6 The linear model

6.1 Simple linear model

The aim of the linear model is to model a response variable y as a function ($f(\cdot)$) of one covariate x_1. For any value of the covariate the expected value of y is given by

$$E(y|x_1) = f(x_1)$$

We model the expectation (E) of y given the covariate x_1.

A special case is the simple linear regression, where $f(x) = \beta_0 + \beta_1 x + \epsilon$

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i \qquad \epsilon_i \sim N(0, \sigma^2)$$

β_0 is the intercept (where the line intercepts the y-axis) and β_1 is the slope, i.e., how much does y change if we increase x by one unit. Finally, ϵ are the residuals (i.e., what ever is left over after fitting the model) and we assume that the residuals follow a normal distribution with mean 0 and a **constant** variance of σ^2. Hence, we can rewrite the simple linear model as:

$$y_i = N(\beta_0 + \beta_1 x_i, \sigma^2)$$

6.1.1 Estimating coefficients

We can obtain estimates for β_0 and β_1, using the methods of least squares. That is we find β_0 and β_1 such that the residuals are minimized. For the simple linear model, we can estimate β_0 by

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

and β_1 by:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i y_i - \bar{y} x_i)}{\sum_{i=1}^{n}(x_i^2 - \bar{x} x_i)}$$

and finaly the residual variances with:

$$\hat{\sigma}^2 = \frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

6.1.2 Assumptions

We take the following assumptions concerning the data:

- $\epsilon_i \sim N(0, \sigma^2)$.
- all ϵ_i are independent of each other.
- additionally covariates (x_is) must not be correlated.

6.2 Multiple linear regression

The multiple linear model is an extension of the simple linear models and allows for more than one predictor (i.e., we have more that one covariate).

We extend the simple linear model

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \quad i = 1, \dots, n$$

to a slighty more complex version with with p covariates

$$y_i = \beta_0 + \beta_1 x_{ik} + \dots + \beta_k x_{ik} + \epsilon_i, \quad i = 1, \dots, n \quad \text{und} \quad k = 1, \dots, p$$

Where: - $i = 1, \dots, n$ are the number of data points. - $k = 1, \dots, p$ are the number of covariates.

6.2.1 Matrix notation

For the multiple linear model it is often easier to use the matrix notation. We can rewrite the simple linear regression

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

in matrix form as:

$$\mathbf{y} = \mathbf{X} \quad +$$

Here, $\mathbf{X}$ is known as the **design matrix** or sometimes also the model matrix.

Some rearranging (not part of this course) lets us solve for as

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

In R, we can use the following commands to solve for the estimates

```
b.hat <- solve(t(X) %*% X) %*% t(X) %*% y
```

Note, we make use of some algebra in R: `%*%` is used for matrix multiplication, `t()` to transpose a matrix and `solve()` to invert a matrix.

Exercise 1: Fitting a linear model

Use the `trees` data set (located in `data/trees.rds`) and select only spruce trees for the year 2020. First, investigate graphically, if you think that *bio18* (percipitation in the warmest month) could have a linear effect on `mean_damage`. Then fit a linear model for this relationship and give an interpretation of the estimated intercept, slope and variance parameter.

6.3 Visualize a linear model

6.3.1 Predicting with `ggeffects`

The `ggeffects` packages makes predictions straight-forward. The function `predict_response()` allows to easily create predictinos for all covariates specified with the `terms`-argument. Terms can be named as character with special syntax. For example `x [20:40]` would predict the variable `x` for an interval from 20 to 40. Alternatively, a list can be passed to term. To achieve the same as above, you would specify `terms = list(x = 20:40)`[1].

Improtant: the covariates specified with `terms` will be labeled as `x`, `group`, `facet`, `panel` and `grid` respectively.

6.3.2 Confidence interval vs Prediction interval

We can calculate the uncertainty for a new prediction by

$$\hat{y} \pm t_{n-p}^{\alpha/2} SE$$

The standard error SE is different for a confidence interval and a prediction interval.

- A **confidence interval** accounts for uncertainties in the predicted conditional mean due to sampling error.
- A **prediction interval** accounts for uncertainties in a predicted data point due to sampling error **and** variability of individuals around the mean.

To calculate the standard error for a confidence interval (SE_c) and a prediction interval (SE_p) use

- $$SE_c = \hat{\sigma^2}\sqrt{x_0^t(X^tX)^{-1}x_0}$$
- $$SE_p = \hat{\sigma^2}\sqrt{x_0^t(X^tX)^{-1}x_0 + 1}$$

with $\hat{\sigma^2}$ being the estimated residual variance and x_0 the design matrix with the new observations.

In R:

- The function `predict()` has an additional argument `interval`.
- The default is `none`, but could also take the values `confidence` or `prediction`.

Exercise 2: Visualize a fitted model

Extent the model from exercise 1, by addeing elevation as an additional covariate. Give a brief interpretation of elevation. Then visualize model by choosing three typical values for elevation (colored lines) and show the values for bio18 on the x-axis and the predicted mean loss on the y-axis. Can you create three variants of the plot? Once without uncertainty, once withe a prediction interval and once with a confidence interval?

[1] See here for more details: https://strengejacke.github.io/ggeffects/articles/introduction_effectsatvalues.html

6.4 Factors and interactions

The effect of one covariates depends at least on an other covariate (i.e., their joint effect might be different).

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \epsilon$$

Ways to specify interactions in R:

- `y = x1 + x2 + I(x1 * x2)`, or
- `y = x1 + x2 + x1:x2`, or
- `y = x1 * x2`

6.5 Dummy variables

To model a categorical variable with c levels using dummy coding, $c - 1$ dummy variables are created. With

$$x_{i,1} = \begin{cases} 1 & x_i = 1 \\ 0 & \text{else} \end{cases} \dots x_{i,c-1} = \begin{cases} 1 & x_i = c - 1 \\ 0 & \text{else} \end{cases}$$

The c-th level is the reference level.

R does this automatically, if the a predictor is a `character` or a `factor`.

Exercise 3: Visualize a fitted model

Extend the data set from Exercise 1 by also including beeches and oaks. Fit two models: 1) a linear model where you try to explain variation in the mean leaf loss as a function of *bio18* and *species*, and 2) second where you allow an interaction between *bio18* and *species*. Give a brief interpretation of the results. Visualize model both model, what do you notice?

6.6 Residuals

Assumptions for the residuals

1. Approximate linearity between x and y.
2. The expectation of the residuals is 0.

3. The variance of the residuals is constant.
4. The residuals are uncorrelated.
5. The residuals $\epsilon_i \sim N(0, \sigma^2)$ distributed.

If the model is good, then

- there should be no obvious pattern in the previous plot.
- all residuals should be centered around 0.
- the variance should be constant.

The Cook's distance D_i for the i-th data point, tells us how much the model changes if the i-th data point is removed.

$$D_i = \frac{\sum_{j=1}^{n} (\hat{y}_j - \hat{y}_{j(-i)})^2}{ps^2}$$

with

- $\hat{y}_{j(-i)}$ being the fitted responds with excluding the i-th data point.
- p is the number of predictors.
- $s^2 = \frac{\mathbf{e}'\mathbf{e}}{n-p}$

Different thresholds exist when a point is considered influential. For example $D_i > 0.5$, $D_i > 1$, or $D_i > 3\bar{D}$.

6.7 Goodness of Fit

6.7.1 Coefficient of determination

$$R^2 = 1 - \frac{RSS}{TSS}$$

$$RSS = \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 = \sum_{i=1}^{n} \epsilon_i^2$$

$$TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

Better to use the adjusted R^2 ($\bar{R}^2$)

$$\bar{R}^2 = 1 - (1 - R^2) \frac{n-1}{n-p-1}$$

With n being the number of data points and p the number of predictors.

The R^2 always ranges from 0 to 1.

Exercise 4: Comparing models

Compare the model of Exercise 1 and 2 with the adjusted R^2, which one do you think is better?

Home work: Fit a linear model in three ways

Try to estimate the coefficients of the model (`mean_loss ~ bio18 + ele`) from exericse 2 in three different ways:

1. With ordinary least square using the `lm()`-function.
2. With the `brms`-package using the function `brm()`.
3. With Stan (optionally, also with matrix notation).