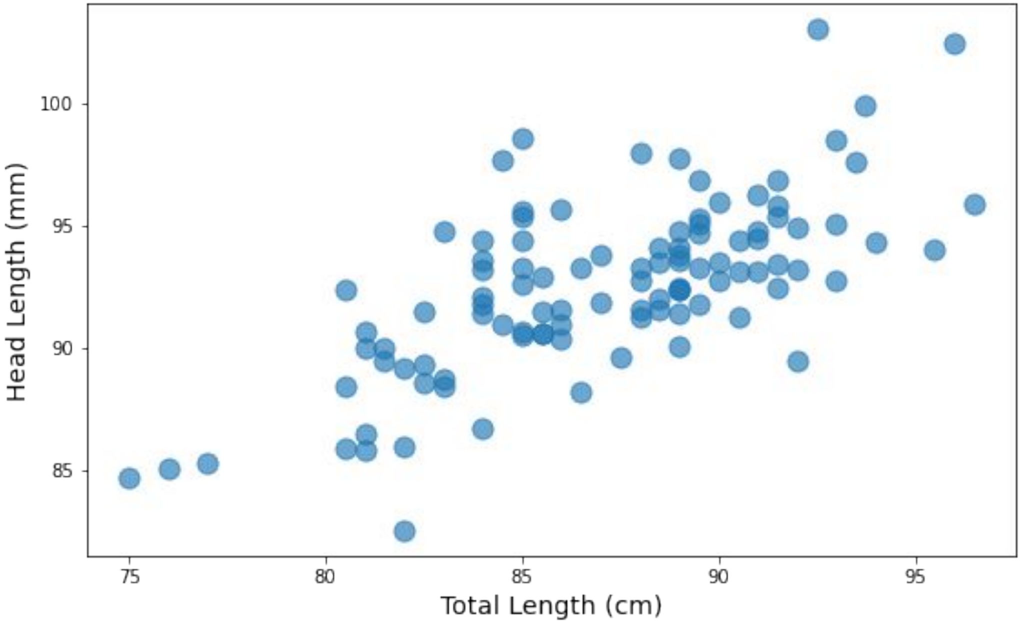
Introduction to Generalized Linear Models

Part 2: Logistic Regression

Recall: Australian brush possums



OpenIntro Statistics, Section 8.1.2

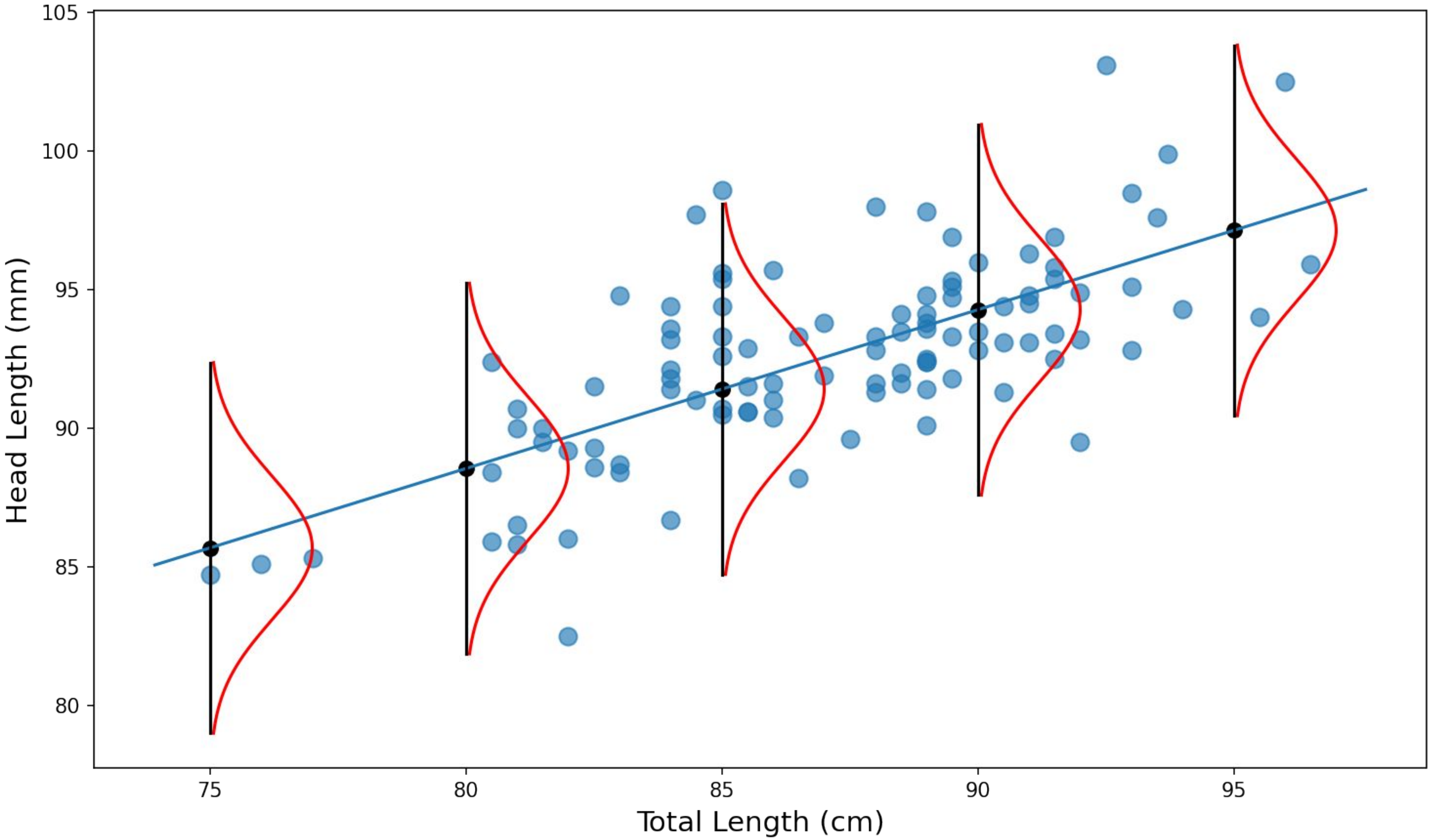
```
linreg_tl.summary()
```

Generalized Linear Model Regression Results

Dep. Variable:	head_l	**No. Observations:**	104
Model:	GLM	**Df Residuals:**	102
Model Family:	Gaussian	**Df Model:**	1
Link Function:	identity	**Scale:**	6.7357
Method:	IRLS	**Log-Likelihood:**	-245.75
Date:	Wed, 15 Sep 2021	**Deviance:**	687.04
Time:	22:16:23	**Pearson chi2:**	687.
No. Iterations:	3		
Covariance Type:	nonrobust		

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
const	42.7098	5.173	8.257	0.000	32.571	52.848
total_l	0.5729	0.059	9.657	0.000	0.457	0.689

For possums with a total length of t, the model estimates that the distribution of head lengths is normal with a mean of $42.7098 + 0.5729t$ and a variance of 6.7357.


Head Length (mm)
105
100
95
90
85
80
75
80
85
90
95
Total Length (cm)

Linear Regression in General

$Y|\vec{x}$ follows a normal distribution with mean

$$\mu = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

Where $\vec{x} = \langle x_1, \ldots, x_n \rangle$ are the values of the predictor variables.

Now what if our target variable is a binary categorical variable?

OpenIntro Statistics, Section 9.5

```
1  resume.head()
```

	received_callback	honors	years_experience
0	0	0	6
1	0	0	6
2	0	0	6
3	0	0	6
4	0	0	22

Goal: Predict the probability of receiving a callback.

OpenIntro Statistics, Section 9.5

```
resume.head()
```

	received_callback	honors	years_experience
0	0	0	6
1	0	0	6
2	0	0	6
3	0	0	6
4	0	0	22

target
column

What type of distribution do we expect the target to follow?

What type of distribution do we expect the target to follow?

Ans: A Bernoulli/Binomial distribution.

This means we just need to determine the probability of success (p).

Approach 1: Ignore the other variables and just focus on the target.

Approach 1: Ignore the other variables and just focus on the target.

```
import statsmodels.api as sm

logreg = (sm.GLM(endog = resume['received_callback'],
                 exog = sm.add_constant(resume[[]]),
                 family = sm.families.Binomial())
          .fit()
         )
```

Approach 1: Ignore the other variables and just focus on the target.

```
import statsmodels.api as sm

logreg = (sm.GLM(endog = resume['received_callback'],
                 exog = sm.add_constant(resume[[]]),
                 family = sm.families.Binomial())
          .fit()
         )
```

We'll be using the *statsmodels* library.

Approach 1: Ignore the other variables and just focus on the target.

```
import statsmodels.api as sm

logreg = (sm.GLM(endog = resume['received_callback'],
                 exog = sm.add_constant(resume[[]]),
                 family = sm.families.Binomial())
          .fit()
         )
```

Fit a Generalized Linear Model (GLM).

Approach 1: Ignore the other variables and just focus on the target.

```
import statsmodels.api as sm

logreg = (sm.GLM(endog = resume['received_callback'] ,
                 exog = sm.add_constant(resume[[]]),
                 family = sm.families.Binomial())
          .fit()
         )
```

This tells the model the target variable.

Approach 1: Ignore the other variables and just focus on the target.

```
import statsmodels.api as sm

logreg = (sm.GLM(endog = resume['received_callback'],
                 exog = sm.add_constant(resume[[]]) ,
                 family = sm.families.Binomial())
          .fit()
         )
```

We are not going to use any other variables in our initial model.

Approach 1: Ignore the other variables and just focus on the target.

```
import statsmodels.api as sm

logreg = (sm.GLM(endog = resume['received_callback'],
                 exog = sm.add_constant(resume[[]]),
                 family = sm.families.Binomial() )
          .fit()
         )
```

We’ll assume that the target follows a Binomial/Bernoulli distribution.

Approach 1: Ignore the other variables and just focus on the target.

```
import statsmodels.api as sm

logreg = (sm.GLM(endog = resume['received_callback'],
                 exog = sm.add_constant(resume[[]]),
                 family = sm.families.Binomial())
          .fit()
         )
```

Go ahead and fit the model after specifying it.

Approach 1: Ignore the other variables and just focus on the target.

```
logreg.summary()
```

Generalized Linear Model Regression Results

Dep. Variable:	received_callback	**No. Observations:**	4870
Model:	GLM	**Df Residuals:**	4869
Model Family:	Binomial	**Df Model:**	0
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-1363.5
Date:	Wed, 15 Sep 2021	**Deviance:**	2726.9
Time:	15:28:44	**Pearson chi2:**	4.87e+03
No. Iterations:	5		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-2.4357	0.053	-46.242	0.000	-2.539	-2.332

Approach 1: Ignore the other variables and just focus on the target.

```
logreg.summary()
```

Generalized Linear Model Regression Results

Dep. Variable:	received_callback	**No. Observations:**	4870
Model:	GLM	**Df Residuals:**	4869
Model Family:	Binomial	**Df Model:**	0
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-1363.5
Date:	Wed, 15 Sep 2021	**Deviance:**	2726.9
Time:	15:28:44	**Pearson chi2:**	4.87e+03
No. Iterations:	5		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-2.4357	0.053	-46.242	0.000	-2.539	-2.332

The estimated value of *p* is *logistic*(-2.4357)

Approach 1: Ignore the other variables and just focus on the target.

```
logreg.summary()
```

Generalized Linear Model Regression Results

Dep. Variable:	received_callback	**No. Observations:**	4870
Model:	GLM	**Df Residuals:**	4869
Model Family:	Binomial	**Df Model:**	0
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-1363.5
Date:	Wed, 15 Sep 2021	**Deviance:**	2726.9
Time:	15:28:44	**Pearson chi2:**	4.87e+03
No. Iterations:	5		
Covariance Type:	nonrobust		

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
const	-2.4357	0.053	-46.242	0.000	-2.539	-2.332

The estimated value of *p* is *logistic*(-2.4357)

$$= \frac{1}{1 + e^{-(-2.4357)}}$$

$$= 0.0805$$

Note: This is identical to the overall proportion of applicants who received a callback.

```
(
    resume['received_callback']
    .value_counts(normalize = True)
)
```

```
0    0.919507
1    0.080493
Name: received_callback, dtype: float64
```

Approach 2: Estimate using the honors column (and a constant) .

Approach 2: Estimate using the honors column (and a constant) .

```
logreg_honors = (sm.GLM(endog = resume['received_callback'],
                        exog = sm.add_constant(resume[['honors']]),
                        family = sm.families.Binomial())
                 .fit()
                )
```

Approach 2: Estimate using the honors column (and a constant) .

```
logreg_honors = (sm.GLM(endog = resume['received_callback'],
                        exog = sm.add_constant(resume[['honors']]) ,
                        family = sm.families.Binomial())
                  .fit()
                )
```

This time, we'll use the honors column as a predictor.

Approach 2: Estimate using the honors column (and a constant) .

```
logreg_honors.summary()
```

Generalized Linear Model Regression Results

Dep. Variable:	received_callback	**No. Observations:**	4870
Model:	GLM	**Df Residuals:**	4868
Model Family:	Binomial	**Df Model:**	1
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-1353.4
Date:	Wed, 15 Sep 2021	**Deviance:**	2706.7
Time:	23:28:29	**Pearson chi2:**	4.87e+03
No. Iterations:	5		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-2.4998	0.056	-44.958	0.000	-2.609	-2.391
honors	0.8668	0.178	4.880	0.000	0.519	1.215

Approach 2: Estimate using the honors column (and a constant) .

```
logreg_honors.summary()
```

Generalized Linear Model Regression Results

Dep. Variable:	received_callback	**No. Observations:**	4870
Model:	GLM	**Df Residuals:**	4868
Model Family:	Binomial	**Df Model:**	1
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-1353.4
Date:	Wed, 15 Sep 2021	**Deviance:**	2706.7
Time:	23:28:29	**Pearson chi2:**	4.87e+03
No. Iterations:	5		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-2.4998	0.056	-44.958	0.000	-2.609	-2.391
honors	0.8668	0.178	4.880	0.000	0.519	1.215

For applicants **without honors**, the model estimates that the distribution of callbacks is Bernoulli with *p* equal to

logistic(-2.4998) = 0.0759

Approach 2: Estimate using the honors column (and a constant) .

```
logreg_honors.summary()
```

Generalized Linear Model Regression Results

Dep. Variable:	received_callback	**No. Observations:**	4870
Model:	GLM	**Df Residuals:**	4868
Model Family:	Binomial	**Df Model:**	1
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-1353.4
Date:	Wed, 15 Sep 2021	**Deviance:**	2706.7
Time:	23:28:29	**Pearson chi2:**	4.87e+03
No. Iterations:	5		
Covariance Type:	nonrobust		

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
const	-2.4998	0.056	-44.958	0.000	-2.609	-2.391
honors	0.8668	0.178	4.880	0.000	0.519	1.215

For applicants **with honors**, the model estimates that the distribution of callbacks was Bernoulli with *p* equal to *logistic*(-2.4998 + 0.8668) = 0.1634

Similar to linear regression, we can add additional predictors.

Approach 3: Estimate using the honors and years_experience column.

```
logreg_full = sm.GLM(endog = resume['received_callback'],
                 exog = sm.add_constant(resume[['honors', 'years_experience']]) ,
                 family = sm.families.Binomial()).fit()
```

Approach 3: Estimate using the honors and years_experience column.

```
logreg_full.summary()
```

Generalized Linear Model Regression Results

Dep. Variable:	received_callback	**No. Observations:**	4870
Model:	GLM	**Df Residuals:**	4867
Model Family:	Binomial	**Df Model:**	2
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-1347.4
Date:	Wed, 15 Sep 2021	**Deviance:**	2694.8
Time:	23:37:09	**Pearson chi2:**	4.86e+03
No. Iterations:	5		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-2.7664	0.096	-28.813	0.000	-2.955	-2.578
honors	0.7612	0.181	4.201	0.000	0.406	1.116
years_experience	0.0332	0.009	3.565	0.000	0.015	0.051

Approach 3: Estimate using the honors and years_experience column.

```
logreg_full.summary()
```

Generalized Linear Model Regression Results

Dep. Variable:	received_callback	**No. Observations:**	4870
Model:	GLM	**Df Residuals:**	4867
Model Family:	Binomial	**Df Model:**	2
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-1347.4
Date:	Wed, 15 Sep 2021	**Deviance:**	2694.8
Time:	23:37:09	**Pearson chi2:**	4.86e+03
No. Iterations:	5		
Covariance Type:	nonrobust		

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
const	-2.7664	0.096	-28.813	0.000	-2.955	-2.578
honors	0.7612	0.181	4.201	0.000	0.406	1.116
years_experience	0.0332	0.009	3.565	0.000	0.015	0.051

For applicants **without honors** and t years of experience, the model estimates that the distribution of callbacks is Bernoulli with p equal to

logistic(-2.7664 + 0.0332 t)

Approach 3: Estimate using the honors and years_experience column.

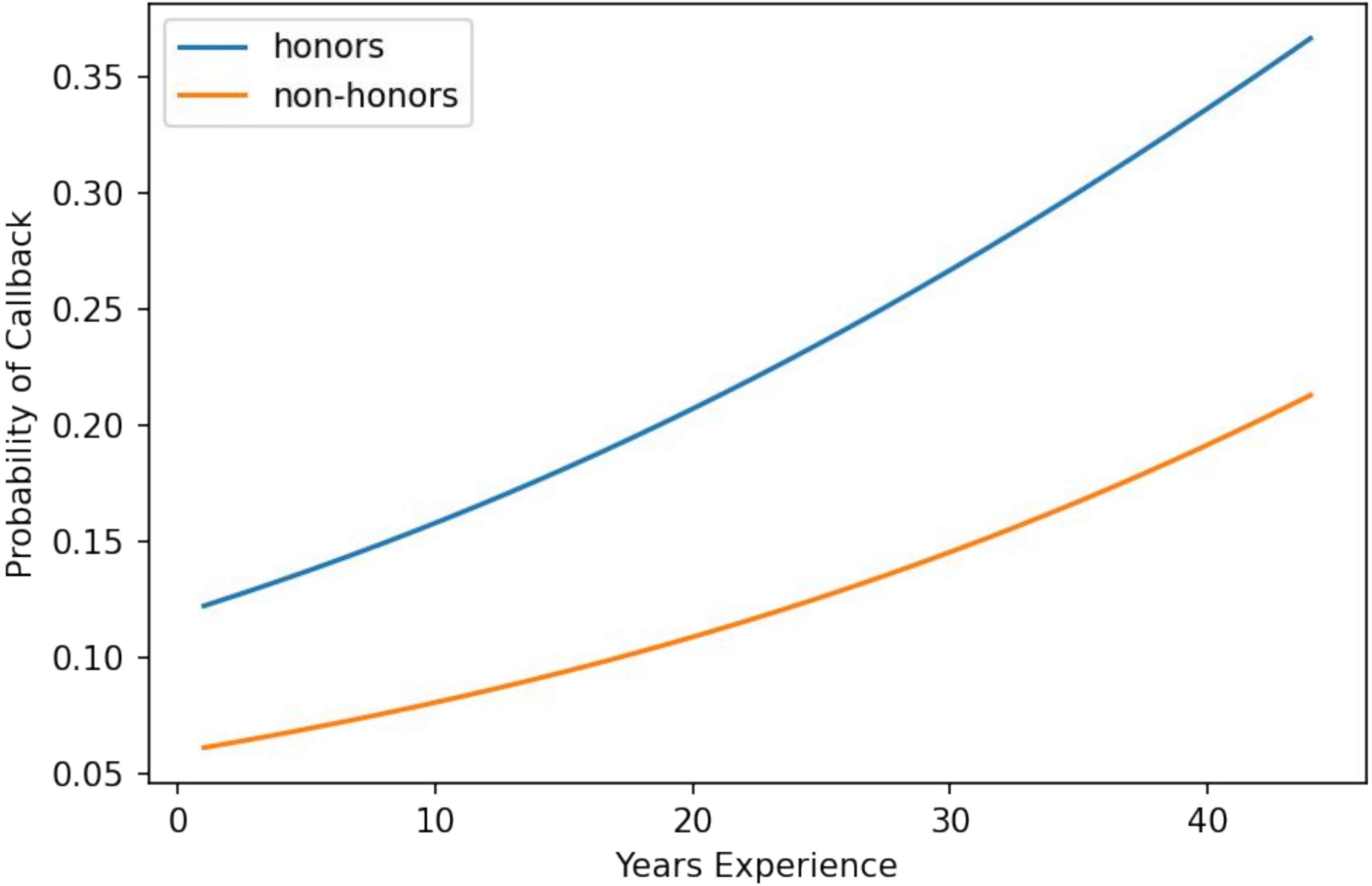
```
logreg_full.summary()
```

Generalized Linear Model Regression Results

Dep. Variable:	received_callback	**No. Observations:**	4870
Model:	GLM	**Df Residuals:**	4867
Model Family:	Binomial	**Df Model:**	2
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-1347.4
Date:	Wed, 15 Sep 2021	**Deviance:**	2694.8
Time:	23:37:09	**Pearson chi2:**	4.86e+03
No. Iterations:	5		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-2.7664	0.096	-28.813	0.000	-2.955	-2.578
honors	0.7612	0.181	4.201	0.000	0.406	1.116
years_experience	0.0332	0.009	3.565	0.000	0.015	0.051

For applicants **with honors** and t years of experience, the model estimates that the distribution of callbacks is Bernoulli with p equal to *logistic*(-2.7664 + 0.7612 + 0.0332t)


honors
non-honors
Probability of Callback
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0
10
20
30
40
Years Experience

Summary - Linear and Logistic Regression

Linear Regression

$Y|\vec{x}$ follows a $\boxed{}$ distribution with mean

$$\mu = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

Linear Regression

$Y|\vec{x}$ follows a normal distribution with mean

$$\mu = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

Logistic Regression

$Y|\vec{x}$ follows a distribution with mean

$$\mu = \quad (\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n)$$

Logistic Regression

$Y|\vec{x}$ follows a Bernoulli distribution with mean

$$\mu = \quad (\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n)$$

Logistic Regression

$Y|\vec{x}$ follows a $\boxed{\text{Bernoulli}}$ distribution with mean

$$\mu = \text{logistic}(\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n)$$

Logistic Regression

$Y|\vec{x}$ follows a $\boxed{\text{Bernoulli}}$ distribution with mean

$$\mu = \text{logistic}(\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n)$$

$$= \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n)}}$$

To Be Continued