

# Logistic Regression Diagnostics

## Fundamental Techniques in Data Science



**Utrecht  
University**

Kyle M. Lang

Department of Methodology & Statistics  
Utrecht University

# Outline

---

Statistical Assumptions

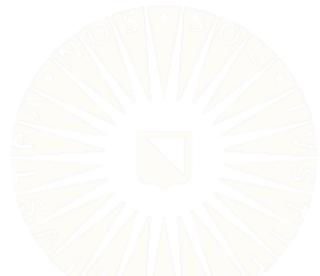
Diagnostics

Residuals

Checking Assumptions

Computational Considerations

Influential Cases

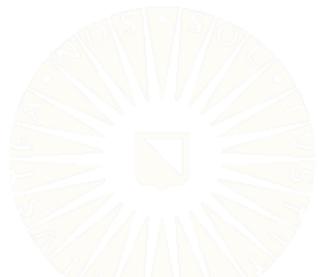


# Assumptions of Logistic Regression

---

The first two assumptions are shared with linear regression.

1. The model is linear in the parameters.
  - This is OK:  $\text{logit}(\pi) = \beta_0 + \beta_1X + \beta_2Z + \beta_3XZ + \beta_4X^2 + \beta_5X^3$
  - This is not:  $\text{logit}(\pi) = \beta_0X^{\beta_1}$
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

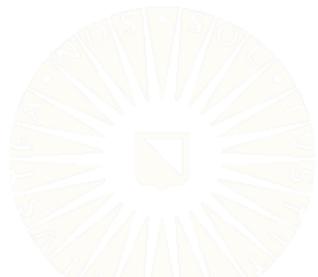
$$Y \sim N(\hat{Y}, \hat{\sigma}^2)$$

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

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

- Logistic regression

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



# Assumptions of Logistic Regression

---

The variance of the binomial distribution is a function of its mean.

- Linear regression

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

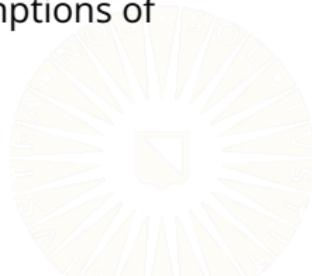
- Logistic regression

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

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

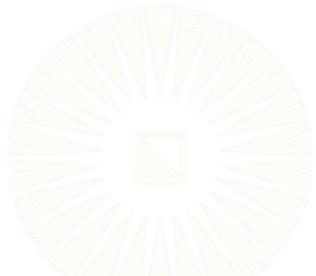
---

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 <- titanic0 <- readRDS(here::here("data", "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)

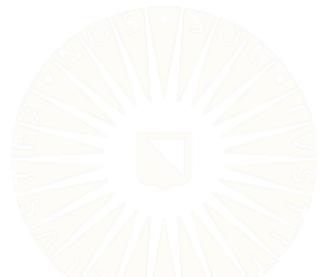
Null deviance: 1182.8 on 886 degrees of freedom

Residual deviance: 881.4 on 883 degrees of freedom

AIC: 889.4

Number of Fisher Scoring iterations: 5

# DIAGNOSTICS



# Raw Residuals

---

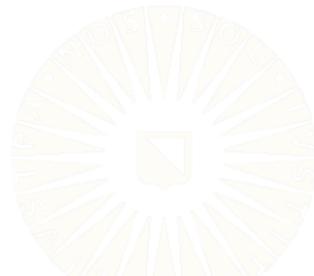
In logistic regression the outcome is binary,  $Y \in \{0, 1\}$ , but the parameter that we're trying to model is continuous,  $\pi \in (0, 1)$ .

- 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.

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

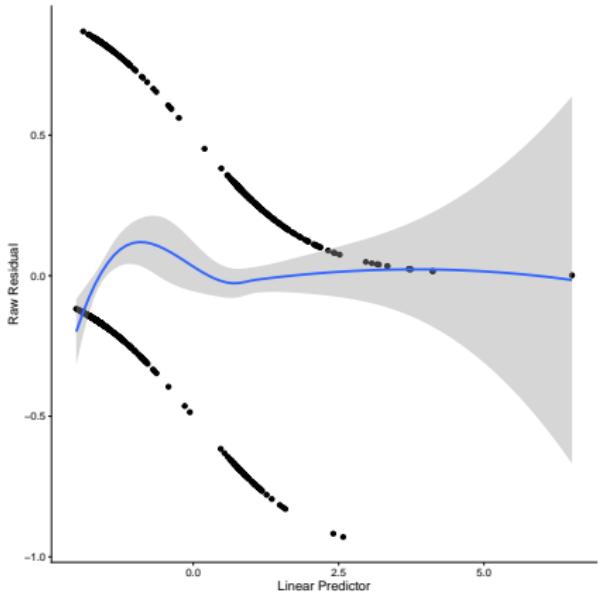


# Raw Residuals

```
library(ggplot)

## 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")
```

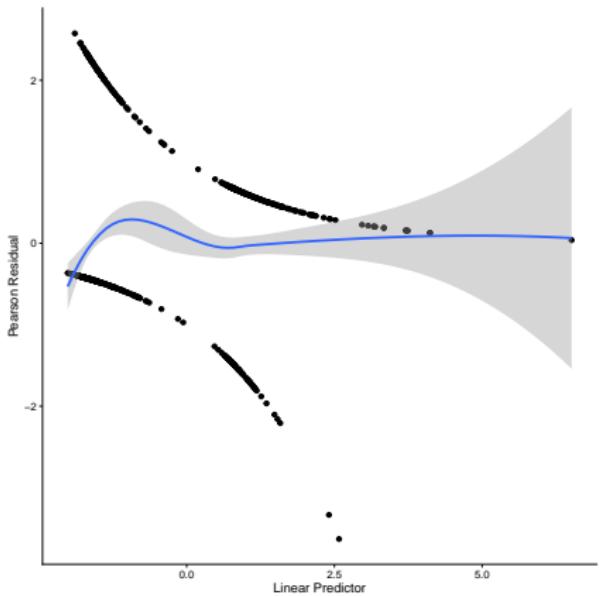


# Pearson Residuals

*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

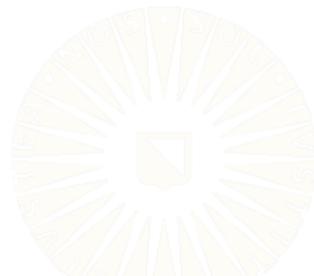
---

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

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

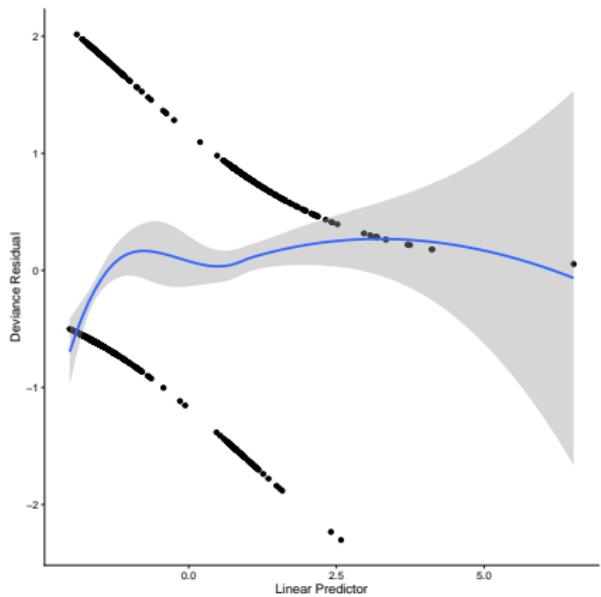
The *residual deviance,  $D$* , is the sum of squared deviance residuals.

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



# 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.

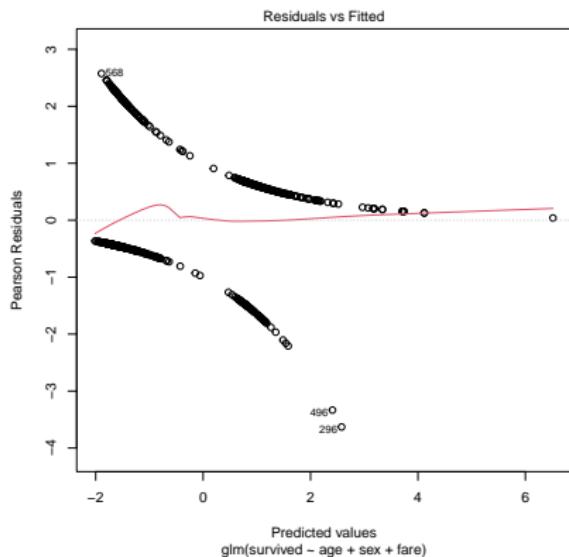
```
## Estimate a null model:  
nullFit <- glm(survived ~ 1, family = binomial, data = titanic)  
  
## Test the fit of our example model:  
anova(nullFit, glmFit, test = "Chisq")  
  
Analysis of Deviance Table  
  
Model 1: survived ~ 1  
Model 2: survived ~ age + sex + fare  
  Resid. Df Resid. Dev 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)
```

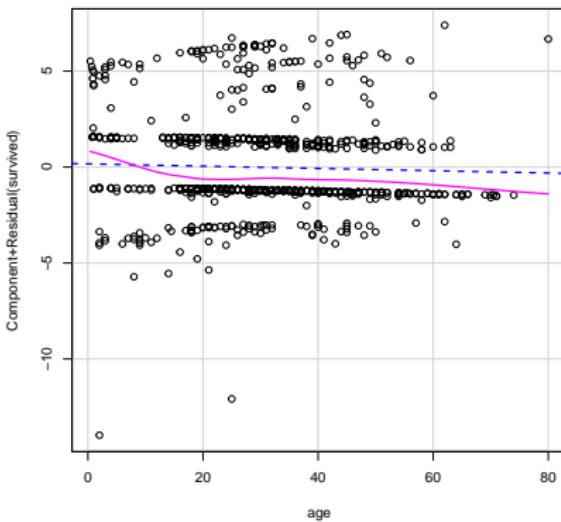
Assumption 1 implies a linear relation between continuous predictors and the *logit of the success probability*.

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

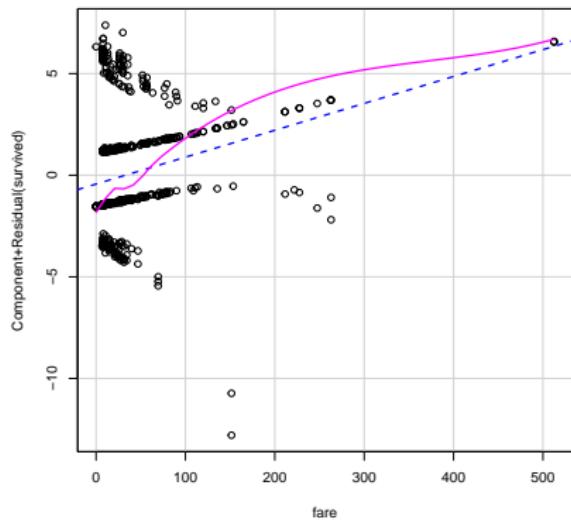


# A1: Linearity

```
car::crPlot(glmFit, "age")
```



```
car::crPlot(glmFit, "fare")
```



## 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  $\pi$ .
  - o No residual clustering after accounting for  $\mathbf{X}$ .
  - o 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
```

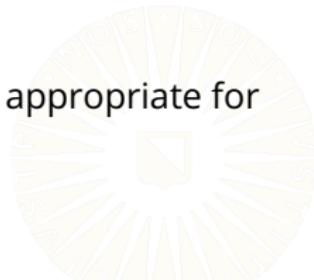
# Alternative Modeling Schemes

---

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.

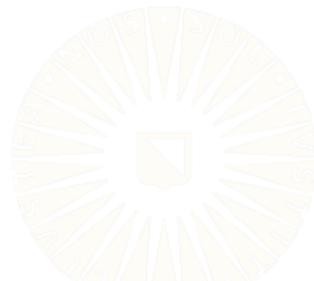


## A3: Clustering

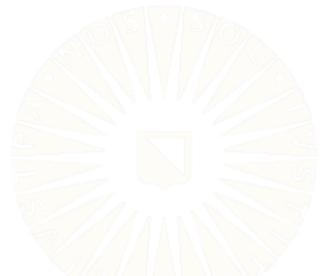
---

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



# Computational Considerations

---

We must also satisfy three computational requirements that were not necessary in linear regression.

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



# Sufficient Sample Size

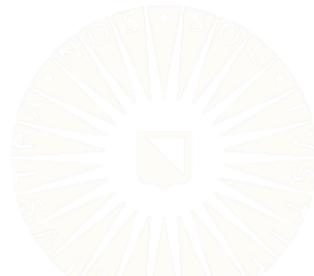
---

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

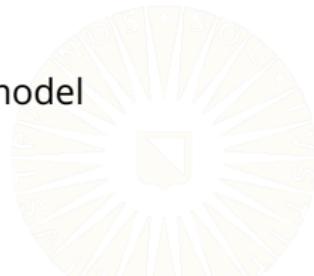
---

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:

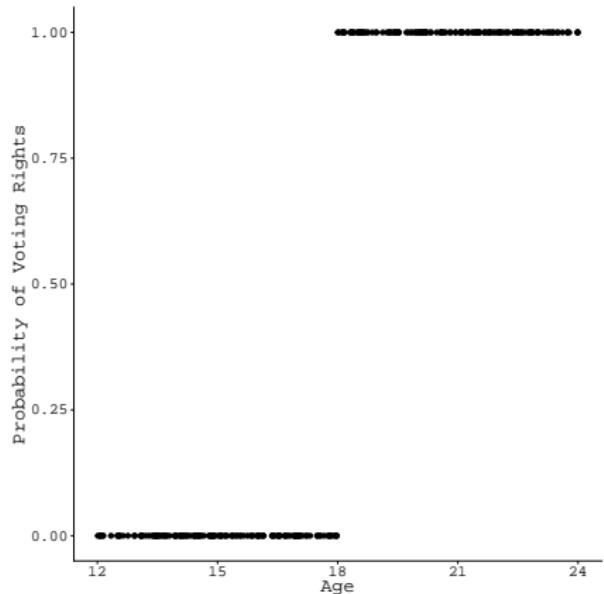
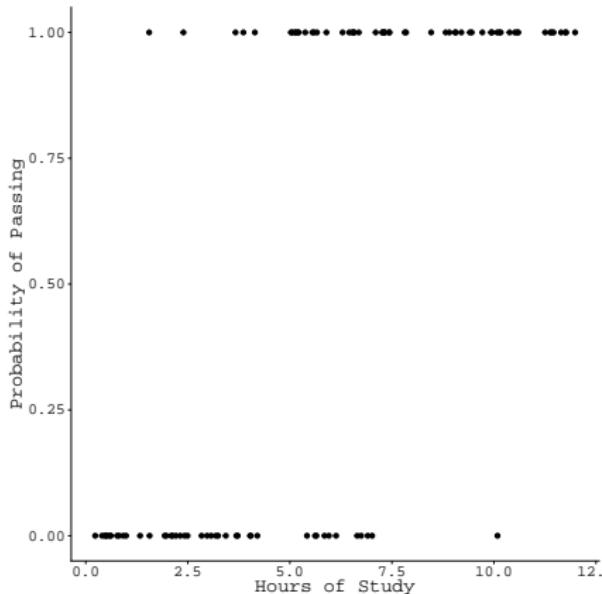
- Down-sampling the majority class
- Up-sampling the minority class
- Use weights when estimating the logistic regression model
  - `weights` argument in `glm()`



# Perfect Prediction

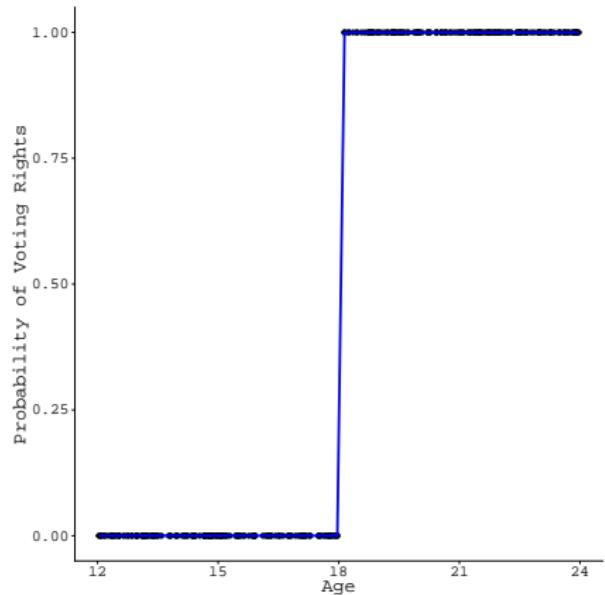
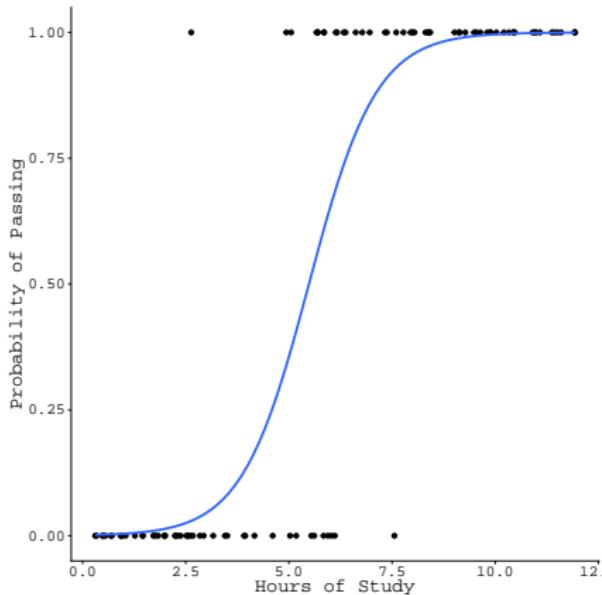
---

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



# Perfect Prediction

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



# Perfect Prediction

The model won't estimate correctly with perfectly separable classes.

```
glm(vote ~ age, family = "binomial") |> summary()
```

Call:

```
glm(formula = vote ~ age, family = "binomial")
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-3366.2	268678.4	-0.013	0.99
age	186.5	14885.4	0.013	0.99

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 4.1503e+02 on 299 degrees of freedom

Residual deviance: 4.0906e-07 on 298 degrees of freedom

AIC: 4

Number of Fisher Scoring iterations: 25

# INFLUENTIAL CASES

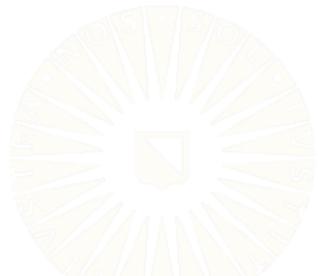


# Influential Cases

---

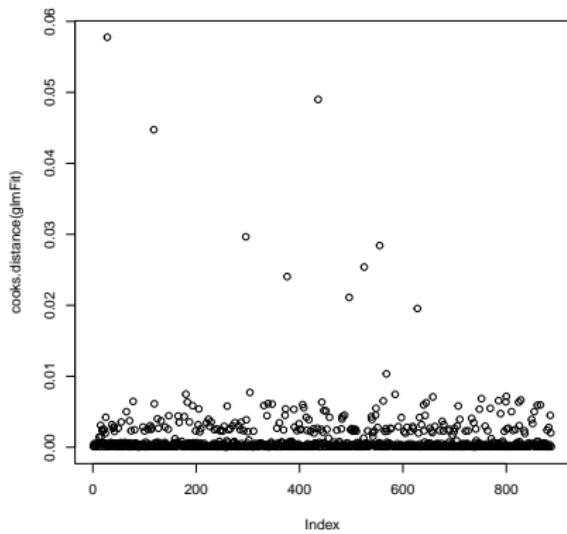
As with linear regression, we need to deal with any influential cases.

- 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.

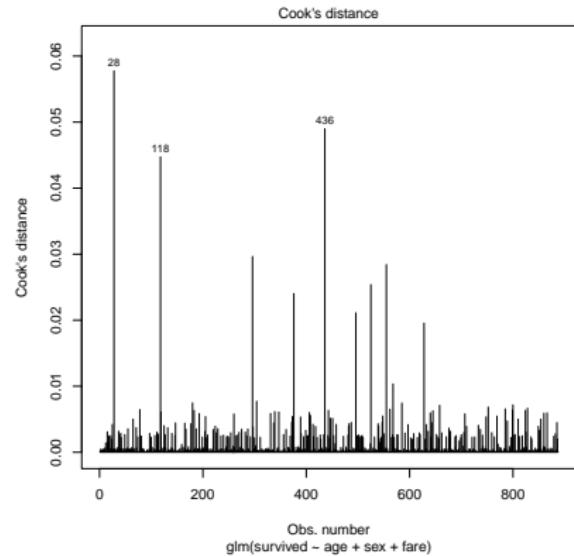


# Influential Cases

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



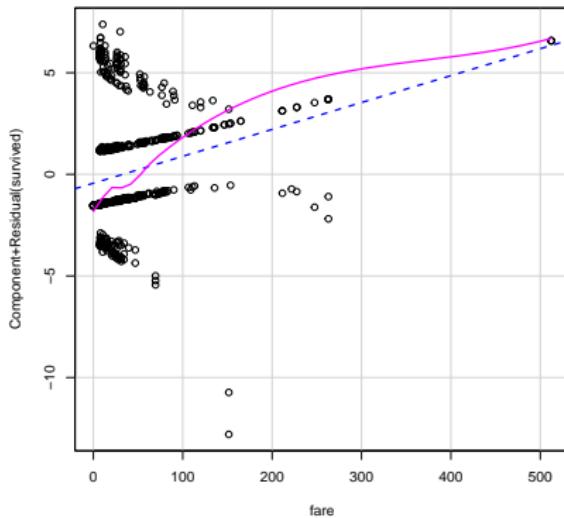
```
plot(glmFit, 4)
```



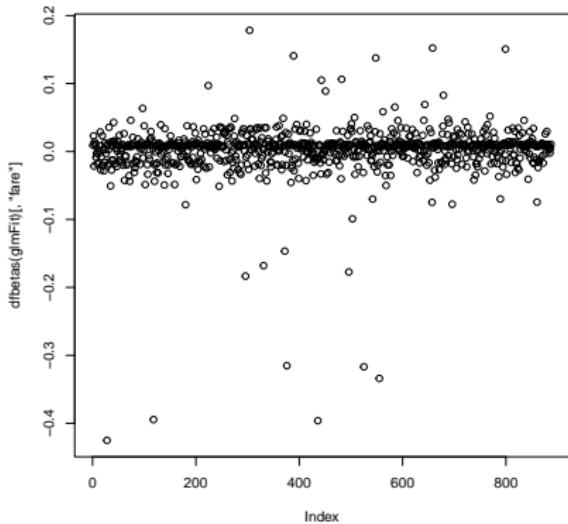
# Influential Cases

Recall the weirdly large ticket fares we saw earlier.

```
car::crPlots(glmFit, "fare")
```



```
dfbetas(glmFit)[ , "fare"] |> plot()
```



# Influential Cases

---

Let's see if the large fares are influential.

```
# Find the three most influential cases:  
mostInf <- which(cooks.distance(glmFit) > 0.04)  
  
# View the problematic case:  
titanic0[mostInf, ]  
  
   survived class          name  sex age  
28       no    1st Mr. Charles Alexander Fortune male  19  
118      no    1st     Mr. Quigg Edmond Baxter male  24  
436      no    1st     Mr. Mark Fortune male  64  
   siblings_spouses parents_children   fare  
28             3                  2 263.0000  
118            0                  1 247.5208  
436            1                  4 263.0000
```

Hmm...the most influential cases don't have especially large fares.

# Influential Cases

---

Let's turn our attention to the high-fare cases.

```
# View the largest 12 fares:  
sortFare <- titanic$fare |> sort(decreasing = TRUE)  
head(sortFare, 12)  
  
[1] 512.3292 512.3292 512.3292 263.0000 263.0000 263.0000  
[7] 263.0000 262.3750 262.3750 247.5208 247.5208 227.5250  
  
# Find the observation number for the three largest fares:  
moneyBags <- which(titanic$fare %in% sortFare[1:3])
```

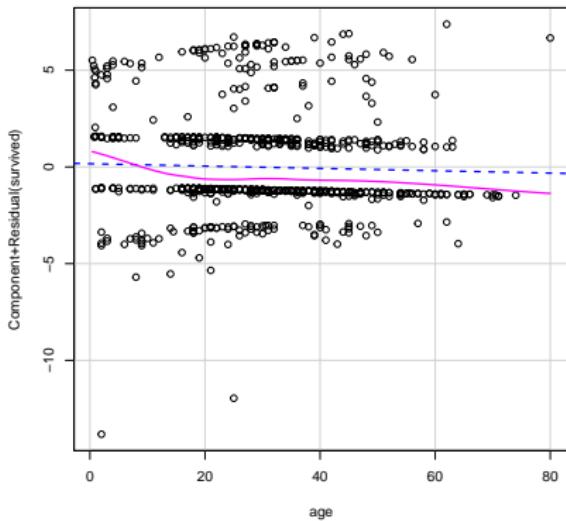
# Influential Cases

---

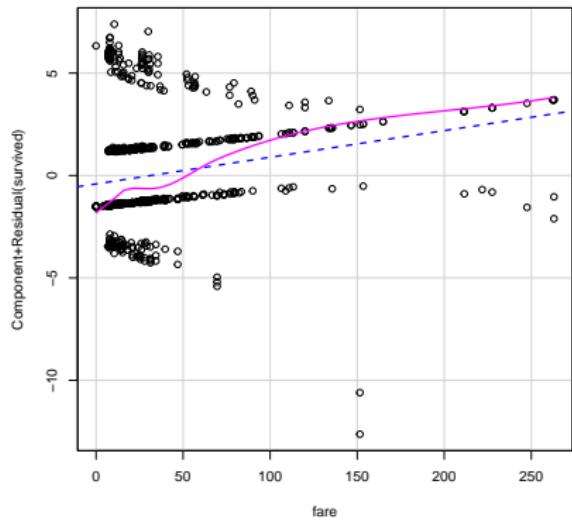
```
# View the cases with the largest fares:  
titanic0[moneyBags, ]  
  
survived class name sex  
258 yes 1st Miss. Anna Ward female  
677 yes 1st Mr. Thomas Drake Martinez Cardeza male  
734 yes 1st Mr. Gustave J Lesurer male  
  
age siblings_spouses parents_children fare  
258 35 0 0 512.3292  
677 36 0 1 512.3292  
734 35 0 0 512.3292  
  
# Refit the model excluding the cases with the three largest fares:  
titanic2 <- titanic[-moneyBags, ]  
glmFit2 <- update(glmFit, data = titanic2)
```

# Influential Cases

```
car::crPlots(glmFit2, "age")
```



```
car::crPlots(glmFit2, "fare")
```



# Influential Cases

---

Nothing much happening with the coefficient estimates.

```
summary(glmFit)$coef |> round(3)

      Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.838     0.215   3.894   0.00
age          -0.007    0.006  -1.226   0.22
sexmale      -2.392    0.171 -13.967   0.00
fare          0.012     0.002   4.955   0.00

summary(glmFit2)$coef |> round(3)

      Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.840     0.215   3.899   0.000
age          -0.007    0.006  -1.221   0.222
sexmale      -2.393    0.171 -13.970   0.000
fare          0.011     0.002   4.824   0.000
```

# Influential Cases

---

The deviances are largely unchanged.

```
partSummary(glmFit, 4)
```

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

```
partSummary(glmFit2, 4)
```

```
Null deviance: 1177.04 on 883 degrees of freedom  
Residual deviance: 881.34 on 880 degrees of freedom  
AIC: 889.34
```

The large-fare cases look weird, but they aren't influential.

- These cases follow the extrapolated trend implied by the other data.

# 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.
- 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.

