

# Logistic regression

## Fundamental Techniques in Data Science



**Utrecht  
University**

Mingyang Cai

Department of Methodology & Statistics  
Utrecht University

# Outline

---

Recap  
Model assumptions  
Residuals  
Confusion matrix



# Generalized linear models (GLMs)

---

- The mean  $E(Y|X) = g^{-1}(\eta) = g^{-1}(\beta_0 + \sum_{p=1}^P \beta_p X_p)$ 
  - A linear function  $\eta$
  - The link function  $g$
- The variance  $V(g^{-1}(\eta))$



# Logistic regression

---

The link function of logistic regression is logit link:

$$\ln\left(\frac{p(Y = 1)}{p(Y = 0)}\right) = \beta_0 + \sum_{p=1}^P \beta_p X_p$$

The interpretation of the coefficients is on log odds units.



# Titanic data

---

```
titanic <- read.csv(file = "data/titanic.csv", header = TRUE,  
                    stringsAsFactors = TRUE)
```

The titanic data describes the survival status of passengers on the Titanic. For the heuristic, We only include four variables.

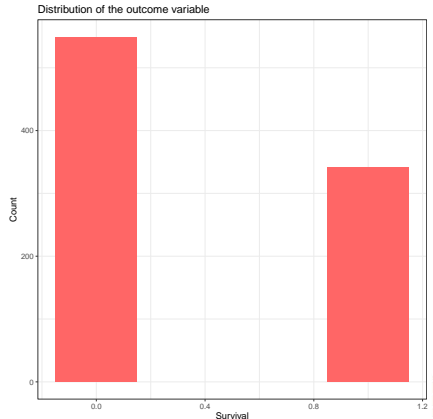
```
titanic %>% head()
```

	Survived	Pclass	Age	Sex
1	0	3	22.00000	male
2	1	1	38.00000	female
3	1	3	26.00000	female
4	1	1	35.00000	female
5	0	3	35.00000	male
6	0	3	29.69912	male

# 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] 0 1

titanic$Survived %>% factor() %>% levels()
[1] "0" "1"
```

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 outcomes

---

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.

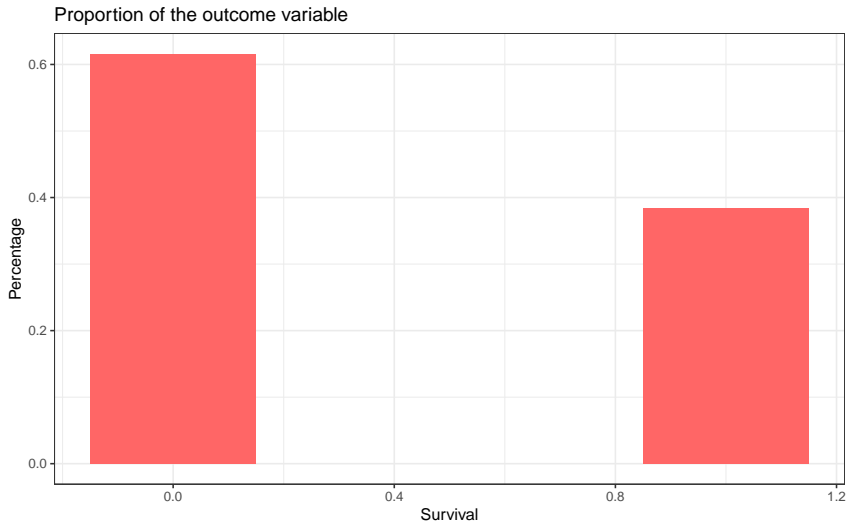
```
titanic$Survived %>% factor() %>% table() %>%  
  prop.table() %>% round(digits = 3)
```

```
.  
  0      1  
0.616 0.384
```



# Balanced outcomes

---



# Balanced outcomes

---

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

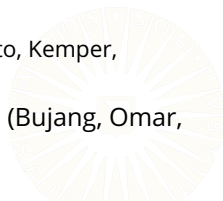
---

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.

```
glm(Survived ~ Age + Sex*Pclass, family = binomial,  
     data = titanic) %>%  
  VIF()
```

Age	Sex	Pclass	Sex:Pclass
1.187231	22.400248	7.749858	20.200886

A VIF value larger than 10 indicates high multicollinearity.



# Linearity

---

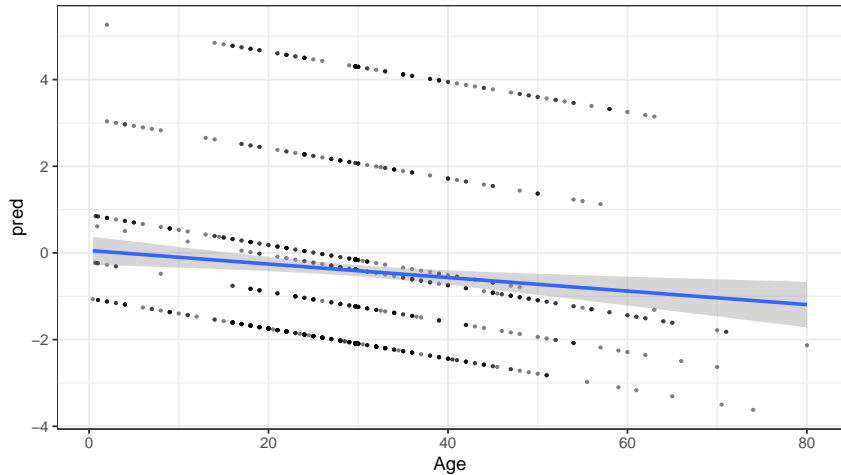
Logistic regression assumes a linear relationship between continuous predictors and *the logit of the response variable*.

```
pred <- glm(Survived ~ Age + Sex*Pclass, family = binomial,
            data = titanic) %>%
  predict(., type = "link")

ggplot(data = data.frame(Age = titanic$Age, pred = pred),
       aes(Age, pred)) +
  geom_point(size = 0.5, alpha = 0.5) +
  geom_smooth(method = "lowess") + theme_bw()
```



# Linearity



# Linearity

---

Some solutions for the violation of linearity:

- transform predictors (log transformation)
- add interaction terms or higher-order terms

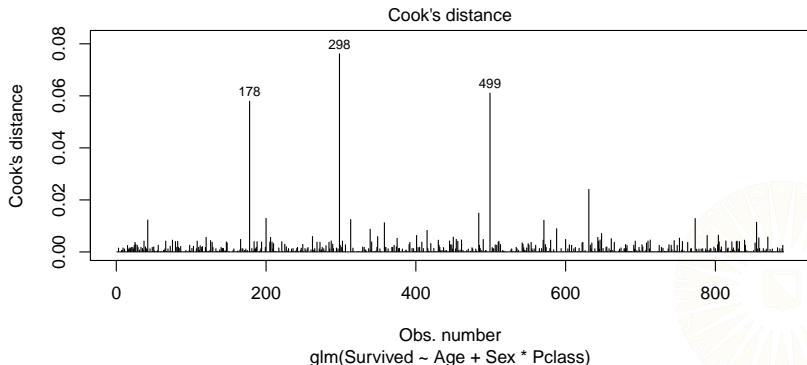




# No influential values or outliers

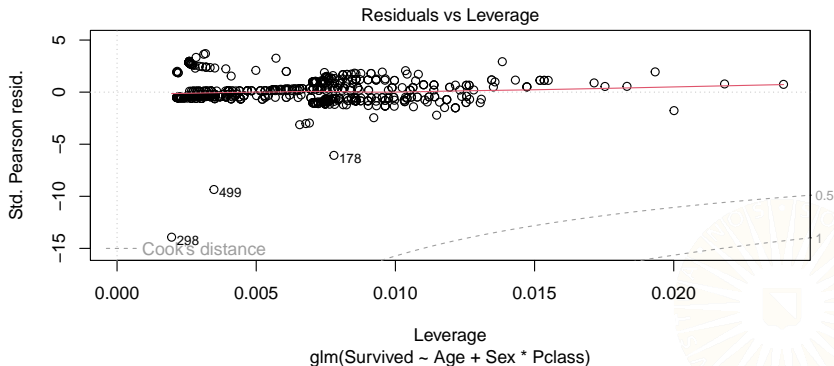
Influential values or outliers can seriously influence the fit of the logistic regression.

```
glm(Survived ~ Age + Sex*Pclass, family = binomial,  
     data = titanic) %>% plot(., which = 4)
```



# No influential values or outliers

```
glm(Survived ~ Age + Sex*Pclass, family = binomial,  
    data = titanic) %>% plot(., which = 5)
```



# No influential values or outliers

---

Some solutions for influential values or outliers:

- remove them
- keep them but mention them in the result
- robust logistic regression ( `robust::glmrob` )



# No influential values or outliers

```
glmRob(Survived ~ Age + Sex + Pclass, family = binomial(),
       data = titanic, method = "cubif") %>%
  summary() %>% .$coefficients
```

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	4.76878301	0.452599082	10.536440	5.867834e-26
Age	-0.03433009	0.007412619	-4.631304	3.633694e-06
Sexmale	-2.60924207	0.187095480	-13.946045	3.325391e-44
Pclass	-1.17625580	0.119412375	-9.850368	6.829363e-23

```
glm(Survived ~ Age + Sex + Pclass, family = binomial, data = titanic) %>%
  summary() %>% .$coefficients
```

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	4.73195642	0.449819406	10.519680	7.011095e-26
Age	-0.03342722	0.007347635	-4.549385	5.380296e-06
Sexmale	-2.61196394	0.186608818	-13.997002	1.625866e-44
Pclass	-1.16846287	0.118940571	-9.823922	8.881931e-23

# Raw residual

---

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

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

```
glm(Survived ~ Age + Sex*Pclass, family = binomial,  
     data = titanic) %>% residuals(., type = "response") %>%  
  head()
```

	1	2	3	4	5
	-0.14000292	0.01770869	0.50655145	0.01598547	-0.09393664
	6				
	-0.11080960				

# Pearson residual

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

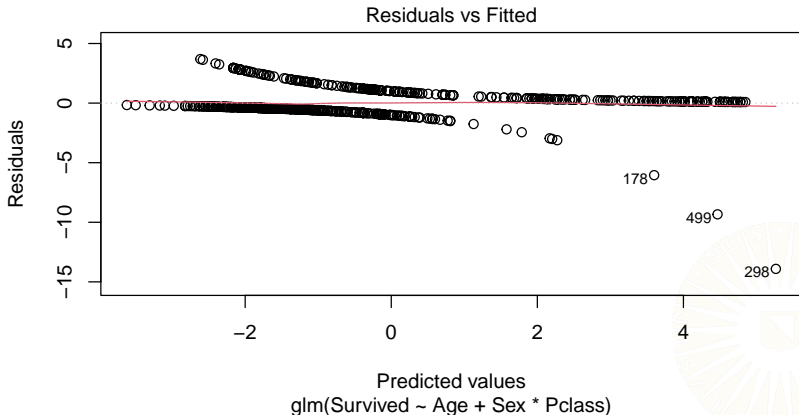
$$r_i = \frac{e_i}{\sqrt{\hat{p}_i(1 - \hat{p}_i)}}$$

```
glm(Survived ~ Age + Sex*Pclass, family = binomial,  
     data = titanic) %>% residuals(., type = "pearson") %>%  
  head()
```

	1	2	3	4	5
	-0.4034782	0.1342682	1.0131899	0.1274565	-0.3219869
	6				
	-0.3530135				

# Pearson residual

```
glm(Survived ~ Age + Sex*Pclass, family = binomial,  
    data = titanic) %>% plot(., which = 1)
```



# Deviance residual

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

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

```
glm(Survived ~ Age + Sex*Pclass, family = binomial,  
    data = titanic) %T>%  
{residuals(., type = "deviance") %>%  
  head(., n = 6) %>%  
  print()} %>%  
summary() %>% .$deviance.resid %>%  
head(., n = 6)
```

1	2	3	4	5
-0.5492291	0.1890363	1.1885594	0.1795250	-0.4441757
6				
-0.4846522				

1	2	3	4	5
-0.5492291	0.1890363	1.1885594	0.1795250	-0.4441757
6				
-0.4846522				



# Residual deviance

---

The residual deviance is the sum of squared deviance residuals.

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

```
glm(Survived ~ Age + Sex*Pclass, family = binomial,  
     data = titanic) %>%  
  summary() %>% .$deviance  
[1] 781.1453
```



# 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.

```
result <- glm(Survived ~ Age + Sex*Pclass, family = binomial,  
              data = titanic) %>% summary()  
1 - pchisq(result$null.deviance - result $deviance,  
           result$df.null - result$df.residual)  
  
[1] 0
```

null :  $\beta_1 = \beta_2 = \dots = \beta_p = 0$

alt : at least one of the  $\beta_i$  is different from 0

Generally, if the value is less than 0.05, the logistic regression is overall significant.

# Confusion matrix

---

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

Predicted	True	
	Not survived	Survived
Not survived	507	144
Survived	42	198

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

---

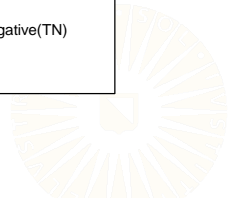
- **true positive(TP)**: A test result that correctly indicates the presence of a condition.
- **true negative(TN)**: A test result that correctly indicates the absence of a condition.
- **false positive(FP)**: A test result which wrongly indicates that a particular condition is present.
- **false negative(FN)**: A test result which wrongly indicates that a particular condition is absent.



# Confusion matrix

---

		Predicted condition	
		Positive(PP)	Negative(PN)
Actual condition	Positive(P)	True positive(TP)	False negative(FN)
	Negative(N)	Flase positive(FP)	True negative(TN)



# Confusion matrix

```
Confusion.matrix<- glm(Survived ~ Age + Sex*Pclass, family = binomial,  
  data = titanic) %>% predict(., type = "response") %>%  
  {ifelse(. > 0.5, 1, 0)} %>% as.factor() %>%  
  confusionMatrix(., reference = as.factor(titanic$Survived),  
    positive = "1")
```

```
Confusion.matrix$table
```

	Reference	
Prediction	0	1
0	507	144
1	42	198

```
Confusion.matrix$overall
```

Accuracy	Kappa	AccuracyLower	AccuracyUpper
7.912458e-01	5.323785e-01	7.630562e-01	8.174942e-01
AccuracyNull	AccuracyPValue	McNemarPValue	
6.161616e-01	2.391263e-29	1.304808e-13	

# Confusion matrix

---

Confusion.matrix\$byClass

Sensitivity	Specificity
0.5789474	0.9234973
Pos Pred Value	Neg Pred Value
0.8250000	0.7788018
Precision	Recall
0.8250000	0.5789474
F1	Prevalence
0.6804124	0.3838384
Detection Rate	Detection Prevalence
0.2222222	0.2693603
Balanced Accuracy	
0.7512223	





# Summary of the confusion matrix

---

## *Accuracy:*

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

## *Error rate:*

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



# Summary of the confusion matrix

---

## *Sensitivity:*

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

## *Specificity:*

- $\text{specificity} = \text{TN} / (\text{TN} + \text{FP})$
- 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) =  $FP / (TN + FP) = 1 - \text{specificity}$
- 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

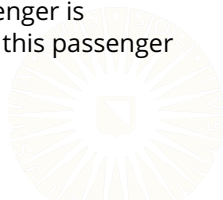
---

## *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 ( $1 - \text{specificity}$ ) at various threshold values.

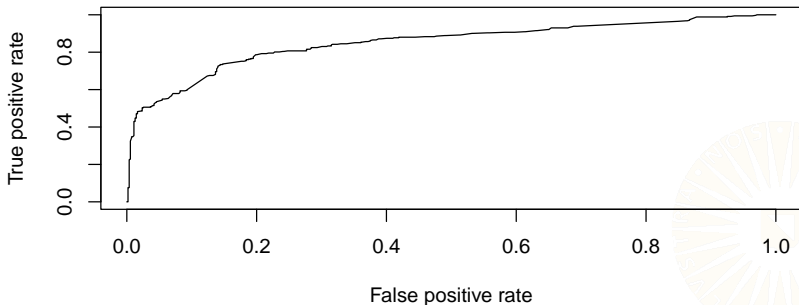
ROC curve is mainly used for:

- evaluating the classification performance
- selecting discrimination threshold



# ROC curve

```
glm(Survived ~ Age + Sex*Pclass, family = binomial,  
    data = titanic) %>% predict(., type = "response") %>%  
  ROCR::prediction(., as.factor(titanic$Survived)) %>%  
  ROCR::performance(., "tpr", "fpr") %>%  
  plot()
```



# Sensitivity versus specificity trade-off

---

Since sensitivity has a positive correlation with the false positive rate, there is a trade-off between sensitivity and specificity.



# ROC curve

---

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

```
glm(Survived ~ Age + Sex*Pclass, family = binomial,  
    data = titanic) %>% predict(., type = "response") %>%  
  pROC::roc(as.factor(titanic$Survived), .) %>%  
  auc()
```

Area under the curve: 0.8497

- AUC value from 0.7-0.8: acceptable
- AUC value from 0.8-0.9: excellent
- AUC value over 0.9: outstanding (Mandrekar, 2010)





# Threshold selection

Sometimes, we do not want to use 0.5 as the threshold.

```
glm(Survived ~ Age + Sex*Pclass, family = binomial,  
    data = titanic) %>% predict(., type = "response") %>%  
mutate(titanic, pred = .) %>%  
OptimalCutpoints::optimal.cutpoints(X = pred ~ Survived,  
                                     tag.healthy = 0,  
                                     control = control.cutpoints(),  
                                     methods = "ROC01",  
                                     data = ., ci.fit = FALSE,  
                                     conf.level = 0.95,  
                                     trace = FALSE) %>%  
  .$ROC01 %>% .$Global %>% .$optimal.cutoff %>% .$cutoff  
  
[1] 0.3530982
```

This threshold minimizes the distance between the selected point on the ROC plot and point (0, 1).

# Weight sensitivity or specificity?

---

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



# Weight sensitivity or specificity?

---

- *When sensitivity is more important*
  - Predict whether a patient has a specific disease.
- *When specificity is more important*
  - Predict whether a person has committed a crime.



# References

---

- Agresti, A. (2018). *An introduction to categorical data analysis*. 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: MJMS*, 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.

