

# Logistic Regression Diagnostics

## Fundamental Techniques in Data Science



**Utrecht  
University**

Kyle M. Lang

Department of Methodology & Statistics  
Utrecht University

# Outline

---

## Assumptions & Diagnostics

- Statistical Assumptions

- Residuals

- Diagnostics

- Computational Considerations

- Influential Cases

## Classification Performance

- Confusion Matrix

- ROC Curve

- Alternative Performance Measures



# Recap: Model Definition

---

We define the logistic regression model as:

$$Y \sim \text{Bin}(\pi, 1)$$
$$\text{logit}(\pi) = \beta_0 + \sum_{p=1}^P \beta_p X_p$$

We denote the untransformed linear predictor as  $\eta$ :

$$\eta = \beta_0 + \sum_{p=1}^P \beta_p X_p$$

The logit link represents the natural log of the odds of success:

$$\text{logit}(\pi) = \ln\left(\frac{\pi}{1 - \pi}\right)$$



# Recap: Inverse Link Function

---

In logistic regression, the inverse link function,  $g^{-1}(\cdot)$ , is the *logistic function*:

$$\text{logistic}(X) = \frac{e^X}{1 + e^X}$$

So, we convert  $\eta$  to  $\pi$  by:

$$\pi = \frac{e^\eta}{1 + e^\eta} = \frac{\exp\left(\beta_0 + \sum_{p=1}^P \beta_p X_p\right)}{1 + \exp\left(\beta_0 + \sum_{p=1}^P \beta_p X_p\right)}$$



# ASSUMPTIONS & DIAGNOSTICS



# Assumptions of Logistic Regression

---

The first two assumptions of logistic regression are shared with linear regression.

1. The model is linear in the parameters.

- This is OK:  $\text{logit}(\pi) = \beta_0 + \beta_1 X + \beta_2 Z + \beta_3 XZ + \beta_4 X^2 + \beta_5 X^3$
- This is not:  $\text{logit}(\pi) = \beta_0 X^{\beta_1}$

2. The predictor matrix is *full rank*.

- $N > P$
- No  $X_p$  can be a linear combination of other predictors.



# Assumptions of Logistic Regression

---

The distributional assumptions of logistic regression are not framed in terms of residuals.

- Linear regression

$$Y \sim N(\hat{Y}, \hat{\sigma}^2)$$

$$Y = \hat{Y} + \hat{\varepsilon}$$

$$\varepsilon \sim N(0, \sigma^2)$$

- Logistic regression

$$Y \sim \text{Bin}(\hat{\pi}, 1)$$



# Assumptions of Logistic Regression

---

The variance of the binomial distribution is a function of it's mean.

- Linear regression

$$\bar{Y} = \hat{Y}, \text{ var}(Y) = \hat{\sigma}^2$$

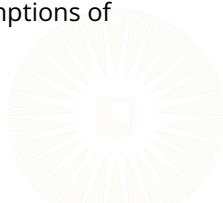
- Logistic regression

$$\bar{Y} = \hat{\pi}, \text{ var}(Y) = \hat{\pi} (1 - \hat{\pi})$$

So, we consider the entire outcome distribution in logistic regression.

- We can succinctly summarize the distributional assumptions of logistic regression as:

$$Y_i \stackrel{iid}{\sim} \text{Bin}(\hat{\pi}_i, 1)$$





# Assumptions of Logistic Regression

---

We end up with three assumptions where the third assumption fills the role played by all residual-related assumptions in linear regression.

1. The model is linear in the parameters.
2. The predictor matrix is *full rank*.
3. The outcome is independently and identically binomially distributed.

$$Y_n \stackrel{iid}{\sim} \text{Bin}(\hat{\pi}_n, 1)$$
$$\hat{\pi}_n = \text{logistic} \left( \hat{\beta}_0 + \sum_{p=1}^P \hat{\beta}_p X_{np} \right)$$



# Example

---

To demonstrate these ideas, we'll fit a logistic regression model that predicts the chances of Titanic passengers surviving based on their age, sex, and ticket price

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

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

## Save the linear predictor estimates:
titanic$etaHat <- predict(glmFit, type = "link")
```

# Example

---

```
partSummary(glmFit, -1)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	0.837621	0.215121	3.894	9.87e-05
age	-0.007404	0.006040	-1.226	0.22
sexmale	-2.392422	0.171288	-13.967	< 2e-16
fare	0.011586	0.002338	4.955	7.23e-07

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1182.8 on 886 degrees of freedom  
Residual deviance: 881.4 on 883 degrees of freedom  
AIC: 889.4

Number of Fisher Scoring iterations: 5

# Raw Residuals

---

In logistic regression the outcome is binary,  $Y \in \{0, 1\}$ , but the parameter that we're trying to model is continuous,  $\pi \in (0, 1)$ .

- Due to this mismatch in measurement levels, we don't have a natural definition of a "residual" in logistic regression.
- We have a few potential operationalizations.

The most basic residual is the *raw residual*,  $e_n$ .

- The difference between the observed outcome value and the predicted probability.

$$e_n = Y_n - \hat{\pi}_n$$

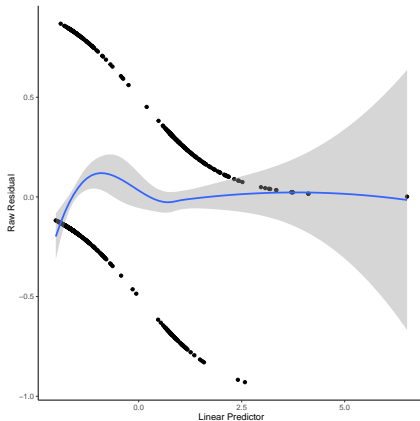


# Raw Residuals

```
library(ggplot2)

## Calculate the raw residuals:
titanic$e <-
  resid(glmFit, type = "response")

## Plot raw residuals vs. fitted
## linear predictor values:
ggplot(titanic, aes(etaHat, e)) +
  geom_point() +
  geom_smooth() +
  theme_classic() +
  xlab("Linear Predictor") +
  ylab("Raw Residual")
```

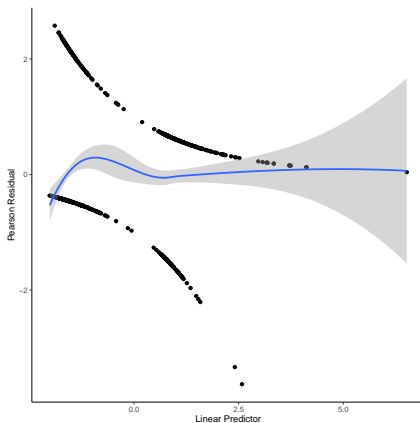


# Pearson Residuals

Pearson residuals,  $r_n$ , are scaled raw residuals.

$$r_n = \frac{e_n}{\sqrt{\hat{\pi}_n(1 - \hat{\pi}_n)}}$$

```
## Calculate the Pearson residuals:  
titanic$r <-  
  resid(glmFit, type = "pearson")
```



# Deviance Residuals

---

*Deviance residuals*,  $d_n$ , are derived directly from the objective function used to estimate the model.

$$d_n = \text{sign}(e_n) \sqrt{-2 [Y_n \ln (\hat{\pi}_n) + (1 - Y_n) \ln (1 - \hat{\pi}_n)]}$$

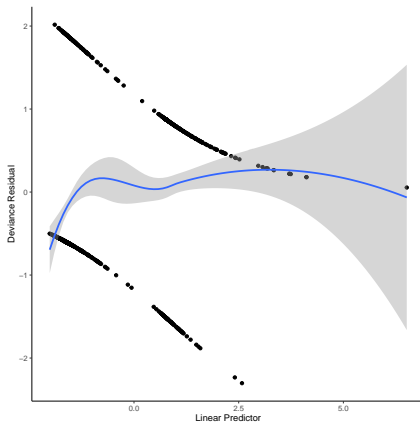
The *residual deviance*,  $D$ , is the sum of squared deviance residuals.

$$D = \sum_{n=1}^N d_n^2$$



# Deviance Residuals

```
## Calculate the deviance residuals:  
titanic$d <-  
  resid(glmFit, type = "deviance")  
  
## Calculate the residual deviance:  
titanic$d^2 %>% sum()  
  
[1] 881.4048  
  
summary(glmFit)$deviance  
  
[1] 881.4048
```





# Residual Deviance

The residual deviance quantifies how well the model fits the data.

```
## Estimate a null model:  
nullFit <- glm(survived ~ 1, family = binomial, data = titanic)
```

```
## Test the fit of our example model:  
anova(nullFit, glmFit, test = "Chisq")
```

Analysis of Deviance Table

Model 1: survived ~ 1

Model 2: survived ~ age + sex + fare

	Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
1	886	1182.8			
2	883	881.4	3	301.37	< 2.2e-16 ***

---

Signif. codes:

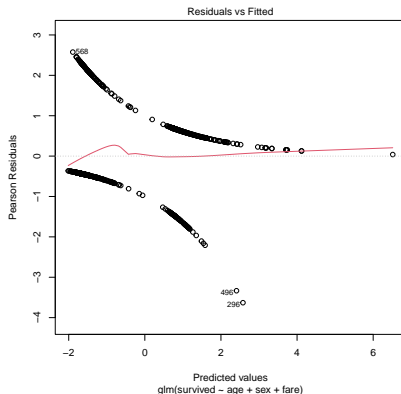
0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# A1: Linearity

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

Assumption 1 implies a linear relation between continuous predictors and the *logit of the success probability*.

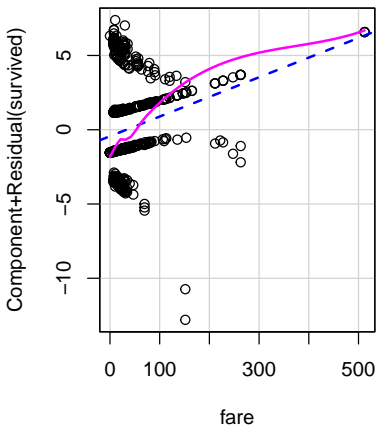
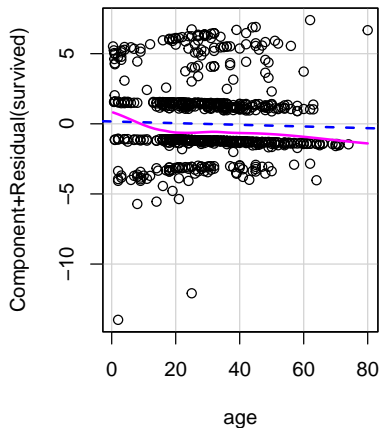
- We can basically evaluate the linearity assumption using the same methods we applied with linear regression.
- $\hat{Y} \rightarrow \hat{\eta} = \text{logit}(\hat{\pi})$



# A1: Linearity

```
car::crPlots(glmFit, terms = ~ age + fare)
```

Component + Residual Plots



## A2: Predictor Matrix Rank

---

Assumption 2 implies two conditions:

1.  $P > N$
2. No severe (multi)collinearity among the predictors

We can quantify multicollinearity with the *variance inflation factor* (VIF).

```
car::vif(glmFit)

      age      sex      fare
1.031829 1.007699 1.026373
```

VIF > 10 indicates severe multicollinearity.

## A3: IID Binomial

---

Assumption 3 implies several conditions.

1. The outcome,  $Y$ , is binary.
2. The linear predictor,  $\eta$ , can explain all the systematic trends in  $\pi$ .
  - No residual clustering after accounting for  $\mathbf{X}$ .
  - No important variables omitted from  $\mathbf{X}$ .

We can easily check the first condition with summary statistics.

```
levels(titanic$survived)

[1] "no"  "yes"

table(titanic$survived)

  no  yes
545 342
```

# Alternative Modeling Schemes

---

If we have a non-binary, categorical outcome, we can use a different type of model.

- Multiclass nominal variables: Multinomial logistic regression
  - `nnet::multinom()`
- Ordinal variables: Proportional odds logistic regression
  - `MASS::polr()`
- Counts: Poisson regression
  - `glm()` with `family = "poisson"`

The binomial distribution (and logistic regression) is also appropriate for modeling the proportion of successes in  $N$  trials.



## A3: Clustering

---

We can check for residual clustering by calculating the ICC using deviance residuals.

```
## Check for residual dependence induced by 'class':  
ICC::ICCbare(x = titanic$class, y = resid(glmFit, type = "deviance"))  
[1] 0.1054665
```



# Computational Considerations

---

In addition to the preceding statistical assumptions, we must satisfy three computational requirements that were not necessary in linear regression.

1. The sample size is large enough to support the necessary numerical estimation.
2. The outcome classes are sufficiently balanced.
3. There is no perfect prediction.





# Sufficient Sample Size

---

Logistic regression models are estimated with numerical methods, so we need larger samples than we would for linear regression models.

- The sample size requirements increase with model complexity.

Some suggested rules of thumb:

- 10 cases for each predictor (Agresti, 2018)
- $N = 10P/\pi_0$  (Peduzzi, Concato, Kemper, Holford, & Feinstein, 1996)
  - $P$ : Number of predictors
  - $\pi_0$ : Proportion of the minority class
- $N = 100 + 50P$  (Bujang, Omar, & Baharum, 2018)



# Balanced Outcomes

---

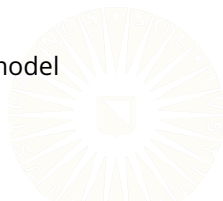
The logistic regression may not perform well when the outcome classes are severely imbalanced.

```
with(titanic, table(survived) / length(survived))
```

```
survived
      no      yes
0.6144307 0.3855693
```

We have a few possible solutions for problematic imbalance:

- Down-sampling the majority class
- Up-sampling the minority class
- Use weights when estimating the logistic regression model
  - `weights` argument in `glm()`



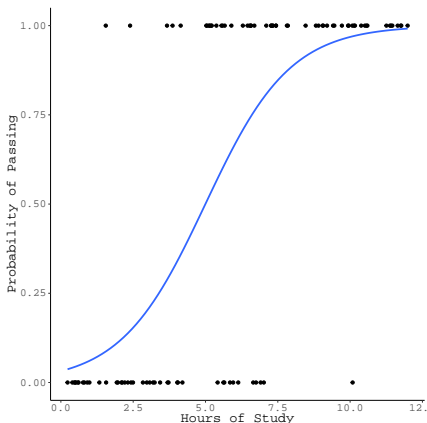
# Perfect Prediction

We don't actually want to perfectly predict class membership.

- The model cannot estimate with perfectly separable classes.

Model regularization (e.g., ridge or LASSO penalty) may help.

- `glmnet::glmnet()`



# Influential Cases

---

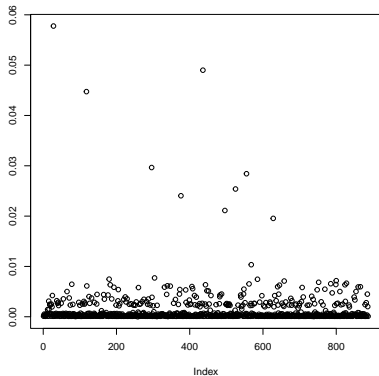
As with linear regression, we need to deal with any overly influential cases.

- We can use the linear predictor values to calculate Cook's Distances.
- Any cases that exerts undue influence on the linear predictor will have the same effect of the predicted success probabilities.

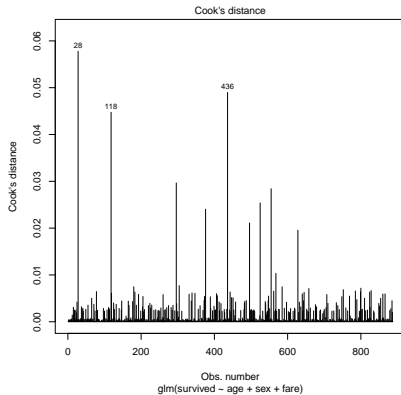


# Influential Cases

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



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



# CLASSIfICATION PERFORMANCE



# Confusion Matrix

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

```
## Add predictions to the dataset:
titanic %<>%
  mutate(piHat = predict(glmFit, type = "response"),
         yHat = as.factor(ifelse(piHat <= 0.5, "no", "yes")))
)
```

Predicted	True	
	no	yes
no	458	106
yes	87	236

Confusion Matrix of Predicted Survival



# Confusion Matrix

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

		True	
Predicted	Died	Died	Survived
Died	True Negative		False Negative
Survived	False Positive		True Positive

Confusion Matrix of Predicted Survival

- **TP:** Correctly predict survival
- **TN:** Correctly predict death
- **FP:** Predict survival for dead people
- **FN:** Predict death for survivors





# Confusion Matrix

---

```
library(caret)

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

cMat$table
```

	Reference	
Prediction	no	yes
no	458	106
yes	87	236

```
cMat$overall
```

Accuracy	Kappa	AccuracyLower	AccuracyUpper
7.824126e-01	5.359709e-01	7.537802e-01	8.091549e-01
AccuracyNull	AccuracyPValue	McNemarPValue	
6.144307e-01	7.471405e-27	1.950898e-01	

# Confusion Matrix

---

cMat\$byClass

Sensitivity	Specificity
0.8403670	0.6900585
Pos Pred Value	Neg Pred Value
0.8120567	0.7306502
Precision	Recall
0.8120567	0.8403670
F1	Prevalence
0.8259693	0.6144307
Detection Rate	Detection Prevalence
0.5163472	0.6358512
Balanced Accuracy	
0.7652127	

# Summaries of the Confusion Matrix

---

$$\text{Accuracy} = (TP + TN) / (P + N)$$

- In our example, Accuracy = 0.78
- 78% are correctly classified

$$\text{Error Rate} = (FP + FN) / (P + N) = 1 - \text{Accuracy}$$

- In our example, Error Rate = 0.22
- 22% are incorrectly classified

$$\text{Sensitivity} = TP / (TP + FN)$$

- In our example, Sensitivity = 0.84
- 84% of survivors are correctly classified

$$\text{Specificity} = TN / (TN + FP)$$

- In our example, Specificity = 0.69
- 69% of deaths are correctly classified



# Summaries of the Confusion Matrix

---

*False Positive Rate (FPR) =  $FP / (TN + FP) = 1 - \text{Specificity}$*

- In our example,  $FPR = 0.31$
- 31% of deaths are incorrectly classified as survivors

*Positive Predictive Value (PPV) =  $TP / (TP + FP)$*

- In our example,  $PPV = 0.81$
- There is an 81% chance that a passenger classified as a survivor was classified correctly

*Negative Predictive Value (NPV) =  $TN / (TN + FN)$*

- In our example,  $NPV = 0.73$
- There is a 73% chance that a passenger classified as dying was classified correctly

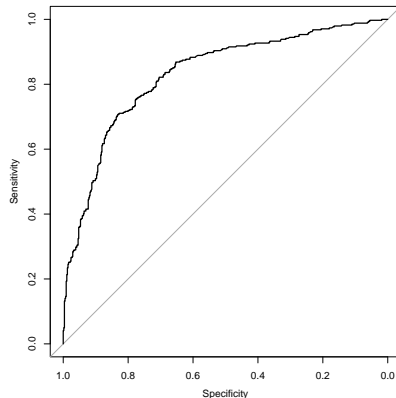


# ROC Curve

```
rocData <- titanic %$%  
  pROC::roc(survived, piHat)  
plot(rocData)
```

A receiver operating characteristic (ROC) curve illustrates the diagnostic ability of a binary classifier for all possible values of the classification threshold.

- The ROC curve plots sensitivity against specificity at different threshold values.



# ROC Curve

---

The *area under the ROC curve* (AUC) is a one-number summary of the potential performance of the classifier.

- The AUC does not depend on the classification threshold.

```
pROC::auc(rocData)
```

```
Area under the curve: 0.8298
```

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

---

We can use numerical methods to estimate an optimal threshold value.

```
library(OptimalCutpoints)

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

# Threshold Selection

---

```
partSummary(ocOut, -1)
```

Area under the ROC curve (AUC): 0.83 (0.802, 0.858)

CRITERION: ROC01

Number of optimal cutoffs: 1

	Estimate
cutoff	0.2360978
Se	0.7543860
Sp	0.7761468
PPV	0.6789474
NPV	0.8343195
DLR.Positive	3.3700029
DLR.Negative	0.3164531
FP	122.0000000
FN	84.0000000
Optimal criterion	0.1104365



# Alternative Performance 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*:

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

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

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.

