Statistics for Linguistics 2021 06 08

Adam Tallman

06/06/2021

```
knitr::opts_chunk$set(echo = TRUE)
```

Logistic regression

Bernouilli distribution

Bernouilli distribuion is a discrete distribution with two possible outcomes. You can simulate a bernouilli distribution with rbinom(). For our purposes it will be useful to think of the Bernouilli distribution as being composed of 0s and 12.

```
data <- rbinom(20, 1,.5)
data
```

```
##  [1] 1 0 0 1 0 1 0 0 1 1 1 1 1 1 1 0 1 0 1 0
```

What we did was that we created data following a probability density function that looks as follows, where p is 0.5.

$$P(y) = \begin{cases} 1-p & \text{for } y = 0 \\ p & \text{for } y = 1 \end{cases}$$

This formula can also be written as.

$$P(y) = p^y(1-p)^{1-y}$$

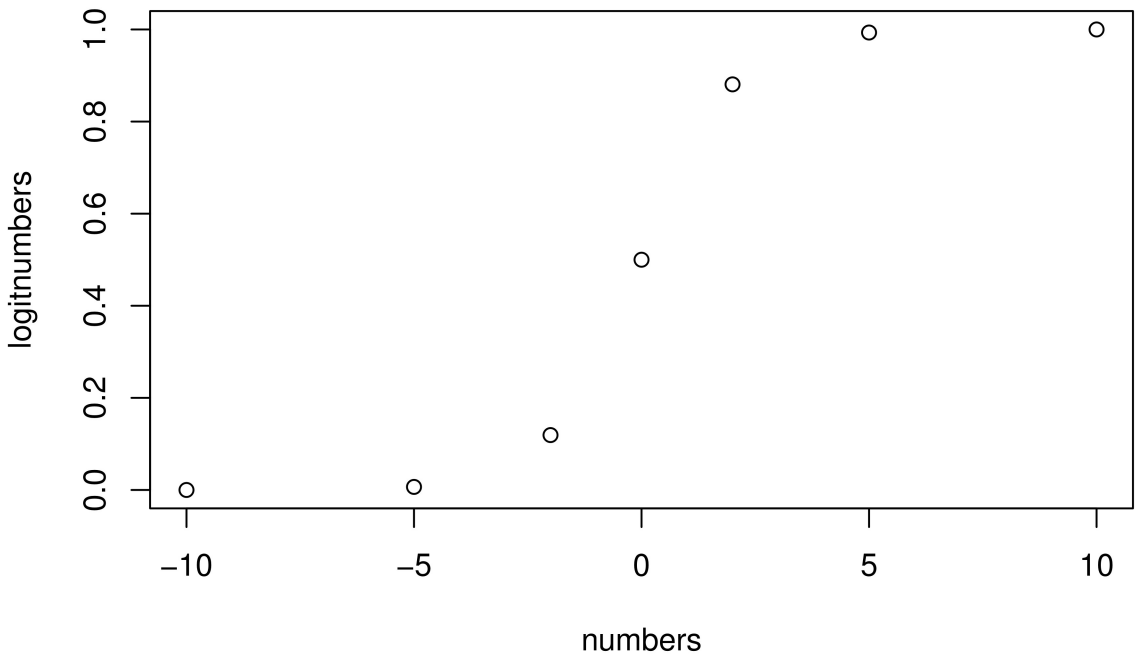
```
data <- rbinom(20, 1,.5)
p <- sum(data)/20

dataodds <- 1 / 1 - p
```

Logistic regression is about predicting the likelihood of a binary outcome,. or predicting some Bernouilli patterned distribution with some factors. Since the outcome is binary rather than a fit line, its best to model the relationship with a sigmoid, which basically means an S-shaped curve. Here's the logistic function. This function will get you numbers that have a ceiling effects like a sigmoid.

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

To understanding why having a function that gives ceiling effects (as x larger and largey, the change in y gets smaller and smaller), it will be useful first, the play around with the formula and then to consider a worked example.

```
numbers <- c(-10, -5, -2, 0,2, 5,  10)
logitnumbers <- 1/(1+exp(-(numbers)))
plot(logitnumbers~numbers)
```



Multilevel modelling

Here's a normal regression model.

$$y = \alpha + \beta x + \epsilon$$

We can put {i} in it to reflect the fact that our model makes predictions about specific data points. For a given x you get a specific y with a specific error. Everything with i as a subscript is a random variable. The coefficients α and β don't have this property, they are just constants.

$$y_i = \alpha + \beta x_i + \epsilon_i$$

But let's say you are running the model repeatedly over different groups. Do you expect the coefficients to be the same or should those also be thought of us random variables with a mean and standard deviation.

Here's a model where the intercept varies by group 'j'.

$$y_i = \alpha_{j[i]} + \beta x_i + \epsilon_i$$

Here's a model where the slope varies by group 'j'.

$$y_i = \alpha + \beta_{j[i]} x_i + \epsilon_i$$

Here's a model where both the intercept and the slope vary by group 'j'.

$$y_i = \alpha_{j[i]} + \beta_{j[i]} x_i + \epsilon_i$$

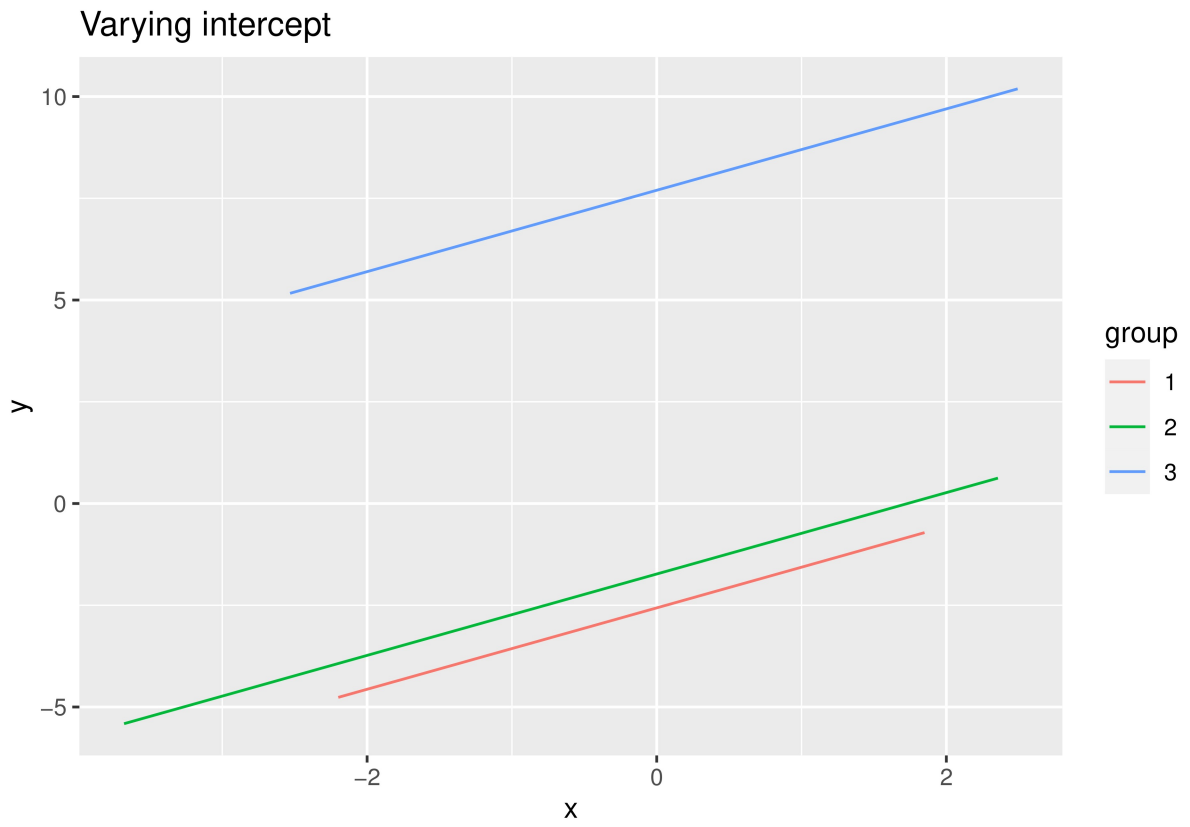
```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
library(ggExtra)
```

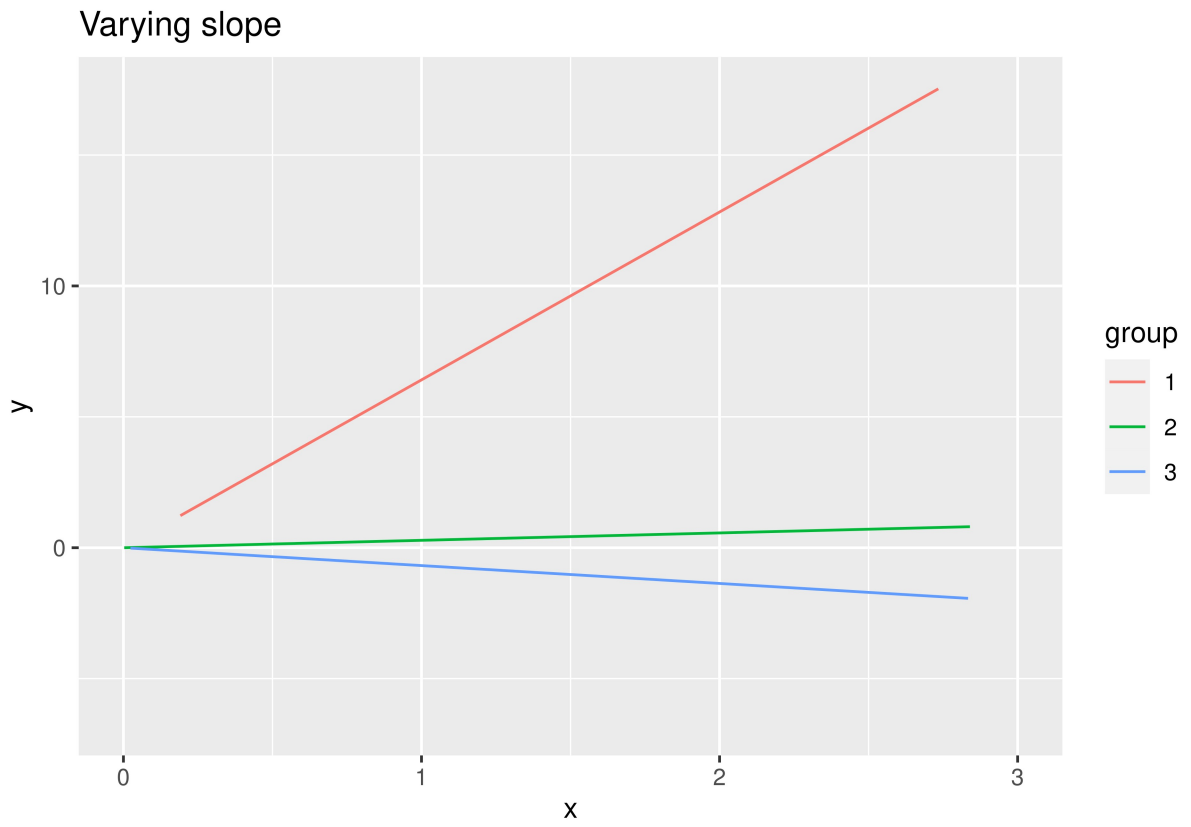
```
## Warning: package 'ggExtra' was built under R version 4.0.5
```

```
a1 <- rnorm(1, 0, 3)
a2 <- rnorm(1, 0, 3)
a3 <- rnorm(1, 0, 3)
b <- 1
x1 <- rnorm(100)
y1 <- a1 + b*x1
x2 <- rnorm(100)
y2 <- a2 + b*x2
x3 <- rnorm(100)
y3 <- a3 + b*x3

d1 <- data.frame(x = x1 ,y = y1, group = "1")
d2 <- data.frame(x = x2,y = y2, group = "2")
d3 <- data.frame(x = x3,y = y3, group = "3")

d <- rbind(d1,d2, d3)
varying.a <- ggplot(d,aes(y = y,x = x,color = group)) +
  geom_line()+
  ggtitle("Varying intercept")
varying.a
```



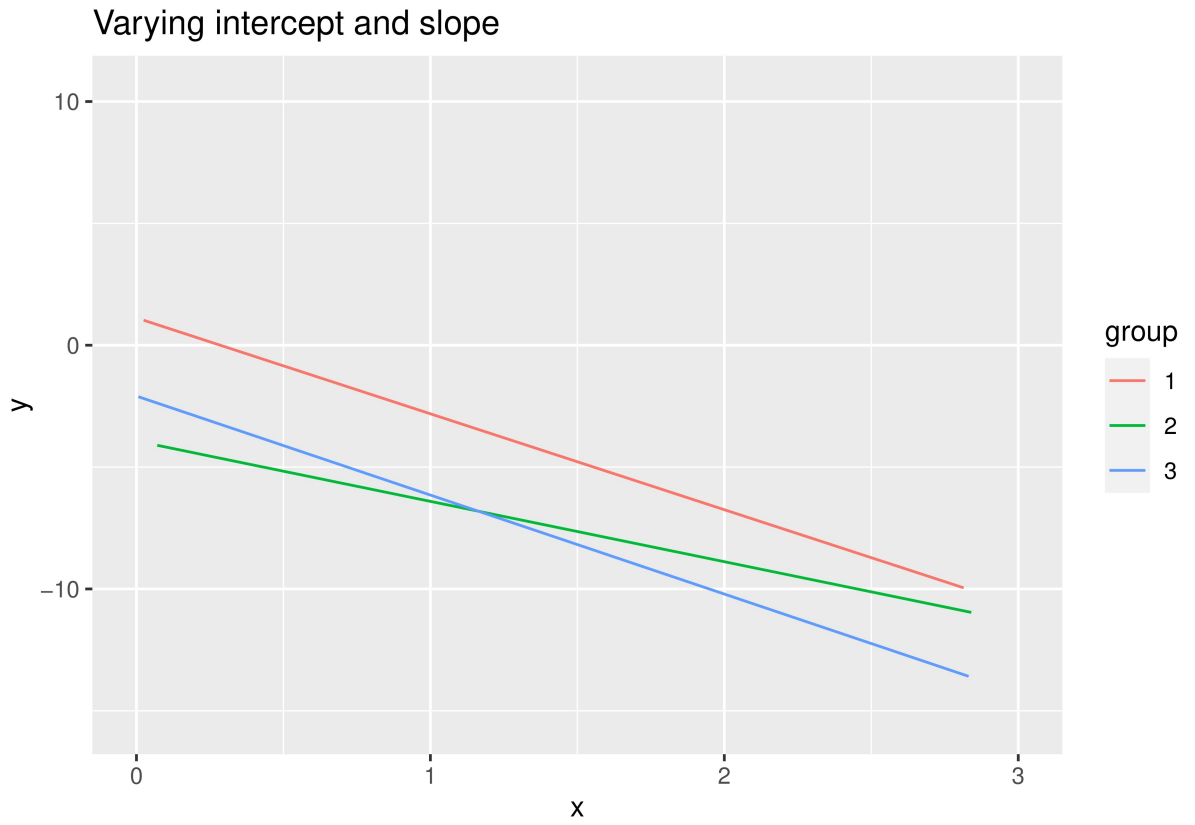

```
a=0
b1 <- rnorm(1, 0, 3)
b2 <- rnorm(1, 0, 3)
b3 <- rnorm(1, 0, 3)

x1 <- rnorm(100, 1)
y1 <- a+b1*x1
x2 <- rnorm(100, 1)
y2 <- a+b2*x2
x3 <- rnorm(100, 1)
y3 <- a+b3*x3

d1 <- data.frame(x = x1 ,y = y1, group = "1")
d2 <- data.frame(x = x2,y = y2, group = "2")
d3 <- data.frame(x = x3,y = y3, group = "3")

d <- rbind(d1,d2, d3)
varying.b <- ggplot(d,aes(y = y,x = x,color = group)) +
  geom_line()+
  xlim(0,3)+
  ggtitle("Varying slope")

varying.b
```

```
## Warning: Removed 63 row(s) containing missing values (geom_path).
```



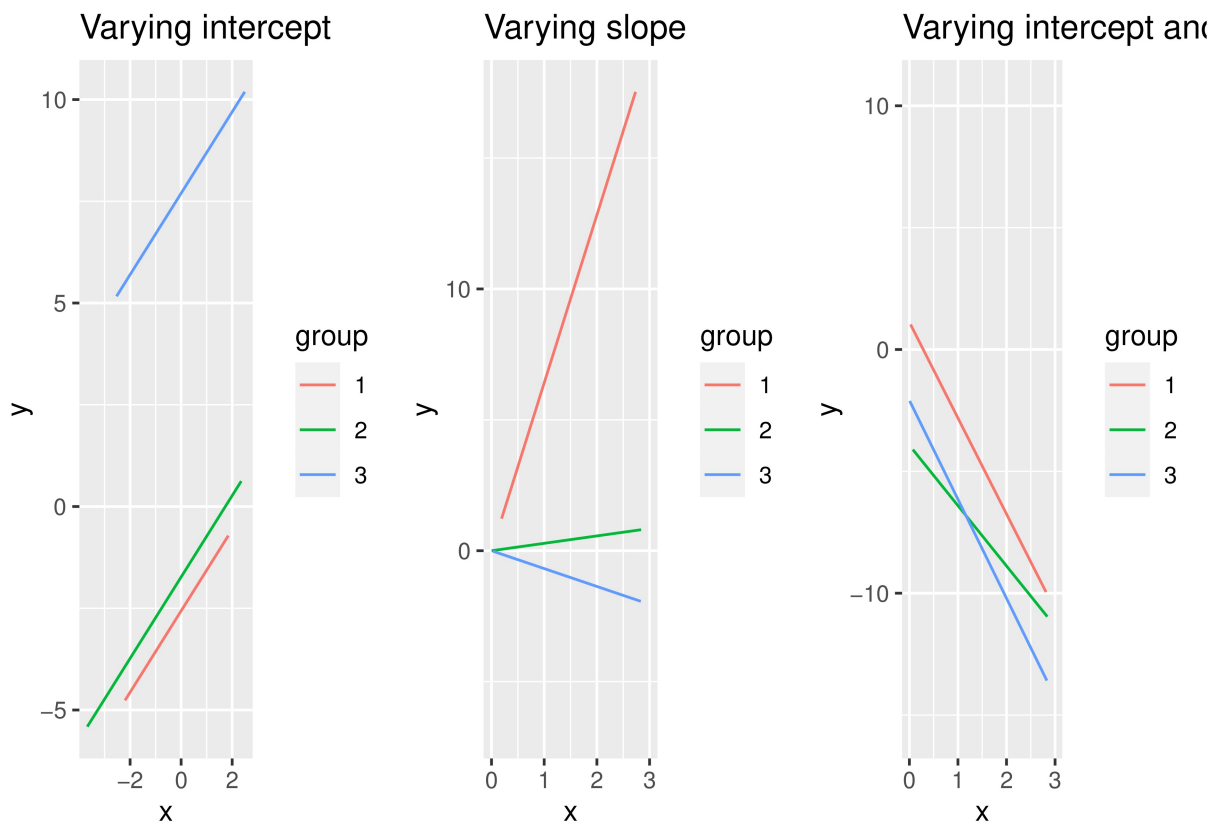

```
a1 <- rnorm(1, 0, 3)
a2 <- rnorm(1, 0, 3)
a3 <- rnorm(1, 0, 3)
b1 <- rnorm(1, 0, 3)
b2 <- rnorm(1, 0, 3)
b3 <- rnorm(1, 0, 3)

x1 <- rnorm(100, 1)
y1 <- a1+b1*x1
x2 <- rnorm(100, 1)
y2 <- a2+b2*x2
x3 <- rnorm(100, 1)
y3 <- a3+b3*x3

d1 <- data.frame(x = x1 ,y = y1, group = "1")
d2 <- data.frame(x = x2,y = y2, group = "2")
d3 <- data.frame(x = x3,y = y3, group = "3")

d <- rbind(d1,d2, d3)
varying.ab <- ggplot(d,aes(y = y,x = x,color = group)) +
  geom_line()+
  xlim(0,3)+
  ggtitle("Varying intercept and slope")

varying.ab
```

```
## Warning: Removed 64 row(s) containing missing values (geom_path).
```




```
library(gridExtra)

grid.arrange(varying.a,varying.b,varying.ab, nrow=1)
```

```
## Warning: Removed 63 row(s) containing missing values (geom_path).
## Warning: Removed 64 row(s) containing missing values (geom_path).
```


Varying intercept
Varying slope
Varying intercept an
y
x
group
1
2
3
10
5
0
−5
−2
0
2
1
3
−10