kit1        Cyfra_Kit2
 Min.   :0.469    Min.   :0.010
 1st Qu.:2.099    1st Qu.:1.640
 Median :3.181    Median :3.018
 Mean   :3.310    Mean   :3.290
 3rd Qu.:4.383    3rd Qu.:5.006
 Max.   :7.375    Max.   :8.818
```

We compute the differences for each measurement and visualize it using different plots. We continue with estimating the normality of the difference.



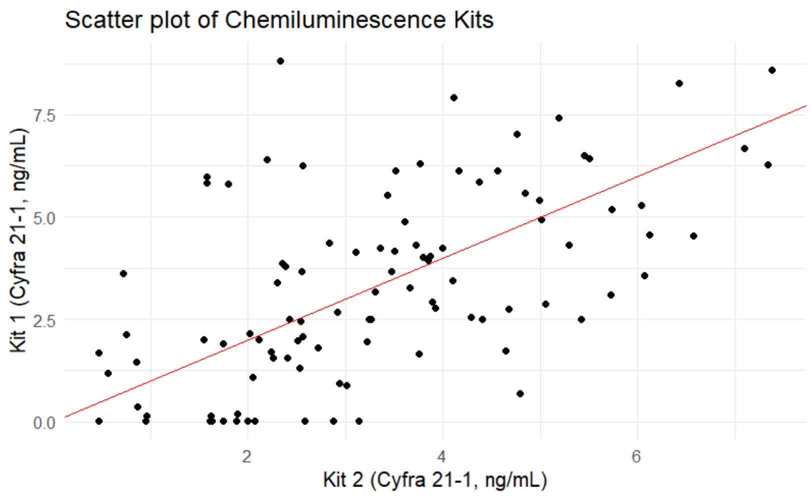


```
> shapiro.test(data$diff)

	Shapiro-Wilk normality test

data:  data$diff
W = 0.96771, p-value = 0.01481
```

According to the normality test we should reject the hypothesis that the difference is normal. However, according to both the histogram and the qq plot we still assume normality. Since approximate normality is something that the bland-altman assumes and not perfect.

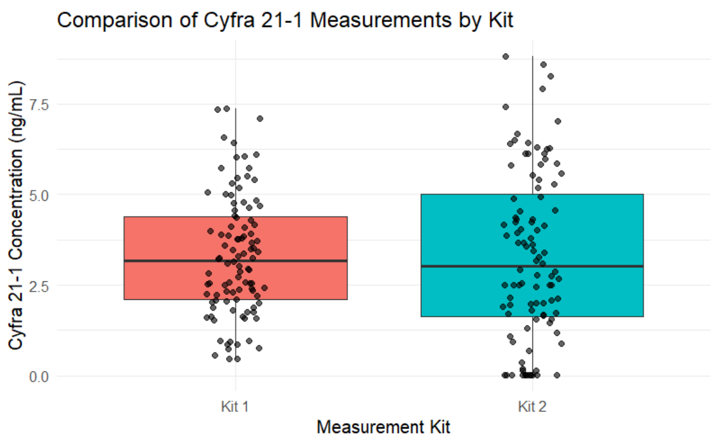
We continue with visualising the agreement of the two kits.

We start with a scatter plot of the two kits.

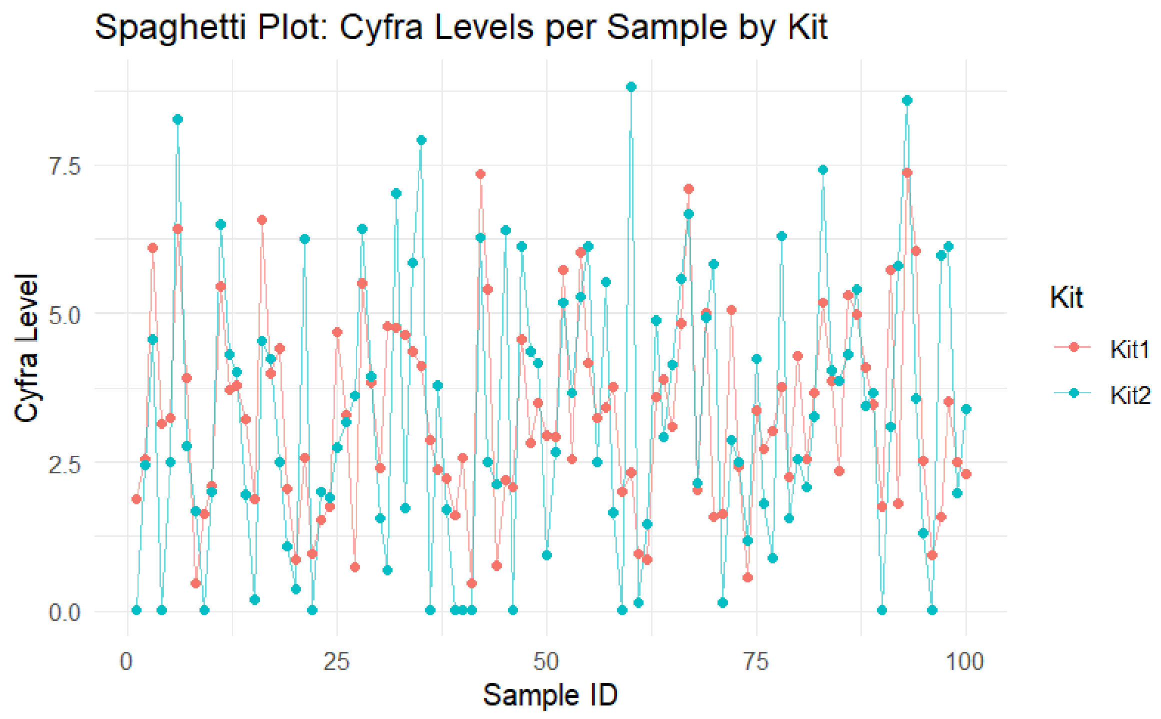


We can not determine a clear dependence between the two kits.

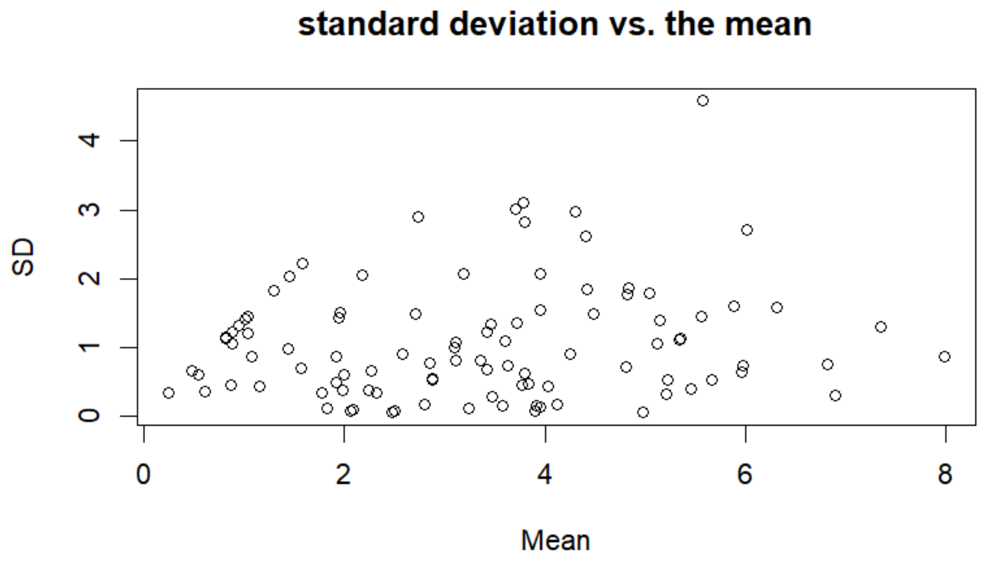
We go on to plot a side by side histogram plot which indicates that kit 2 has larger variability.



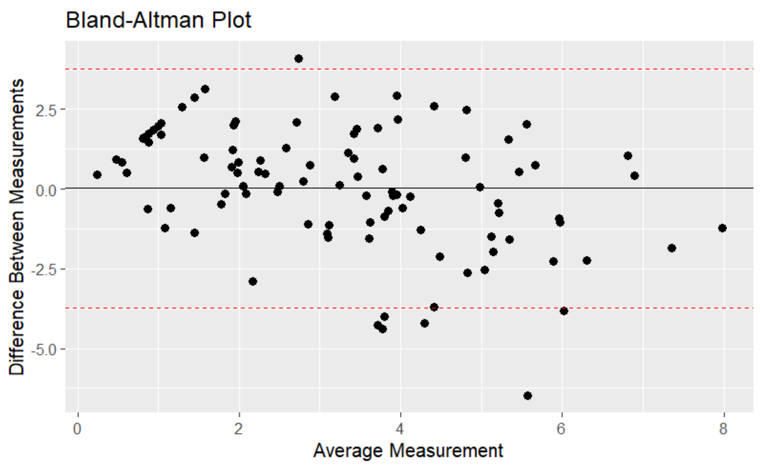
A spaghetti plot of individual samples measured by both kits shows that Kit 2 consistently produces either systematically higher or lower values for each sample, suggesting a difference in spread as seen in the boxplot.



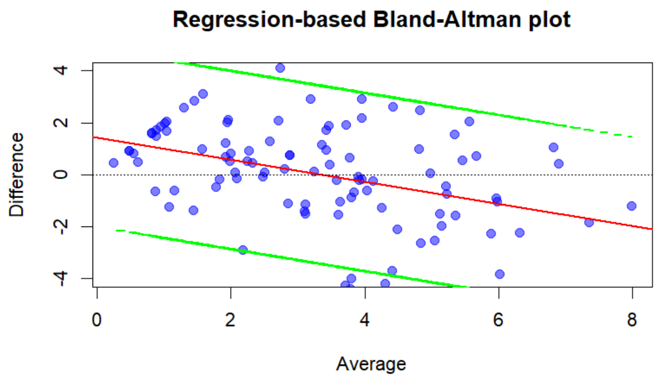
Plotting the standard deviation vs. the mean reveals no strong evidence of proportional heteroscedasticity, but we observe slightly wider variability at higher concentrations. This hints that a regression-based agreement model may be more appropriate.



The standard BA plot shows a downward trend in differences across the concentration range. This indicates: Proportional bias and violation of the constant difference assumption which motivates trying a regression-based method.



We try a regression-based Bland-Altman plot, and we clearly see that this is more appropriate.



CCC tells you whether two methods truly agree, while ICC tells you whether measurements are consistent — you can have a high ICC and still have poor agreement. This is why we compute Lin´s CCC.

```
> # Lin's CCC from epiR
> epi.ccc(data$Cyfra_Kit1, data$Cyfra_Kit2)
$rho.c
        est     lower     upper
1 0.5518658 0.4130112 0.6656461
```

0.5518 -> This value indicates poor to moderate agreement. CCC values closer to 1.0 indicate perfect agreement.

The CL interval is far from 1.0, confirming the poor agreement is statistically significant

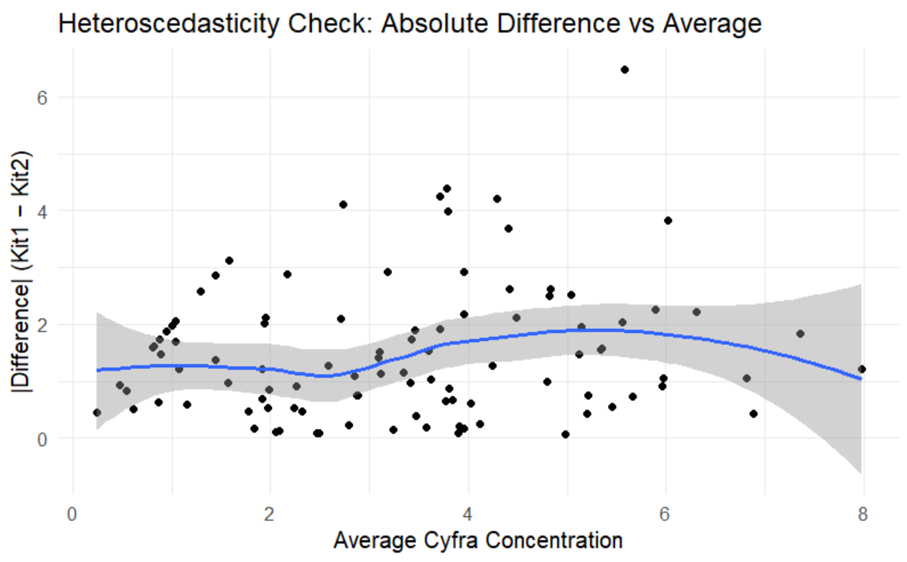
```
$s.shift
[1] 1.408829

$l.shift
[1] -0.01027646
```

1.41 -> relates to the ratio of the Standard Deviations of the two methods SD_Kit2 /SD_Kit1. A value >1 means Kit2 has a wider spread higher SD than Kit1.

-0.01 -> This relates to the difference between the means of the two methods Mean_Kit2 - Mean_Kti1. A small value means the means are very similar. Despite the near-zero mean difference, there is large random error and poor precision.

Plotting the difference against measurement magnitude shows:

- Larger differences at higher concentration
- No major curvature, but visible proportional bias
- Supports the regression-based BA method



Which method is most suitable given the data?

Given the observations:

- Non-constant variability
- Downward trend in the standard BA plot
- Evidence of proportional bias
- Larger variability in Kit 2
- CCC confirming poor precision

The regression-based Bland–Altman method is the most appropriate for this dataset.

It properly accounts for:

- Proportional bias
- Heteroscedasticity
- Non-constant limits of agreement

The classical Bland–Altman would misrepresent the agreement because it assumes constant variability.

task3_report

December 7, 2025

[86]:
```
#https://www.emilyzabor.com/survival-analysis-in-r.html
data <- read.csv("data_t3.csv")
```

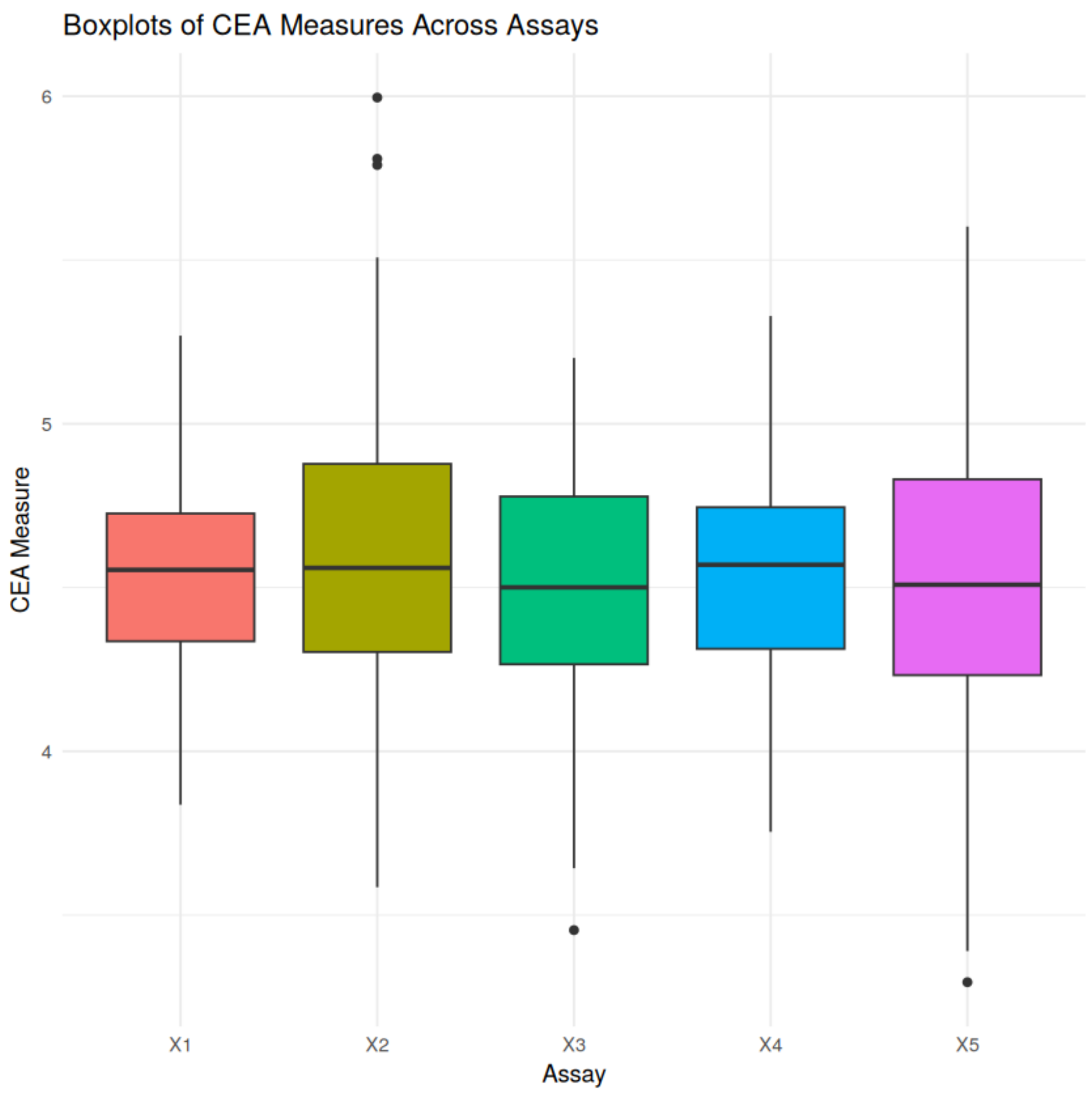
[78]:
```
# a first look at the data
head(data)
summary(data)
```

A data.frame: 6 × 6

	X <int>	X1 <dbl>	X2 <dbl>	X3 <dbl>	X4 <dbl>	X5 <dbl>
1	1	4.248	5.321	4.185	4.318	4.123
2	2	4.492	5.022	4.275	4.209	4.083
3	3	4.954	4.848	4.771	4.847	5.337
4	4	4.492	4.766	4.300	4.516	4.793
5	5	4.419	4.352	4.453	4.666	4.008
6	6	5.031	4.800	5.097	4.701	4.986

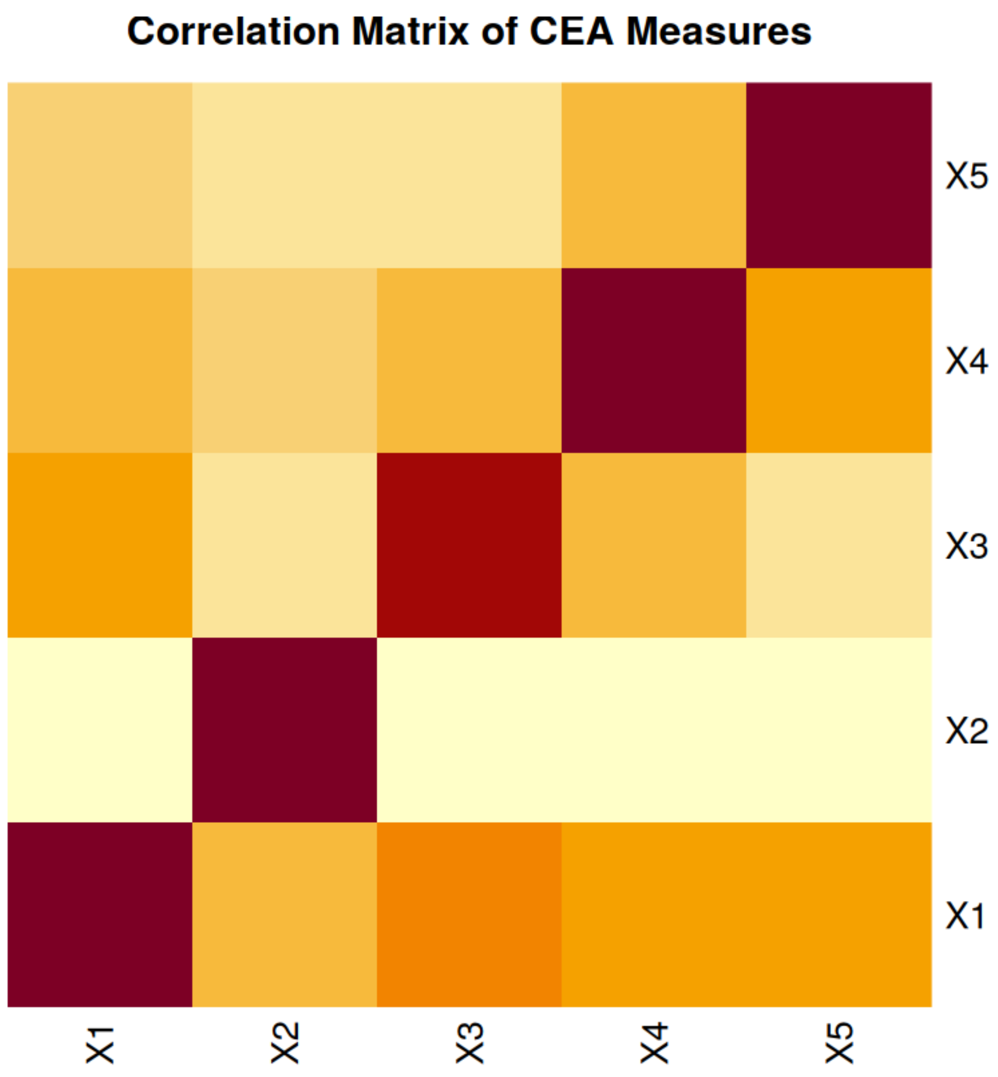
```
       X                X1              X2              X3
 Min.   :  1.00   Min.   :3.837   Min.   :3.585   Min.   :3.454
 1st Qu.: 25.75   1st Qu.:4.336   1st Qu.:4.303   1st Qu.:4.266
 Median : 50.50   Median :4.554   Median :4.560   Median :4.500
 Mean   : 50.50   Mean   :4.534   Mean   :4.581   Mean   :4.529
 3rd Qu.: 75.25   3rd Qu.:4.726   3rd Qu.:4.878   3rd Qu.:4.778
 Max.   :100.00   Max.   :5.269   Max.   :5.996   Max.   :5.201
       X4              X5
 Min.   :3.754   Min.   :3.295
 1st Qu.:4.313   1st Qu.:4.232
 Median :4.569   Median :4.508
 Mean   :4.547   Mean   :4.517
 3rd Qu.:4.745   3rd Qu.:4.830
 Max.   :5.329   Max.   :5.602
```

[83]:
```
ggplot(data_long, aes(x = Assay, y = CEA, fill = Assay)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Boxplots of CEA Measures Across Assays",
       y = "CEA Measure",
       x = "Assay") +
  theme(legend.position = "none")
```



We can observe there is an average rater agreement across all the raters by observing the box plots.

[84]:
```
# Compute the correlation matrix
cor_matrix <- cor(data[-1])

# Plotting the heatmap
heatmap(cor_matrix, main = "Correlation Matrix of CEA Measures",
        Colv = NA, Rowv = NA, scale = "column", margins = c(5, 5))
```



We can observe some slight correlation between raters but nothing concrete further testing is requried to form any concrete observations.

[85]:
```
library(psych)

# Calculate the ICC
icc_result <- ICC(data[-1])

# Convert to data frame (if not already done)
icc_df <- as.data.frame(icc_result$results)

# View the structure to confirm
print(icc_df)
```

```
boundary (singular) fit: see help('isSingular')
```

```
                         type       ICC        F df1 df2            p
Single_raters_absolute   ICC1 0.4367499 4.877052  99 400 5.016847e-30
Single_random_raters     ICC2 0.4367499 4.877052  99 396 6.621512e-30
Single_fixed_raters      ICC3 0.4367499 4.877052  99 396 6.621512e-30
Average_raters_absolute ICC1k 0.7949581 4.877052  99 400 5.016847e-30
Average_random_raters   ICC2k 0.7949581 4.877052  99 396 6.621512e-30
Average_fixed_raters    ICC3k 0.7949581 4.877052  99 396 6.621512e-30
                        lower bound upper bound
Single_raters_absolute    0.3439548   0.5355998
Single_random_raters      0.3439500   0.5356031
Single_fixed_raters       0.3438419   0.5356767
Average_raters_absolute   0.7238657   0.8522148
Average_random_raters     0.7238615   0.8522164
Average_fixed_raters      0.7237657   0.8522537
```

Here above we can see the p values are quite significant for all the cases. ICC1 - One-way random-effects model. In this model, each subject is rated by a different set of randomly chosen raters. Here, raters are considered as the random effects ICC2 - Two-way random-effects model. A set of k raters are randomly selected, then, each subject is measured by the same set of k raters with similar characteristics. In this model, both subjects and raters are viewed as random effects. ICC3 - Two-way mixed effects model. Here the raters are considered as fixed.

We get the Similar ICC of 0.4467 for all the 3 models showing moderate agreement.

Similarly for Average ICC 1 to 3 the ICC is same and it's 0.749 showing good agreement i.e. the raters are usually consistent when averages are compared.

0.1 Alternative

```
[97]: kripp.alpha(t(as.matrix(data[-1])), method = "interval")
```

```
 Krippendorff's alpha

 Subjects = 100
   Raters = 5
    alpha = 0.434
```

Krippoendorff's alpha shows a value of 0.434 which shows moderate agreement across all 5 raters.

task4

zhexuan

2025-12-06

Task 4

In this task, we evaluated a classification model for the early detection of lung cancer using primary care data. The model outputs a predicted probability of lung cancer for each patient, and it was externally validated on a retrospective dataset of 1,000 matched patients with and without a lung cancer diagnosis. The main goals of the analysis were to assess the performance of the model, and then choose appropriate thresholds, taking into account the severe consequences of missing true lung cancer cases.

Data

The dataset contains 1,000 patients with observed outcome (0 for no lung cancer and 1 for lung cancer) and the model-predicted probability of lung cancer.

```
dat <- read.csv("data_t4.csv")

str(dat)
```

```
## 'data.frame':    1000 obs. of  3 variables:
##  $ X         : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ labels_obs: int  1 0 1 0 0 1 0 0 0 1 ...
##  $ prob_pred : num  0.586 0.348 0.729 0.655 0.432 ...
```

```
summary(dat)
```

```
##        X            labels_obs      prob_pred
##  Min.   :   1.0   Min.   :0.000   Min.   :0.0588
##  1st Qu.: 250.8   1st Qu.:0.000   1st Qu.:0.4770
##  Median : 500.5   Median :1.000   Median :0.6207
##  Mean   : 500.5   Mean   :0.507   Mean   :0.6121
##  3rd Qu.: 750.2   3rd Qu.:1.000   3rd Qu.:0.7567
##  Max.   :1000.0   Max.   :1.000   Max.   :1.0000
```

Model Performance

To assess how well the model can distinguish between patients with and without lung cancer, we first computed a receiver operating characteristic (ROC) curve. The x-axis is actually specificity, which is plotted from 1 down to 0, so it looks reversed compared with the standard “FPR on the x-axis” ROC plots. The dashed diagonal line represents the performance of a classifier that randomly guesses class labels.

The resulting area under the ROC curve (AUC) was 0.8533. An AUC of 0.5 would indicate a model that performs no better than random guessing, while an AUC of 1.0 would correspond to perfect discrimination. An AUC around 0.85 therefore suggests that the model has good discriminative ability.

```
roc_obj <- roc(response = dat$labels_obs,
               predictor = dat$prob_pred)
```

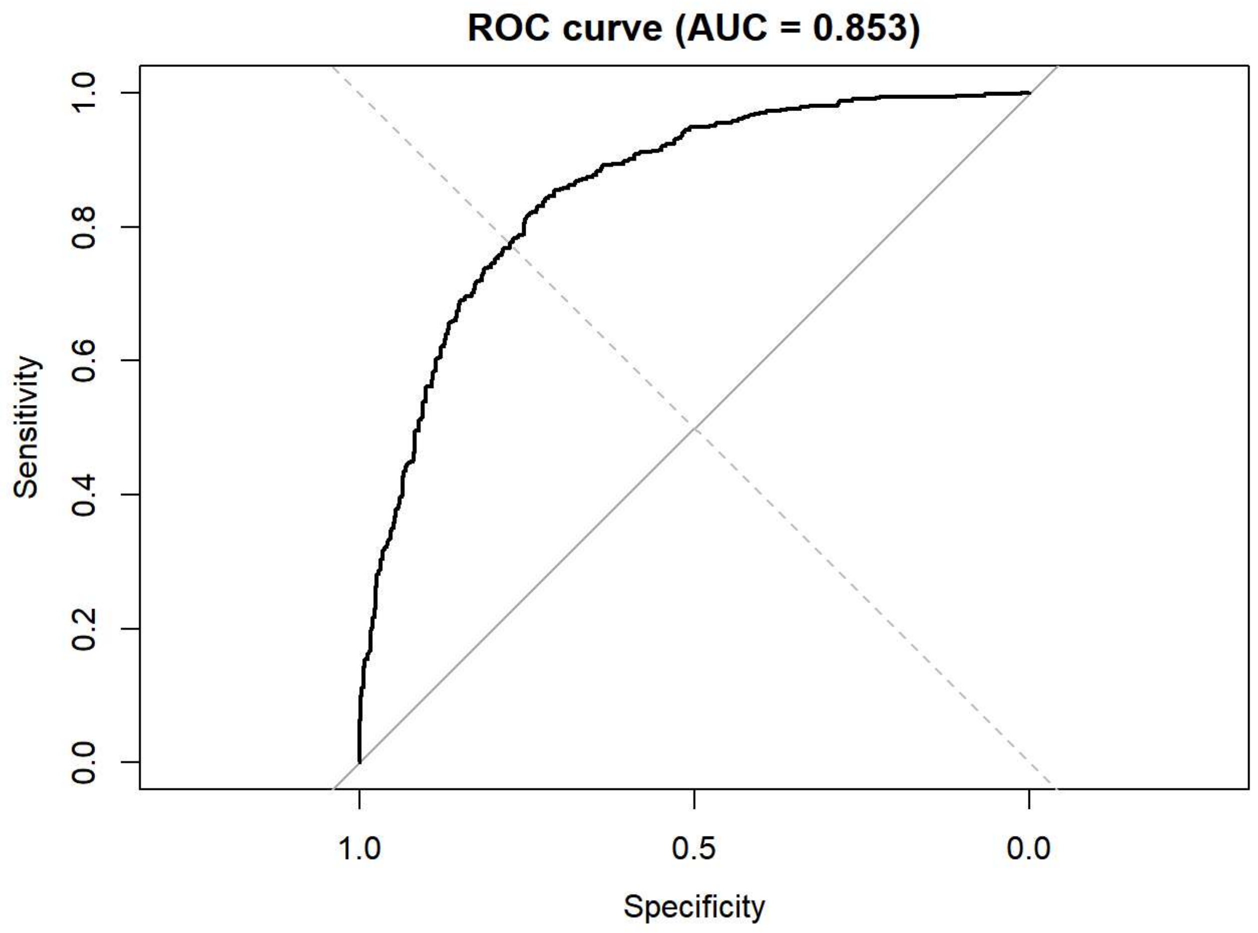
```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
# AUC
auc_value <- auc(roc_obj)
auc_value
```

```
## Area under the curve: 0.8533
```

```
# ROC curve
plot(roc_obj,
     main = paste0("ROC curve (AUC = ", round(auc_value, 3), ")"))
abline(a = 0, b = 1, lty = 2, col = "gray")
```



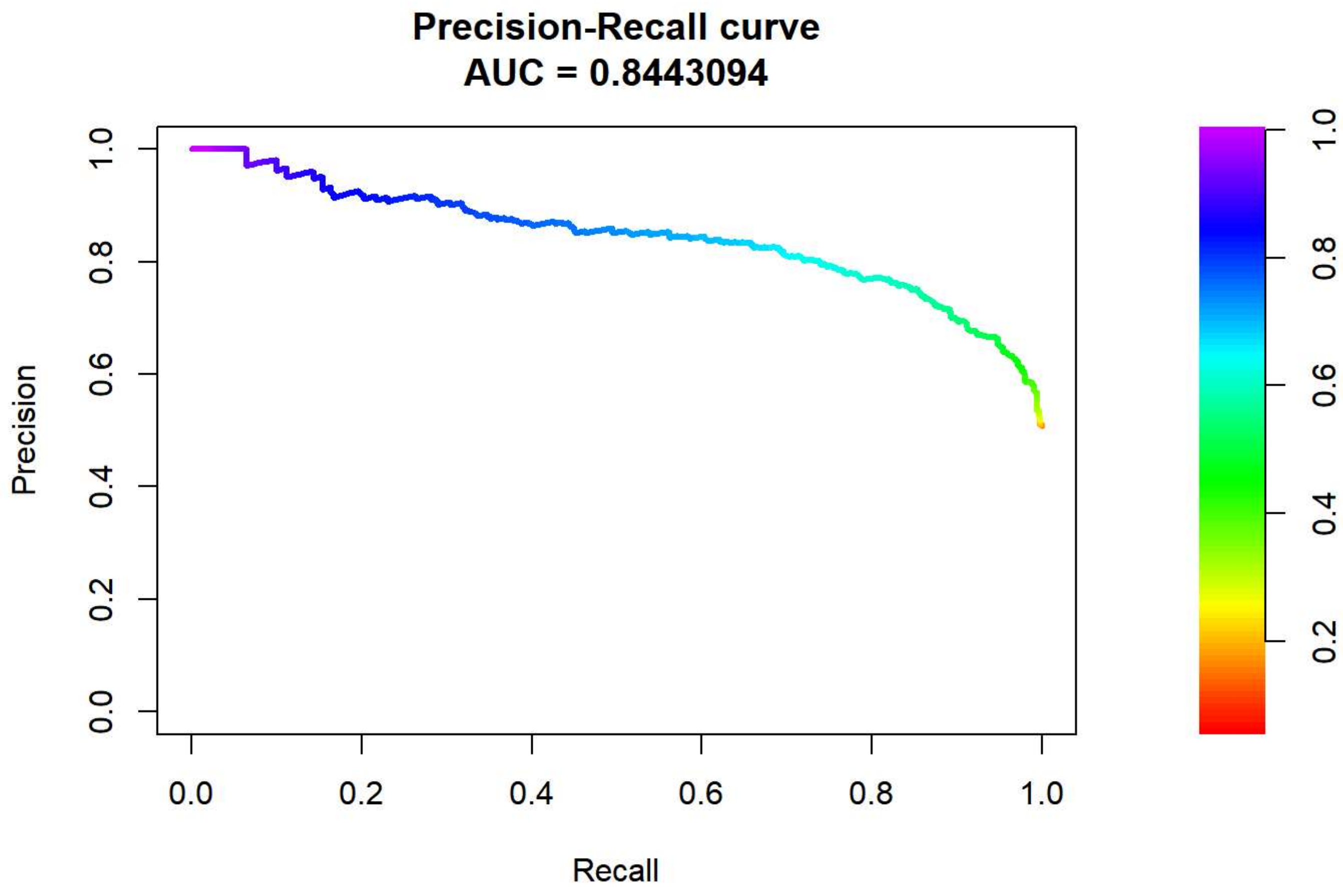
Precision–recall analysis could link recall (sensitivity) to the precision of the positive predictions. This is especially relevant in an early detection context, where the clinical priority is usually to maximize recall while keeping precision at a clinically acceptable level.

The area under the PR curve was 0.844, indicating that the model maintains a favorable trade-off between recall and precision across a range of thresholds.

```
scores_pos <- dat$prob_pred[dat$labels_obs == 1]  # cancer
scores_neg <- dat$prob_pred[dat$labels_obs == 0]  # non-cancer

## PR curve
pr_obj <- pr.curve(scores.class0 = scores_pos,
                   scores.class1 = scores_neg,
                   curve = TRUE)

pr_obj$auc.integral  # PR AUC
```

```
## [1] 0.8443094
```

```
plot(pr_obj,
     main = paste0("Precision-Recall curve"))
```



For clinical decision-making, it is also important that the predicted probabilities themselves are well-calibrated. To assess calibration, we first computed the Brier score, which is the mean squared error between the predicted probabilities and the observed outcomes (coded as 0 or 1). A lower value indicates better probabilistic predictions, with 0 representing perfect predictions. A Brier score of around 0.18 suggests that the model’s probability estimates are reasonably good but not perfect, leaving some room for miscalibration.

To visualize calibration in more detail across the risk spectrum, we grouped the patients into ten bins based on the deciles of the predicted risk. Because the sample size is 1000 and the probabilities are fairly smoothly distributed, each bin ended up containing exactly 100 observations.

For each bin, we computed the average predicted probability and the observed proportion of lung cancer cases. And we plotted these as a calibration plot. In this plot, the dashed diagonal line corresponds to perfect calibration, where the predicted and observed probabilities are equal.

In our results, the points for the lower-risk bins lie below the diagonal, which means that in those groups the observed cancer rate is slightly lower than the predicted risk. In other words, for low predicted probabilities, the model tends to overestimate the absolute risk a bit. For higher predicted risk levels (around a mean predicted probability of 0.7 and above), the points move above the diagonal, which indicates that in those high-risk groups the observed cancer rate is actually higher than the predicted risk. This pattern suggests that the model becomes somewhat too conservative at the highest risk levels, underestimating risk in those patients who are truly at very high risk.

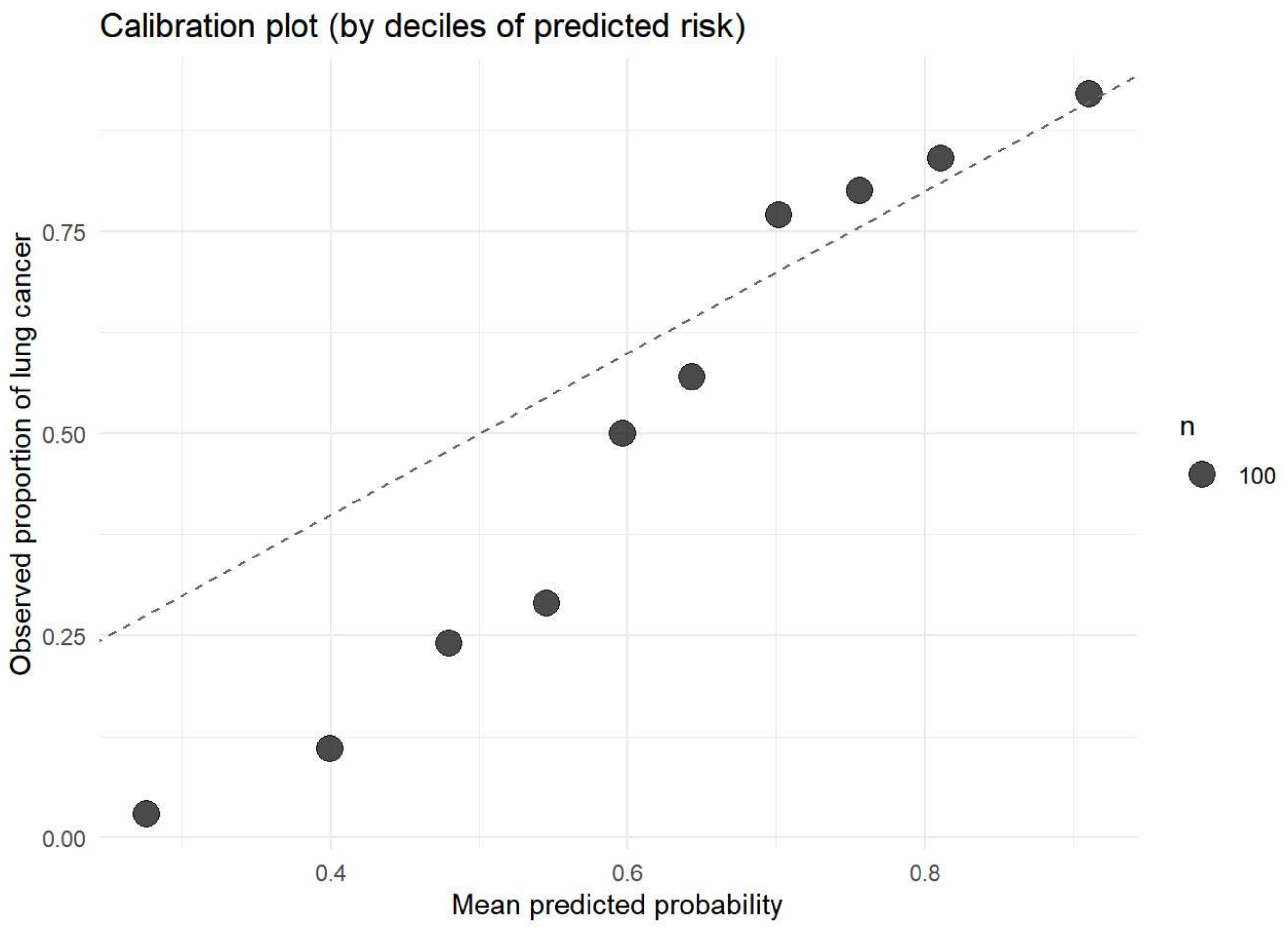
```
# Brier score: MSE
brier_score <- mean( (dat$prob_pred - dat$labels_obs)^2 )
brier_score
```

```
## [1] 0.1836193
```

```
# split prob_pred into 10 groups
dat_cal <- dat %>%
  mutate(bin = cut(prob_pred,
                   breaks = quantile(prob_pred, probs = seq(0, 1, by = 0.1)),
                   include.lowest = TRUE)) %>%
  group_by(bin) %>%
  summarise(
    mean_pred = mean(prob_pred),
    obs_rate  = mean(labels_obs),
    n = n()
  )

dat_cal
```

```
## # A tibble: 10 × 4
##    bin            mean_pred obs_rate     n
##    <fct>              <dbl>    <dbl> <int>
##  1 [0.0588,0.345]     0.276     0.03   100
##  2 (0.345,0.443]      0.400     0.11   100
##  3 (0.443,0.52]       0.480     0.24   100
##  4 (0.52,0.57]        0.546     0.29   100
##  5 (0.57,0.621]       0.597     0.5    100
##  6 (0.621,0.674]      0.643     0.57   100
##  7 (0.674,0.729]      0.702     0.77   100
##  8 (0.729,0.782]      0.756     0.8    100
##  9 (0.782,0.845]      0.811     0.84   100
## 10 (0.845,1]          0.911     0.92   100
```

```
# calibration plot
ggplot(dat_cal, aes(x = mean_pred, y = obs_rate, size = n)) +
  geom_point(alpha = 0.7) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "gray40") +
  labs(x = "Mean predicted probability",
       y = "Observed proportion of lung cancer",
       title = "Calibration plot (by deciles of predicted risk)") +
  theme_minimal()
```



Threshold Determination

The choice of threshold has direct clinical implications, especially in a setting such as lung cancer detection where the cost of missing a true cancer case is substantial.

We first used the ROC curve to identify the Youden-optimal threshold, which maximizes the quantity sensitivity + specificity − 1. This threshold represents the point on the ROC curve that achieves the best overall balance between correctly identifying cancer cases and correctly identifying non-cancer cases. Based on our data, the Youden index was maximized at a threshold of 0.6025, providing a statistically well-balanced operating point. However, this balance does not necessarily reflect clinical priorities, because it treats false negatives and false positives as equally costly.

```
tpr <- roc_obj$sensitivities
fpr <- 1 - roc_obj$specificities
thresholds <- roc_obj$thresholds

# Youden
youden <- tpr - fpr

# threshold for the maximum Youden
optimal_threshold <- thresholds[which.max(youden)]

optimal_threshold
```

```
## [1] 0.6025
```

To better understand how the model behaves across the full range of thresholds, we evaluated performance at every value from 0 to 1 in increments of 0.01. For each threshold we calculated sensitivity, specificity, PPV, NPV, and accuracy, producing a detailed map of how model performance shifts as the threshold changes. As a reference point, we also examined the commonly used default threshold of 0.5. At this threshold, the sensitivity was high (0.949), but specificity dropped to 0.507. It indicated that although most cancer cases were correctly detected, the model produced many false alarms in practice. At the threshold 0.6, where we derived maximum Youden index, the sensitivity was 0.82 and the specificity was 0.74, with a PPV of 0.77, an NPV of 0.80, and an overall accuracy of 0.78. These values indicate a well-balanced operating point from a purely statistical perspective, with both sensitivity and specificity at reasonably high levels.

The comparison between 0.50 and 0.60 illustrates an important trade-off: raising the threshold improves specificity and reduces false positives, but at the cost of missing more true cancer cases.

```r
# calculate all the metrics
metrics_at_threshold <- function(threshold, labels, probs) {
  pred_class <- ifelse(probs >= threshold, 1, 0)

  TP <- sum(pred_class == 1 & labels == 1)
  FP <- sum(pred_class == 1 & labels == 0)
  TN <- sum(pred_class == 0 & labels == 0)
  FN <- sum(pred_class == 0 & labels == 1)

  sensitivity <- ifelse((TP + FN) > 0, TP / (TP + FN), NA)
  specificity <- ifelse((TN + FP) > 0, TN / (TN + FP), NA)
  PPV <- ifelse((TP + FP) > 0, TP / (TP + FP), NA)
  NPV <- ifelse((TN + FN) > 0, TN / (TN + FN), NA)
  accuracy <- (TP + TN) / (TP + TN + FP + FN)

  ## Youden index (J)
  youden <- sensitivity + specificity - 1

  data.frame(
    threshold = threshold,
    TP = TP, FP = FP, TN = TN, FN = FN,
    sensitivity = sensitivity,
    specificity = specificity,
    PPV = PPV,
    NPV = NPV,
    accuracy = accuracy,
    youden = youden
  )
}

# list of thresholds
thresholds <- seq(0, 1, by = 0.01)

# calculate metrics of all the thresholds
metrics_all <- do.call(rbind,
                       lapply(thresholds, metrics_at_threshold,
                              labels = dat$labels_obs,
                              probs = dat$prob_pred))

head(metrics_all)
```

```
##   threshold  TP  FP TN FN sensitivity specificity   PPV NPV accuracy youden
## 1      0.00 507 493  0  0           1           0 0.507  NA    0.507      0
## 2      0.01 507 493  0  0           1           0 0.507  NA    0.507      0
## 3      0.02 507 493  0  0           1           0 0.507  NA    0.507      0
## 4      0.03 507 493  0  0           1           0 0.507  NA    0.507      0
## 5      0.04 507 493  0  0           1           0 0.507  NA    0.507      0
## 6      0.05 507 493  0  0           1           0 0.507  NA    0.507      0
```

```
# reference: threshold = 0.5
metrics_at_threshold(0.5, dat$labels_obs, dat$prob_pred)
```

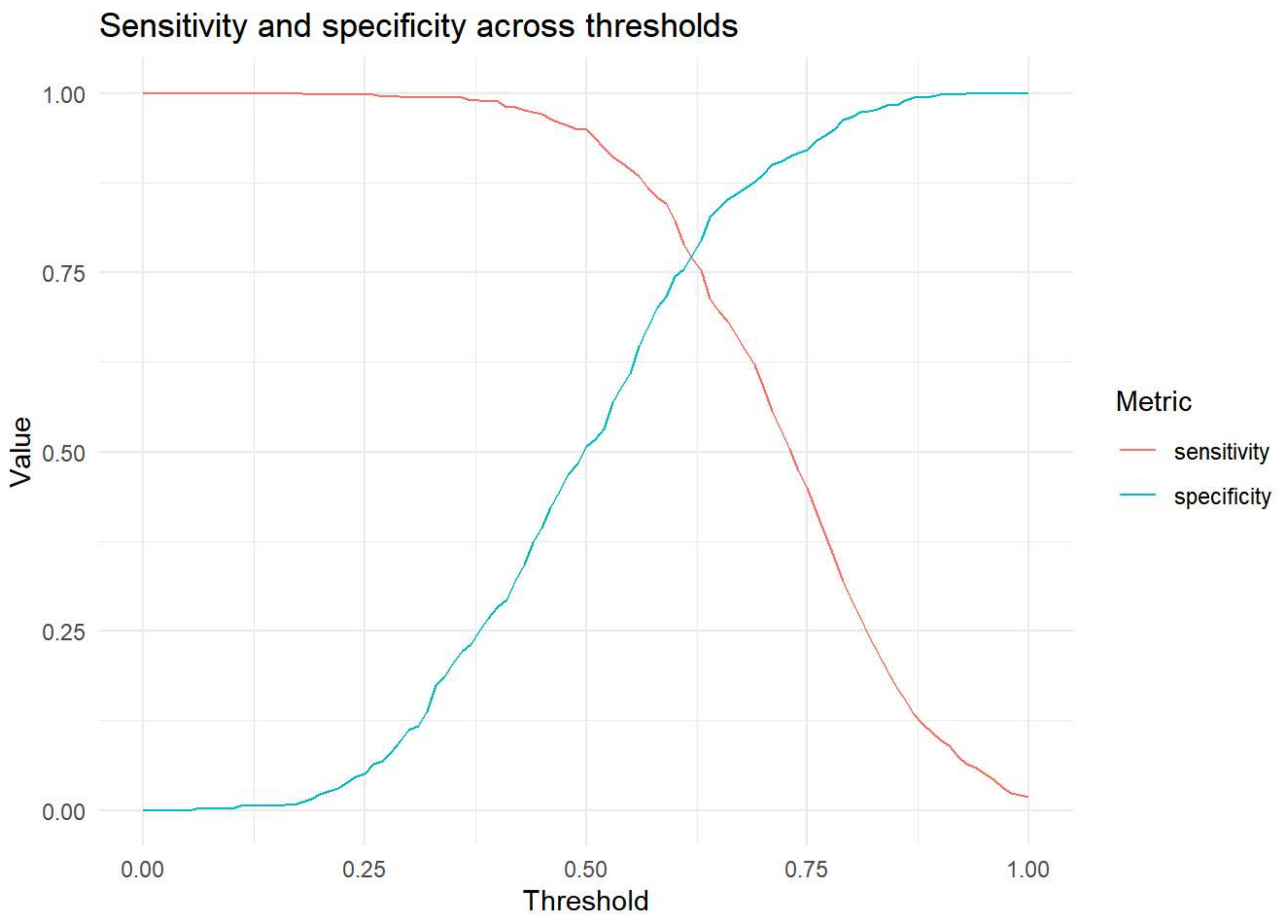
```
##   threshold  TP  FP  TN FN sensitivity specificity       PPV       NPV accuracy
## 1       0.5 481 243 250 26   0.9487179   0.5070994 0.6643646 0.9057971    0.731
##      youden
## 1 0.4558173
```

```
# threshold with maximum Youden
metrics_at_threshold(0.6, dat$labels_obs, dat$prob_pred)
```

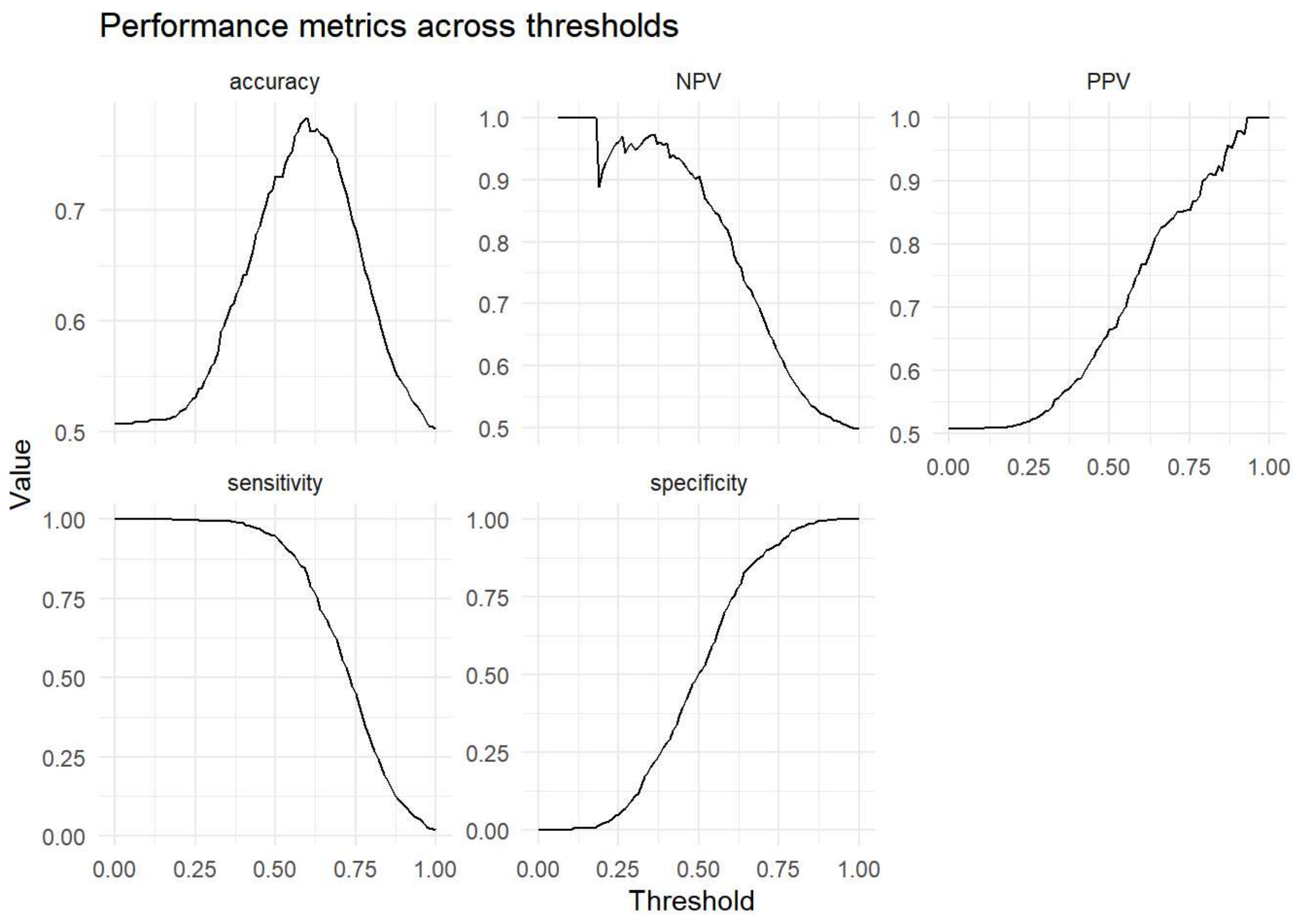
```
##   threshold  TP  FP  TN FN sensitivity specificity       PPV       NPV accuracy
## 1       0.6 417 126 367 90   0.8224852   0.7444219 0.7679558 0.8030635    0.784
##      youden
## 1 0.5669071
```

We plotted sensitivity and specificity versus threshold. The two curves intersected at a threshold of approximately 0.62. We also plotted accuracy, PPV, and NPV across thresholds. Accuracy peaked close to 0.60, which is consistent with the Youden-optimal threshold and reflects the fact that accuracy is maximized when sensitivity and specificity are jointly high in a balanced dataset. PPV increased steadily with higher thresholds. NPV showed a gradual downward trend with a noticeable drop at lower thresholds.

```
metrics_plot <- metrics_all %>%
  select(threshold, sensitivity, specificity) %>%
  pivot_longer(cols = c(sensitivity, specificity),
               names_to = "metric",
               values_to = "value")

ggplot(metrics_plot, aes(x = threshold, y = value, color = metric)) +
  geom_line() +
  labs(x = "Threshold",
       y = "Value",
       color = "Metric",
       title = "Sensitivity and specificity across thresholds") +
  theme_minimal()
```



```
metrics_plot_all <- metrics_all %>%
  select(threshold, sensitivity, specificity, PPV, NPV, accuracy) %>%
  pivot_longer(cols = -threshold,
               names_to = "metric",
               values_to = "value")

ggplot(metrics_plot_all, aes(x = threshold, y = value)) +
  geom_line() +
  facet_wrap(~ metric, scales = "free_y") +
  labs(x = "Threshold",
       y = "Value",
       title = "Performance metrics across thresholds") +
  theme_minimal()
```



Given the severe consequences of missing a lung cancer diagnosis, we then focused on thresholds that ensured high sensitivity (>= 0.90). Within this subset of thresholds, we selected the one with the highest specificity. It could minimize unnecessary false positives while still meeting the clinical requirement for high sensitivity. The threshold identified through this procedure was 0.54, with sensitivity 0.903 and specificity 0.590. Compared with the Youden-optimal threshold, this threshold sacrifices some specificity and overall accuracy but aligns more closely with the clinical objective of prioritizing patient safety and reducing the likelihood of missed cancer cases.

```
# thresholds with sensitivity >= 0.9
high_sens_candidates <- metrics_all %>%
  filter(sensitivity >= 0.9)

# threshold with maximum specificity
high_sens_best <- high_sens_candidates %>%
  arrange(desc(specificity)) %>%
  slice(1)

high_sens_best
```

```
##   threshold  TP  FP  TN FN sensitivity specificity       PPV       NPV accuracy
## 1      0.54 458 202 291 49   0.9033531   0.5902637 0.6939394 0.8558824    0.749
##      youden
## 1 0.4936167
```

We also explored an alternative strategy focused on the negative predictive value (NPV). In a screening context, a high NPV ensures that individuals classified as negative are indeed very unlikely to have the disease, reducing unnecessary follow-up investigations. We therefore identified all thresholds achieving NPV ≥ 0.90 and selected the one with the highest specificity within this set. And this procedure returned a threshold of

0.50, corresponding to sensitivity 0.95 and specificity 0.51. This result differs from the high-sensitivity threshold of 0.54 primarily because NPV depends not only on sensitivity but also on the underlying prevalence of the disease. In our balanced dataset, NPV remains relatively high over a broad range of thresholds, so the constraint NPV >= 0.90 does not push the threshold as low as in real-world low-prevalence screening settings. As a consequence, the NPV-based threshold resembles the default 0.50 threshold and places greater emphasis on avoiding false negatives while tolerating more false positives.

```
# thresholds with NPV >= 0.9
high_NPV_candidates <- metrics_all %>%
  filter(NPV >= 0.9)

# threshold with maximum specificity
high_NPV_best <- high_NPV_candidates %>%
  arrange(desc(specificity)) %>%
  slice(1)

high_NPV_best
```

```
##   threshold  TP  FP  TN FN sensitivity specificity       PPV       NPV accuracy
## 1       0.5 481 243 250 26   0.9487179   0.5070994 0.6643646 0.9057971    0.731
##      youden
## 1 0.4558173
```

Finally, we examined a cost-based approach to threshold selection, which explicitly encodes the idea that false negatives are more harmful than false positives in the context of lung cancer detection. We explored how the optimal threshold changes when different penalties are assigned to false negatives. As the relative cost of missing a cancer case increased, the cost-minimizing threshold shifted progressively downward, favoring thresholds that achieve higher sensitivity at the expense of specificity.

```
# set cost_FP = 1, try different cost_FN
cost_test <- function(cost_FN) {
  metrics_all %>%
    mutate(cost = cost_FN * FN + 1 * FP) %>%
    arrange(cost) %>%
    slice(1) %>%
    mutate(FN_cost = cost_FN)
}

cost_results <- bind_rows(
  cost_test(2),
  cost_test(4),
  cost_test(6),
  cost_test(10),
  cost_test(20)
)

cost_results
```

```
##   threshold  TP  FP  TN FN sensitivity specificity       PPV       NPV accuracy
## 1      0.56 448 174 319 59   0.8836292   0.6470588 0.7202572 0.8439153    0.767
## 2      0.50 481 243 250 26   0.9487179   0.5070994 0.6643646 0.9057971    0.731
## 3      0.44 494 309 184 13   0.9743590   0.3732252 0.6151930 0.9340102    0.678
## 4      0.40 501 353 140  6   0.9881657   0.2839757 0.5866511 0.9589041    0.641
## 5      0.36 504 384 109  3   0.9940828   0.2210953 0.5675676 0.9732143    0.613
##      youden cost FN_cost
## 1 0.5306880  292       2
## 2 0.4558173  347       4
## 3 0.3475841  387       6
## 4 0.2721413  413      10
## 5 0.2151782  444      20
```