Introduction to Generalized Linear Models

Filippo Gambarota

University of Padova

2022/2023

```
devtools::load_all()
library(tidyverse)
library(kableExtra)
library(patchwork)
```

Outline

1. Beyond the Gaussian distribution
2. Generalized Linear Models
3. Relevant distributions
4. Data simulation *[EXTRA]*
5. Binomial GLM
6. Binomial GLM

1. Beyond the Gaussian distribution

2. Generalized Linear Models

3. Relevant distributions

4. Data simulation *[EXTRA]*

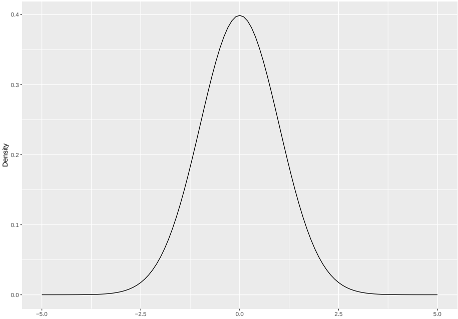
5. Binomial GLM

6. Binomial GLM

Quick recap about Gaussian distribution

- function
- parameters
- support

“‘r ggnorm(0, 1) “‘

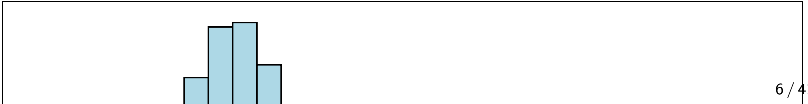


But not always gaussian-like variables!

Reaction times

Measuring reaction times during a cognitive task. Non-negative and proably skewed data.

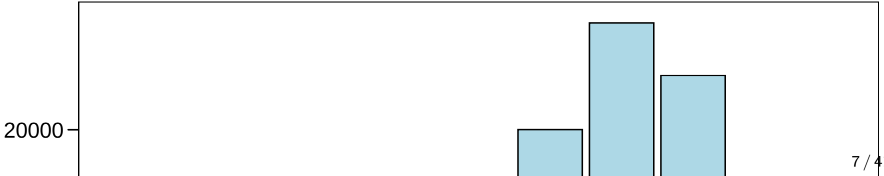
```r
dat <- data.frame(
    x = rgamma(1e5, 9, scale = 0.5)*100
)

dat |>
    ggplot(aes(x = x)) +
    geom_histogram(fill = "lightblue",
                   color = "black") +
    xlab("Reaction Times (ms)") +
    ylab("Count") +
    mytheme()
```



Binary outcomes

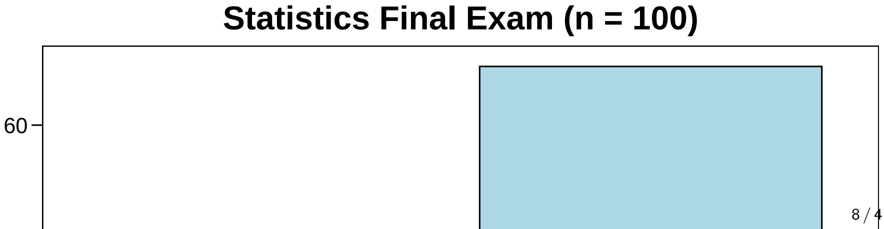
Counting the number of people passing the exam out of the total.
Discrete and non-negative. A series of binary (i.e., *bernoulli*) experiments.

```
dat <- data.frame(x = rbinom(1e5, 10, 0.7))

dat |>
    ggplot(aes(x = factor(x))) +
    geom_bar(fill = "lightblue",
             color = "black") +
    xlab("Number of success out of 10 trials") +
    ylab("Count") +
    mytheme()
```



Binary outcomes

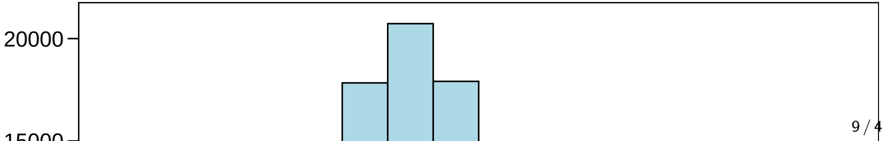
```
dat <- data.frame(y = c(70, 30), x = c("Passed", "Failed"))

dat |>
    ggplot(aes(x = x, y = y)) +
    geom_col(color = "black",
             fill = "lightblue") +
    ylab("Count") +
    mytheme() +
    theme(axis.title.x = element_blank()) +
    ggtitle("Statistics Final Exam (n = 100)")
```



Counts

Counting the number of new patients per week. Discrete and non-negative values.

```r
dat <- data.frame(x = rpois(1e5, 15))

dat |>
    ggplot(aes(x = x)) +
    geom_histogram(binwidth = 2,
                   color = "black",
                   fill = "lightblue") +
    xlab("Number of new patients per week") +
    ylab("Count") +
    mytheme()
```



Should we use a linear model for these variables?

```
# y = number of exercises solved in 1 semester
# x = percentage of attended lectures

n <- 30
x <- rep(0:10, each = n)
b0 <- 0.01
b1 <- 0.8
y <- rbinom(length(x), 1, plogis(qlogis(b0) + b1*x))

dat <- data.frame(x, y)

dat |>
    ggplot(aes(x = x, y = y)) +
    geom_hline(yintercept = 0, linetype = "dashed", col = "rec
    geom_point(size = 3,
               alpha = 0.5,
               position = position_jitter(height = 0.03)) +
```

Should we use a linear model for these variables?

```r
# y = number of exercises solved in 1 semester
# x = percentage of attended lectures

n <- 30
x <- runif(n, 0, 1)
b0 <- 0
b1 <- 4
y <- rpois(n, exp(b0 + b1*x))

dat <- data.frame(x, y)

dat |>
    ggplot(aes(x = x*100, y = y)) +
    geom_hline(yintercept = 0, linetype = "dashed", col = "rec
    geom_point(size = 3) +
    geom_smooth(method = "lm",
                se = F) +
```

Should we use a linear model for these variables?

```
fit <- lm(y ~ x, data = dat)

dfit <- data.frame(
    fitted = fitted(fit),
    residuals = residuals(fit)
)

qqn <- dfit |>
    ggplot(aes(sample = residuals)) +
    stat_qq() +
    stat_qq_line() +
    xlab("Theoretical Quantiles") +
    ylab("Residuals") +
    mytheme()

res_fit <- dfit |>
    ggplot(aes(x = fitted, y = residuals)) +
```

A new class of models

- We need that our model take into account the **features of our response variable**
- We need a model that, **with appropriate transformation**, keep **properties of standard linear models**
- We need a model that is **closer to the true data generation process**

Let's switch to Generalized Linear Models!

Main references

add books here

General idea

- models that assume distributions other than the normal distributions
- models that considers non-linear relationships

Recipe for a GLM

- **Random Component**
- **Systematic Component**
- **Link Function**

Random Component

Systematic Component

Link Function

1. Beyond the Gaussian distribution

2. Generalized Linear Models

3. Relevant distributions

4. Data simulation *[EXTRA]*

5. Binomial GLM

6. Binomial GLM

Binomial distribution

Poisson distribution

1. Beyond the Gaussian distribution

2. Generalized Linear Models

3. Relevant distributions

4. Data simulation *[EXTRA]*

5. Binomial GLM

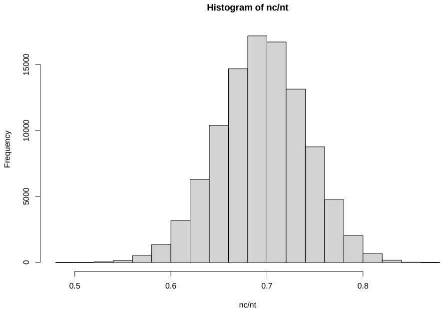
6. Binomial GLM

Data simulation *[EXTRA]*

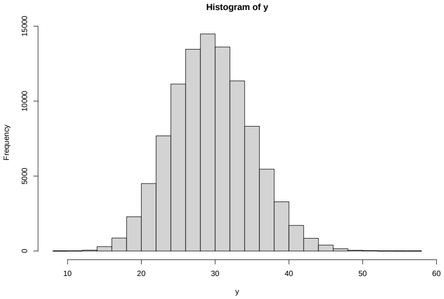
- During the course we will try to simulate some data. Simulating data is an amazing education tool to understand a statistical model.
- By simulating from a **generative model** we are doing **Monte Carlo Simulations** [1]

Data simulation *[EXTRA]*

```r
n <- 1e5 # number of experiments
nt <- 100 # number of subjects
p <- 0.7 # probability of success
nc <- rbinom(n, nt, p)
```




```r
n <- 1e5 # number of subjects
lambda <- 30 # mean/variance
y <- rpois(n, lambda)
```



Example: Passing the exam

We want to measure the impact of **watching tv-shows** on the probability of **passing the statistics exam**.

- `exam`: **passing the exam** (1 = “passed”, 0 = “failed”)
- `tv_shows`: **watching tv-shows regularly** (1 = “yes”, 0 = “no”)

```
head(dat)
```

```
##   tv_shows exam
## 1        1    0
## 2        1    1
## 3        1    1
## 4        1    1
## 5        1    0
## 6        1    1
```

Example: Passing the exam

We can create the **contingency table**

```
xtabs(~exam + tv_shows, data = dat) |>
    addmargins()
```

```
##       tv_shows
## exam    0   1 Sum
##   0    32  18  50
##   1    18  32  50
##   Sum  50  50 100
```

Example: Passing the exam

Each cell probability π_{ij} is computed as π_{ij}/n

```
(xtabs(~exam + tv_shows, data = dat)/n) |>
    addmargins()
```

```
##       tv_shows
## exam      0    1  Sum
##   0    0.32 0.18 0.50
##   1    0.18 0.32 0.50
##   Sum 0.50 0.50 1.00
```

Example: Passing the exam - Odds

The most common way to analyze a 2x2 contingency table is using the **odds ratio** (OR). Firsly let's define *the odds of success* as:

$$odds = \frac{\pi}{1-\pi} \quad \pi = \frac{odds}{odds+1}$$

- the **odds** are non-negative, ranging between 0 and $+\infty$
- an **odds** of e.g. 3 means that we expect 3 *success* for each *failure*

Example: Passing the exam - Odds

For the `exam` example:

```
odds <- function(p) p / (1 - p)
p11 <- mean(with(dat, exam[tv_shows == 1])) # passing exam | t
odds(p11)
```

```
## [1] 1.777778
```

Example: Passing the exam - Odds Ratio

The OR is a ratio of odds:

$$OR = \frac{\frac{\pi_1}{1-\pi_1}}{\frac{\pi_2}{1-\pi_2}}$$

- OR ranges between 0 and $+\infty$. When $OR = 1$ the odds for the two conditions are equal
- An e.g. $OR = 3$ means that being in the condition at the numerator increase 3 times the odds of success

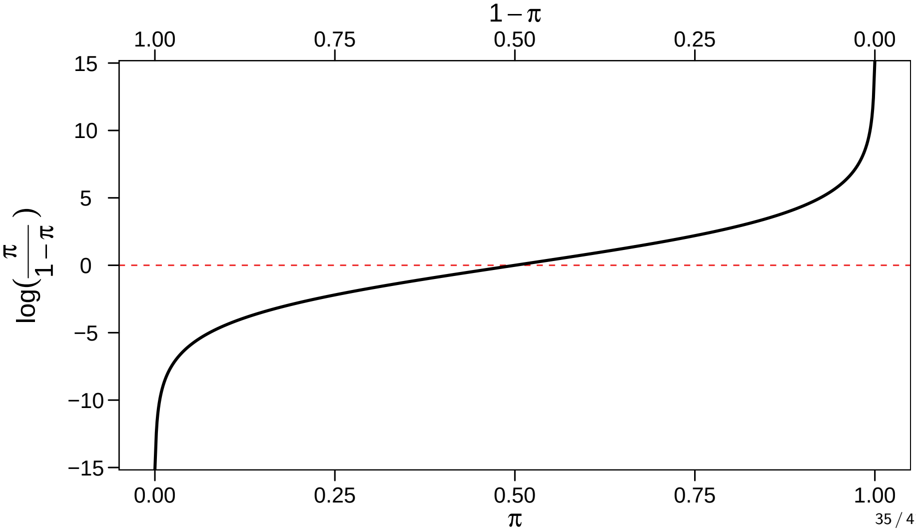
Example: Passing the exam - Odds Ratio

```
odds_ratio <- function(p1, p2) odds(p1) / odds(p2)
p11 <- mean(with(dat, exam[tv_shows == 1])) # passing exam | t
p10 <- mean(with(dat, exam[tv_shows == 0])) # passing exam | r
odds_ratio(p11, p10)
```

```
## [1] 3.160494
```

Why using these measure?

The odds have an interesting property when taking the logarithm. We can express a probability π using a scale ranging $[-\infty, +\infty]$



Binomial GLM

- The **random component** of a Binomial GLM the binomial distribution with parameter π
- The **systematic component** is a linear combination of predictors and coefficients $\beta \boldsymbol{X}$
- The **link function** is a function that map probabilities into the $[-\infty, +\infty]$ range.

Binomial GLM - Logit Link

The **logit** link is the most common link function when using a binomial GLM:

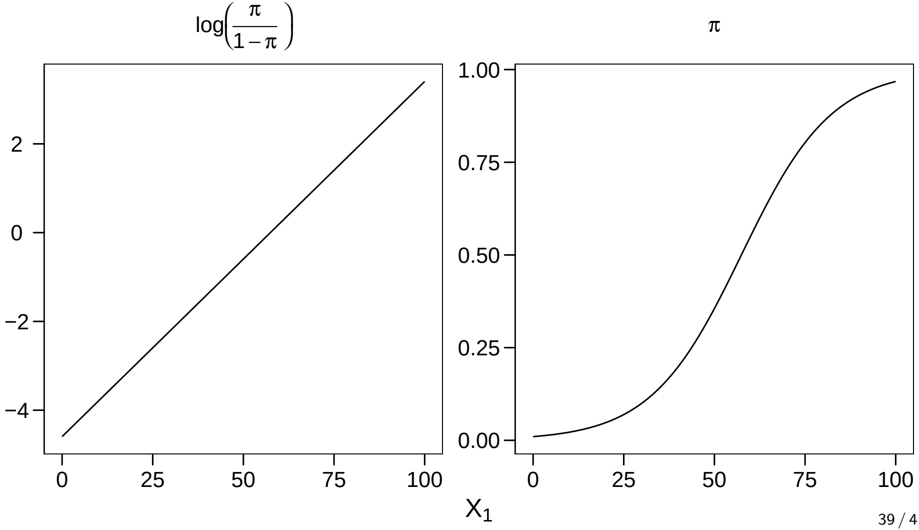
$$log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 X_1 + ...\beta_p X_p$$

The inverse of the **logit** maps again the probability into the $[0,1]$ range:

$$\pi = \frac{e^{\beta_0+\beta_1 X_1+...\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+...\beta_p X_p}}$$

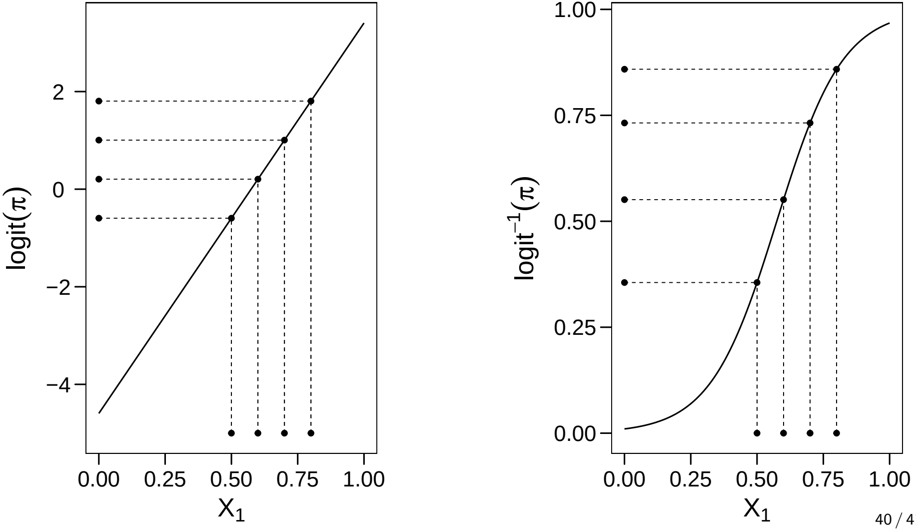
Binomial GLM - Logit Link

Thus with a single numerical predictor x the relationship between x and π in non-linear on the probability scale but linear on the logit scale.



Binomial GLM - Logit Link

The problem is that effects are non-linear, thus is more difficult to interpret and report model results



References

[1] J. E. Gentle, “Monte carlo methods for statistical inference,” in *Computational statistics*, J. E. Gentle, Ed., New York, NY: Springer New York, 2009, pp. 417–433. doi: 10.1007/978-0-387-98144-4_11.