

# Multiple logistic regression

INTERMEDIATE REGRESSION IN R



**Richie Cotton**

Learning Solutions Architect at  
DataCamp

# Bank churn dataset

has_churned	time_since_first_purchase	time_since_last_purchase
0	0.3993247	-0.5158691
1	-0.4297957	0.6780654
0	3.7383122	0.4082544
0	0.6032289	-0.6990435
...	...	...
<i>response</i>	<i>length of relationship</i>	<i>recency of activity</i>

<sup>1</sup> <https://www.rdocumentation.org/packages/bayesQR/topics/Churn>

# glm()

```
glm(response ~ explanatory, data = dataset, family = binomial)
```

```
glm(response ~ explanatory1 + explanatory2, data = dataset, family = binomial)
```

```
glm(response ~ explanatory1 * explanatory2, data = dataset, family = binomial)
```

# Prediction flow

```
explanatory_data <- expand_grid(  
  explanatory1 = some_values,  
  explanatory2 = some_values  
)  
prediction_data <- explanatory_data %>%  
  mutate(  
    has_churned = predict mdl, explanatory_data, type = "response")  
  )
```

# The four outcomes

	actual false	actual true
predicted false	correct	false negative
predicted true	false positive	correct

<sup>1</sup> <https://campus.datacamp.com/courses/introduction-to-regression-in-r/simple-logistic-regression?ex=10>

# Confusion matrix

```
actual_response <- dataset$response  
predicted_response <- round(fitted mdl))
```

```
outcomes <- table(predicted_response, actual_response)
```

```
confusion <- conf_mat(outcomes)
```

```
autoplot(confusion)
```

```
summary(confusion, event_level = "second")
```

# Visualization

- Use faceting for categorical variables.
- For 2 numeric explanatory variables, use color for response.
- Give responses below 0.5 one color; responses above 0.5 another color.

```
scale_color_gradient2(midpoint = 0.5)
```

# Let's practice!

INTERMEDIATE REGRESSION IN R



# The logistic distribution

INTERMEDIATE REGRESSION IN R



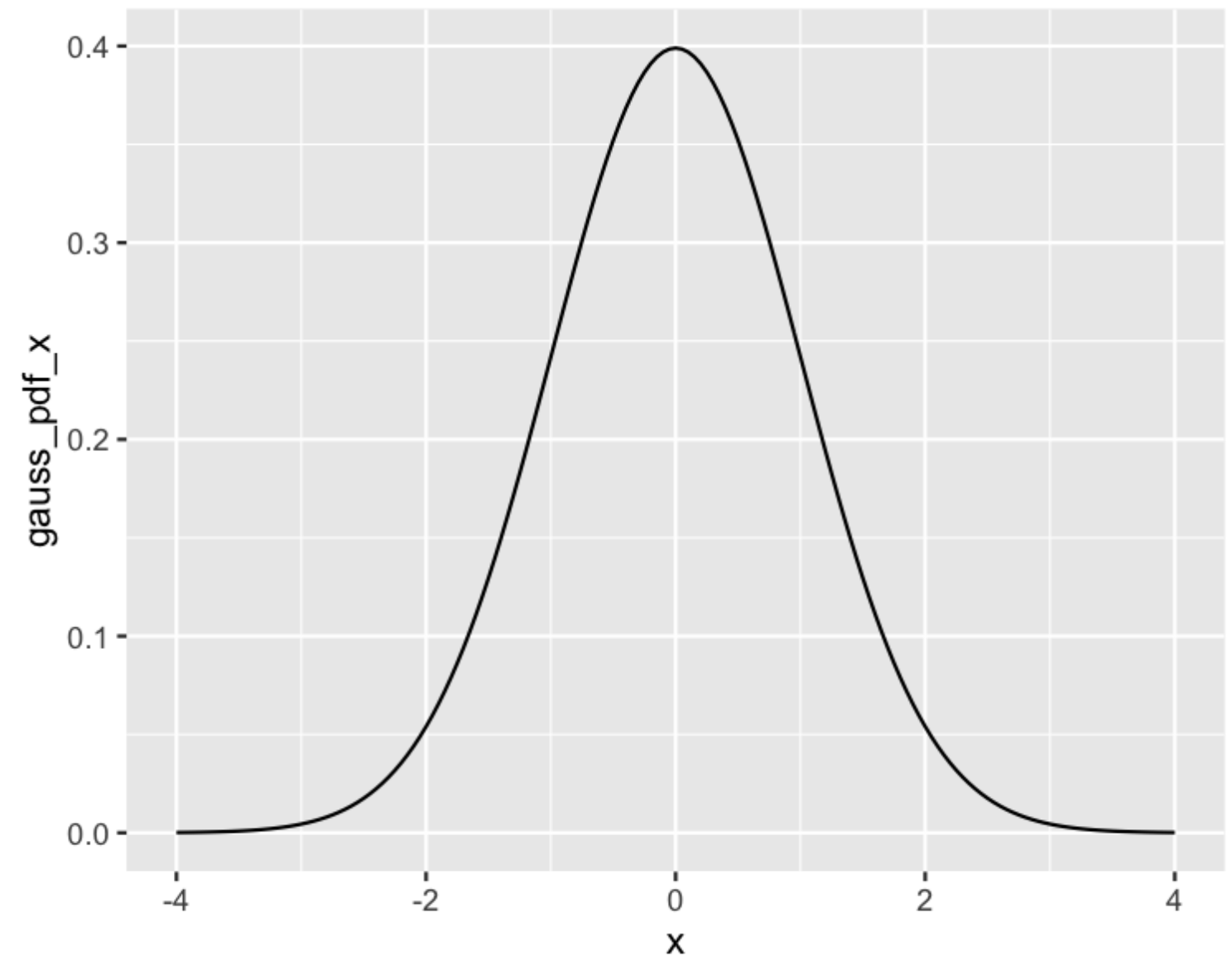
**Richie Cotton**

Learning Solutions Architect at  
DataCamp

# Gaussian probability density function (PDF)

```
gaussian_distn <- tibble(  
  x = seq(-4, 4, 0.05),  
  gauss_pdf_x = dnorm(x)  
)
```

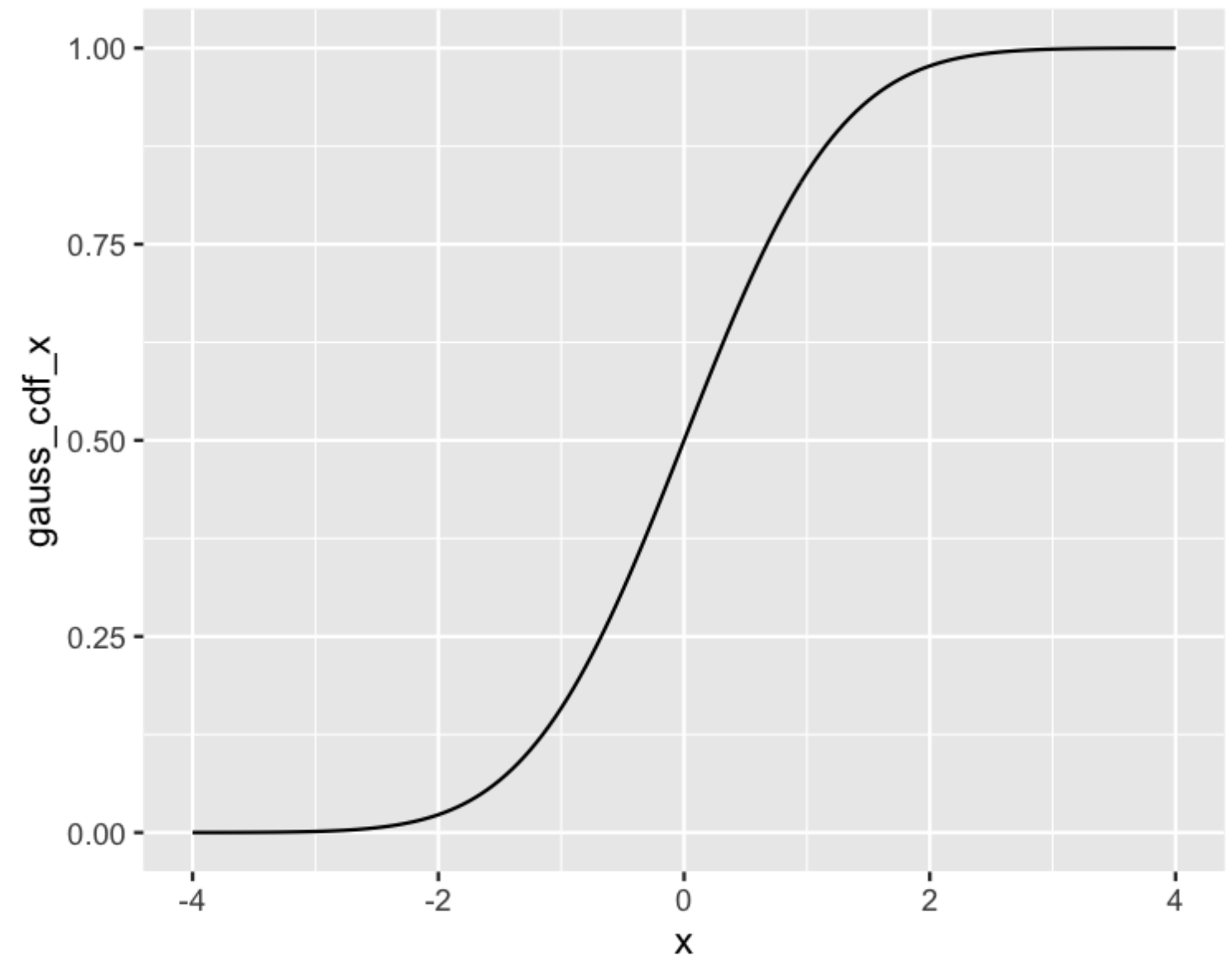
```
ggplot(gaussian_distn, aes(x, gauss_pdf_x)) +  
  geom_line()
```



# Gaussian cumulative distribution function (CDF)

```
gaussian_distn <- tibble(  
  x = seq(-4, 4, 0.05),  
  gauss_pdf_x = dnorm(x),  
  gauss_cdf_x = pnorm(x)  
)
```

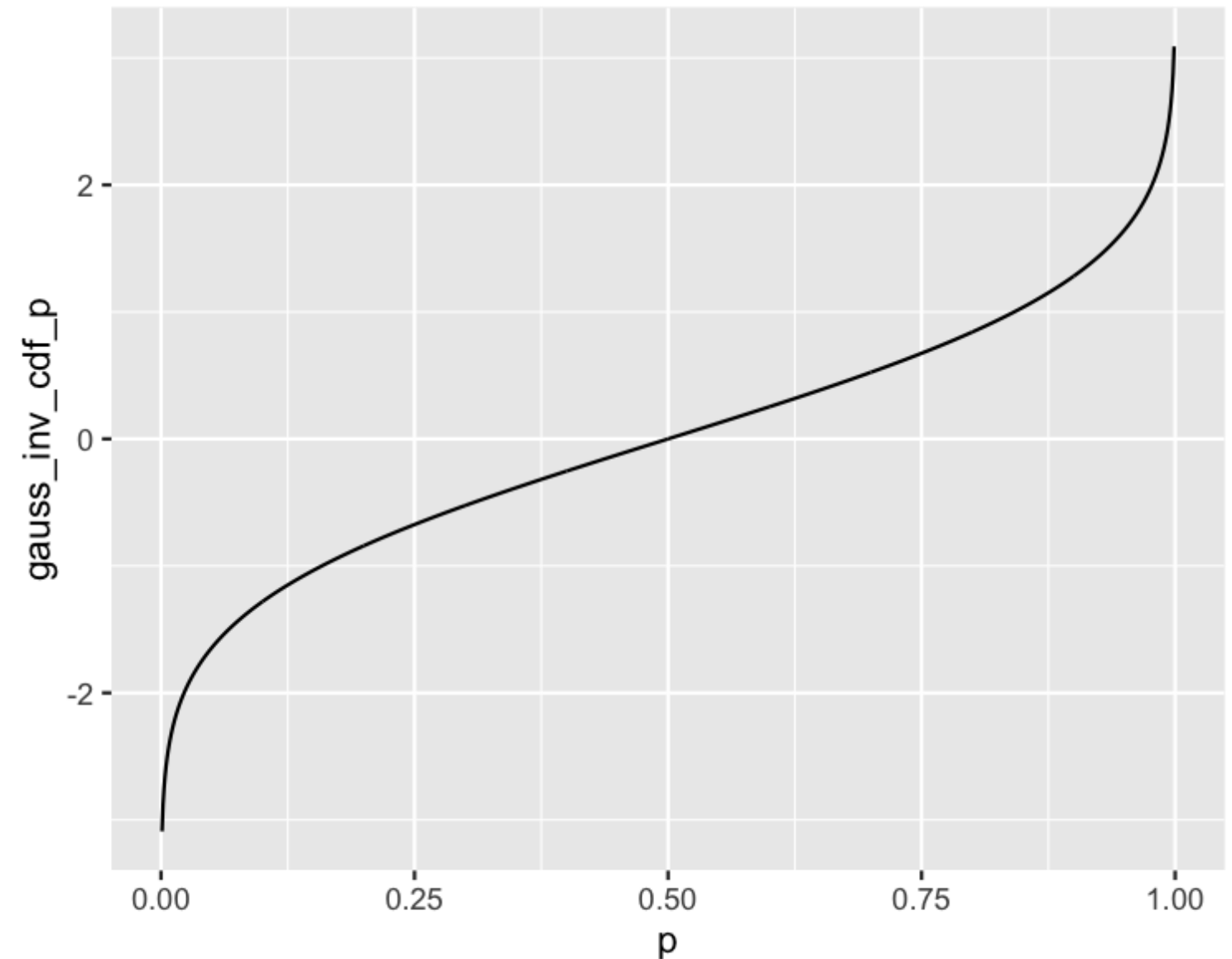
```
ggplot(gaussian_distn, aes(x, gauss_cdf_x)) +  
  geom_line()
```



# Gaussian inverse CDF

```
gaussian_distn_inv <- tibble(  
  p = seq(0.001, 0.999, 0.001),  
  gauss_inv_cdf_p = qnorm(p)  
)
```

```
ggplot(gaussian_distn_inv, aes(p, gauss_inv_cdf_p)) +  
  geom_line()
```



# Distribution function names

curve	prefix	normal	logistic	nmemonic
PDF	d	<code>dnorm()</code>	<code>dlogis()</code>	"d" for differentiate - you differentiate the CDF to get the PDF
CDF	p	<code>pnorm()</code>	<code>plogis()</code>	"p" is backwards "q" so it's the inverse of the inverse CDF
Inv. CDF	q	<code>qnorm()</code>	<code>qlogis()</code>	"q" for quantile

# glm()'s family argument

```
lm(response ~ explanatory, data = dataset)
```

```
glm(response ~ explanatory, data = dataset, family = gaussian)
```

```
glm(response ~ explanatory, data = dataset, family = binomial)
```

<sup>1</sup> <https://campus.datacamp.com/courses/introduction-to-regression-in-r/simple-logistic-regression?ex=1>

# gaussian()

```
str(gaussian())
```

```
List of 11
```

```
$ family      : chr "gaussian"
```

```
$ link        : chr "identity"
```

```
$ linkfun     :function (mu)
```

```
$ linkinv     :function (eta)
```

```
$ variance    :function (mu)
```

```
$ dev.resids: function (y, mu, wt)
```

```
$ aic         :function (y, n, mu, wt, dev)
```

```
$ mu.eta      :function (eta)
```

```
$ initialize:  expression({ n <- rep.int(1, nobs) if (is.null(etastart) && is.null(start) &&  
    is.null(mustart) && ((family$lin| __truncated__
```

```
$ validmu     :function (mu)
```

```
$ valideta    :function (eta)
```

```
- attr(*, "class")= chr "family"
```

# linkfun and linkinv

*Link function* is a transformation of the response variable

```
gaussian()$linkfun
```

```
function (mu)
mu
```

```
gaussian()$linkinv
```

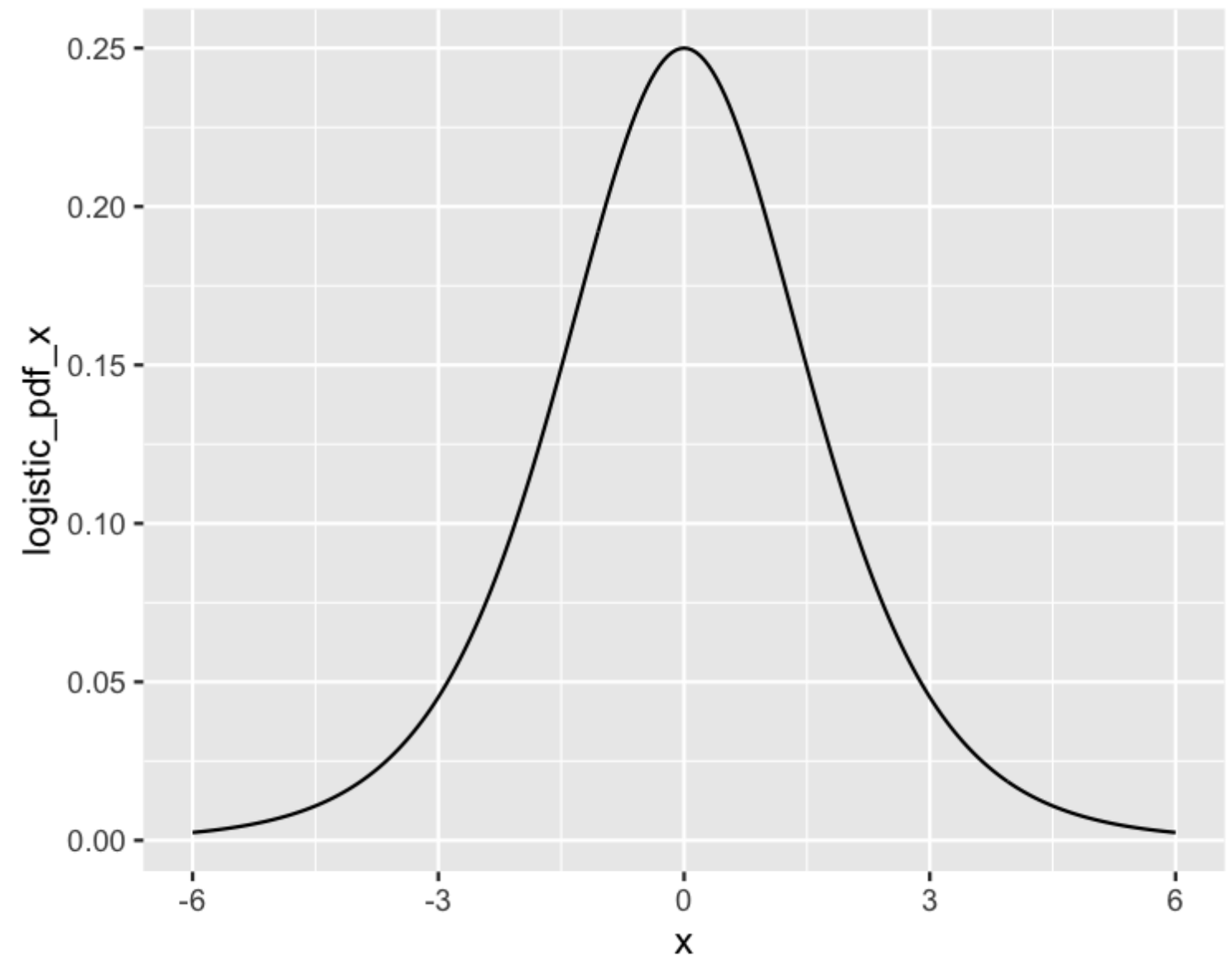
```
function (eta)
eta
```



# Logistic PDF

```
logistic_distn <- tibble(  
  x = seq(-6, 6, 0.05),  
  logistic_pdf_x = dlogis(x)  
)
```

```
ggplot(logistic_distn, aes(x, logistic_pdf_x)) +  
  geom_line()
```



# Logistic distribution

- Logistic distribution CDF is also called the *logistic function*.
- $\text{cdf}(x) = \frac{1}{(1+\exp(-x))}$
- Logistic distribution inverse CDF is also called the *logit function*.
- $\text{inverse\_cdf}(p) = \log\left(\frac{1}{(1-p)}\right)$

# Let's practice!

INTERMEDIATE REGRESSION IN R

# How logistic regression works

INTERMEDIATE REGRESSION IN R



**Richie Cotton**

Learning Solutions Architect at  
DataCamp

# Sum of squares doesn't work

```
sum((y_pred - y_actual) ^ 2)
```

`y_actual` is always `0` or `1`.

`y_pred` is between `0` and `1`.

There is a better metric than sum of squares.

# Likelihood

```
y_pred * y_actual
```

# Likelihood

```
y_pred * y_actual + (1 - y_pred) * (1 - y_actual)
```

# Likelihood

```
sum(y_pred * y_actual + (1 - y_pred) * (1 - y_actual))
```

When `y_actual = 1`

$$y_{\text{pred}} * 1 + (1 - y_{\text{pred}}) * (1 - 1) = y_{\text{pred}}$$

When `y_actual = 0`

$$y_{\text{pred}} * 0 + (1 - y_{\text{pred}}) * (1 - 0) = 1 - y_{\text{pred}}$$



# Log-likelihood

- Computing likelihood involves adding many very small numbers, leading to numerical error.
- Log-likelihood is easier to compute.

```
log(y_pred) * y_actual + log(1 - y_pred) * (1 - y_actual)
```

Both equations give the same answer.

# Negative log-likelihood

Maximizing log-likelihood is the same as minimizing negative log-likelihood.

```
-sum(log_likelihoods)
```

# Logistic regression algorithm

```
calc_neg_log_likelihood <- function(coeffs) {  
  intercept <- coeffs[1]  
  slope <- coeffs[2]  
  # More calculation!  
}
```

```
optim(  
  par = ???,  
  fn = ???  
)
```

**Let's practice!**  
INTERMEDIATE REGRESSION IN R

# Congratulations

INTERMEDIATE REGRESSION IN R



**Richie Cotton**

Learning Solutions Architect

# You learned things

## Chapter 1

- Fit/visualize/predict/assess parallel slopes

## Chapter 3

- Extend to many explanatory variables
- Implement linear regression algorithm

## Chapter 2

- Interactions between explanatory variables
- Simpson's Paradox

## Chapter 4

- Logistic regression with multiple explanatory variables
- Logistic distribution
- Implement logistic regression algorithm

# There is more to learn

- Training and testing sets
- Cross validation
- P-values and significance

# Advanced regression

- Modeling with Data in the Tidyverse
- Generalized Linear Models in R
- Machine Learning with caret in R
- Bayesian Regression Modeling with rstanarm



# Let's practice!

INTERMEDIATE REGRESSION IN R