Statistics for linguists

2023-12-06

Chi-squared test, loglinear models

From last class

- Linear models
- Errors and residuals
- Sum of squares
- ANOVA

For this class

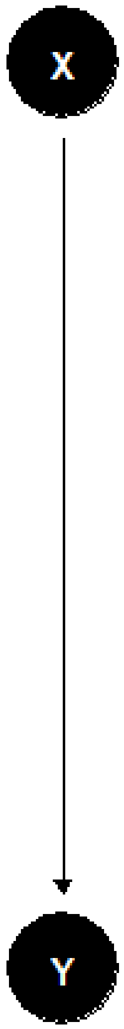
- Directed acyclic graphs (introduction)
- Chi-squared test
- Loglinear model / logistic regression
- Some data wrangling (splitstr etc.)

Packages for today

```
library(tidyverse)
library(ggdag)
library(V8)
library(dagitty)
library(glm2)
library(Rling)
library(rms)
library(visreg)
library(car)
```

Direct acyclic graph

- A directed acyclic graph represents our assumptions about causal direction.
- When you use a statistical model you imply assumptions about causation between your variables.
- So far, we have just been dealing with very simple causal relations (simple 'bivariate' models)

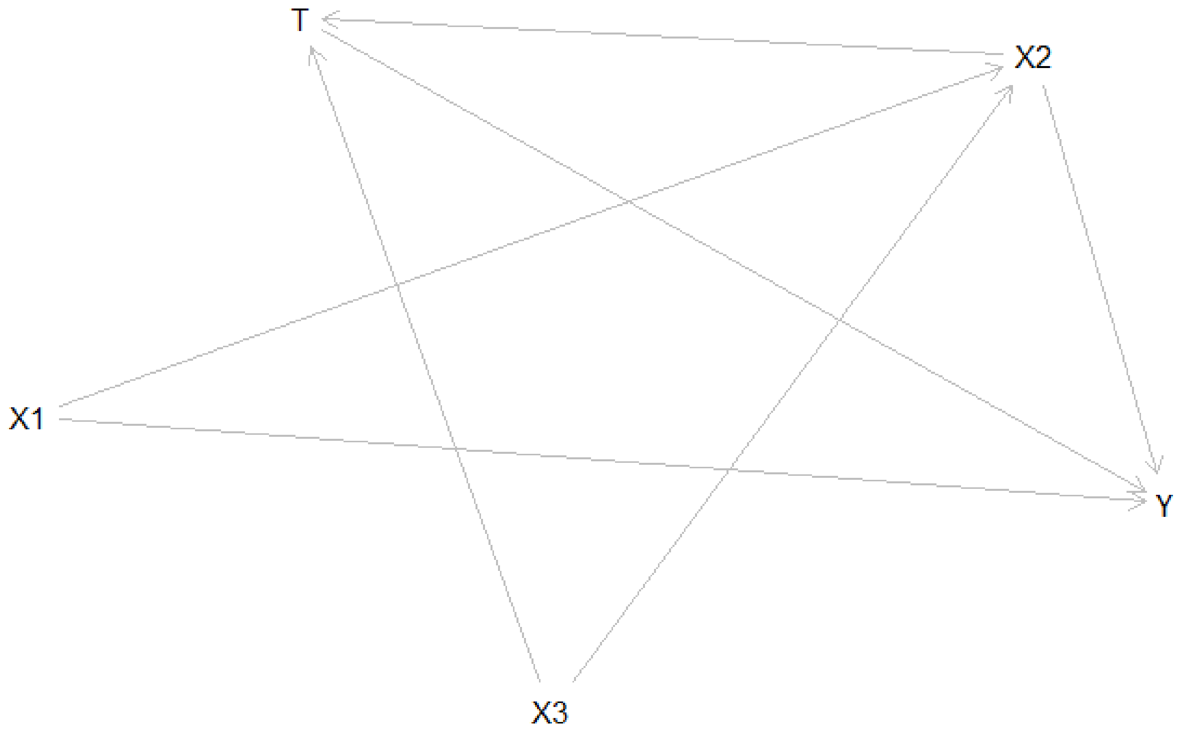


read as 'X causes Y'

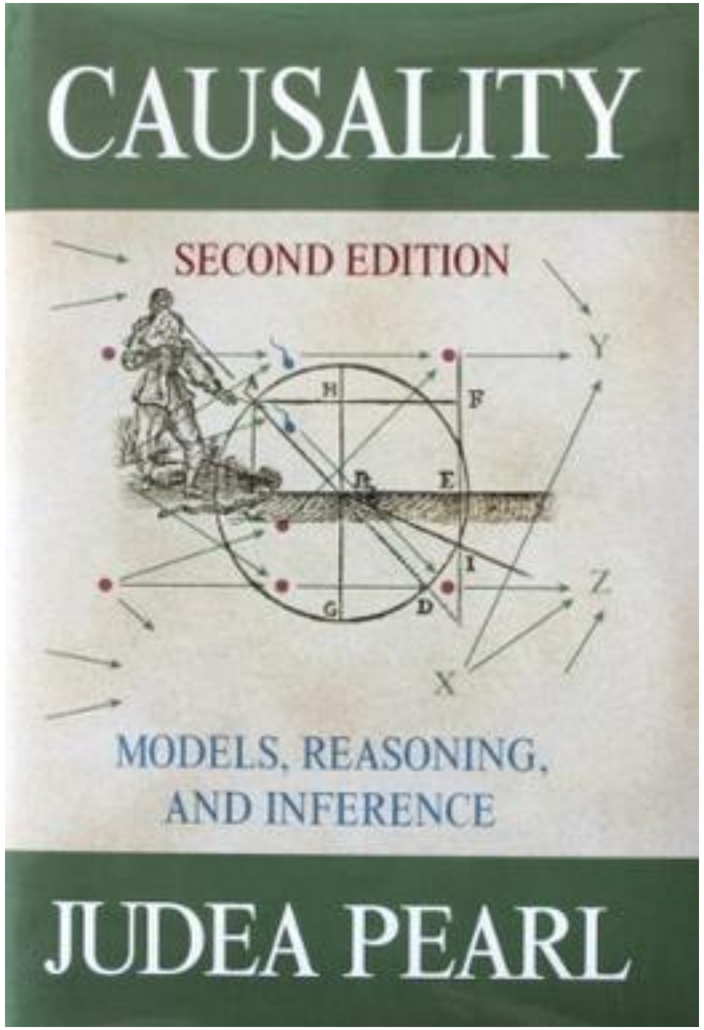
Directed acyclic graph (DAG)

- The causal relations can get very complicated though.

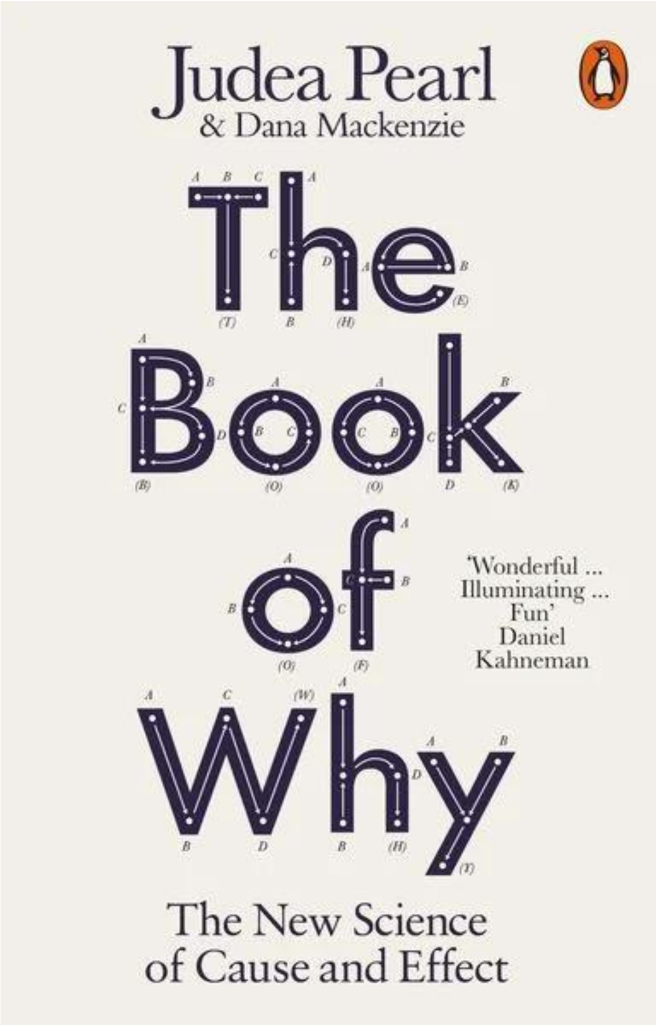
- DAGs are used for **causal inference** – if your statistical model is paired with DAG it means you can infer causation from your model**,** rather than just correlation



Pearl, Judea. 2009. *Causality: Models, Reasoning and Inference.* Cambridge: Cambridge University Press.



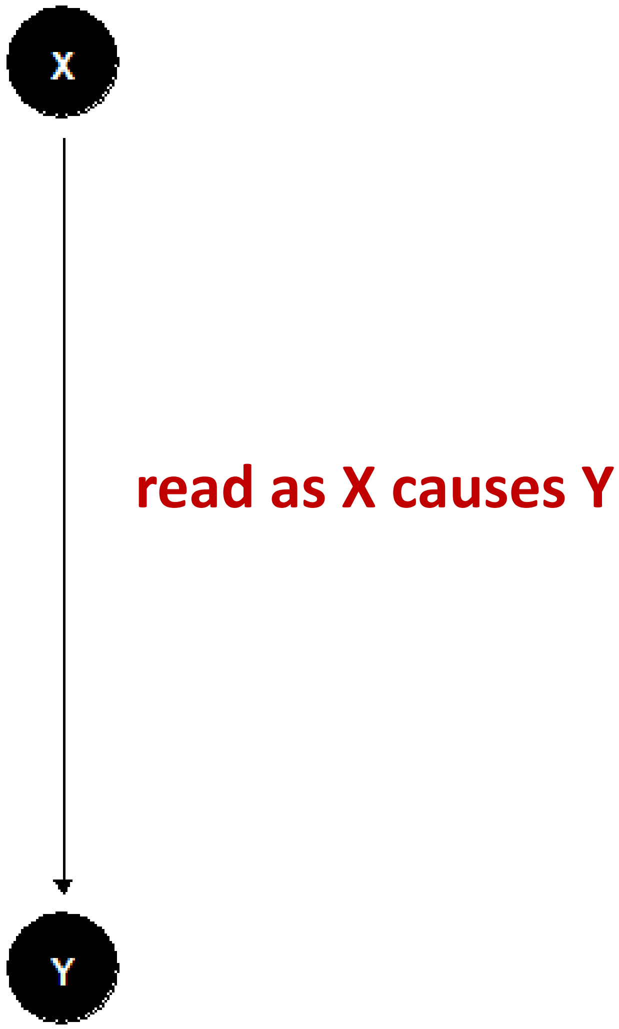
Causal inference & the Causal revolution



Pearl, Judea. 2009. *Causality: Models, Reasoning and Inference.* Cambridge University Press.
Pearl, Judea (with Dana Mackenzie). *The book of Why: The new science of cause and effect.* Penguin.

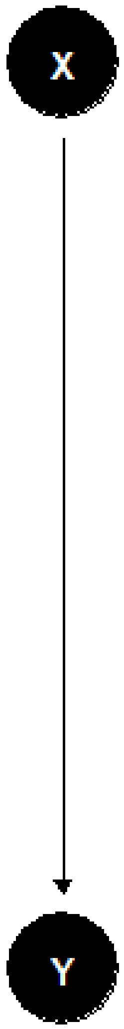
Direct acyclic graph

- Linear model / Regression
  - X = continuous
  - Y = continuous
- ANOVA
  - X is categorical
  - Y is continuous



Direct acyclic graph

- Linear model / Regression
  - X = continuous
  - Y = continuous
- ANOVA
  - X is categorical
  - Y is continuous
- Chi-squared
  - X is categorical
  - Y is continuous
- Loglinear model / Logistic regression
  - X is continuous
  - Y is categorical

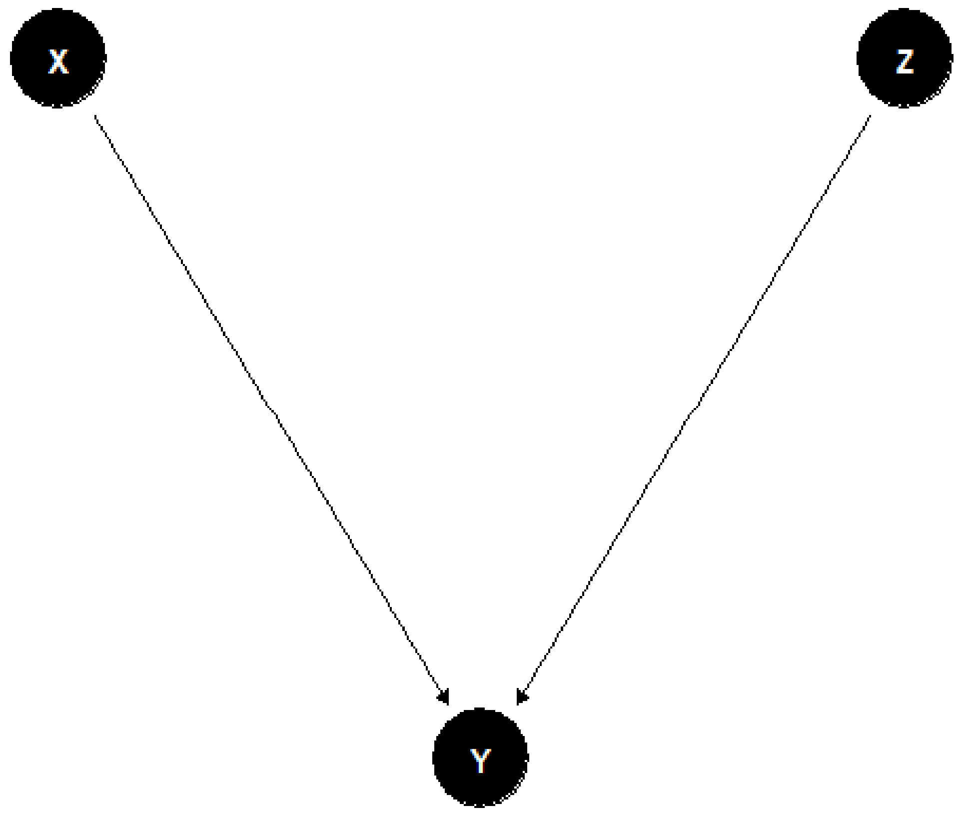


read as X causes Y

Directed acyclic graph

- Naturally there can be more than one cause

- Next week we will start looking at *multivariate* models in more detail



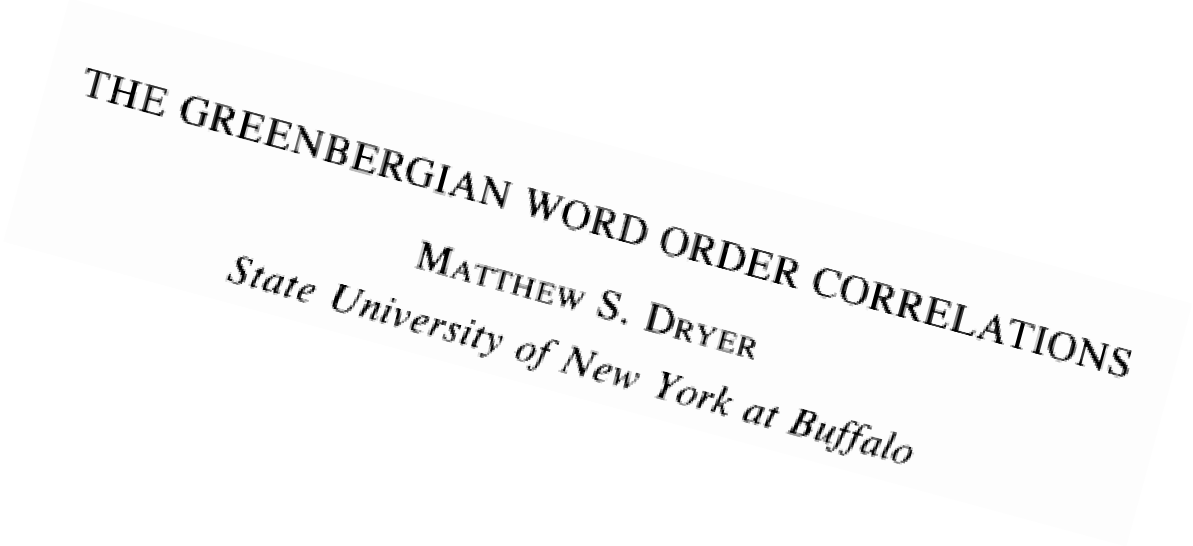
Counts & contingency tables

- A lot of statistical information comes from counts (e.g. frequency of words in different texts)
- The data are usually presented in a contingency table.
- Data from Matthew Dryer (1992)
- OV = Object-Verb words order / VO = Verb-Object word order
- Postp = positions / Prep = prepositions
- H1:postpositions are associated with OV word order and prepositions are associated with VO order

Dryer, Matthew S. 1992. The Greenbergian Word Order Correlations. *Languages* 68:1, 81-138.

Contingency table (word order associations)

	OV	VO
Postp	107	12
Prep	7	70


THE GREENBERGIAN WORD ORDER CORRELATIONS
MATTHEW S. DRYER
State University of New York at Buffalo

```
adpos <- matrix(c(107,12,7,70),ncol=2,byrow=TRUE)
rownames(adpos)<-c("PostP","Prep")
colnames(adpos)<-c("OV","VO")
adpos
```

Dryer, Matthew S. 1992. The Greenbergian Word Order Correlations. *Languages* 68:1, 81-138.

Counts and probabilities

- These are **observed frequencies**
- We now need a model that predicts the **expected frequencies**
- Using these data, what is the probability of a random language from this sample having OV?

Dryer, Matthew S. 1992. The Greenbergian Word Order Correlations. *Languages* 68:1, 81-138.

Counts and probability

```r
wordorder <- cbind(c(107, 7), c(12, 70))
rownames(wordorder) <- c("Postp", "Prep")
colnames(wordorder) <- c("OV", "VO")
wordorder <- rbind(wordorder, c(114,82))
wordorder <- cbind(wordorder, c(119,77,196))
rownames(wordorder) <- c("PostP", "Prep", "Column Total")
colnames(wordorder) <- c("OV", "VO", "Row total")
wordorder
```

Crawley, Michael J. 2015. *Statistics: An Introduction using R.* Wiley.

Expected Frequency

- Raw total (Postp = 119, Prep = 77)
- Column total (OV = 114, VO = 82)
- Grand total = 196
- Expected = (Raw total * Column total) / Grand total

Crawley, Michael J. 2015. *Statistics: An Introduction using R.* Wiley.

Dryer, Matthew S. 1992. The Greenbergian Word Order Correlations. *Languages* 68:1, 81-138.

Expected frequency

- The expected frequency refers to what the values would be if VO/OV and Postp/Prep were independent.

	OV	VO
Postp	(114*119)/196	(82*119)/196
Prep	(114*77)/196	(82*77)/196

```
E <- cbind(c((114*119)/196, (114*77)/196),
           c(82*119/196,(82*77)/196))
rownames(E) <- c("Postp", "Prep")
colnames(E) <- c("OV", "VO")
E
```

Crawley, Michael J. 2015. *Statistics: An Introduction using R.* Wiley.

Expected frequency

- What we've done is created a hypothetical "null distribution" against which we can measure how surprising our actual data are.

	OV	VO
Postp	69.21429	49.78571
Prep	44.78571	32.21429

Crawley, Michael J. 2015. *Statistics: An Introduction using R.* Wiley.

Expected frequency vs. real frequencies

- What we've done is created a hypothetical "null distribution" against which we can measure how surprising our actual data are.

	OV	VO
Postp	69.21429	49.78571
Prep	44.78571	32.21429

	OV	VO
Postp	107	12
Prep	7	70

Chi-squared test

- The classical way of doing this is Karl Pearson's chi-squared test.

$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$

https://upload.wikimedia.org/wikipedia/commons/2/21/Karl_Pearson_2.jpg

Crawley, Michael J. 2015. *Statistics: An Introduction using R.* Wiley.

```r
E <- cbind(c((114*119)/196, (114*77)/196),
           c(82*119/196,(82*77)/196))
rownames(E) <- c("Postp", "Prep")
colnames(E) <- c("OV", "VO")
E
```

$$Expected = \frac{Rowtotal * Columntotal}{Grandtotal}$$

```r
E.df <- melt(E)
colnames(E.df)<-c("Adposition", "Verb.Object",
"Expected.Frequency")
E.df$Observed.Frequency <- c(107,7,12,70)
E.df
```

Putting expected and observed data in the same data set

```r
E.df$oe <- ((E.df$Observed.Frequency - E.df$Expected.Frequency)^2) / E.df$Expected.Frequency
sum(E.df$oe)
```

$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$

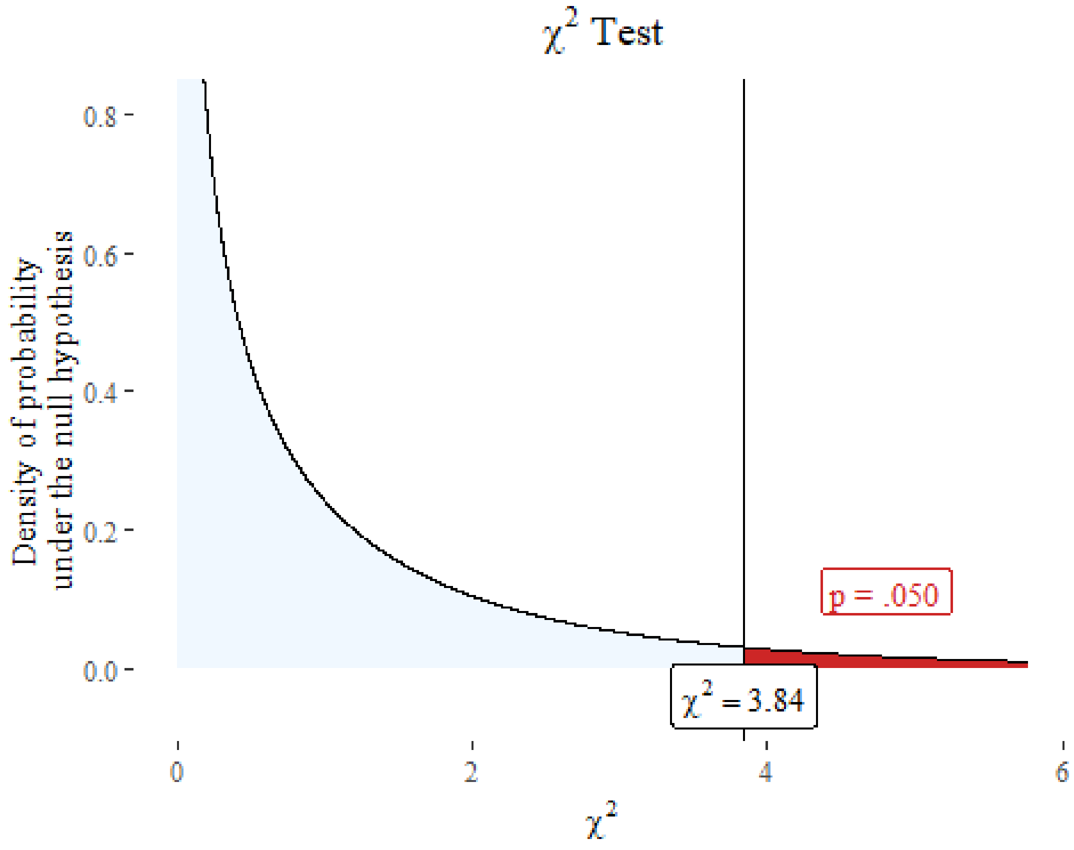
Chi-squared test and p values

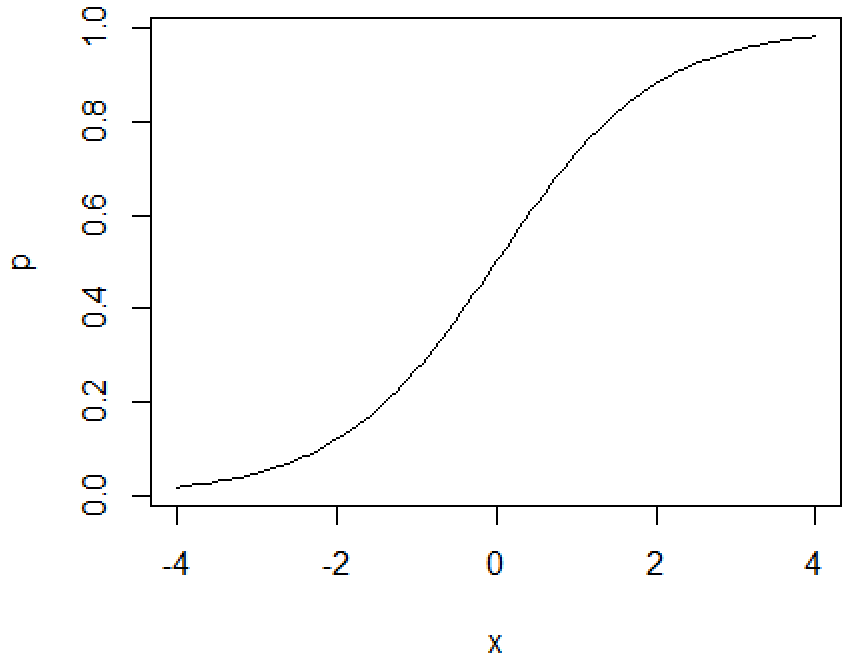
- Is this number big?
- What is the critical value of the chi-squared test?
- To calculate this, we need the degrees of freedom and the cut off area you want.
- R = number of rows
- C = number of columns

Crawley, Michael J. 2015. *Statistics: An Introduction using R.* Wiley.

Chi-squared test and p-value

- We have another hypothetical distribution based like the t distribution.





Logistic regression

Logistic regression

- The logistic regression is a method for modelling binary data.

- The basic ideas can be extended to non-binary data as long as they are organized into levels.

- It is typically used when the dependent variable is binary and there is an interest in knowing how a change in *x* effects the probability that something is *y.*

Logistic regression

- Psycholingustic experiments where subjects have to give yes or no answers.
- Various uses in natural language processing
- Predict the risk of developing a specific disease
- Predict the probability that someone will vote for a particular political party.
- ...

Field experiment

- Let’s look at data with a binary response variable.
- A **multiple forced choice** test was used with Chácobo speakers in 2022 using Praat
- Pitch and duration values of a form *janáquë* [hanákɨ] were manipulated systematically to create a matrix of duration and pitch values.
- Speakers Heard these and had to guess whether they were hearing *jánaquë* ’s/he left it’ or *janáquë* ’s/he vomited’

https://www.fon.hum.uva.nl/praat/manual/ExperimentMFC.html

Multiple forced choice experiment

- You’ll have to wrangle the data a bit.

```
mcf <- read.csv(“YourPathway/chacobo.mcf.df.csv")

response <- ifelse(mcf$response=="janáquë vomitó", 1, 0)
stimulus <- str_remove_all(mcf$stimulus, "janaquë_")
df <- as.data.frame(cbind(response, stimulus))
df[c("Pitch_Hz", "Duration_ms")] <- str_split_fixed(df$stimulus, "_", 2)
df$Pitch_Hz<-as.numeric(df$Pitch_Hz)
df$response <- as.numeric(df$response)
head(df)
```

```r
mcf <- read.csv(“YourPathway/chacobo.mcf.df.csv")
response <- ifelse(mcf$response=="janáquë vomitó", 1, 0)

stimulus <- str_remove_all(mcf$stimulus, "janaquë_")
df <- as.data.frame(cbind(response, stimulus))
df[c("Pitch_Hz", "Duration_ms")] <- str_split_fixed(df$stimulus, "_", 2)

df$Pitch_Hz<-as.numeric(df$Pitch_Hz)
df$response <- as.numeric(df$response)

head(df)
```

Change values so that they are 0 or 1

Extract the pitch and duration values from the stimuli

Make the values numeric

Linear model

- You could run a normal linear model on these data once the response variable is coded as 0 or 1.

- It is not very realistic or well-fit – but let‘s illustrate this anyways.

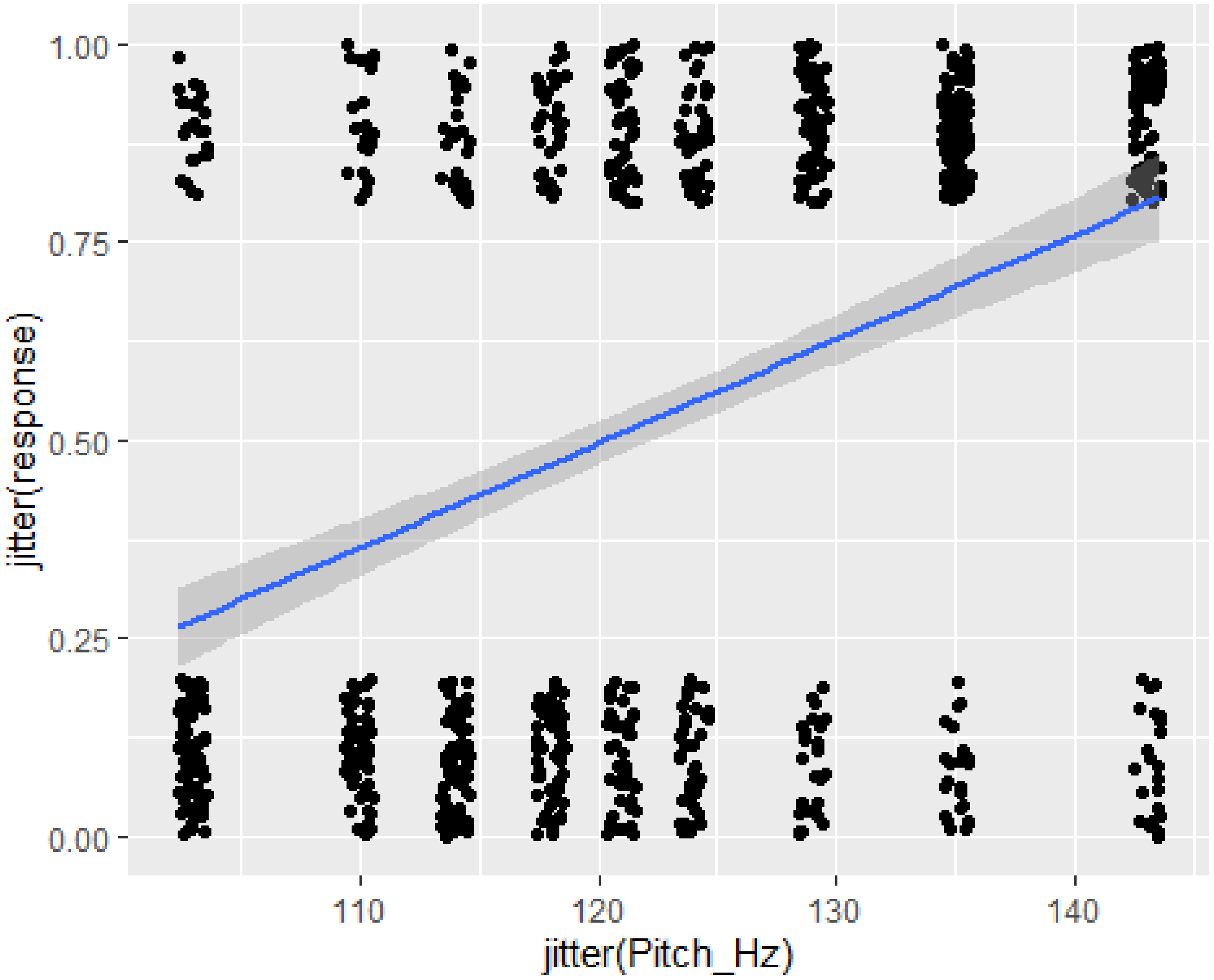
```
ggplot(df, aes(jitter(Pitch_Hz), jitter(response))) +
  geom_point()+
  stat_smooth(method="lm",
              formula = y ~ x,
              geom= "smooth")+
  ylim(0,1)
```

Note that we use *jitter()* when lots of data points are overlapping so we can see the distribution.

Normal linear model

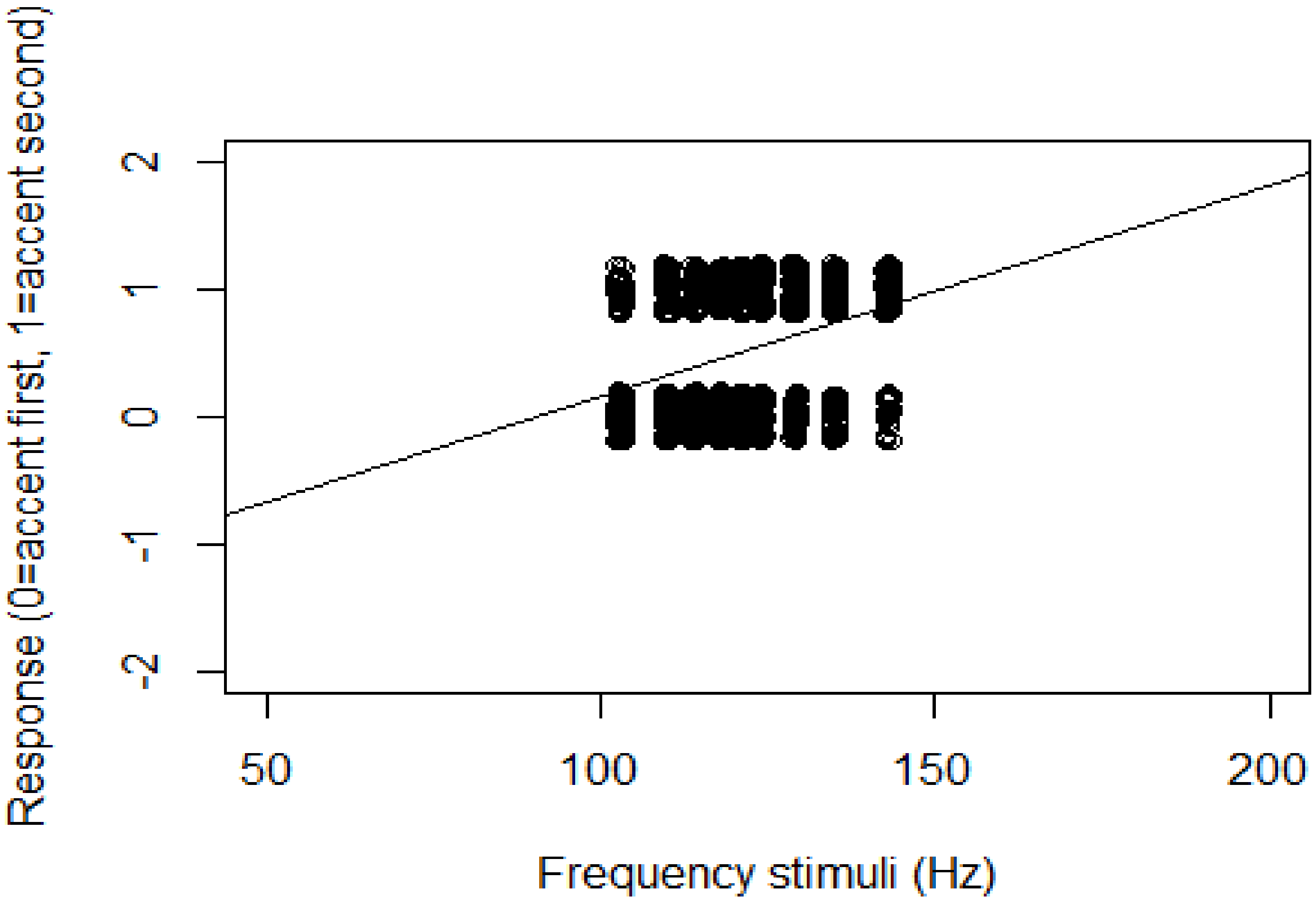
- You can see that as the pitch increases on the second vowel, the probability that ***janáquë 'vomit' = 1*** is chosen over ***jánaquë 'leave' = 0*** increases.

- But the linear model makes predictions way beyond the structure of the data.



```r
plot(jitter(df$response)~jitter(df$Pitch_Hz), ylab="Response (0=accent first, 1=accent second)", xlab ="Frequency stimuli (Hz)", xlim=c(50,200), ylim=c(-2,2)) +

abline(lm(response ~ Pitch_Hz, data = df))
```



Asymptotic function

- To model probability of one answer over another in relation to some data point *x,* we want a line that has **asymptotic** properties

- **Asymptotic function:** A function which gets closer and closer to a value or condition but never reaches it.

- Why? The difference between 125 and 150 Hz should be huge in terms of predicting the response but the difference between 150 and 175 Hz should be smaller.

Logistic regression

- A logistic regression or logit model can be represented with the following equations ....

$$logit(y) = \alpha + \beta_1 x_1 + \beta_2 x_2 \; \ldots + e$$

$$logit(p) = \log \frac{p}{1-p}$$

$$Prob\{y = 1|x\} = \frac{1}{1 + \exp(-x\beta)}$$

Logistic regression

- A logistic regression or logit model can be represented with the following equations ....

$$\boldsymbol{logit(y) = \alpha + \beta_1 x_1 + \beta_2 x_2 \;\ldots + e}$$

The intercept plus the coefficient $\boldsymbol{\beta}$ times *x* (plus error etc.) is equal to *y* output from a logit function

$$logit(p) = \log\frac{p}{1-p}$$

$$Prob\{y = 1|x\} = \frac{1}{1 + \exp(-x\beta)}$$

Logistic regression

- A logistic regression or logit model can be represented with the following equations ....

$$logit(y) = \alpha + \beta_1 x_1 + \beta_2 x_2 \; \ldots + e$$

$$\boldsymbol{logit(p) = \log \frac{p}{1-p}}$$

The logit function is the log of a probability divided by 1 minus that probability – the "log odds"

$$Prob\{y = 1|x\} = \frac{1}{1 + \exp(-x\beta)}$$

Logistic regression

- A logistic regression or logit model can be represented with the following equations ....

$$logit(y) = \alpha + \beta_1 x_1 + \beta_2 x_2 \ \ldots + e$$

$$logit(p) = \log \frac{p}{1-p}$$

$$\boldsymbol{Prob\{y = 1|x\}} = \frac{\mathbf{1}}{\mathbf{1 + \exp(-x\beta)}}$$

So to get the probability that y is 1 given x, you divide the loglinear variables by an equation that reverses the logit function.

Logistic regression

- A logistic regression or logit model can be represented with the following equations ....

$$logit(y) = \alpha + \beta_1 x_1 + \beta_2 x_2 \; \ldots + e$$

Just algebra

$$logit(p) = \log \frac{p}{1-p}$$

$$\boldsymbol{Prob\{y = 1 | x\}} = \frac{\mathbf{1}}{\mathbf{1 + \exp(-x\beta)}}$$

So to get the probability that y is 1 given x, you divide the loglinear variables by an equation that reverses the logit function.

Bernouilli distribution

- A Bernouilli distribution is a **discrete distribution** with **two possible outcomes**.
- We want to account for the distribution of *y.*
- You can model it with rbinom()

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

By Niklaus Bernoulli (1662-1716) - [2] [3], Public Domain, https://commons.wikimedia.org/w/index.php?curid=266673

Bernouilli distribution

- A Bernouilli distribution is a discrete distribution with two possible outcomes.
- We want to account for the distribution of *y.*
- You can model it with rbinom()

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

‘The probability of y being 1 is p and the probability of y being 0 is p – 1’

Bernouilli distribution

- A logistic regression or logit model can be represented with the following equations ....

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

```
bernouilli_data <- rbinom(20, 1, 0.5)
print(bernouilli_data)

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

Logistic function

$$logit(p) = \log \frac{p}{1-p}$$

- The important point about log odds ratios as that they take any numbers and transform them to a number from 0 to 1 along a sigmoid shape.

- Why is this good?

- Because we are interested in modelling a binary outcome,0 or 1, and we want to have a function that translates the effect of predictors into that scale for *y.*

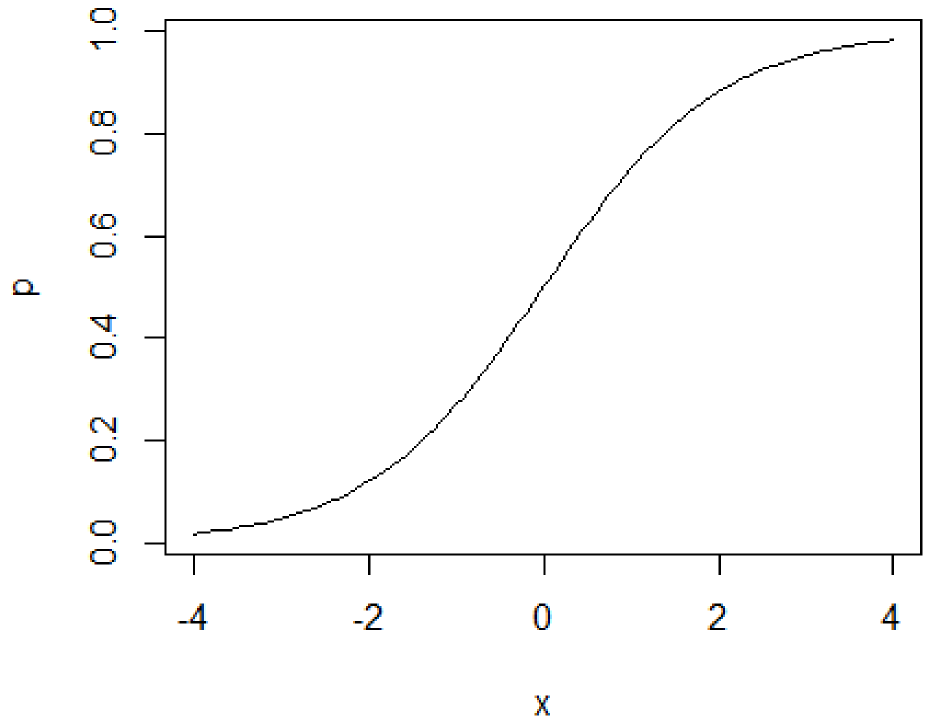
Logistic function

- Any value of *x* will be transformed into a number that varies from 0 to 1 with ceiling effects.

$$Prob\{y = 1|x\} = \frac{1}{1 + \exp(-x\beta)}$$

Logistic regression

- *y* is bounded to 0 or 1
- The relationship between x and y has a ceiling effect (like logarithms)
- Let’s run a simulation model to get the feel for it.



Functions

- There are two functions in R

- The logarithmic function and the exponential function – they are basically mirror-images of each other.

```r
t <- log(10)
t

## [1] 2.302585

exp(t)

## [1] 10
```

- The *odds* function is pretty straightforward

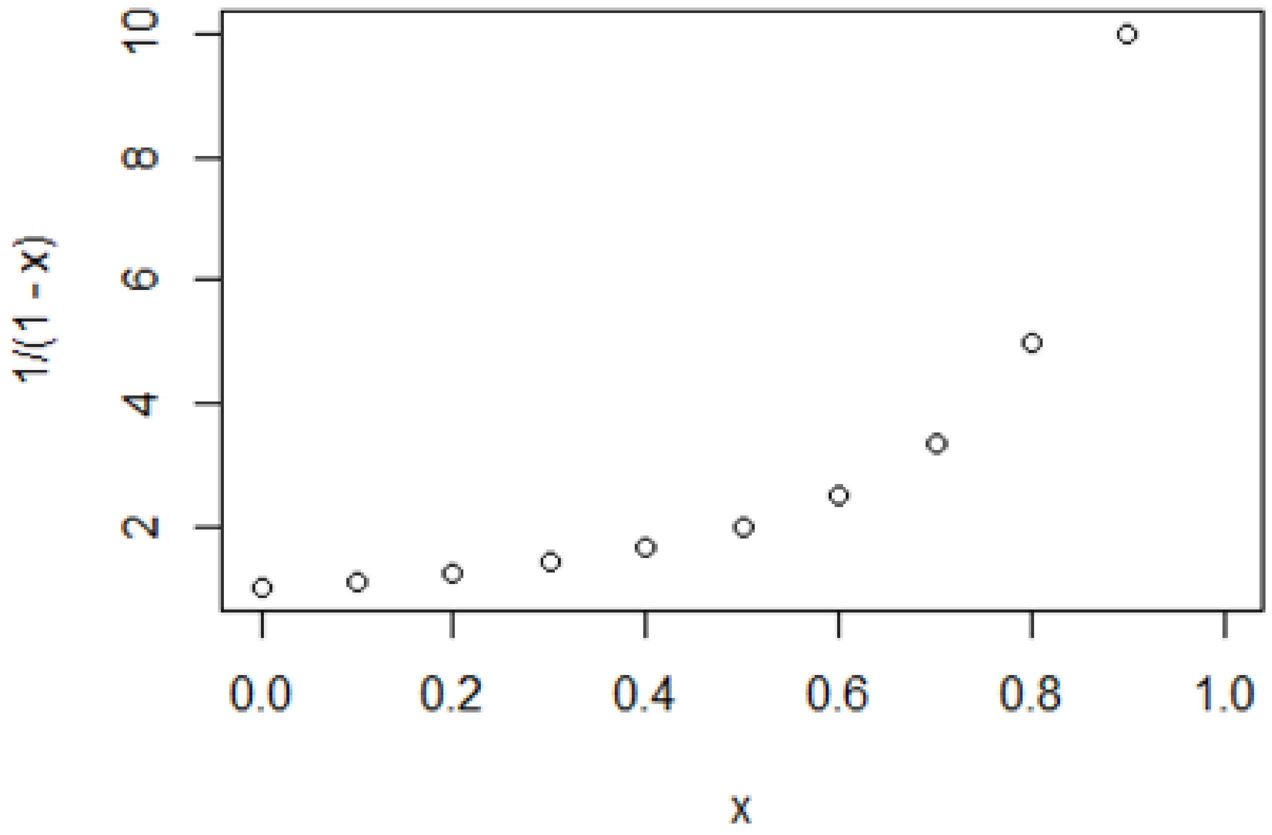
$$\frac{1}{1-p}$$

Odds

- If you plot probabilities versus odds, you can see that odds has an asymptotic property in relation to probability

```
par(mfrow=c(1,1))
x <- seq(0,1,by=0.1)
plot(x, 1/(1-x))
```

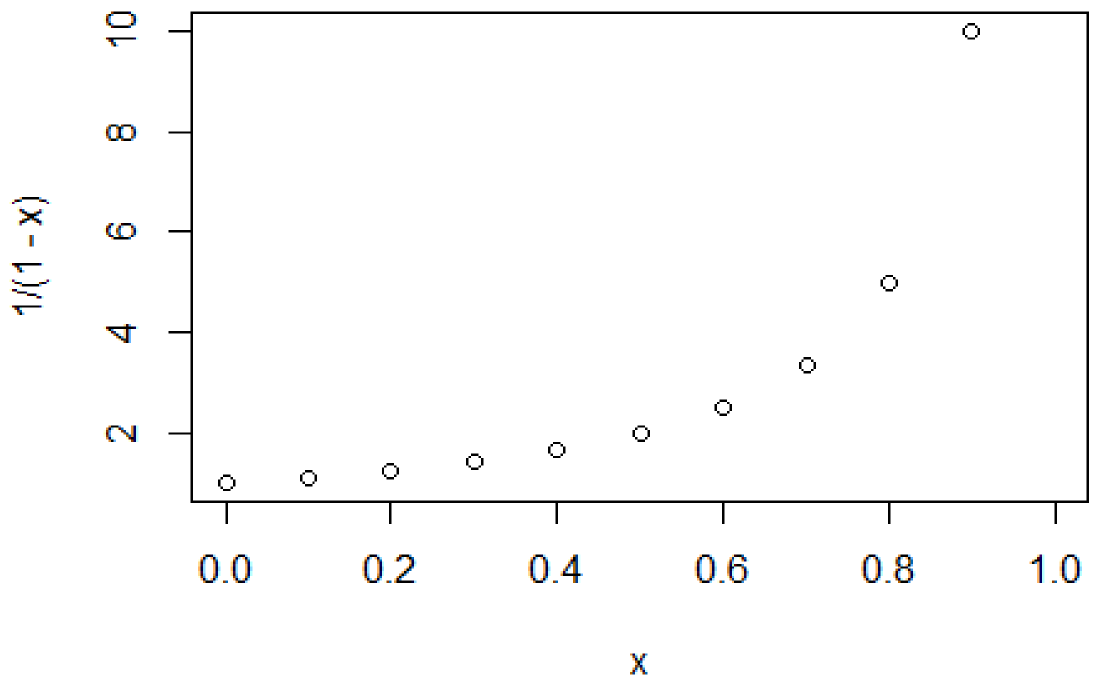
$$\frac{1}{1-p}$$

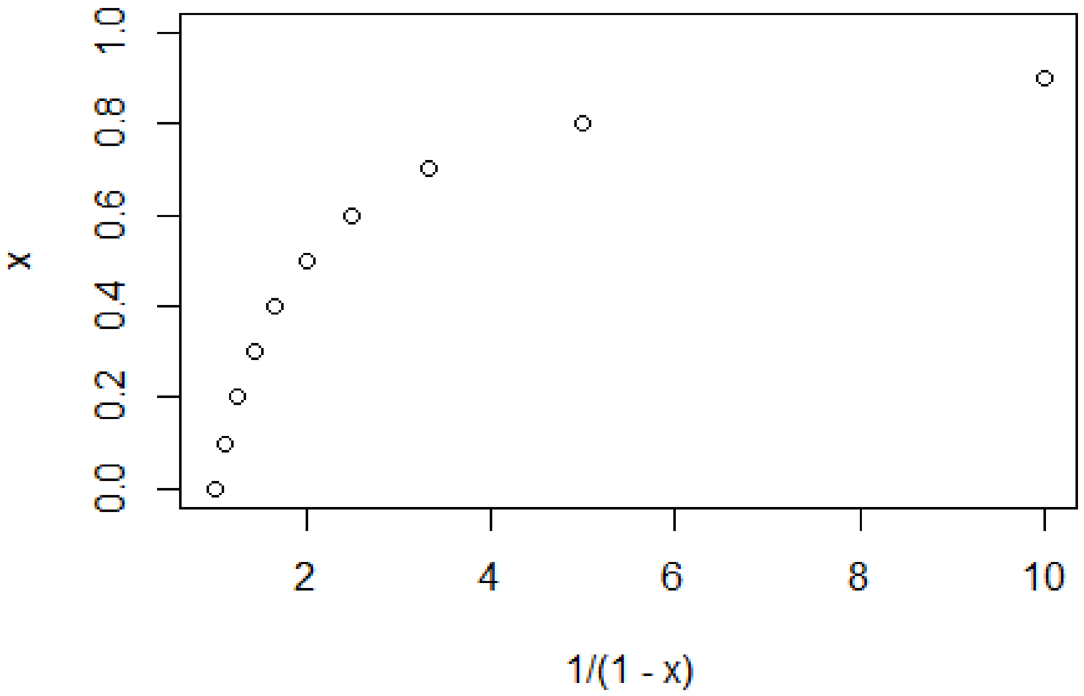


Plotting odds

```
plot(x, 1/(1-x))
```

```
plot(1/(1-x),x)
```





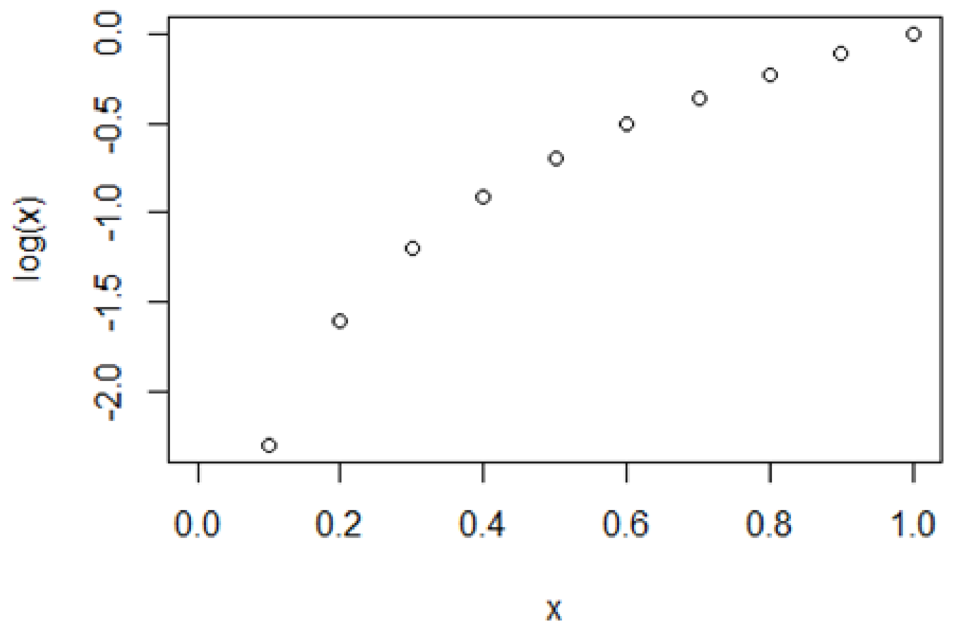
Logs

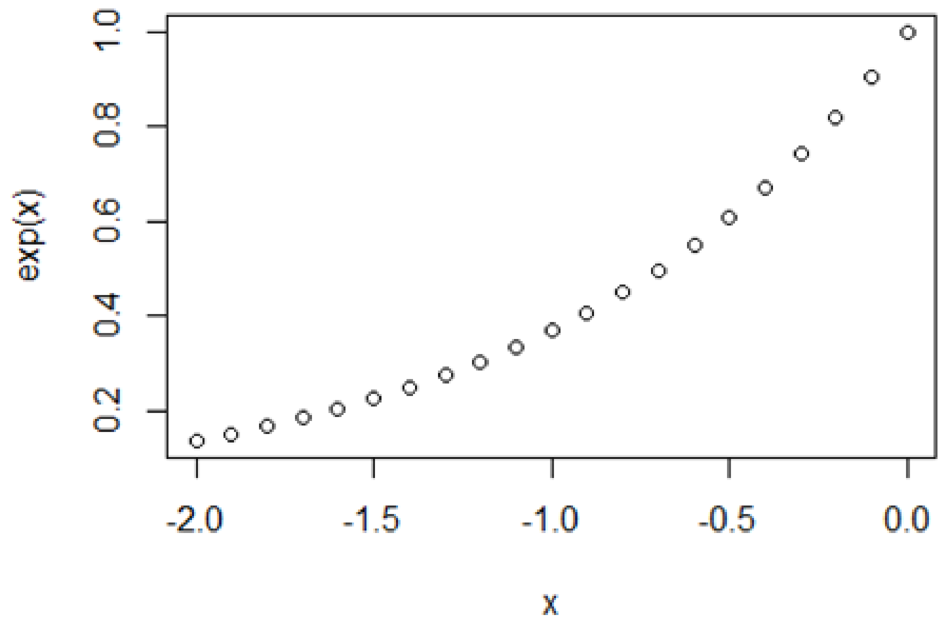
- Logs also have this asymptotic property on both sides.

```
x <- seq(0,1,by=0.1)
plot(x,log(x))
```

```
x <- seq(-2,0, by =.1)
plot(x, exp(x))
```

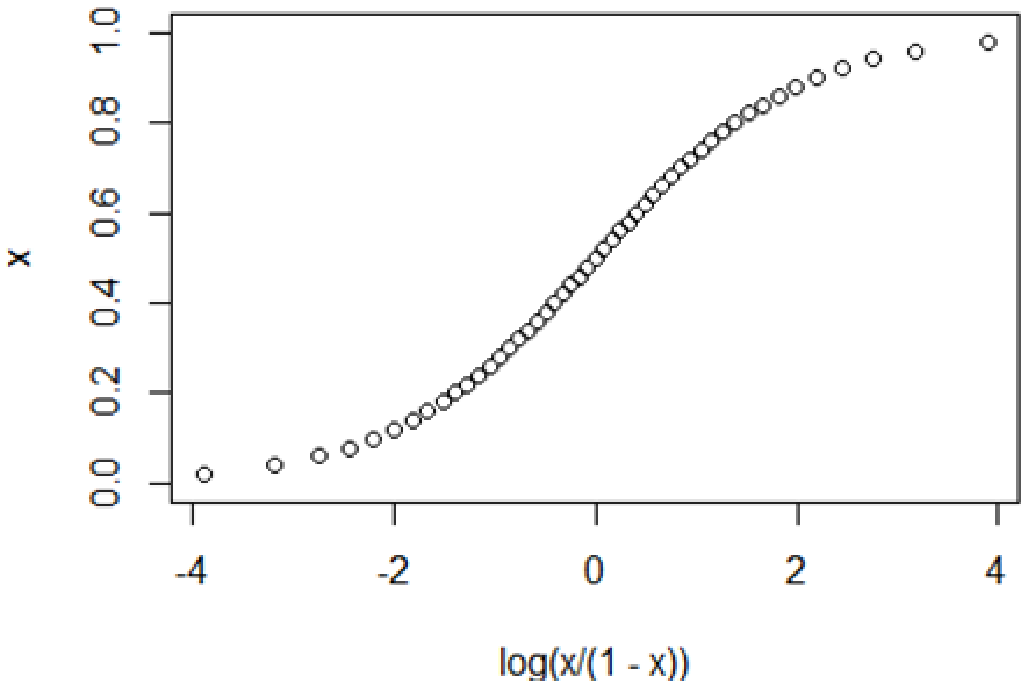
- But not exactly how we want because we want a ceiling effect approaching 0 and approaching 1



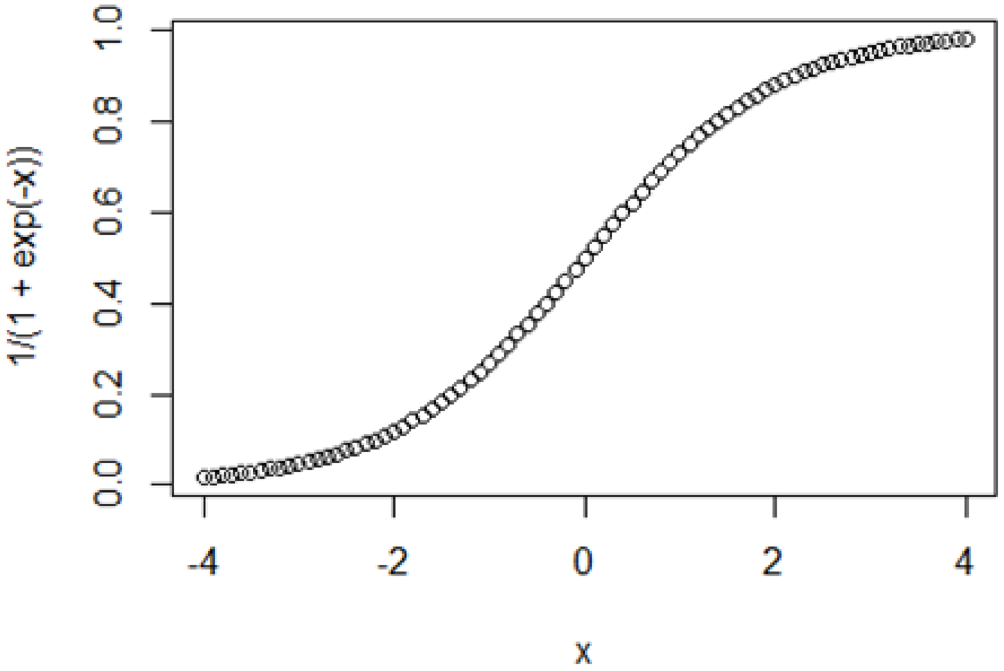


Log-odds

```r
par(mfrow=c(1,1))
x <- seq(0,1,by=.02)
plot(log(x/(1-x)),x)
```




```r
par(mfrow=c(1,1))
x <- seq(-4,4,by=.1)
plot(x, 1/(1+exp(-x)))
```



Slope of an S-curve

- Does a slope make sense?

- The slope changes as x changes – so there's not a single slope.

- There are concepts we can make use of (e.g. the slope at the steepest point of the curve)

- It is hard to interpret coefficients in logistic regressions

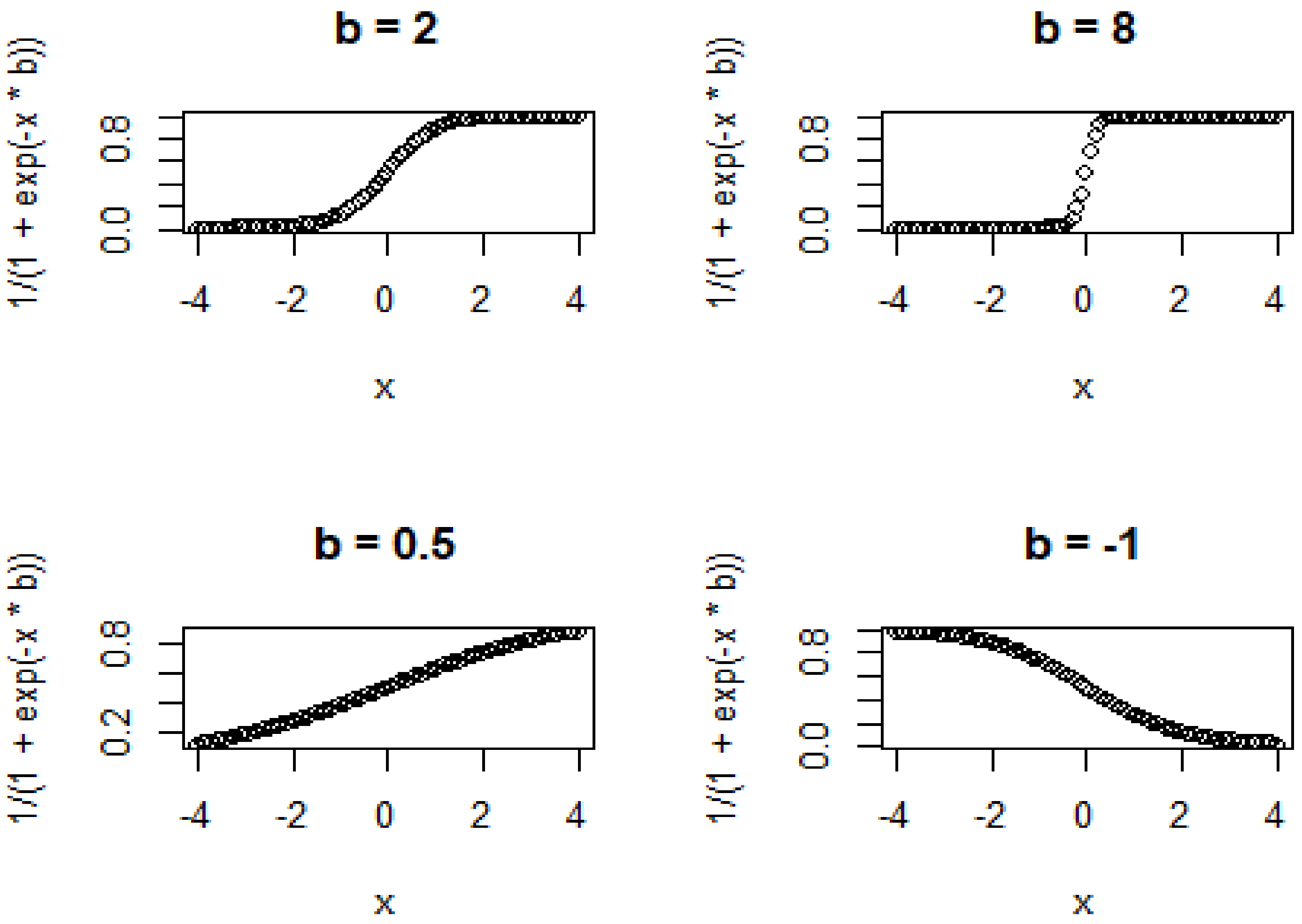
Gelman, Andrew & Jennifer Hill. 2007. *Data Analysis Using Regression and Multilevel/Hierarchical Models.* Cambridge: Cambridge University Press.

Slope

- Adding a slope coefficient

```
par(mfrow=c(2,2))
x <- seq(-4,4,by=.1)
b <- 2
plot(x, 1/(1+exp(-x*b)), main = "b = 2")
b <- 8
plot(x, 1/(1+exp(-x*b)), main = "b = 8")
b <- 0.5
plot(x, 1/(1+exp(-x*b)), main = "b = 0.5")
b <- -1
plot(x, 1/(1+exp(-x*b)), main = "b = -1")
```

Slope of a logistic function



- Intuitively, the slope corresponds to how steep the curve is around its center point
- Large slope = small change in *x* is equal to a large change in the probability that *y* = 1 **around the middle of the S-curve (where the slope is changing the fastest)**

Simulation exercise

- Simulate whether someone will go to class other not

```r
set.seed(1)
coffee <- rnorm(100, 15, 5)
happiness <- rnorm(100, 10, 2)
a <- -5
b1 <- -1
b2 <- 1
```

```r
xb <- a + b1 * happiness + b2 * coffee + rnorm(100, 0, 0.1)

p <- 1/ (1+exp(-xb))
```

Simulation exercise

- Simulate whether someone will go to class other not

```
set.seed(1)
coffee <- rnorm(100, 15, 5)
happiness <- rnorm(100, 10, 2)
a <- -5
b1 <- -1
b2 <- 1
```

$$Prob\{y = 1|x\} = \frac{1}{1 + \exp(-x\beta)}$$

```
xb <- a + b1 * happiness + b2 * coffee + rnorm(100, 0, 0.1)

p <- 1/ (1+exp(-xb))
```

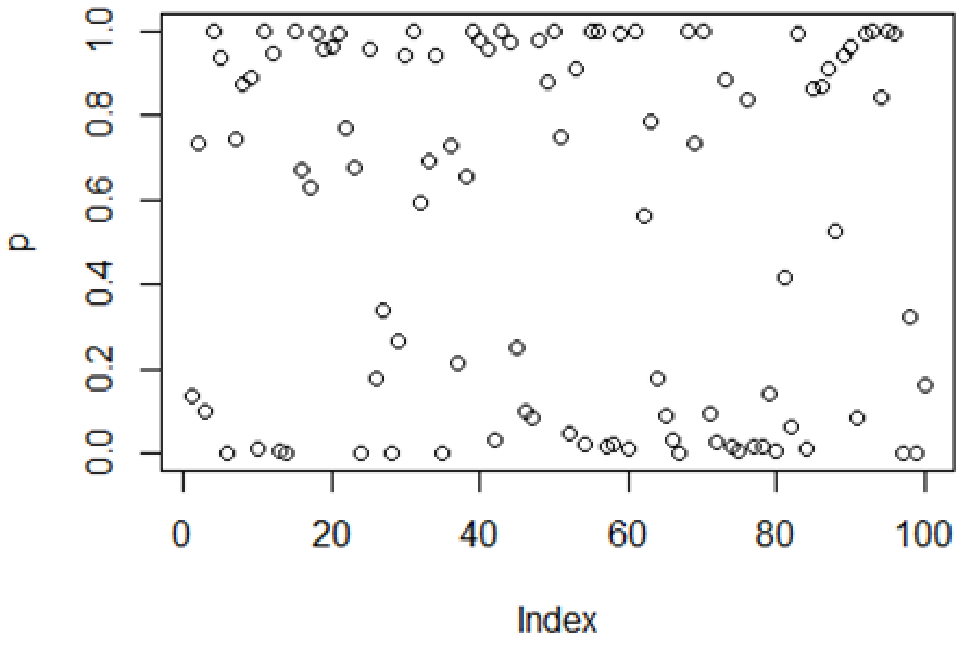
We have to subject our *y* to the inverse of the logit function to get the actual probabilities

Simulation exercise

```
summary(p)

##       Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0000497 0.0780173 0.7105522 0.5493012 0.9611723 0.9999948

par(mfrow=c(1,1))
plot(p)
```



Simulation exercise

```
gotoclass <- rbinom(n=100, size=1, prob=p)
gotoclass

##   [1] 0 0 0 1 0 0 1 1 0 0 1 1 0 0 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 1 1 1 1 1 0 1 0
##  [38] 1 1 1 1 0 1 1 1 0 0 1 1 1 1 0 1 0 1 1 0 0 1 0 1 0 1 0 0 0 0 1 1 1 0 0 1 0
##  [75] 0 1 0 0 0 0 1 0 1 0 1 1 1 1 1 1 0 1 1 1 1 1 0 0 0 0
```

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

We are creating a Bernouilli distribution from our probabilities which are produced from the stochastic model we simulated earlier

Generalized linear models

- Logistic regressions are a type of **generalized linear models**
- Unlike normal linear models, glms do not use ordinary least squared (OLS) but rather use **Maximum Likelihood** to estimate parameters.
  - (We will discuss Maximum Likelihood later in in class)
- Use the package glm2()

```r
library(glm2)
model.logit <- glm(gotoclass~happiness+coffee, family="binomial")
```

```r
summary(model.logit)
```

```
##
## Call:
## glm(formula = gotoclass ~ happiness + coffee, family = "binomial")
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.1157     2.8897  -1.424  0.15438
## happiness    -1.6263     0.5221  -3.115  0.00184 **
## coffee        1.3600     0.3348   4.062 4.87e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 137.186  on 99  degrees of freedom
## Residual deviance:  36.172  on 97  degrees of freedom
## AIC: 42.172
##
## Number of Fisher Scoring iterations: 8
```

Interpreting logistic regression coefficients

- It is hard to interpret logistic regression coefficients because the relationship is non-linear
- The intercept is interpreted assuming 0 for other predictors
  - But sometimes 0 it not interesting
  - Alternatively we can interpret the intercept at the center point
- Rather than consider a discrete change in *x* we can compute an approximation of the derivative of the logistic curve at the central value (where the relationship is steepest)
  - You get this by dividing the coefficient by 4

Gelman, Andrew & Jennifer Hill. 2007. *Data Analysis Using Regression and Multilevel/Hierarchical Models.* Cambridge: Cambridge University Press.

Interpreting logistic regression coefficients

- **Divide by 4 rule (this will give you the coefficient)**

- Dividing by 4 gives you the **maximum difference** in *y* corresponding to a **unit of difference** in *x*

```
-1.6263/4
## [1] -0.406575
```

A decrease in 0.4 of happiness corresponds to the steepest rise in the probability of going to class where coffee is at its mean value

Gelman, Andrew & Jennifer Hill. 2007. *Data Analysis Using Regression and Multilevel/Hierarchical Models.* Cambridge: Cambridge University Press.

Interpreting logistic regression

- The intercept is what the logit probability of y is when all the predictors are 0.

```
invlogit <- function (x) {1/(1+exp(-x))}
invlogit(-4.1157)

## [1] 0.01605263
```

This is a function I wrote in R (actually in Gelman & Hill 2007), to calculate the probability from the logit probability

$$Prob\{y = 1|x\} = \boxed{\frac{1}{1 + \exp(-x\beta)}}$$

Gelman, Andrew & Jennifer Hill. 2007. *Data Analysis Using Regression and Multilevel/Hierarchical Models.* Cambridge: Cambridge University Press.

Interpreting logistic regression

- The intercept is what the logit probability of y is when all the predictors are 0.

```
invlogit <- function (x) {1/(1+exp(-x))}
invlogit(-4.1157)

## [1] 0.01605263
```

The probability of *y* (going to class) when happiness is 0 and coffee consumption is 0.

If someone has 0 happiness score and they have had no coffee, there is a 1.6% chance they will go to class

Gelman, Andrew & Jennifer Hill. 2007. *Data Analysis Using Regression and Multilevel/Hierarchical Models.* Cambridge: Cambridge University Press.

Interpreting logistic regression

- The intercept is what the logit probability of y is when all the predictors are 0.

```
invlogit <- function (x) {1/(1+exp(-x))}
invlogit(-4.1157 + -1.6263*mean(happiness) + 1.3600*mean(coffee))

## [1] 0.707797
```

If someone has a mean happiness score and they have had the average amount of coffee, there is a 70.1% chance they will go to class

The probability of *y* (going to class) when happiness is at its mean value (-16.41) and coffee consumption is at its mean value (21.1)

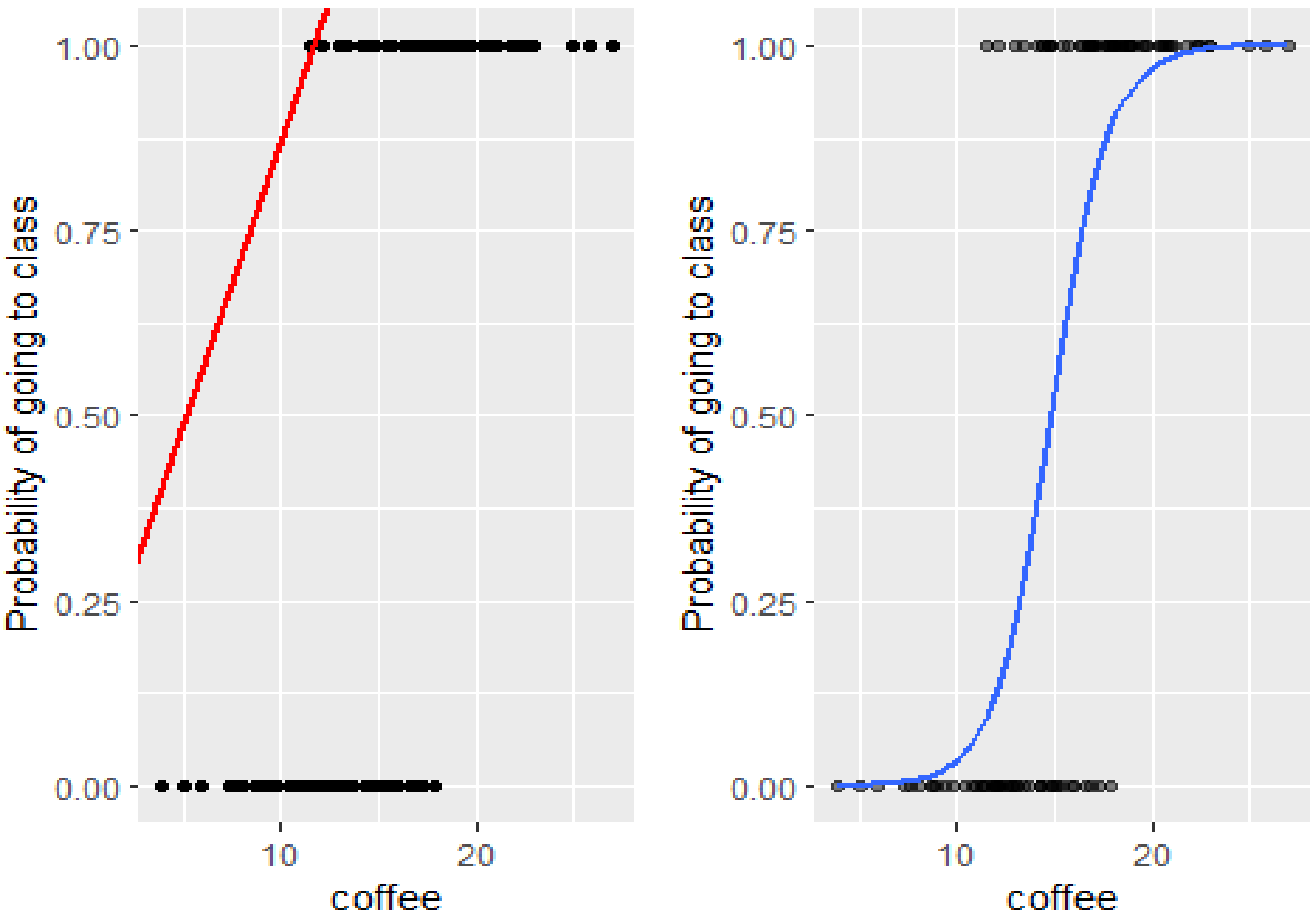
Gelman, Andrew & Jennifer Hill. 2007. *Data Analysis Using Regression and Multilevel/Hierarchical Models.* Cambridge: Cambridge University Press.

Visualizing logistic regression in ggplot

```
par(mfrow=c(1,2))
data <- data.frame(gotoclass, happiness, coffee)

plotline<-ggplot(data, aes(x=coffee, y = gotoclass))+geom_point()+
  geom_abline(intercept = 0.11105, slope = 0.076446, color="red", size=1)+
  ylab("Probability of going to class")
```

```
plotS<-ggplot(data, aes(x=coffee, y= gotoclass))+
  geom_point(alpha=.5)+
  stat_smooth(method="glm", se=FALSE, method.args = list(family=binomial))+
  ylab("Probability of going to class")
plotS
```



Probability of going to class
1.00
0.75
0.50
0.25
0.00
10
20
coffee
Probability of going to class
1.00
0.75
0.50
0.25
0.00
10
20
coffee

Logistic regression on Chácobo MFC

- Try running a logistic regression model on the Chacobo Forced Experiment data, with the response as the dependent variable and the simulated Pitch (Hz) as the independent variable.

```
logit_model_01 <- glm(response~Pitch_Hz, data=df, family="binomial")
summary(logit_model_01)
```

```
## Call:
## glm(formula = response ~ Pitch_Hz, family = "binomial", data = df)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.109549   0.626233  -14.55   <2e-16 ***
## Pitch_Hz     0.075658   0.005149   14.70   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2242.6  on 1619  degrees of freedom
## Residual deviance: 1980.1  on 1618  degrees of freedom
## AIC: 1984.1
##
## Number of Fisher Scoring iterations: 4
```

Interpreting the results

- Apply the Divide by 4 rule to the slope coefficient
- Interpret the intercept at the mean value for the Pitch (Hz)
- Plot the logistic regression using ggplot()

Interpreting the slope

```
0.075658/4

## [1] 0.0189145
```

An increase in 0.02 Hz of pitch corresponds to the steepest rise in the probability of a Chácobo speaker chosing *janáquë* ’s/he vomited’ over *jánaquë* ‘ s/he left’ in the MCF

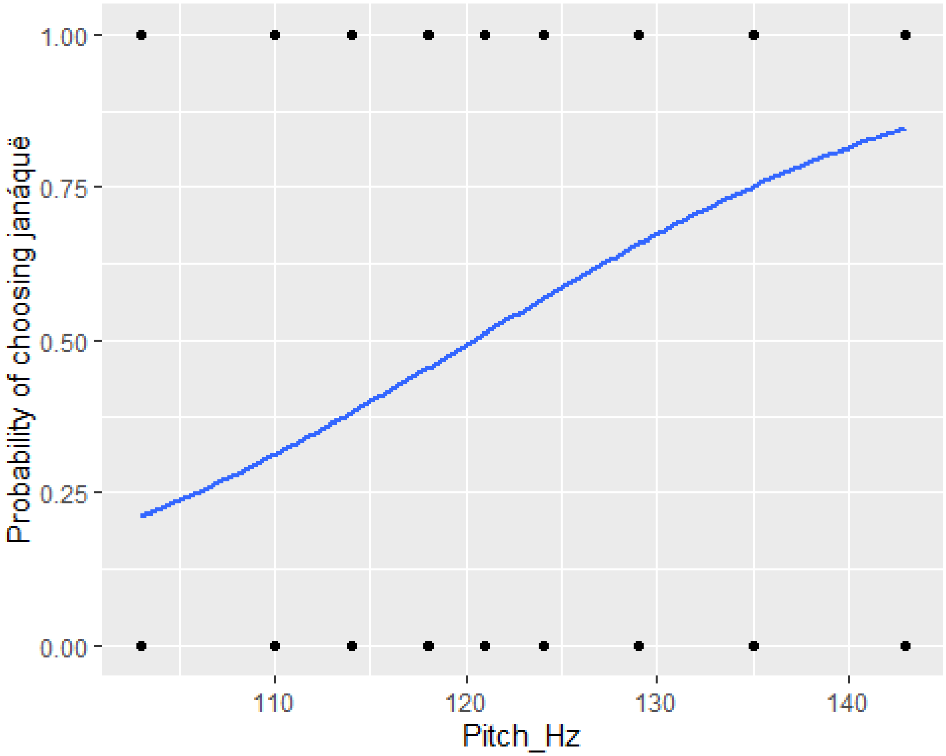
Interpreting the intercept

```
invlogit(-9.109549+ 0.075658*mean(df$Pitch_Hz))
## [1] 0.5287489
```

The probability of *y* (a Chácobo speaker chosing *janáquë* 's/he vomited') when pitch (Hz) is at its mean for the experiment (121.93 Hz) is 52% (when the first syllable is fixed at 80 Hz)

Plotting the loglinear relation

```
plotS<-ggplot(dat=df, aes(x=Pitch_Hz, y= response))+
  geom_point(alpha=.5)+
  stat_smooth(method="glm", se=FALSE, method.args = list(family=binomial))+
  ylab("Probability of choosing janáquë")
plotS
```



Dutch causative constructions

- Predict which causative occurs in Dutch

```
data(doenLaten)
d <- doenLaten
head(d)

##     Aux Country  Causation EPTrans EPTrans1
## 1 laten      NL   Inducive    Intr     Intr
## 2 laten      NL   Physical    Intr     Intr
## 3 laten      NL   Inducive      Tr       Tr
## 4  doen      BE  Affective    Intr     Intr
## 5 laten      NL   Inducive      Tr       Tr
## 6 laten      NL Volitional    Intr     Intr
```

Dutch causative constructions

- Predict which causative occurs in Dutch

```r
data(doenLaten)
d <- doenLaten
head(d)

##      Aux Country  Causation EPTrans EPTrans1
## 1 laten      NL    Inducive    Intr     Intr
## 2 laten
## 3 laten
## 4  doen
## 5 laten
## 6 laten
```

***Doen* relates to direct causation**

***Laten* relates to indirect causation**

(1)	*Hij*	*deed*	*me*	*denken*		*aan*	*mijn*	*vader*
	He	**did**	me	think		at	my	father
	'He reminded me of my father.'							
(2)	*Ik*	*liet*	*hem*	*mijn*	*huis*	*schilderen*		
	I	**let**	him	my	house	Paint		
	'I had him paint my house.'							