

## Part 4

Running binary logistic regression models in R

Interpreting binary logistic regression output



# Steps

- 
1. Prepare our data for analysis
  2. Explore our data
  3. Run the binary logistic regression model
  4. Evaluate the model
  5. Evaluate the individual predictors
  6. Predicted probabilities
  - ~~7. Check residuals~~

# 1. Preparing our data for analysis

## The outcome

---

- The binary outcome should be stored as a numeric value with outcomes coded as 0 and 1
- Set 1 as the outcome level you are interested in:
  - If you are interested in whether an individual is **happy**, set 1 as “happy”, set 0 as “not happy”
  - If you are interested in what predicts passing an exam, set 1 as “pass” and set 0 as “fail”

# 1. Preparing our data for analysis

## The outcome

	Participant_ID	Hamster	Happy	Happy_numeric
19	19	Yes	Yes	1
20	20	Yes	Yes	1
21	21	Yes	Yes	1
22	22	Yes	Yes	1
23	23	Yes	Yes	1
24	24	Yes	Yes	1
25	25	Yes	Yes	1
26	26	Yes	No	0
27	27	Yes	No	0
28	28	Yes	No	0
29	29	Yes	No	0
30	30	Yes	No	0
31	31	Yes	No	0
32	32	Yes	No	0
33	33	Yes	No	0

**Numeric variable:**

1 = Happy = "yes"

0 = Happy = "no"

# 1. Preparing our data for analysis

## The predictor(s)

---

- Categorical predictor should be a factor
- Set the first factor level as the level you want to be the reference category
  - If you are interested in the impact of having a hamster, set Hamster=No as the reference category
  - The coefficients will then tell you the impact of going from HamsterNo to HamsterYes

# 1. Preparing our data for analysis

## The predictor(s)

```
str(happiness_data$Hamster)
```

Tells you about the structure of the “Hamster” variable

Hamster is a factor with two levels – factor level 1 is “No”. Factor level 2 is “Yes”.

```
> str(happiness_data$Hamster)
Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2
```

## 2. Explore our data



```
table(happiness_data$Hamster, happiness_data$Happy_numeric)
```

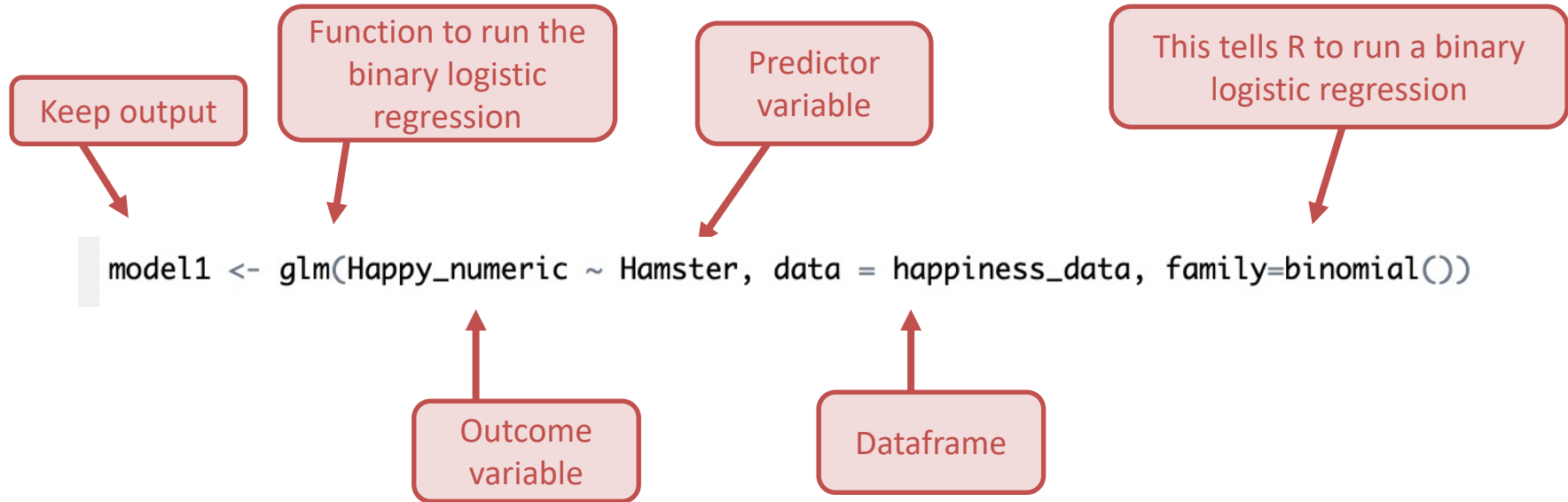
		Happy_numeric	
		0	1
Hamster	No	12	8
	Yes	8	25

No evidence of  
quasi-complete  
separation or  
complete  
separation

### 3. Run the binary logistic regression model

#### Code to run the binary logistic regression model

---





### 3. Run the binary logistic regression model `summary(model)`

`summary(model1)`

```
Call:
glm(formula = Happy_numeric ~ Hamster, family = binomial(), data = happiness_data)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.6835	-1.0108	0.7452	0.7452	1.3537

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-0.4055	0.4564	-0.888	0.3744
HamsterYes	1.5449	0.6110	2.528	0.0115 *

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(Dispersion parameter for binomial family taken to be 1)

```
Null deviance: 70.252  on 52  degrees of freedom
Residual deviance: 63.475  on 51  degrees of freedom
AIC: 67.475
```

```
Number of Fisher Scoring iterations: 4
```

As with linear regression, we can obtain statistics that allow us to evaluate the model and the individual predictors

## 4. Evaluating the model

### Comparing to the intercept only model

---

- To assess the fit of our model, we can compare our specified model to a model containing only the intercept (no predictors)
- We do this by looking at a measure called the “deviance”:
  - This is a measure of goodness of fit of the model
  - It tells you how much your model deviates from a model that perfectly predicts the data

## 4. Evaluating the model

### Comparing to the intercept only model

Null deviance:  
Deviance for a  
model  
containing only  
the intercept

```
HamsterYes    1.5449    0.6110    2.528    0.0115 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 70.252  on 52  degrees of freedom
Residual deviance: 63.475  on 51  degrees of freedom
AIC: 67.475

Number of Fisher Scoring iterations: 4
```

Residual deviance:  
Deviance for the  
specified model (i.e.  
containing 'hamster'  
as a predictor)

Residual deviance is lower than  
null deviance in our example

If our model is better than the model containing only the intercept,  
the “residual deviance” should be lower than the “null deviance”

## 4. Evaluating the model

### ...But is our model **significantly** better?

To assess this, we need to work out the model chi square (test statistic) and its p-value

#### Step 1: Calculate the chi square test statistic

```
model1_chi <- model1$null.deviance - model1$deviance
model1_chi
```

```
> model1_chi
[1] 6.777043
```

Produces the model chi square value (equal to the null deviance minus the deviance)

This is the improvement of the new model over the intercept only model

Our model chi square is 6.78

## 4. Evaluating the model

### ...But is our model **significantly** better?

To assess this, we need to work out the model chi square (test statistic) and its p-value

#### Step 2: Calculate the degrees of freedom

```
model1_chi_df <- model1$df.null - model1$df.residual
model1_chi_df
```

```
> model1_chi_df
[1] 1
```

Produces the degrees of freedom for the model (equal to the df for the intercept only model minus the deviance for our model)

Our model df is 1

## 4. Evaluating the model

### ...But is our model **significantly** better?

To assess this, we need to work out the model chi square (test statistic) and it's p-value

**Step 3: Use the test statistic and the degrees of freedom to calculate the p-value**

```
model1_p <- 1 - pchisq(model1_chi, model1_chi_df)
model1_p
```

Produces p-value for the model

```
> model1_p
[1] 0.009233774
> |
```

The p-value is .009

## 4. Evaluating the model

### ...But is our model **significantly** better?

---

```
> model1_chi  
[1] 6.777043
```

```
> model1_chi_df  
[1] 1
```

```
> model1_p  
[1] 0.009233774  
>
```

Put together:  $X^2(1) = 6.78, p = .009$

This indicates that adding the hamster variable to our model significantly improved the fit, compared to the null model containing intercept only

## 4. Evaluating the model

### Does binary logistic regression have $R^2$ ?

---

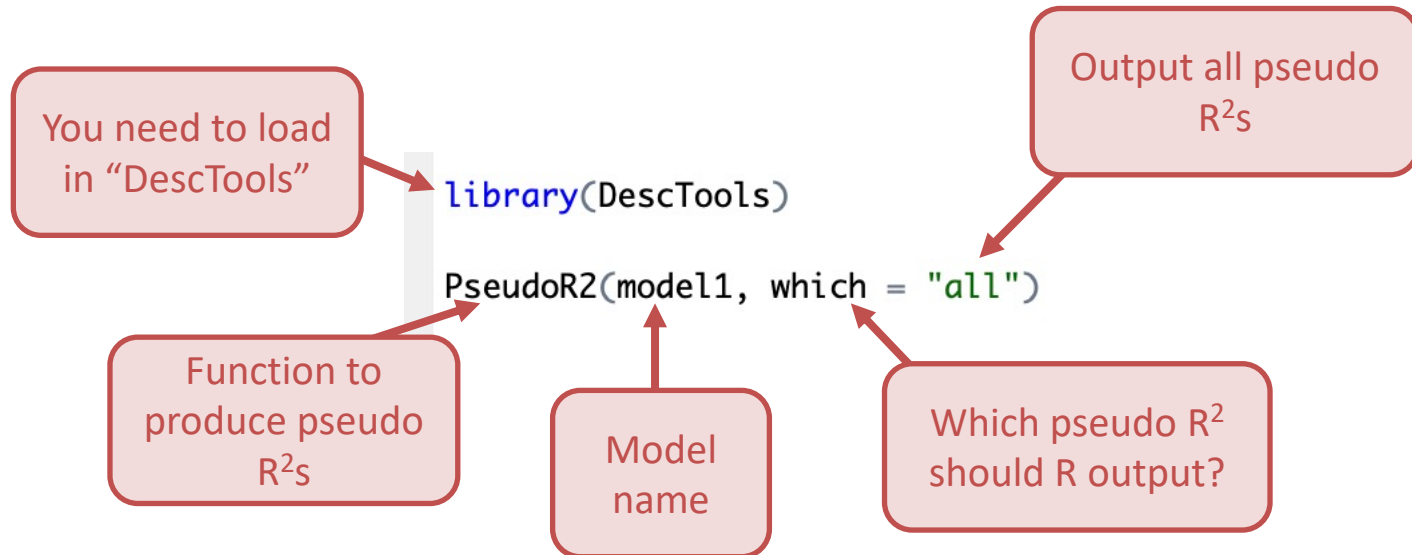
- $R^2$  in linear regression = the proportion of variance explained by the model
- In logistic regression, this doesn't exist
- But several statisticians have developed measures that work in a similar way to  $R^2$  for logistic regression. These are called pseudo  $R^2$ s.
- They all give you an indication of how well the model explains the outcome variable



## 4. Evaluating the model

### Computing pseudo $R^2$ s

- Often, researchers report several measures of pseudo  $R^2$  as there is little consensus on the best method



## 4. Evaluating the model

### Computing pseudo $R^2$ s

```

> PseudoR2(model1, which = "all")

```

McFadden	McFaddenAdj	CoxSnell	Nagelkerke	AldrichNelson	VeallZimmermann	Efron	McKelveyZavoina
0.09646741	0.03952965	0.12003112	0.16345594	0.11337200	0.19890271	0.12786042	0.14563265
1jur	AIC	BIC	logLik	logLik0	G2		
0.12786042	67.47510983	71.41569365	-31.73755491	-35.12607642	6.77704301		

- McFadden, CoxSnell and Nagelkerke are often reported

- McFadden = 0.10
- CoxSnell = 0.12
- Nagelkerke = 0.16

- Not an easy interpretation of pseudo  $R^2$ s, but higher values equal better model fit

# 5. Evaluating individual predictors

## What is the intercept?

```
Call:
glm(formula = Happy_numeric ~ Hamster, family = binomial(), data = happiness_data)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.6835	-1.0108	0.7452	0.7452	1.3537

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-0.4055	0.4564	-0.888	0.3744
HamsterYes	2.5113	0.6116	4.122	0.0115 *

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 70.252 on 52 degrees of freedom  
Residual deviance: 63.475 on 51 degrees of freedom  
AIC: 67.475

Number of Fisher Scoring iterations: 4

The log odds  
of someone  
with a  
Hamster  
value of "No"  
(No hamster)  
having a  
happiness  
value of "Yes"

The log odds that Happy = yes  
for the reference category of  
our predictor variable (e.g.  
Hamster = No)

## 5. Evaluating individual predictors Hamster

```
Call:
glm(formula = Happy_numeric ~ Hamster, family = binomial(), data = happiness_data)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.6835	-1.0108	0.7452	0.7452	1.3537

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	0.4055	0.4564	0.888	0.3744
HamsterYes	1.5449	0.6110	2.528	0.0115 *

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 70.252 on 52 degrees of freedom  
Residual deviance: 63.475 on 51 degrees of freedom  
AIC: 67.475

Number of Fisher Scoring iterations: 4

For our hamster variable, we can see the variable is called HamsterYes.

This tells us the the **change in the log odds** of having a happiness value of “Yes” when going from the reference category (HamsterNo) to HamsterYes

## 5. Evaluating individual predictors Hamster

```
Call:
glm(formula = Happy_numeric ~ Hamster, family = binomial(), data = happiness_data)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.6835	-1.0108	0.7452	0.7452	1.3537

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-0.4055	0.4564	-0.888	0.3744
HamsterYes	1.5449	0.6110	2.528	0.0115 *

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 70.252 on 52 degrees of freedom  
Residual deviance: 63.475 on 51 degrees of freedom  
AIC: 67.475

Number of Fisher Scoring iterations: 4

Going from the reference category (HamsterNo) to “HamsterYes” results in a 1.54 unit increase in the log odds of having a happiness value of “Yes”

## 5. Evaluating individual predictors

### How do we interpret log odds...?!

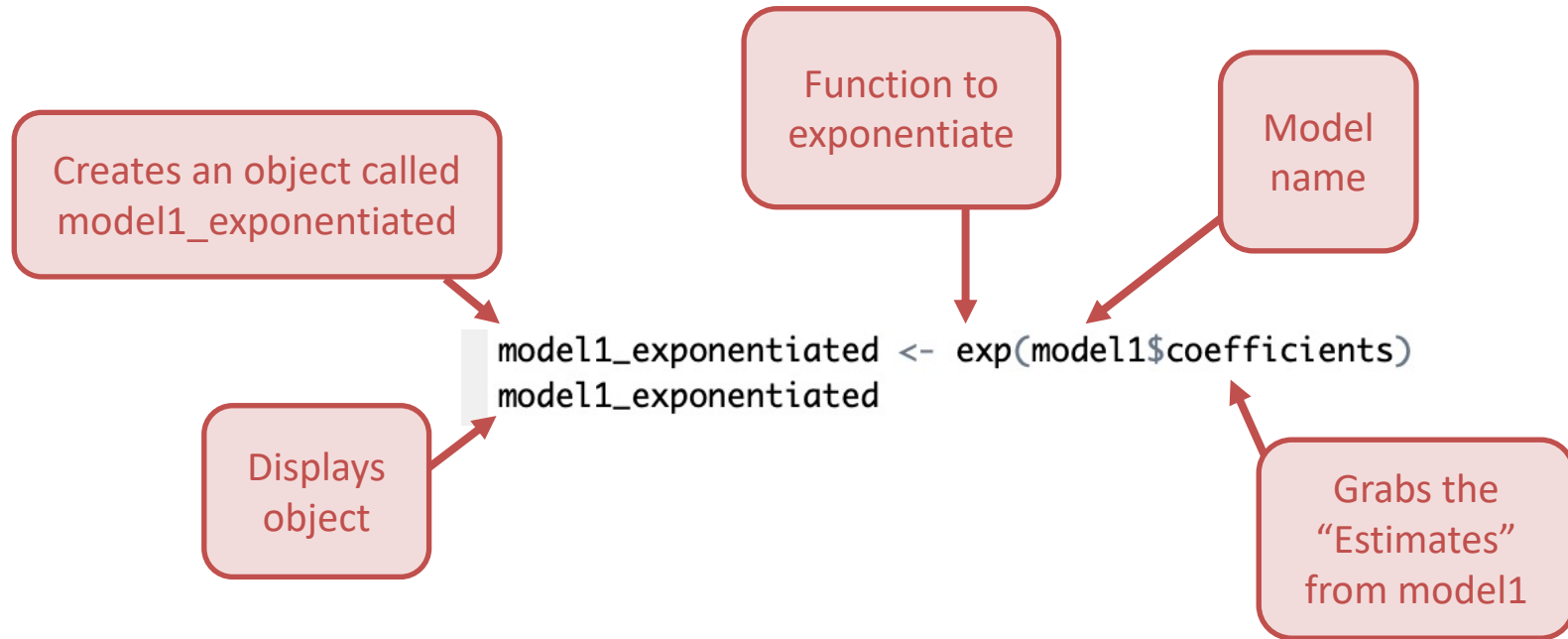
---

- Log odds are very difficult to interpret.... so we don't usually do this.
- **I won't be asking you to interpret the log odds in the lab/WBA/class test**
- Instead we convert back from the log scale, which makes interpretation a little easier

## 5. Evaluating individual predictors

### We need to convert back from the log scale

- To convert back from the log scale, we exponentiate our log odds (*“Estimate”*). This gives us our odds ratio



# 5. Evaluating individual predictors

## This produces an odds ratio

Do these values look familiar?!

(Intercept)	HamsterYes
0.666667	4.687500

Odds in the no hamster group

$$\text{Odds} = \frac{\text{Probability event occurs}}{\text{Probability event does not occur}}$$

	Happy - No	Happy - Yes	Total
Hamster - No	12	8	20

What's the probability they are NOT happy?  
12/20 = 0.6

$$\text{Odds} = \frac{\text{Probability happy}}{\text{Probability not happy}}$$

What's the probability they are happy?  
8/20 = 0.4

$$\text{Odds} = \frac{0.4}{0.6} = 0.667$$

- Intercept: the odds that happy = yes in the reference group

Change in odds (or odds ratio)

$$\text{Odds ratio} = \frac{\text{Odds after a unit change in the predictor}}{\text{Original odds}}$$

No hamster:  
Odds = 0.6667

Hamster:  
Odds = 3.125

$$\text{Odds ratio} = \frac{3.125}{0.6667} = 4.69$$

- Odds ratio: the change in odds after a unit change in the predictor (Hamster - No -> Yes)



## 5. Evaluating individual predictors

### This produces an odds ratio

---

(Intercept)	HamsterYes
0.666667	4.687500

- These values are easier to interpret than log odds!
- With a categorical outcome, this tells us the change in odds from a unit change in the predictor
- The odds of being happy are 4.69x higher if you have a hamster than if you do not have a hamster

## 5. Evaluating individual predictors

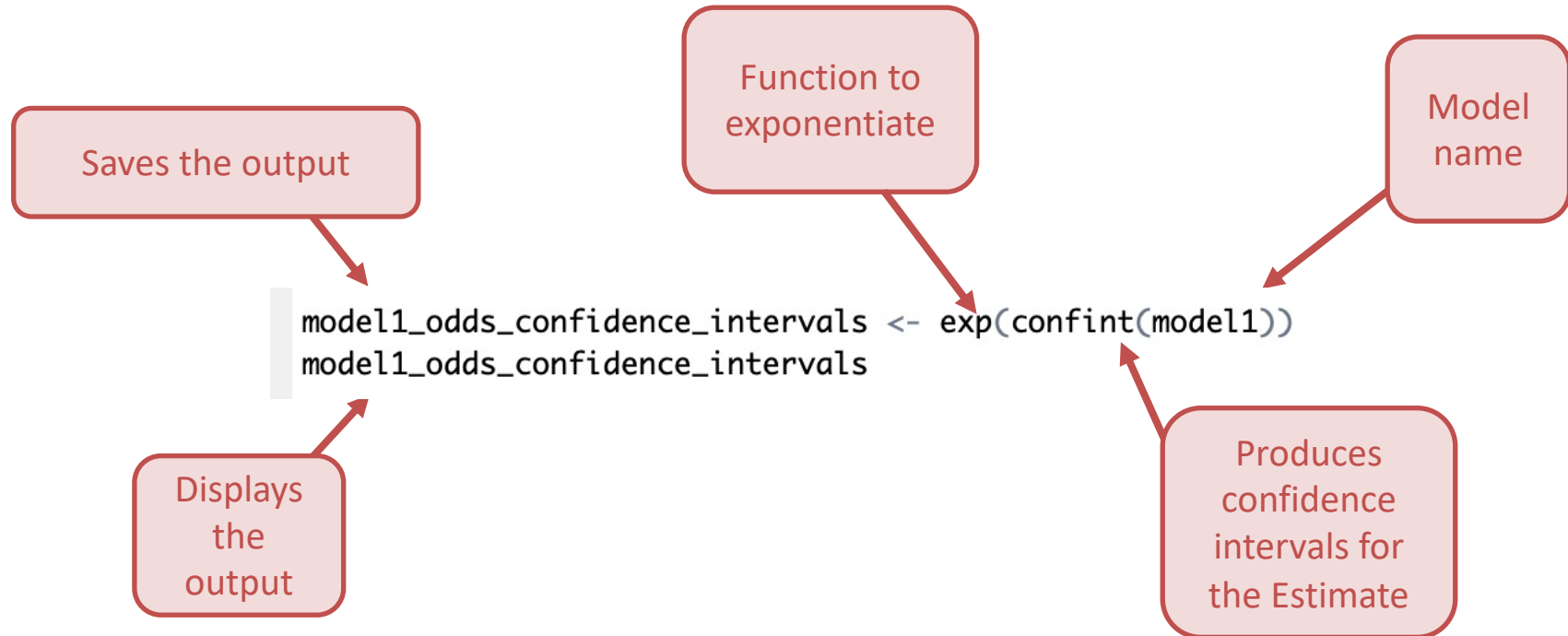
### Odds ratio confidence interval

---

- We also want a confidence interval arounds the odds ratio
- 95% confidence interval tells us the likely range the true odds ratio in the population is contained in

## 5. Evaluating individual predictors

### Odds ratio confidence interval



## 5. Evaluating individual predictors

### Odds ratio confidence interval

	Lower bound of the confidence interval		Higher bound of the confidence interval
	2.5 %	97.5 %	
(Intercept)	0.2612025	1.611465	
HamsterYes	1.4571273	16.307968	

Odds ratio 95% confidence interval = 1.46-16.31

## 5. Evaluating individual predictors

### Is our p-value significant?

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.4055     0.4564  -0.888   0.3744
HamsterYes     1.5449     0.6110   2.528   0.0115 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

$p = .012$

Whether or not an individual has a hamster is a significant predictor of happiness

## 6. Predicted probabilities

---

- So far we've been talking in odds...
- But we can obtain probabilities from our model too. For instance:
  - If an individual has a hamster, what's the probability they will be happy?
  - If an individual does not have a hamster, what's the probability they will be happy?

## 6. Predicted probabilities

```
happiness_data$m1_predicted_probabilities <- fitted(model1)
```

Adds a column called `m1_predicted_probabilities` to our happiness dataframe

Participant ID	Hamster	Happy	Happy numeric	m1_predicted_probabilities
21	Yes	Yes	1	0.7575758
22	Yes	Yes	1	0.7575758
23	Yes	Yes	1	0.7575758
24	Yes	Yes	1	0.7575758
25	Yes	Yes	1	0.7575758
42	No	No	0	0.4000000
43	No	No	0	0.4000000
44	No	No	0	0.4000000
45	No	No	0	0.4000000
46	No	No	0	0.4000000
47	No	No	0	0.4000000
48	No	No	0	0.4000000
49	No	No	0	0.4000000
50	No	No	0	0.4000000
51	No	No	0	0.4000000
52	No	No	0	0.4000000
53	No	No	0	0.4000000
34	No	Yes	1	0.4000000
35	No	Yes	1	0.4000000
36	No	Yes	1	0.4000000
37	No	Yes	1	0.4000000
38	No	Yes	1	0.4000000
39	No	Yes	1	0.4000000

Value ranges between 0 and 1

- When Hamster = Yes, predicted probability = 0.76

→ When an individual has a hamster, there is a probability of 0.76 that they will be happy (76% of people with a hamster will be happy)

## 6. Predicted probabilities

```
happiness_data$m1_predicted_probabilities <- fitted(model1)
```

Adds a column called `m1_predicted_probabilities` to our happiness dataframe

	Participant_ID	Hamster	Happy	Happy_numeric	m1_predicted_probabilities
21	21	Yes	Yes	1	0.7575758
22	22	Yes	Yes	1	0.7575758
23	23	Yes	Yes	1	0.7575758
24	24	Yes	Yes	1	0.7575758
25	25	Yes	Yes	1	0.7575758
42	42	No	No	0	0.400000
43	43	No	No	0	0.400000
44	44	No	No	0	0.400000
45	45	No	No	0	0.400000
46	46	No	No	0	0.400000
47	47	No	No	0	0.400000
48	48	No	No	0	0.400000
49	49	No	No	0	0.400000
50	50	No	No	0	0.400000
51	51	No	No	0	0.400000
52	52	No	No	0	0.400000
53	53	No	No	0	0.400000
34	34	No	Yes	1	0.400000
35	35	No	Yes	1	0.400000
36	36	No	Yes	1	0.400000
37	37	No	Yes	1	0.400000
38	38	No	Yes	1	0.400000
39	39	No	Yes	1	0.400000

- When Hamster = No, predicted probability = 0.40

→ When an individual does **NOT** have a hamster, there is a probability of 0.40 that they will be happy (40% of people with no hamster will be happy)



# Reporting logistic regression in APA format

A binary logistic regression was conducted to examine whether having a hamster (yes/no) is a significant predictor of happiness (yes/no). The model predicted happiness significantly better than the intercept-only model ( $X^2(1) = 6.78, p = .009$ ; McFadden Pseudo  $R^2 = 0.10$ , CoxSnell Pseudo  $R^2 = 0.12$ , Nagelkerke Pseudo  $R^2 = 0.16$ ). The model revealed that individuals who have a hamster had a significantly higher odds being happy relative to individuals who do not have a hamster (Odds ratio = 4.69, 95% confidence interval arounds the odds ratio = 1.46-16.31,  $p = .012$ ).

			95% confidence interval	
	B (SE)	Odds ratio	Lower	Upper
Constant	-0.41 (0.46)			
Hamster	1.55 (0.61)	4.69	1.46	16.31

## Lab preparation (~10 minutes)

---

- Please watch the short lab preparation video prior to your lab
- We will walk through an R script that runs a binary logistic regression model

# Post-lecture activities

---

- Now available on Moodle

Thank you for listening!

Please post any questions on the relevant Qualtrics link on Moodle.