Introduction to GLMs

Contents

1 Introduction **1**

Configuring R . . . 1

1.1 Welcome . . . 2

1.2 What you should already know . . . 2

1.2.1 Epi 202: probability models . . . 2

1.2.2 Epi 203: inference for one or several homogenous populations . . . 3

1.2.3 Stat 108: linear regression models . . . 3

1.3 What we will cover in this course . . . 4

1.4 Motivations for regression models . . . 4

1.4.1 Example: Adelie penguins . . . 4

1.4.2 Linear regression . . . 4

1.4.3 Curved regression lines . . . 5

1.4.4 Multiple regression . . . 6

1.4.5 Modeling non-Gaussian outcomes . . . 8

1.4.6 Why don't we use linear regression? . . . 9

1.4.7 Zoom out . . . 10

1.4.8 log transformation of dose? . . . 11

1.4.9 Logistic regression . . . 12

1.5 Structure of regression models . . . 12

1 Introduction

Configuring R

Functions from these packages will be used throughout this document:

```
library(conflicted) # check for conflicting function definitions
# library(printr) # inserts help-file output into markdown output
library(rmarkdown) # Convert R Markdown documents into a variety of formats.
library(pander) # format tables for markdown
library(ggplot2) # graphics
library(ggfortify) # help with graphics
library(dplyr) # manipulate data
library(tibble) # `tibble`s extend `data.frame`s
library(magrittr) # `%>%` and other additional piping tools
library(haven) # import Stata files
library(knitr) # format R output for markdown
library(tidyr) # Tools to help to create tidy data
library(plotly) # interactive graphics
library(dobson) # datasets from Dobson and Barnett 2018
library(parameters) # format model output tables for markdown
library(haven) # import Stata files
```

```
library(latex2exp) # use LaTeX in R code (for figures and tables)
library(fs) # filesystem path manipulations
library(survival) # survival analysis
library(survminer) # survival analysis graphics
library(KMsurv) # datasets from Klein and Moeschberger
library(parameters) # format model output tables for
library(webshot2) # convert interactive content to static for pdf
library(forcats) # functions for categorical variables ("factors")
library(stringr) # functions for dealing with strings
library(lubridate) # functions for dealing with dates and times
```

Here are some R settings I use in this document:

```
rm(list = ls()) # delete any data that's already loaded into R

conflicts_prefer(dplyr::filter)
ggplot2::theme_set(
  ggplot2::theme_bw() +
        # ggplot2::labs(col = "") +
    ggplot2::theme(
      legend.position = "bottom",
      text = ggplot2::element_text(size = 12, family = "serif")))

knitr::opts_chunk$set(message = FALSE)
options('digits' = 6)

panderOptions("big.mark", ",")
pander::panderOptions("table.emphasize.rownames", FALSE)
pander::panderOptions("table.split.table", Inf)
conflicts_prefer(dplyr::filter) # use the `filter()` function from dplyr() by default
legend_text_size = 9
run_graphs = TRUE
```

1.1 Welcome

Welcome to Epidemiology 204: Quantitative Epidemiology III (Statistical Models).

Epi 204 is a course on **regression modeling**.

1.2 What you should already know

> **Warning**
>
> Epi 202, Epi 203, and Sta 108 are prerequisites for this course. If you haven't passed one of these courses, talk to me ASAP.

1.2.1 Epi 202: probability models

- Probability distributions
  - binomial
  - Poisson
  - Gaussian
  - exponential

- Characteristics of probability distributions

  - Mean, median, mode, quantiles
  - Variance, standard deviation, overdispersion

- Characteristics of samples
  - independence, dependence, covariance, correlation
  - ranks, order statistics
  - identical vs nonidentical distribution (homogeneity vs heterogeneity)
  - Laws of Large Numbers
  - Central Limit Theorem for the mean of an iid sample

1.2.2 Epi 203: inference for one or several homogenous populations

- the maximum likelihood inference framework:
  - likelihood functions
  - log-likelihood functions
  - score functions
  - estimating equations
  - information matrices
  - point estimates
  - standard errors
  - confidence intervals
  - hypothesis tests
  - p-values

- Hypothesis tests for one, two, and >2 groups:
  - t-tests/ANOVA for Gaussian models
  - chi-square tests for binomial and Poisson models
  - nonparametric tests:
    - Wilcoxon signed-rank test for matched pairs
    - Mann–Whitney/Kruskal-Wallis rank sum test for ≥ 2 independent samples
    - Fisher's exact test for contingency tables
    - Cochran–Mantel–Haenszel-Cox log-rank test

For all of the quantities above, and especially for confidence intervals and p-values, you should know how **both**:

- how to compute them
- how to interpret them

1.2.3 Stat 108: linear regression models

- building models for Gaussian outcomes
  - multiple predictors
  - interactions
- regression diagnostics
- fundamentals of R programming; e.g.:
  - Wickham, Çetinkaya-Rundel, and Grolemund (2023)
  - Dalgaard (2008)
- RMarkdown or Quarto for formatting homework[1]
  - LaTeX for writing math in RMarkdown/Quarto

[1] https://r4ds.hadley.nz/quarto

1.3 What we will cover in this course

- Linear (Gaussian) regression models (review and more details)
- Regression models for non-Gaussian outcomes
  - binary
  - count
  - time to event
- Statistical analysis using R

We will start where Epi 203 left off: with linear regression models.

1.4 Motivations for regression models

Exercise 1.1. Why do we need regression models?

Solution 1.1.

- when there's not enough data to analyze every subgroup of interest individually
- especially when subgroups are defined using continuous predictors

1.4.1 Example: Adelie penguins

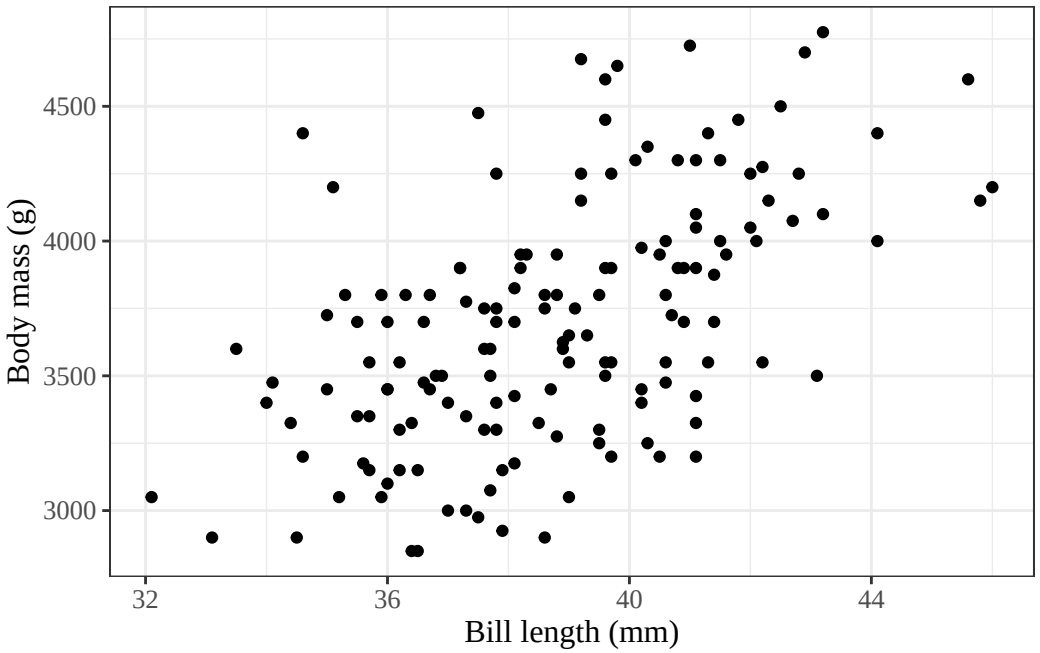

Figure 1: Palmer penguins

1.4.2 Linear regression

```
ggpenguins2 <-
  ggpenguins +
  stat_smooth(
    method = "lm",
```

```
    formula = y ~ x,
    geom = "smooth"
  )

ggpenguins2 |> print()
```

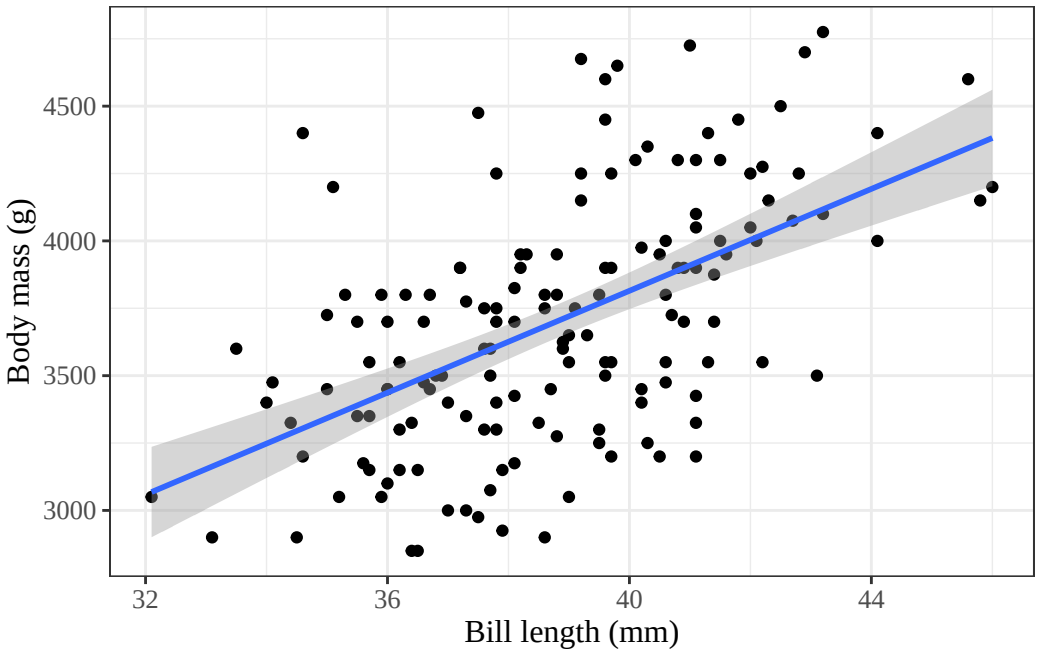

Figure 2: Palmer penguins with linear regression fit

1.4.3 Curved regression lines

```
ggpenguins2 <- ggpenguins +
  stat_smooth(
    method = "lm",
    formula = y ~ log(x),
    geom = "smooth"
  ) +
  xlab("Bill length (mm)") +
  ylab("Body mass (g)")
ggpenguins2
```

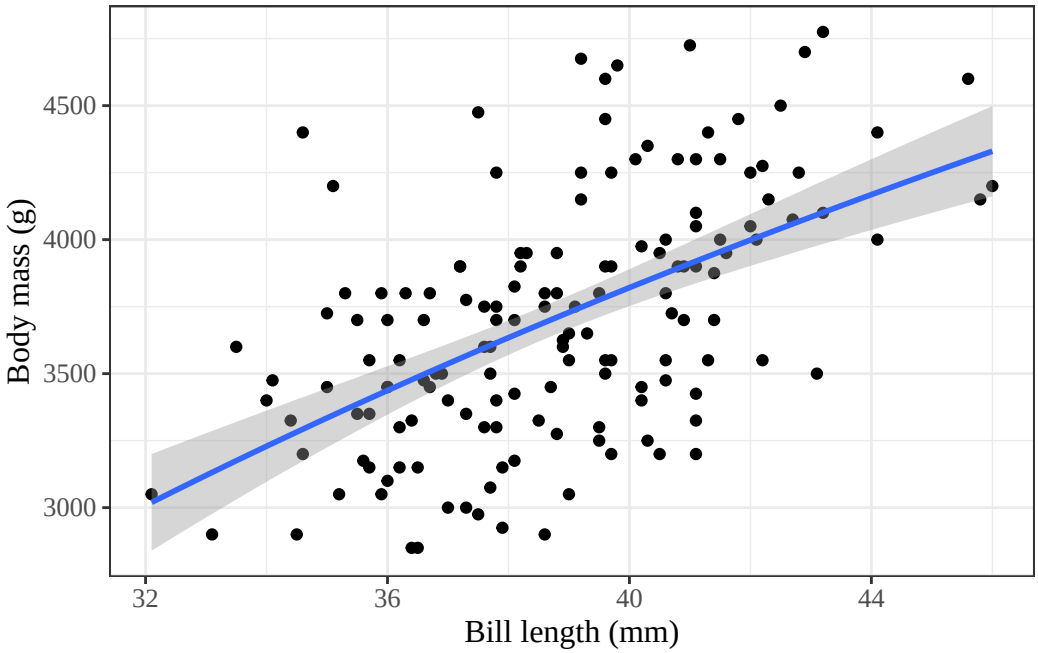

Figure 3: Palmer penguins - curved regression lines

1.4.4 Multiple regression

```
ggpenguins <-
  palmerpenguins::penguins |>
  ggplot(
    aes(
      x = bill_length_mm,
      y = body_mass_g,
      color = species
    )
  ) +
  geom_point() +
  stat_smooth(
    method = "lm",
    formula = y ~ x,
    geom = "smooth"
  ) +
  xlab("Bill length (mm)") +
  ylab("Body mass (g)")
ggpenguins |> print()
```

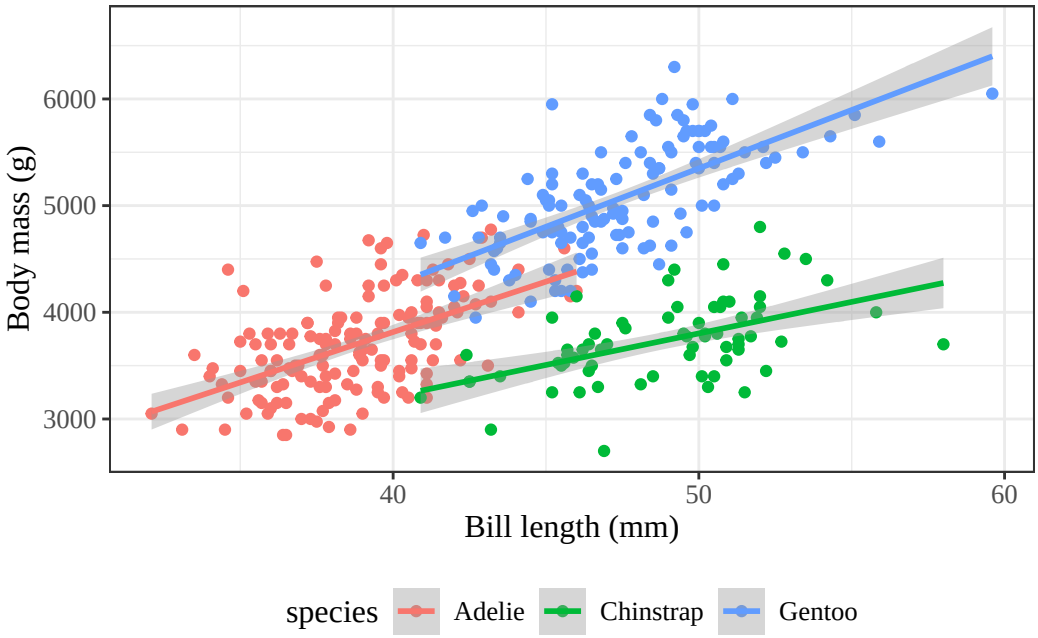

Figure 4: Palmer penguins - multiple groups

1.4.5 Modeling non-Gaussian outcomes

```
library(glmx)
data(BeetleMortality)
beetles <- BeetleMortality |>
  mutate(
    pct = died / n,
    survived = n - died
  )

plot1 <-
  beetles |>
  ggplot(aes(x = dose, y = pct)) +
  geom_point(aes(size = n)) +
  xlab("Dose (log mg/L)") +
  ylab("Mortality rate (%)") +
  scale_y_continuous(labels = scales::percent) +
  # xlab(bquote(log[10]), bquote(CS[2])) +
  scale_size(range = c(1, 2))

print(plot1)
```

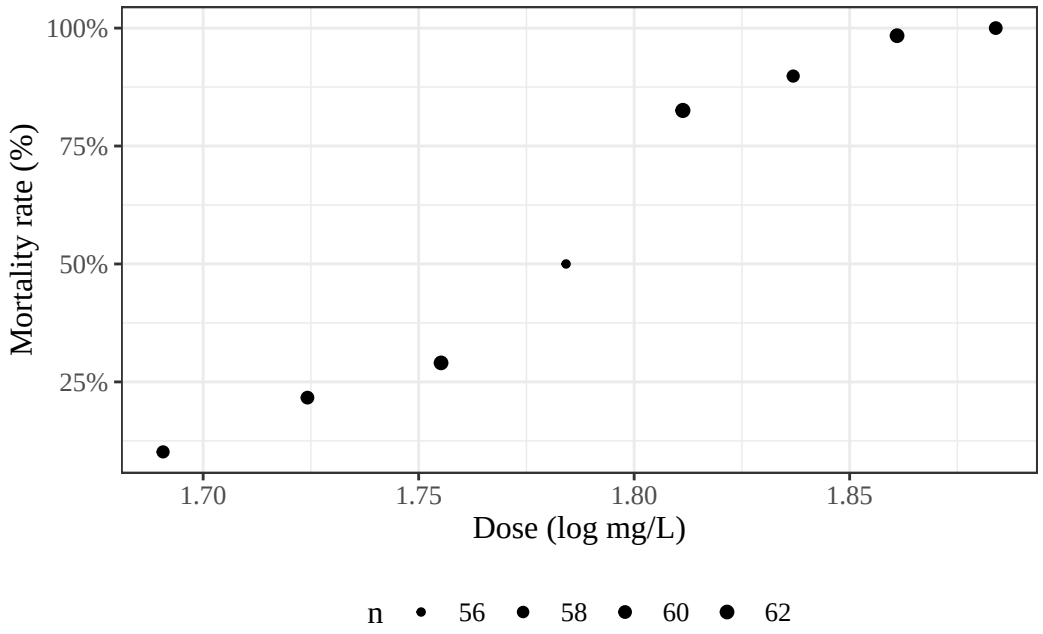

Figure 5: Mortality rates of adult flour beetles after five hours' exposure to gaseous carbon disulphide (Bliss 1935)

1.4.6 Why don't we use linear regression?

```
beetles_long <-
  beetles |>
  reframe(
    .by = everything(),
    outcome = c(
      rep(1, times = died),
      rep(0, times = survived)
    )
  )

lm1 <-
  beetles_long |>
  lm(
    formula = outcome ~ dose,
    data = _
  )


range1 <- range(beetles$dose) + c(-.2, .2)

f_linear <- function(x) predict(lm1, newdata = data.frame(dose = x))

plot2 <-
  plot1 +
  geom_function(fun = f_linear, aes(col = "Straight line")) +
  labs(colour = "Model", size = "")
print(plot2)
```

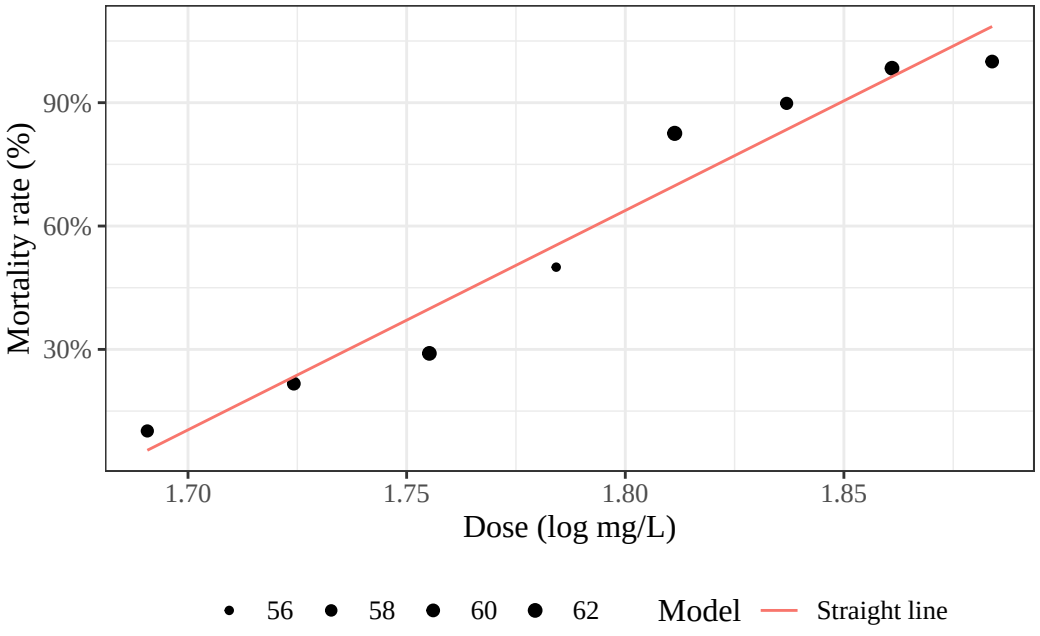

Figure 6: Mortality rates of adult flour beetles after five hours' exposure to gaseous carbon disulphide (Bliss 1935)

1.4.7 Zoom out

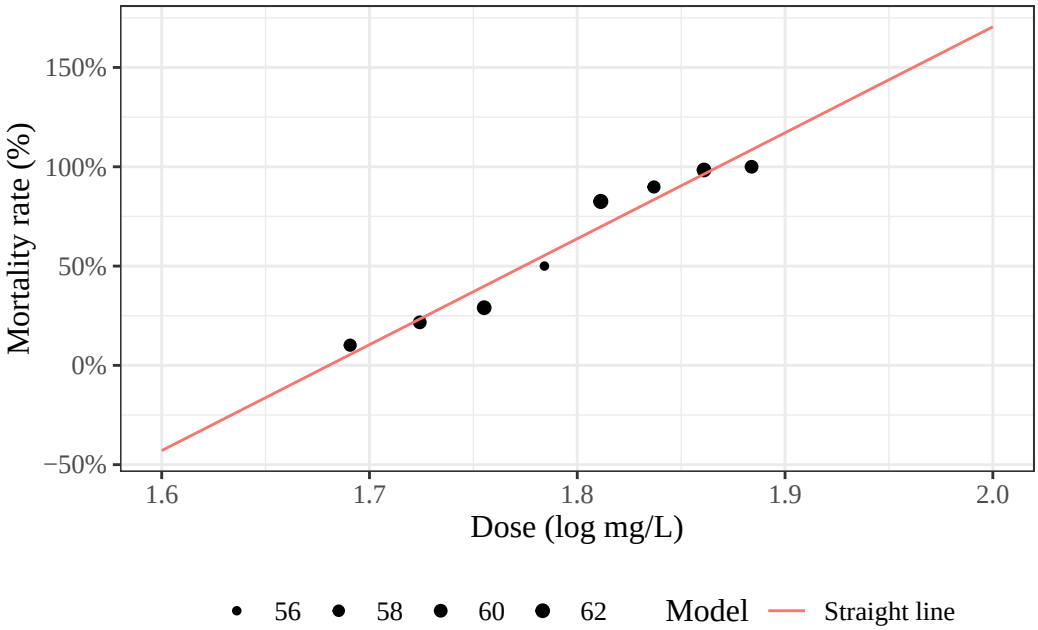

Figure 7: Mortality rates of adult flour beetles after five hours' exposure to gaseous carbon disulphide (Bliss 1935)

1.4.8 log transformation of dose?

```
lm2 <-
  beetles_long |>
  lm(formula = outcome ~ log(dose), data = _)

f_linearlog <- function(x) predict(lm2, newdata = data.frame(dose = x))

plot3 <- plot2 +
  expand_limits(x = c(1.6, 2)) +
  geom_function(fun = f_linearlog, aes(col = "Log-transform dose"))

print(plot3 + expand_limits(x = c(1.6, 2)))
```

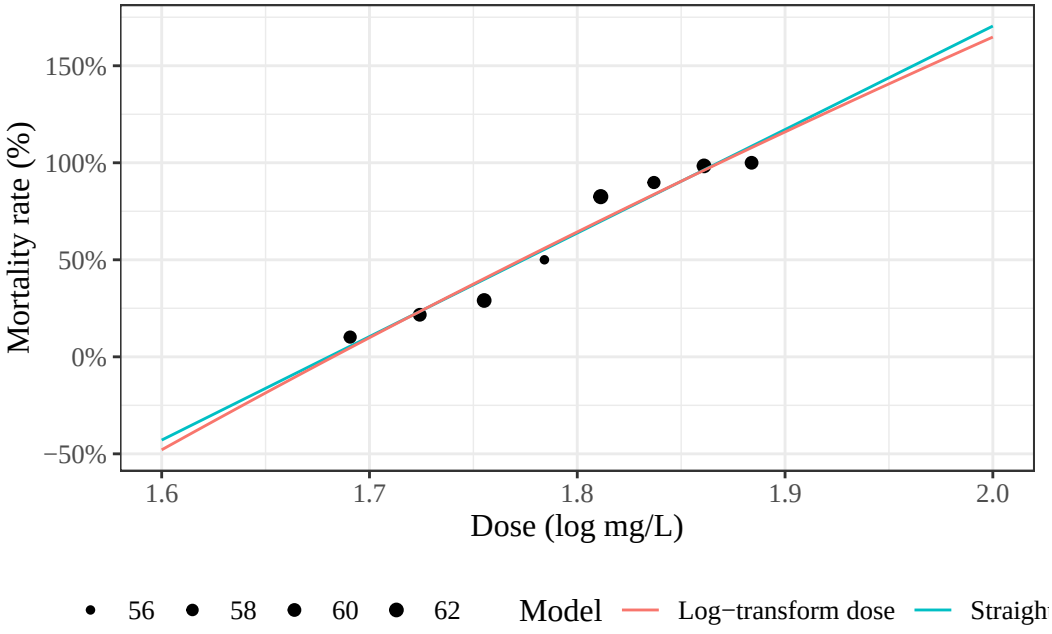

Figure 8: Mortality rates of adult flour beetles after five hours' exposure to gaseous carbon disulphide (Bliss 1935)

1.4.9 Logistic regression

```
glm1 <- beetles |>
  glm(formula = cbind(died, survived) ~ dose, family = "binomial")

f <- function(x) {
  glm1 |>
    predict(newdata = data.frame(dose = x), type = "response")
}

plot4 <- plot3 + geom_function(fun = f, aes(col = "Logistic regression"))
print(plot4)
```

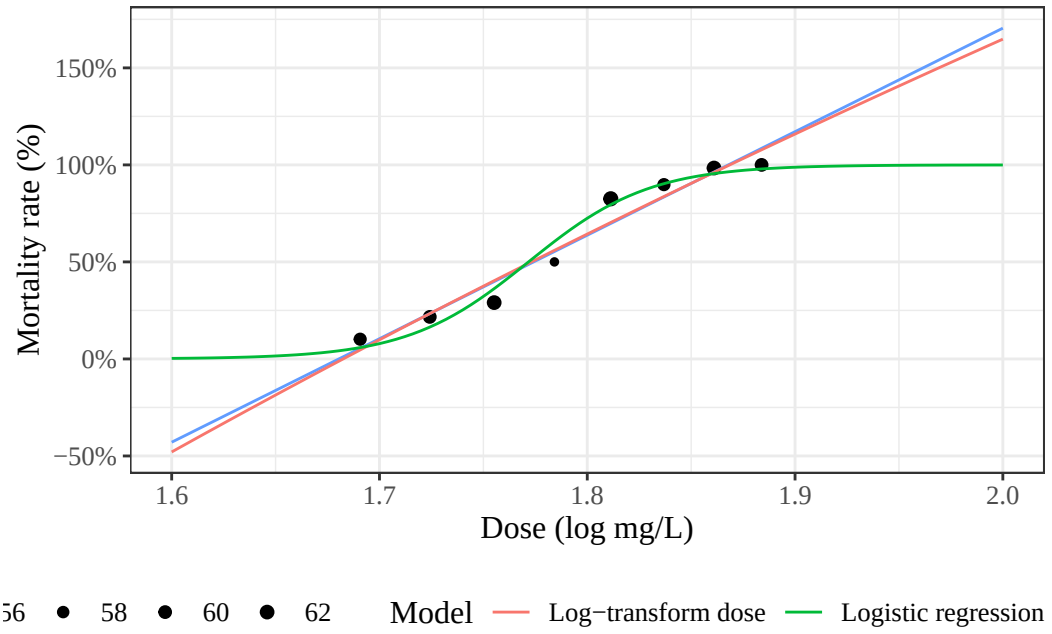

Figure 9: Mortality rates of adult flour beetles after five hours' exposure to gaseous carbon disulphide (Bliss 1935)

1.5 Structure of regression models

Exercise 1.2. What is a regression model?

Definition 1.1 (Regression model)**.** Regression models are conditional probability distribution models:

$$\mathrm{P}(Y|\tilde{X})$$

Exercise 1.3. What are some of the names used for the variables in a regression model $\mathrm{P}(Y|\tilde{X})$?

Definition 1.2 (Outcome)**.** The outcome variable in a regression model is the variable whose distribution is being described; in other words, the variable on the left-hand side of the "|" ("pipe") symbol.

The outcome variable is also called the **response variable**, **regressand**, **predicted variable**, **explained variable**, **experimental variable**, **output variable**, **dependent variable**, **endogenous variables**, **target**, or **label**.

and is typically denoted Y.

Definition 1.3 (Predictors)**.** The predictor variables in a regression model are the conditioning variables defining subpopulations among which the outcome distribution might vary.

Predictors are also called **regressors**, **covariates**, **independent variables**, **explanatory variables**, **risk factors**, **exposure variables**, **input variables**, **exogenous variables**, **candidate variables** (Dunn and Smyth (2018)), **carriers** (Dunn and Smyth (2018)), **manipulated variables**, or **features** and are typically denoted $\tilde{X}$. [2]

Table 1: Common pairings of terms for variables $\tilde{X}$ and Y in regression models $P(Y|\tilde{X})$ [4]

$\tilde{X}$	Y	usual context
input	output	
independent	dependent	
predictor	predicted or response	
explanatory	explained	
exogenous	endogenous	econometrics
manipulated	measured	randomized controlled experiments
exposure	outcome	epidemiology
feature	label or target	machine learning

Exercise 1.4. What is the general structure of a generalized linear model?

Solution 1.2. Generalized linear models have three components:

1. The **outcome distribution** family: $\mathrm{p}(Y|\mu(\tilde{x}))$
2. The **link function**: $g(\mu(\tilde{x})) = \eta(\tilde{x})$
3. The **linear component**: $\eta(\tilde{x}) = \tilde{x} \cdot \beta$

---

1. The **outcome distribution** family (a.k.a. the **random component** of the model)

- Gaussian (normal)
- Binomial
- Poisson
- Exponential
- Gamma
- Negative binomial

---

2. The **linear component** (a.k.a. the *linear predictor* or *linear functional form*) describing how the covariates combine to define subpopulations:

$$\eta(\tilde{x}) \stackrel{\text{def}}{=} \tilde{x}^\top \tilde{\beta} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ...$$

[2] The "~" ("tilde") symbol in the notation $\tilde{X}$ indicates that $\tilde{X}$ is a vector. See the appendices[3] for a table of notation used in these notes.

[4] adapted from https://en.wikipedia.org/wiki/Dependent_and_independent_variables#Synonyms

3. The **link function** relating the outcome distribution to the linear component, typically through the mean:

- identity: $\mu(y) = \eta(\tilde{x})$
- logit: $\log\left\{\frac{\mu(y)}{1-\mu(y)}\right\} = \eta(\tilde{x})$
- log: $\log\{\mu(y)\} = \eta(\tilde{x})$
- inverse: $(\mu(y))^{-1} = \eta(\tilde{x})$
- clog-log: $\log\{-\log\{1-\mu(y)\}\} = \eta(\tilde{x})$

Components 2 and 3 together are sometimes called the **systematic component** of the model (for example, in Dunn and Smyth (2018)).

Dalgaard, Peter. 2008. *Introductory Statistics with r*. New York, NY: Springer New York. https://link.springer.com/book/10.1007/978-0-387-79054-1.

Dunn, Peter K, and Gordon K Smyth. 2018. *Generalized Linear Models with Examples in R*. Vol. 53. Springer. https://link.springer.com/book/10.1007/978-1-4419-0118-7.

Wickham, Hadley, Mine Çetinkaya-Rundel, and Garrett Grolemund. 2023. *R for Data Science*. " O'Reilly Media, Inc.". https://r4ds.hadley.nz/.