Logistic Regression Diagnostics

Fundamental Techniques in Data Science

Utrecht University

Kyle M. Lang

Department of Methodology & Statistics
Utrecht University

Outline

Assumptions of MLR

The assumptions of the linear model can be stated as follows:

1. The model is linear in the parameters.
   - This is OK: $Y = \beta_0 + \beta_1 X + \beta_2 Z + \beta_3 XZ + \beta_4 X^2 + \beta_5 X^3 + \varepsilon$
   - This is not: $Y = \beta_0 X^{\beta_1} + \varepsilon$
2. The predictor matrix is *full rank*.
   - $N > P$
   - No X_p can be a linear combination of other predictors.

Assumptions of MLR

3. The predictors are strictly exogenous.
   - The predictors do not correlated with the errors.
   - $\text{Cov}(\hat{Y}, \varepsilon) = 0$
   - $\text{E}[\varepsilon_n] = 0$
4. The errors have constant, finite variance.
   - $\text{Var}(\varepsilon_n) = \sigma^2 < \infty$
5. The errors are uncorrelated.
   - $\text{Cov}(\varepsilon_i, \varepsilon_j) = 0, \ i \neq j$
6. The errors are normally distributed.
   - $\varepsilon \sim \text{N}(0, \sigma^2)$

Assumptions of MLR

The assumption of *spherical errors* combines Assumptions 4 and 5.

$$\text{Var}(\varepsilon) = \begin{bmatrix} \sigma^2 & 0 & \cdots & 0 \\ 0 & \sigma^2 & \cdots & 0 \\ & & & \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & \sigma^2 \end{bmatrix} = \sigma^2 \mathbf{I}_N$$

We can combine Assumptions 3, 4, 5, and 6 by assuming independent and identically distributed normal errors:

- $\varepsilon \overset{iid}{\sim} \text{N}(0, \sigma^2)$

Example

We'll start by using logistic regression to predict the chances that Titanic passengers survived the sinking based on their age, sex, and ticket class.

```
## Read the data:
titanic <- readRDS(paste0(dataDir, "titanic.rds"))

## Estimate the logistic regression model:
glmFit <- glm(survived ~ age + sex + class,
              data = titanic,
              family = "binomial")
```

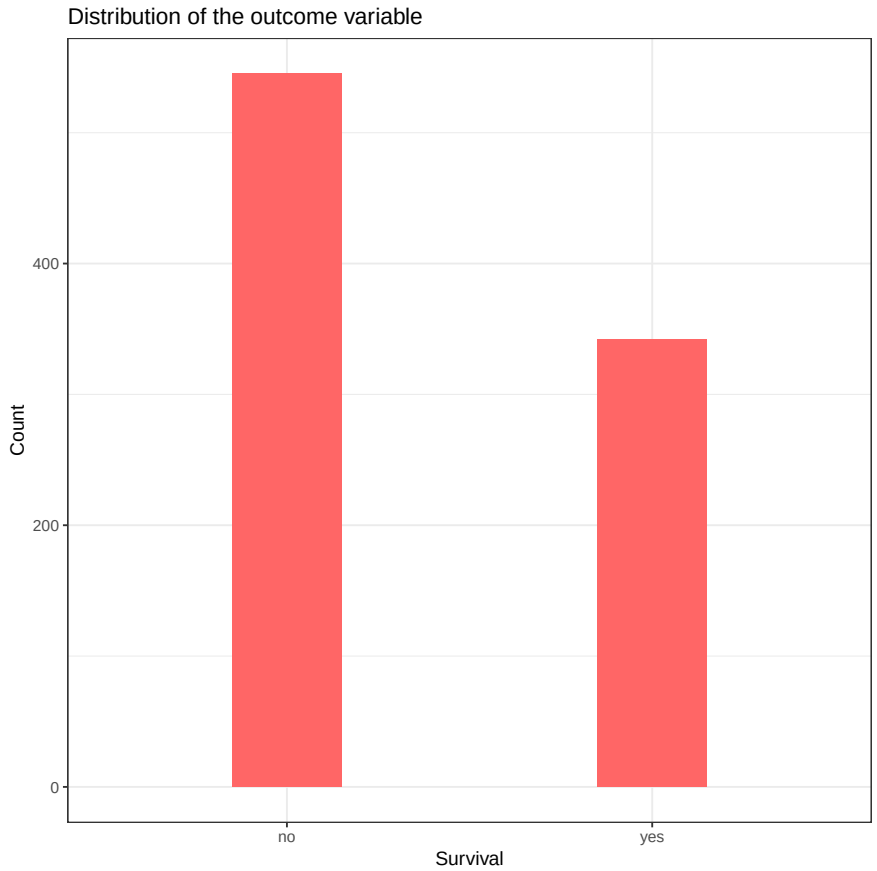
Example

```
partSummary(glmFit, -1)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.63492    0.37045   9.812  < 2e-16
age         -0.03427    0.00716  -4.787 1.69e-06
sexmale     -2.58872    0.18701 -13.843  < 2e-16
class2nd    -1.19911    0.26158  -4.584 4.56e-06
class3rd    -2.45544    0.25322  -9.697  < 2e-16

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1182.77  on 886  degrees of freedom
Residual deviance:  801.59  on 882  degrees of freedom
AIC: 811.59

Number of Fisher Scoring iterations: 5
```

Binary response variable

Logistic regression assumes that the response variable only has two possible outcomes.

For example, “survived” describes the passenger’s survival status where 0 indicates did not survive and 1 indicates survived.



Binary response variable

We can also check the levels of the response variable.

```
titanic$survived %>% unique()

[1] no  yes
Levels: no yes

titanic$survived %>% factor() %>% levels()

[1] "no"  "yes"
```

The assumption is violated when the outcome is

- a multiclass categorical variable. (multinomial logistic regression `mnet::mutinom()`)
- an ordinal categorical variable. (ordered logistic regression `MASS::polr()`)

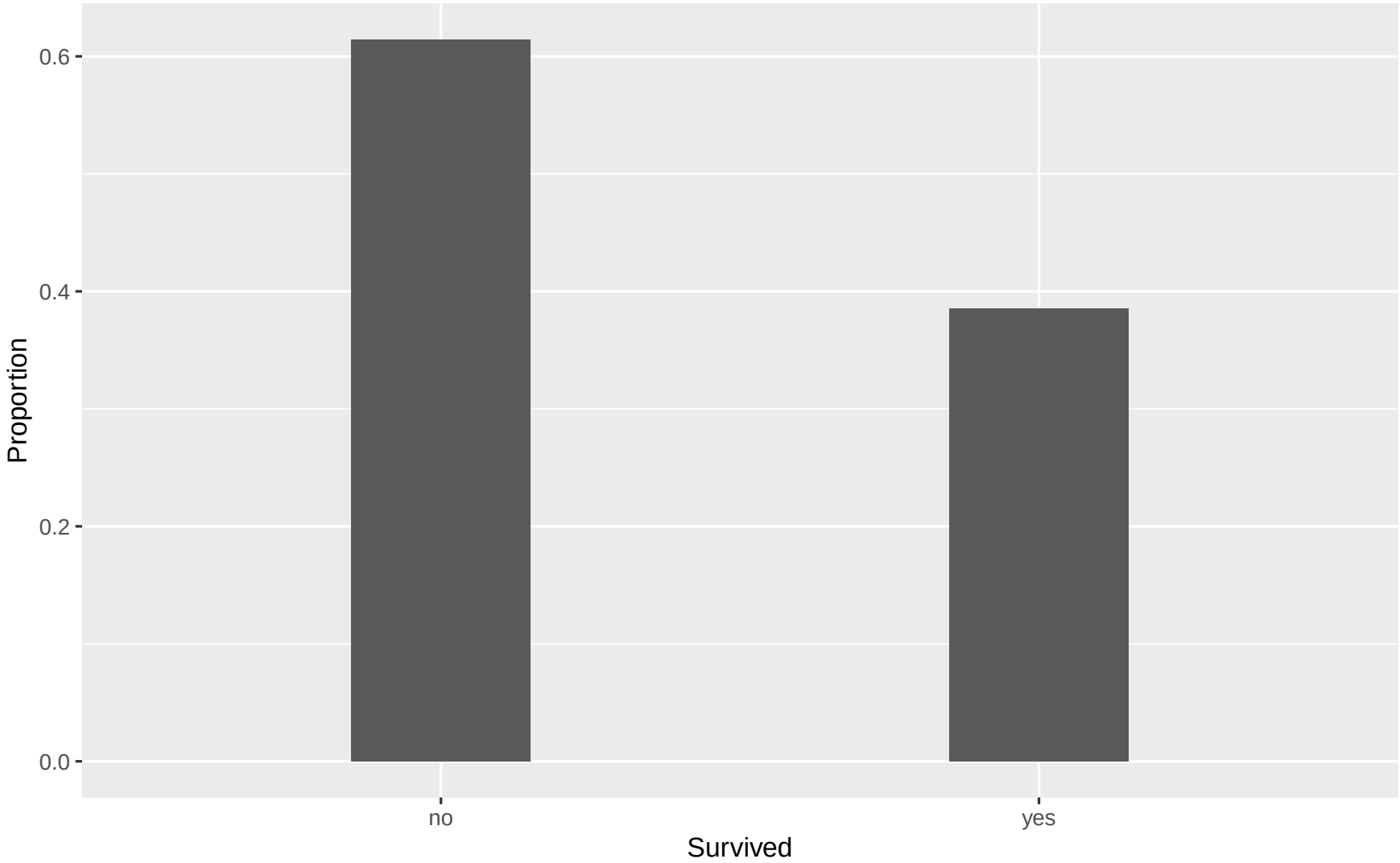
Balanced outomes

The logistic regression may not perform well when there is an imbalance in the classes of the binary response. A possible consequence is the inaccurate classification.

A certain amount of imbalance is normal and can be handled well by the logistic model in most cases. However, we should care about the severe imbalance, for instance, 1000 cases in the majority class and 1 case in the minority class.

```
titanic$survived %>% table() %>% prop.table() %>% round(3)

.
   no   yes
0.614 0.386
```

Balanced outomes



Balanced outomes

Some solutions:

- down-sampling the majority class
- up-sampling the minority class
- adding weights to logistic regression (`weights` argument in `glm()`)

Sufficiently large sample size

Sample size in logistic regression is a complex issue. It depends on:

- the number of predictors
- the sample space of predictors
- the distribution of the binary response variable
- the scientific interests

Some suggestions for the sample size

- 10 cases for each predictor in the model (Agresti, 2018)
- $N = \frac{10*k}{p}$, where
  - k is the number of predictors
  - p is the proportion of the minority class (Peduzzi, Concato, Kemper, Holford, & Feinstein, 1996)
- $N = 100 + 50 * k$, where k is the number of predictors (Bujang, Omar, & Baharum, 2018)

Issue : perfect prediction

Imbalanced outcomes and a small sample size may cause perfect prediction. The `glm()` may show warnings messages:

- glm.fit: algorithm did not converge
- fitted probabilities numerically 0 or 1 occurred

One possible solution is to fit the logistic regression with regularization (`glmnet::glmnet`).

No multicollinearity

The same assumption as in linear regression is that is no multicollinearity among the linear predictors.

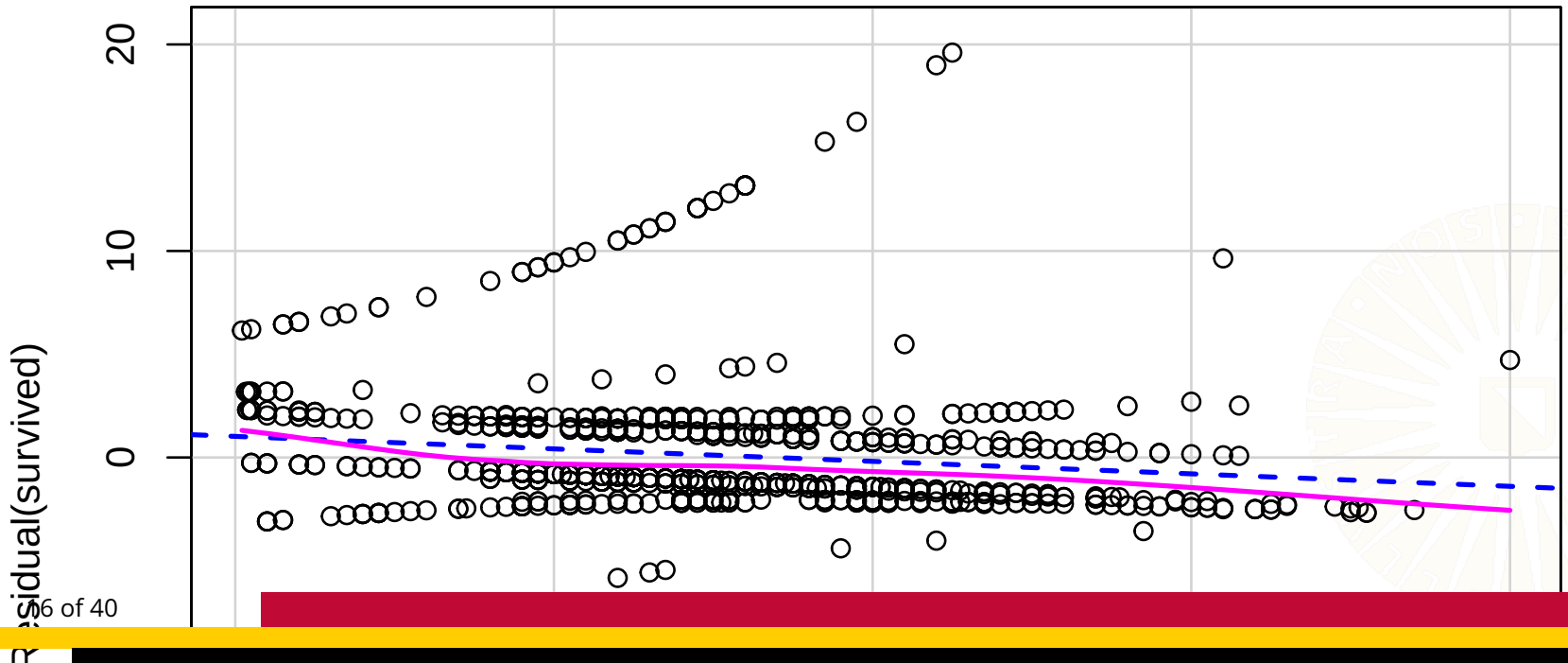
```
VIF(glmFit)

          GVIF Df GVIF^(1/(2*Df))
age   1.351874  1        1.162701
sex   1.085634  1        1.041938
class 1.448873  2        1.097129
```

A VIF value larger than 10 indicates high multicollinearity.

Linearity

Logistic regression assumes a linear relationship between continuous predictors and *the logit of the response variable*.

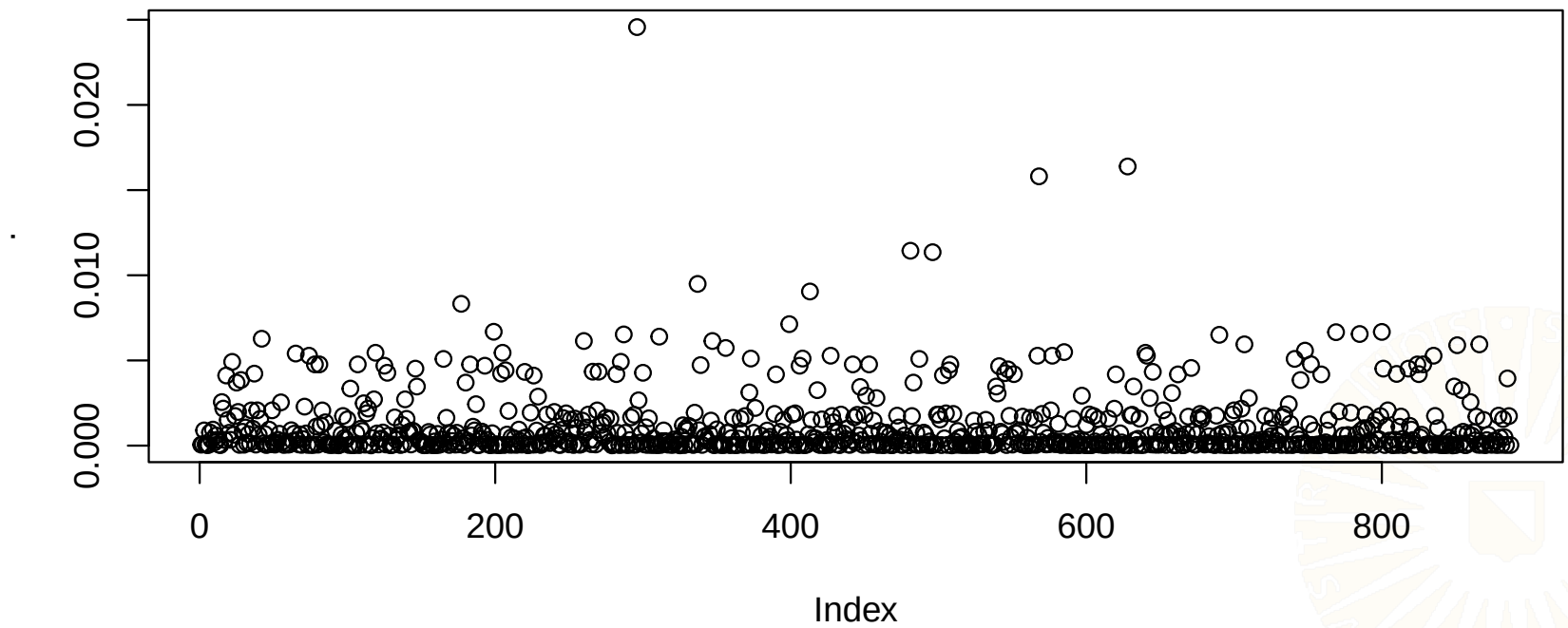
```
library(car)

crPlot(glmFit, "age")
```



No influential values or outliers

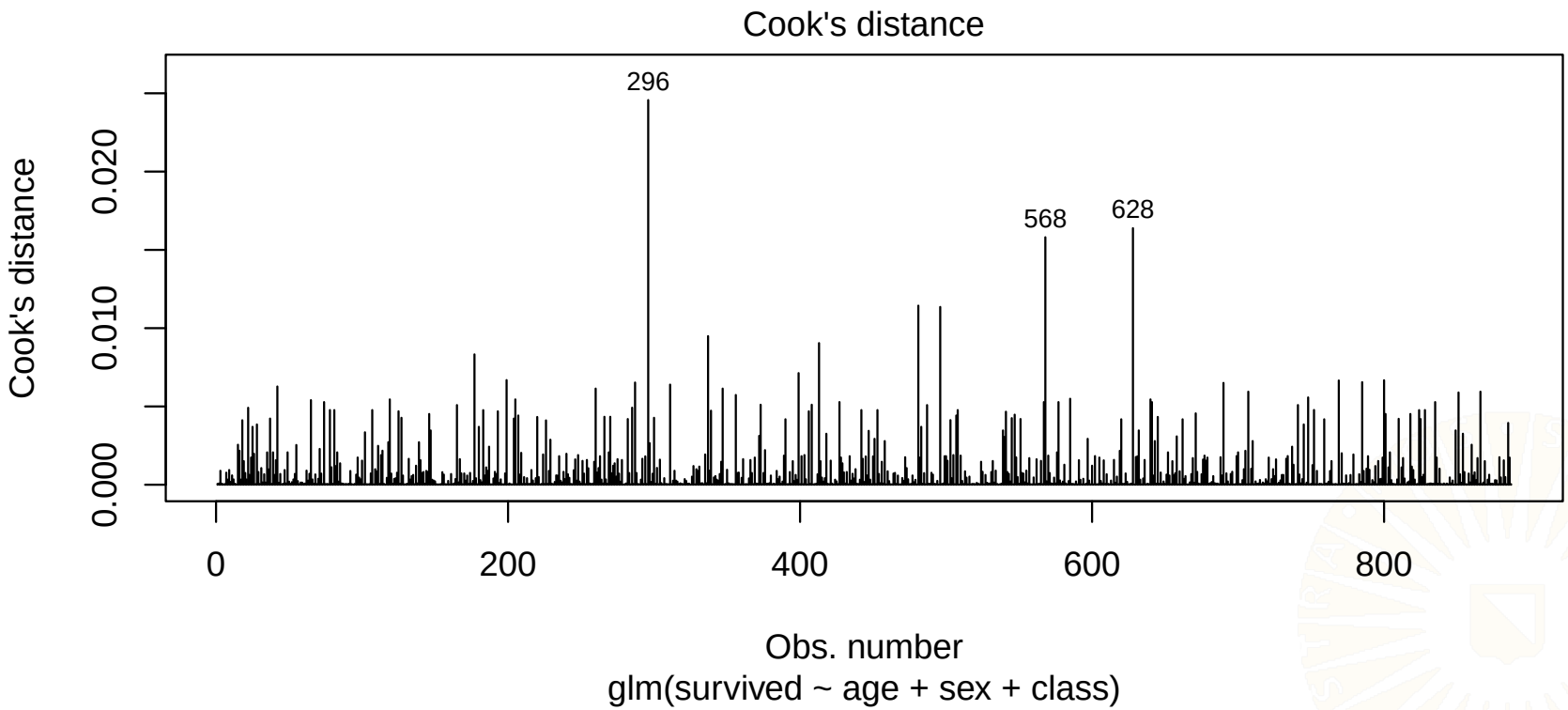
Influential values or outliers can seriously influence the fit of the logistic regression.

```
cooks.distance(glmFit) %>% plot()
```



No influential values or outliers

```
plot(glmFit, 4)
```



Raw Residuals

The most basic residual is the *raw residual*, which is the difference between the observed value and the predicted probability:

$$e_i = y_i - \hat{p}_i$$

```
titanic$r0 <- resid(glmFit, type = "response")

ggplot(titanic, aes(etaHat, r0)) + geom_point() + geom_smooth()

Error in ‘geom_point()‘:
!  Problem while computing aesthetics.
i Error occurred in the 1st layer.
Caused by error:
!  object ’etaHat’ not found
```

Pearson Residuals

The Pearson residual is a scaled version of the raw residual.

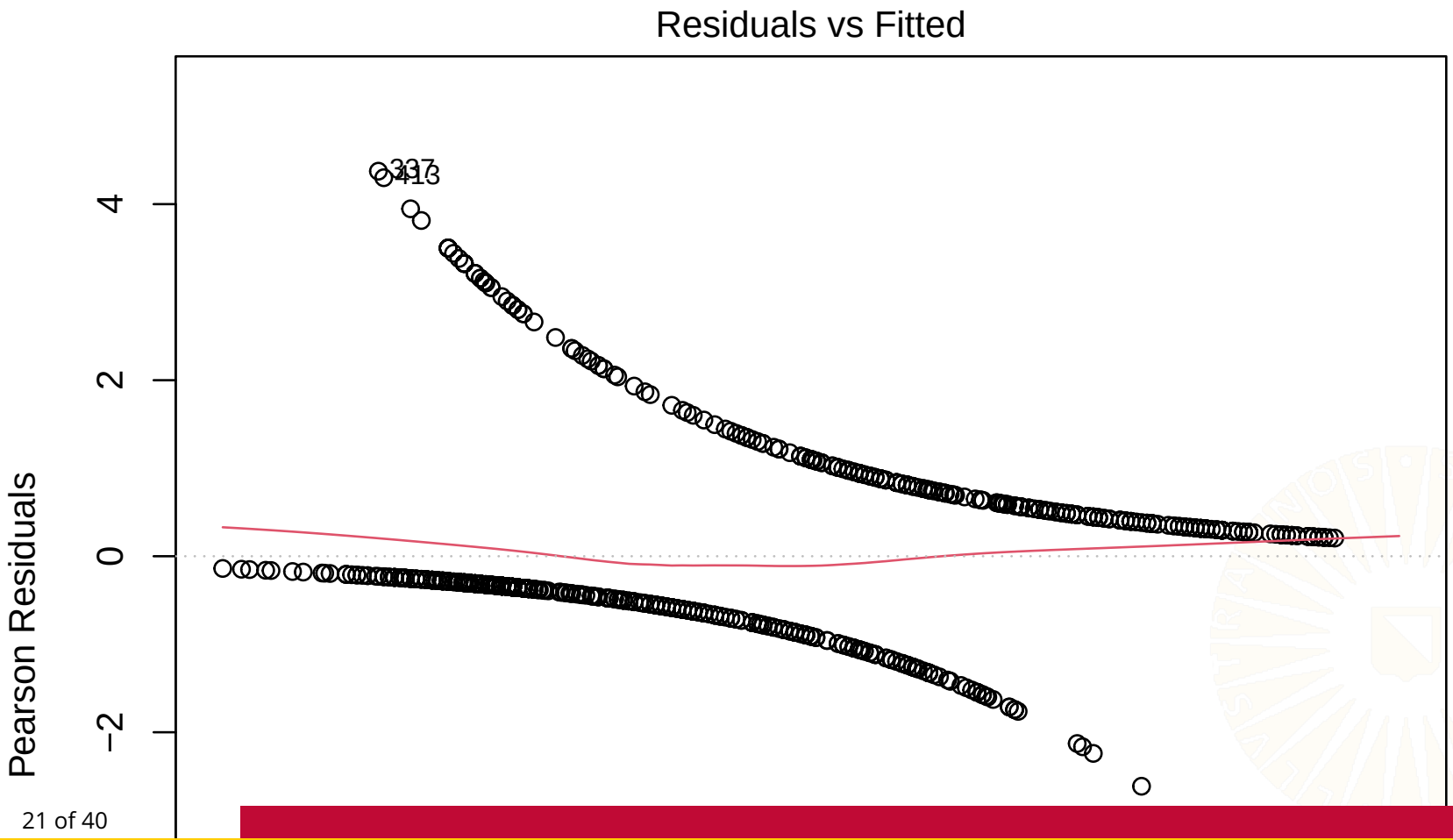
$$r_i = \frac{e_i}{\sqrt{\hat{p}_i(1-\hat{p}_i)}}$$

```
titanic$r1 <- resid(glmFit, type = "pearson")
ggplot(titanic, aes(etaHat, r1)) + geom_point() + geom_smooth()
```

```
Error in ‘geom_point()‘:
!  Problem while computing aesthetics.
i Error occurred in the 1st layer.
Caused by error:
!  object ’etaHat’ not found
```

Pearson Residual

```
plot(glmFit, 1)
```



Deviance Residuals

Deviance residuals can be approximated with a standard normal distribution if the model fits well.

$$d_i = sign(e_i)\sqrt{-2[y_i \ln(\hat{p}_i) + (1 - y_i)ln(1 - \hat{p}_i)]}$$

```
titanic$r2 <- resid(glmFit, type = "deviance")
ggplot(titanic, aes(etaHat, r2)) + geom_point() + geom_smooth()
```

```
Error in ‘geom_point()‘:
!  Problem while computing aesthetics.
i Error occurred in the 1st layer.
Caused by error:
!  object ’etaHat’ not found
```

Residual Deviance

The residual deviance is the sum of squared deviance residuals.

$$D = \sum_{i=1}^{N} d_i^2$$

```
titanic$r2^2 %>% sum()

[1] 801.594

summary(glmFit)$deviance

[1] 801.594
```

Residual Deviance

The residual deviance is used to measure how well the model fits the data. It is similar to the sum of squared errors in linear regression.

```
s    <- summary(glmFit)
dev0 <- s$null.deviance
dev1 <- s$deviance
df0  <- s$df.null
df1  <- s$df.residual

d0 - d1

Error in eval(expr, envir, enclos):  object 'd0' not found

df0 - df1

[1] 4

anova(glmFit, test = "Chisq")

Analysis of Deviance Table

Model: binomial, link: logit

Response: su
```

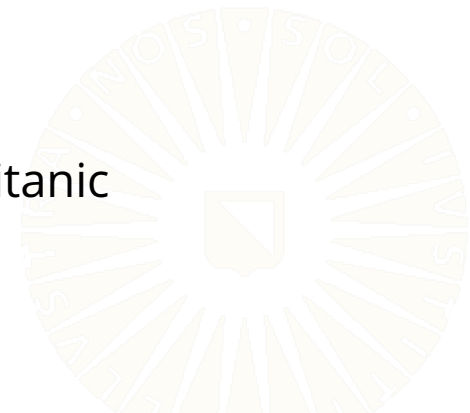
Confusion Matrix

One of the most direct ways to evaluate classification performance is the *confusion matrix*.

```
titanic %<>%
  mutate(etaHat = predict(glmFit, type = "link"),
         piHat = predict(glmFit, type = "response"),
         yHat = as.factor(ifelse(piHat <= 0.5, "no", "yes"))
        )
```

	True	
Predicted	no	yes
no	463	100
yes	82	242

Confusion Matrix of passengers' survival on the Titanic



Confusion Matrix

Each cell in the confusion matrix represents a certain classification outcome.

	True	
Predicted	Died	Survived
1	True Negative	False Positive
2	False Negative	True Positive

Confusion Matrix of passengers' survival on the Titanic

Confusion matrix

In the titanic example,

- **TP**: correctly predict people that survived
- **TN**: correctly predict people that did not survive
- **FP**: predict people survived, when they did not
- **FN**: predict people did not survive, but they did

Confusion Matrix

```
(cMat <- titanic %$% confusionMatrix(data = yHat, reference = survived))

Confusion Matrix and Statistics

          Reference
Prediction  no yes
       no  463 100
       yes  82 242

               Accuracy : 0.7948
                 95% CI : (0.7667, 0.8209)
    No Information Rate : 0.6144
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.5627

 Mcnemar's Test P-Value : 0.2076

            Sensitivity : 0.8495
            Specificity : 0.7076
         Pos Pred Value : 0.8224
         Neg Pred Value : 0.7469
```

Summary of the confusion matrix

Accuracy:

- accuracy = (TP + TN) / (P + N)
- In titanic example, accuracy = 0.79, meaning that 79% are correctly classified.

Error rate:

- error rate = (FP + FN) / (P + N) = 1 - accuracy
- In titanic example, error rate = 0.21, meaning that 21% are not correctly classified.

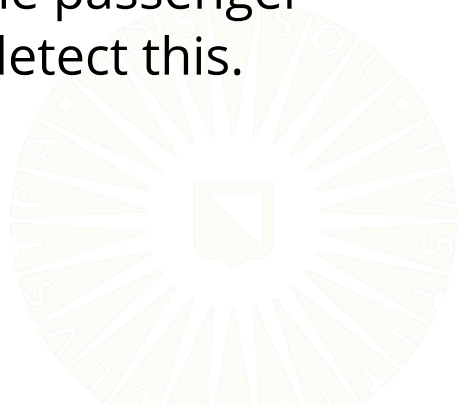
Summary of the confusion matrix

Sensitivity:

- sensitivity = TP / (TP + FN)
- In titanic example, sensitivity = 0.58, meaning that if the passenger did survive, there is a 58% chance the model will detect this.

Specificity:

- specificity = TN / (TN + FP)
- In titanic example, specificity = 0.92, meaning that if the passenger did not survive, there is a 92% chance the model will detect this.



Summary of the confusion matrix

False positive rate:

- false positive rate (FPR) = FP / (TN + FP) = 1 - specificity
- In titanic example, FPR = 0.08, meaning that if a passenger did not survive, there is an 8% chance that the model predicts this passenger as surviving.

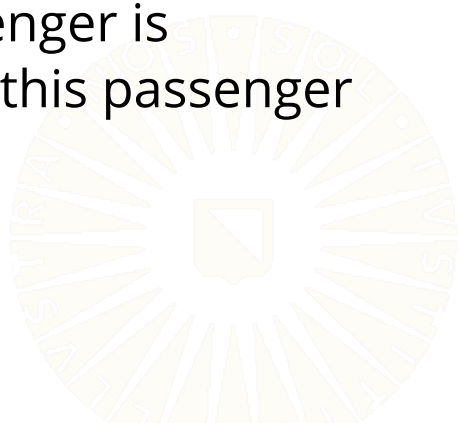
Summary of the confusion matrix

Positive predictive value:

- positive predictive value (PPV) = TP / (TP + FP)
- In titanic example, PPV = 0.83, meaning that if the passenger is predicted as surviving, there is an 83% chance that this passenger indeed survived.

Negative predictive value:

- negative predictive value (NPV) = TN / (TN + FN)
- In titanic example, NPV = 0.78, meaning that if a passenger is predicted as not surviving, there is a 78% chance that this passenger indeed does not survive.



ROC curve

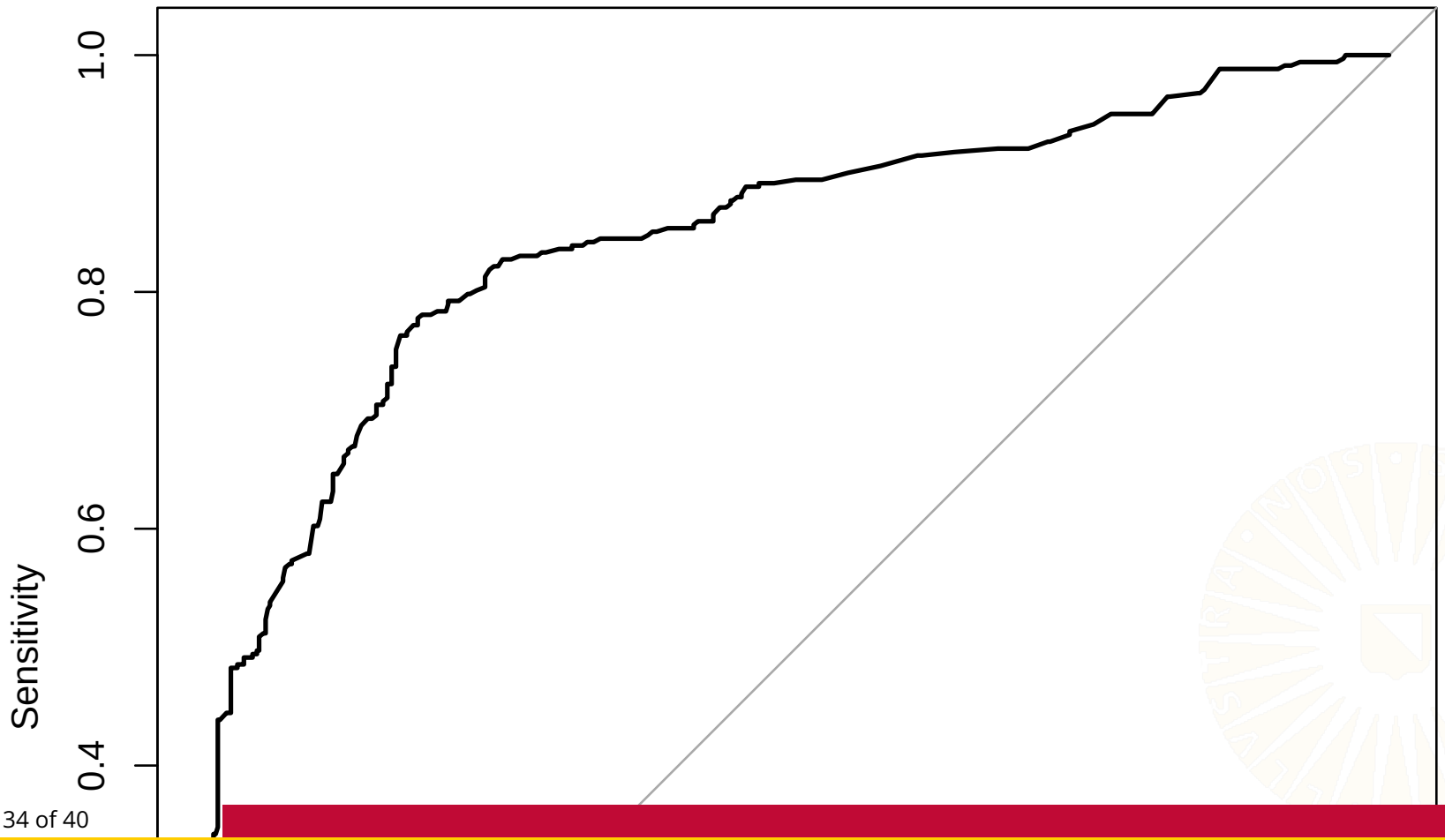
A receiver operating characteristic curve (ROC curve) is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The ROC curve is created by plotting sensitivity against FPR (1 - specificity) at various threshold values.

ROC curve is mainly used for:

- evaluating the classification performance
- selecting discrimination threshold

ROC curve

```
rocData <- titanic %$% roc(response = survived, predictor = piHat)
plot(rocData)
```



ROC curve

The Area Under the ROC Curve (AUC) summarizes the performance of the classification.

```
auc(rocData)

Area under the curve: 0.8483
```

According to Mandrekar (2010):

- AUC value from 0.7-0.8: acceptable
- AUC value from 0.8-0.9: excellent
- AUC value over 0.9: outstanding

Threshold Selection

Sometimes, we do not want to use $\hat{pi} = 0.5$ as the threshold.

```
library(OptimalCutpoints)

ocOut <- optimal.cutpoints(X = "piHat",
                           status = "survived",
                           tag.healthy = "no",
                           data = titanic,
                           method = c("ROC01", "MaxEfficiency")
                           )

summary(ocOut)


Call:
optimal.cutpoints.default(X = "piHat", status = "survived", tag.healthy = "no",
    methods = c("ROC01", "MaxEfficiency"), data = titanic)

Area under the ROC curve (AUC):  0.848 (0.82, 0.876)

CRITERION: ROC01
Number of optimal cutoffs: 1
```

Weight sensitivity or specificity?

Selecting a point with the smallest distance to the point (0, 1) is to maximize $sensitivity^2 + specificity^2$. This optimized function has equal weights to sensitivity and specificity. However, in some scenarios, we care more about sensitivity or specificity.

More Sensitive Measures

Measuring classification performance from a confusion matrix can be problematic.

- Sometimes too coarse.

We can also base our error measure on the residual deviance with the *Cross-Entropy Error*:

$$CEE = -N^{-1} \sum_{n=1}^{N} Y_n \ln(\hat{\pi}_n) + (1 - Y_n) \ln(1 - \hat{\pi}_n)$$

$$CEE = -N^{-1} \sum_{n=1}^{N} \sum_{l=1}^{L} \mathbf{I}(Y_n = l) \ln(\hat{\pi}_{nl})$$

- The CEE is sensitive to classification confidence.
- Stronger predictions are more heavily weighted.

Benefits of CEE

The misclassification rate is a naïvely appealing option.

- The proportion of cases assigned to the wrong group

Consider two perfect classifiers:

1. $P(\hat{Y}_n = 1|Y_n = 1) = 0.90, P(\hat{Y}_n = 1|Y_n = 0) = 0.10, n = 1, 2, \ldots, N$
2. $P(\hat{Y}_n = 1|Y_n = 1) = 0.55, P(\hat{Y}_n = 1|Y_n = 0) = 0.45, n = 1, 2, \ldots, N$

Both of these classifiers will have the same misclassification rate.

- Neither model ever makes an incorrect group assignment.

The first model will have a lower CEE.

- The classifications are made with higher confidence.
- $CEE_1 = 0.105, CEE_2 = 0.598$

References

Agresti, A. (2018). *An introduction to categorical data analysis*. Hoboken, NJ: John Wiley & Sons.

Bujang, M. A., Omar, E. D., & Baharum, N. A. (2018). A review on sample size determination for cronbach's alpha test: a simple guide for researchers. *The Malaysian Journal of Medical Sciences*, *25*(6), 85.

Mandrekar, J. N. (2010). Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology*, *5*(9), 1315–1316.

Peduzzi, P., Concato, J., Kemper, E., Holford, T. R., & Feinstein, A. R. (1996). A simulation study of the number of events per variable in logistic regression analysis. *Journal of Clinical Epidemiology*, *49*(12), 1373–1379.