# Chapter 6

# Practical

# 6.1 Practical 1: Linear Mixed Models with R

- We will illustrate some basic linear mixed models analysis

- We will use the PBC dataset; this is available as the object `pbc2` in the R workspace you have received

- We will need the following variables
  * `id`: patient id number
  * `serBilir`: serum bilirubin (the response variable of interest)
  * `year`: follow-up times in years
  * `drug`: the randomized treatment
  * `sex`: the gender of the patients
  * `age`: the age of the patients

# 6.1 Practical 1: Lin. Mixed Models with R (cont'd)

- The response variable we will use will be the natural logarithm of `serBilir`
- We start with some descriptive plots; load the **lattice** package using: `library("lattice")` (or your favorite graphics package, e.g., **ggplot2**)
- T1: Plot the average longitudinal evolutions of the two treatment groups using loess. Should we or should we not trust this plot?
- T2: Do the same plot for sex

# 6.1 Practical 1: Lin. Mixed Models with R (cont'd)

- T3: Create the plot of the subject-specific longitudinal trajectories
  - ▷ it will be useful to save the plots in a pdf, using `pdf()` before executing the plot and `dev.off()` afterwards

- T4: As an initial analysis we will test for a treatment effect using the AUC
  - ▷ calculate the AUC for each subject (see p. 31)
  - ▷ do a $t$-test for the difference in the AUC between the two treatment groups

# 6.1 Practical 1: Lin. Mixed Models with R (cont'd)

- We will proceed by fitting appropriate linear mixed models to the data

- One approach to graphically investigate the variance function over time is to smooth the squared OLS residuals
  - ▷ in order the OLS residuals to correctly reflect the properties of the marginal covariance matrix of the response variable, it is important to remove all systematic trends
  - ▷ hence we want to fit an elaborate mean structure linear model
  - ▷ we will allow for nonlinear time evolutions using natural cubic splines
  - ▷ correct for `sex`, `drug` and `age` + interactions of the time effect with `sex` and `drug`

# 6.1 Practical 1: Lin. Mixed Models with R (cont'd)

- A bit of motivation and background for splines: When modeling continuous covariates it is customary to assume that such covariates affect linearly the response

- However, this assumption is very restrictive, and in many real applications it may not hold
  - ▷ increasing age from 20y to 25y does not increase the risk in the same amount as increasing age from 60y to 65y
  - ▷ similar conjectures also can be made for the time effect in a longitudinal setting

- Wrongly assuming linearity may affect the resulting inference for such covariates as well as the predictive ability of the model

# 6.1 Practical 1: Lin. Mixed Models with R (cont'd)

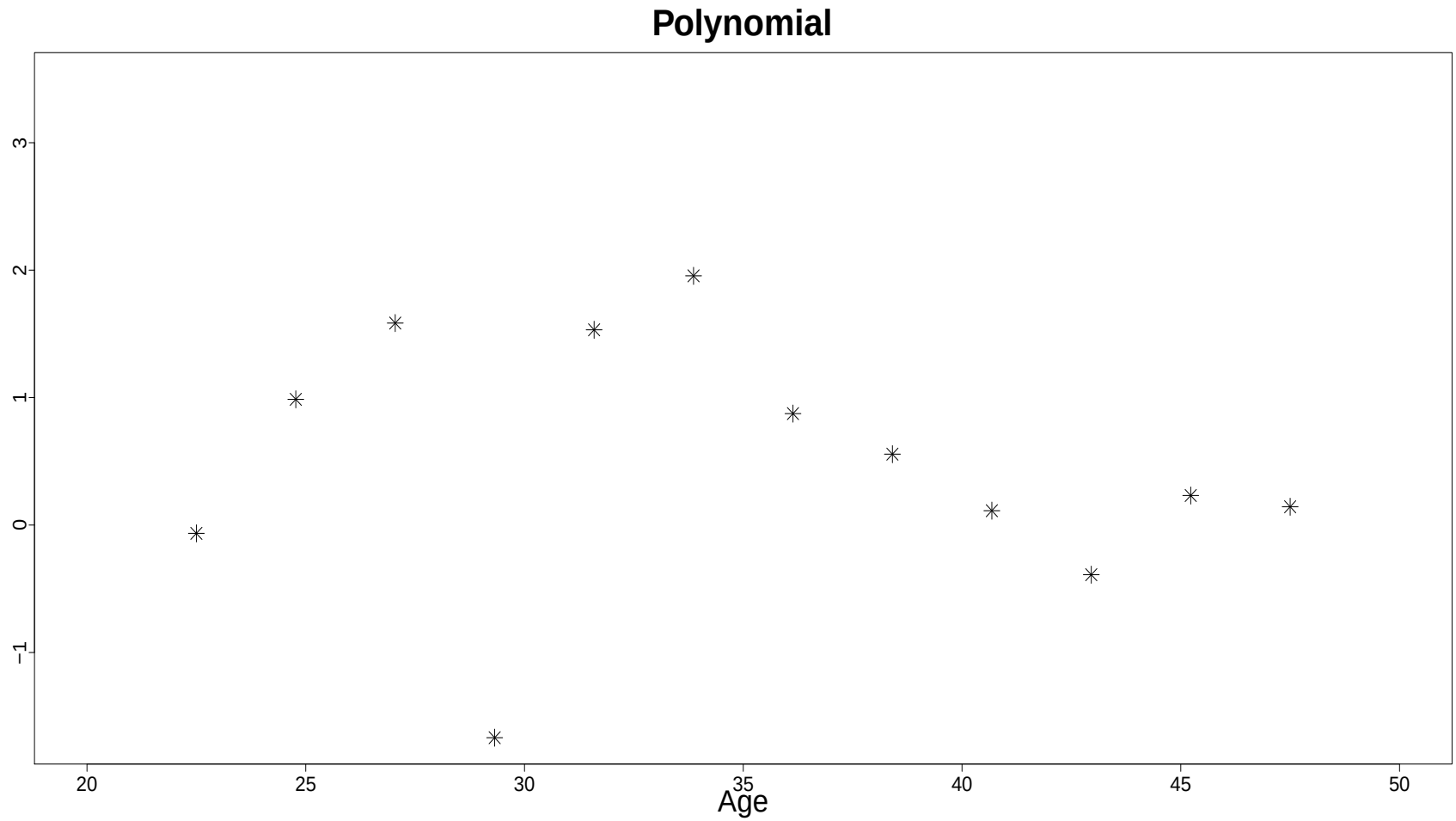
- Therefore, it is highly advisable not to restrict a priori the effects of continuous predictors to be linear and let the data tell you the true story

- The easiest way to relax linearity is to assume polynomial effects

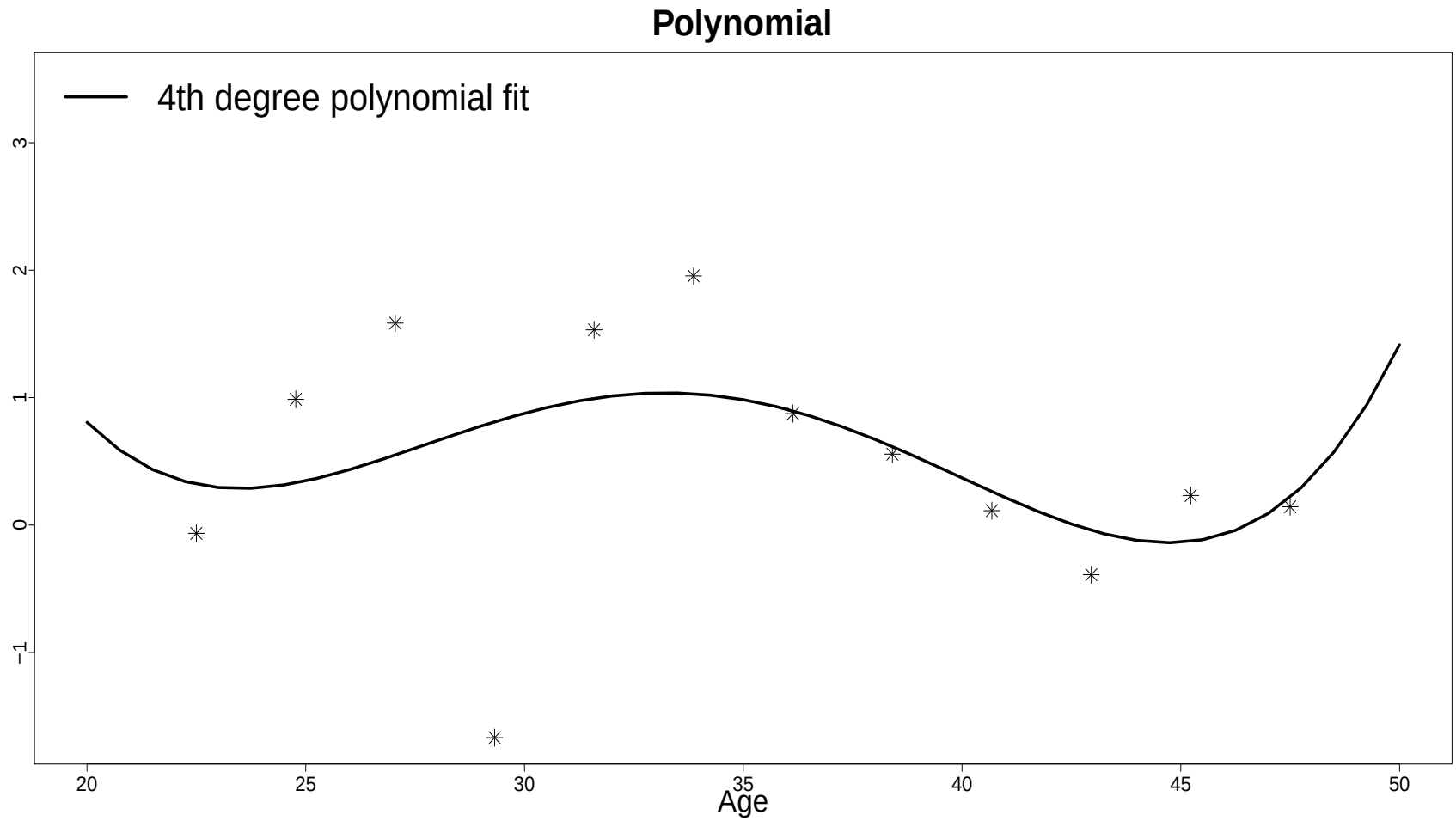
$$\beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \beta_3 X_i^3 + \ldots$$

- However, polynomials have some disadvantages, namely
  - ▷ they are not local $\Rightarrow$ changing one data point will affect the overall fit
  - ▷ numerically ill-conditioned (however, not too worrisome with modern software)

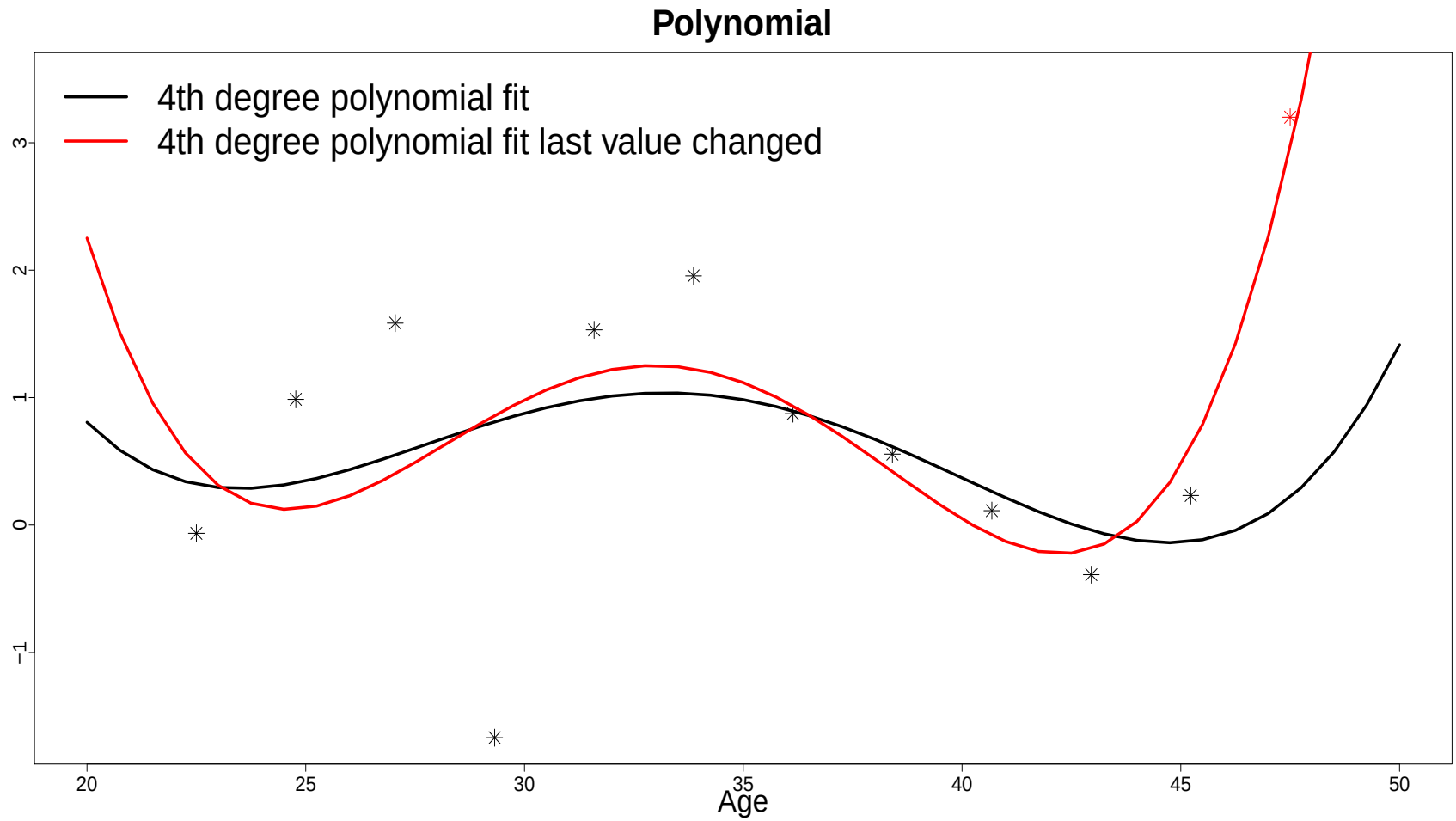
# 6.1 Practical 1: Lin. Mixed Models with R (cont'd)



# 6.1 Practical 1: Lin. Mixed Models with R (cont'd)



# 6.1 Practical 1: Lin. Mixed Models with R (cont'd)

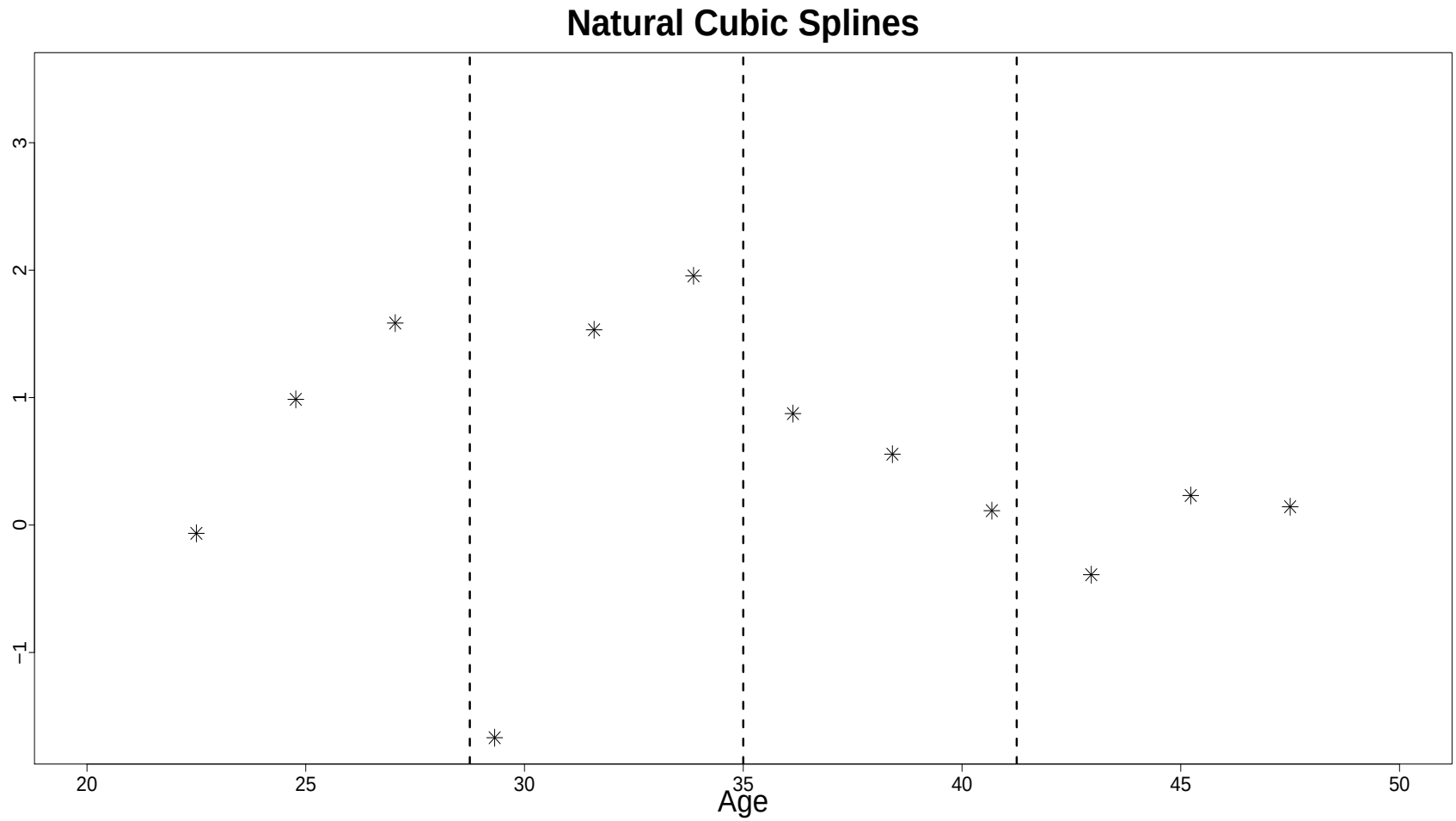


# 6.1 Practical 1: Lin. Mixed Models with R (cont'd)

- An alternative approach to relax the linearity assumption of continuous predictors is to use regression splines

- Idea behind regression splines: use polynomials but locally
  - ▷ split the range of values of the continuous predictor into subintervals using a series of knots
  - ▷ within each subinterval assume that the effect of the predictor is nonlinear and can be approximated by a cubic polynomial
  - ▷ put extra smoothness assumptions, i.e., the cubic polynomial fits between neighboring subintervals must be connected

# 6.1 Practical 1: Lin. Mixed Models with R (cont'd)

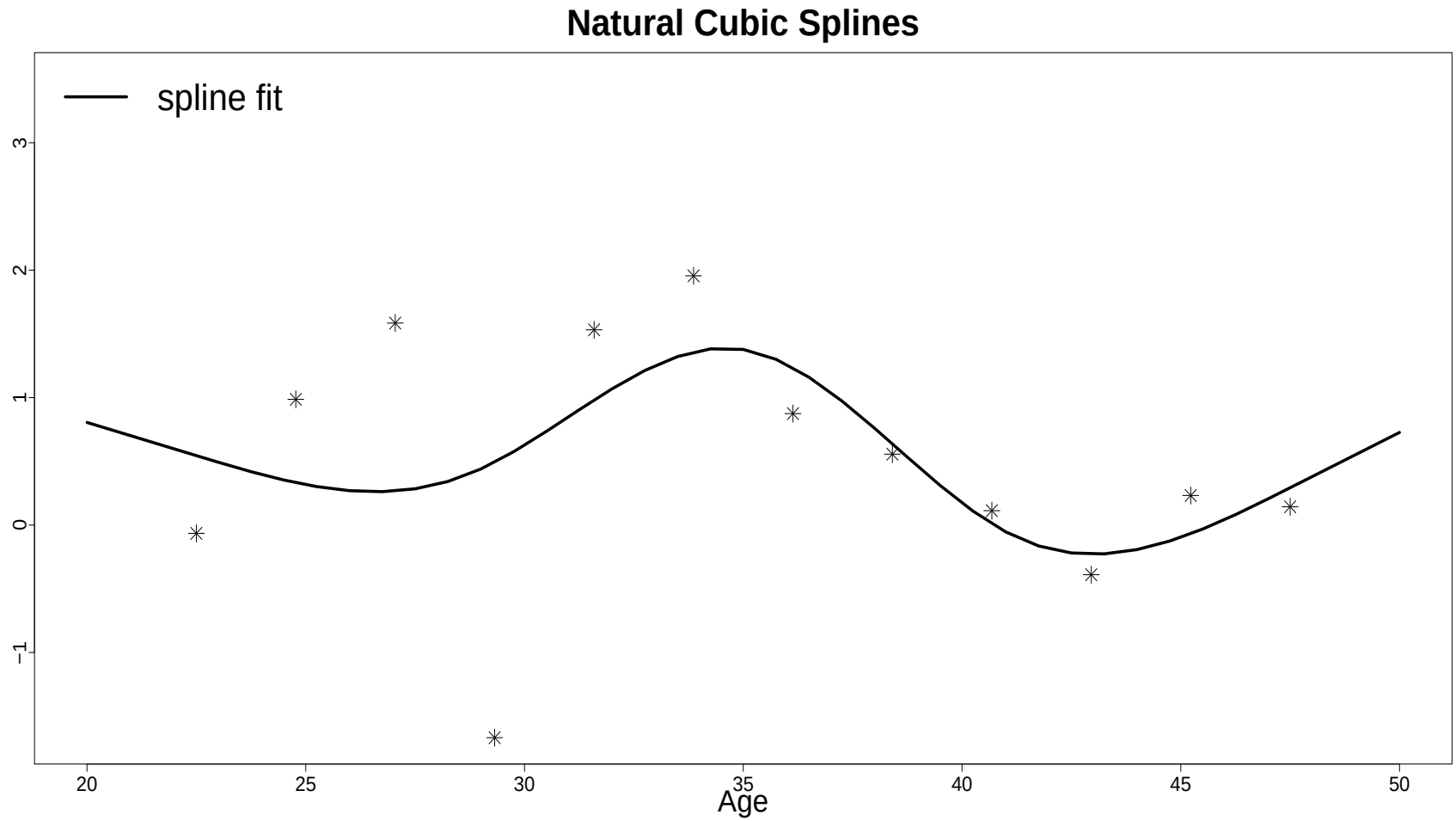


# 6.1 Practical 1: Lin. Mixed Models with R (cont'd)

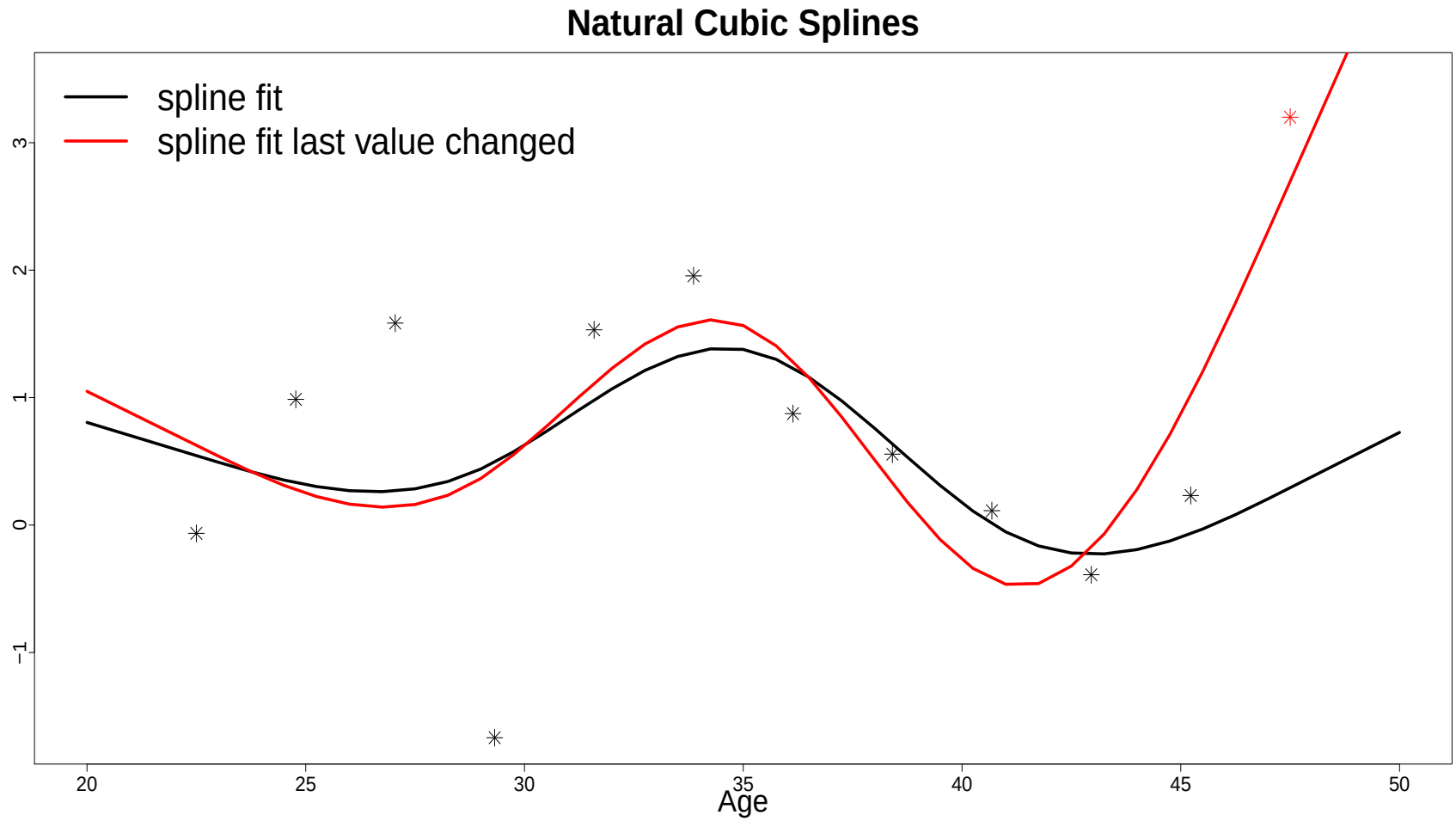
- There are several types of regression splines available
  - ▷ advisable to use natural cubic splines, which assume linearity outside the boundary knots – better statistical properties

- Other approaches (we are not going to discuss them here)
  - ▷ penalized splines
  - ▷ local regression
  - ▷ wavelets
  - ▷ ...

# 6.1 Practical 1: Lin. Mixed Models with R (cont'd)



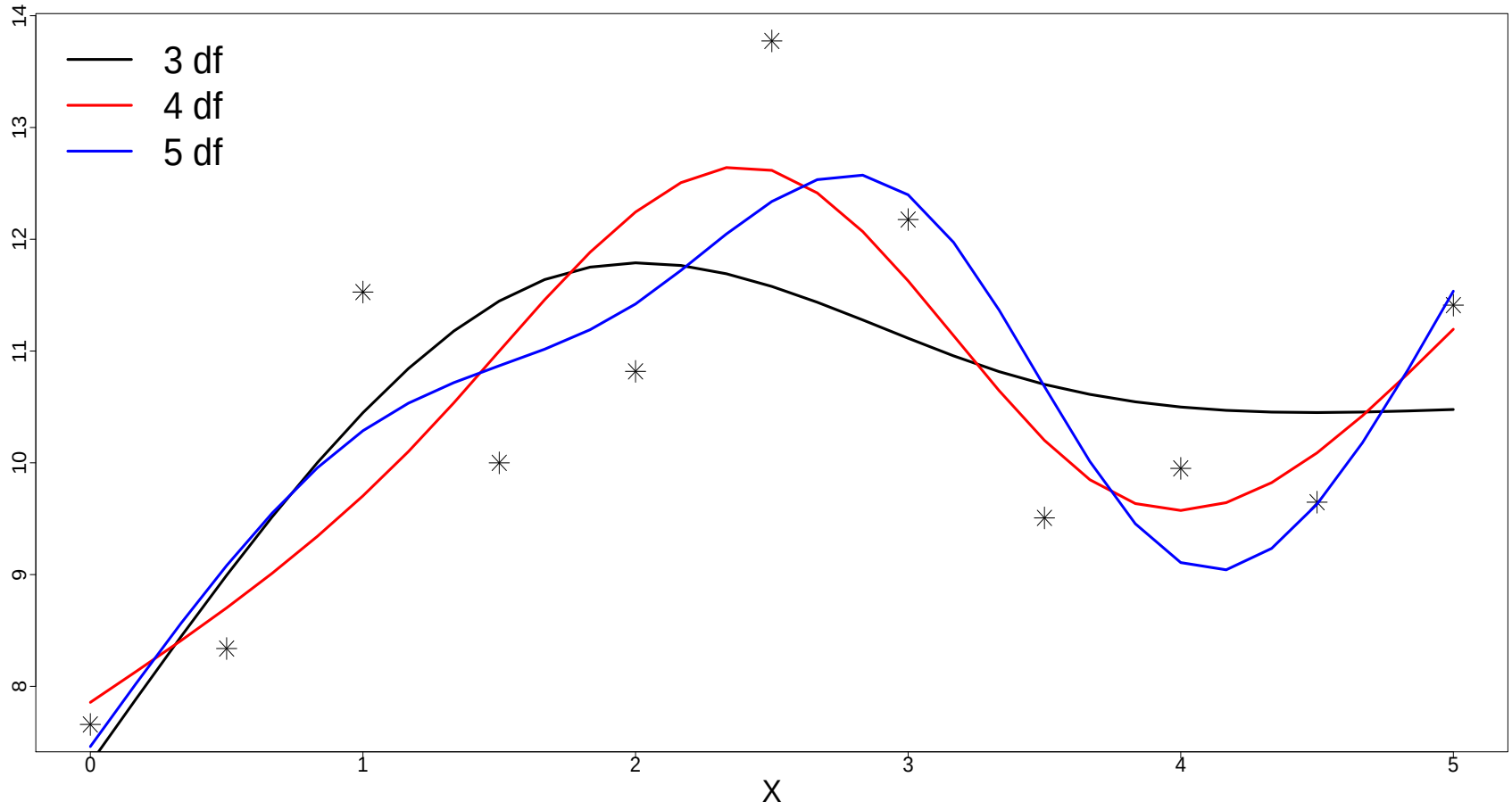
# 6.1 Practical 1: Lin. Mixed Models with R (cont'd)



# 6.1 Practical 1: Lin. Mixed Models with R (cont'd)

- As also in the case of the polynomials, we can tune the degree of nonlinearity by specifying the degrees of freedom for the spline
  - ▷ increasing the degrees of freedom results in more flexible modeling
  - ▷ bias-variance tradeoff

# 6.1 Practical 1: Lin. Mixed Models with R (cont'd)



# 6.1 Practical 1: Lin. Mixed Models with R (cont'd)

- T5: Calculate the squared OLS residuals for the above defined linear regression model, and do the loess plot
  - ▷ load package **splines** using `library("splines")` in order to make the spline functions available
  - ▷ the function that can be used to fit natural cubic splines is `ns()` and it can be directly included in a model formula
  - ▷ fit the above defined model using function `lm()`
  - ▷ extract the residuals using function `resid()`
  - ▷ make the plot of the squared residuals using `xyplot()` (or your favorite plotting function)

# 6.1 Practical 1: Lin. Mixed Models with R (cont'd)

- We will start our model-building exercise...

- General recipe: First model the covariance structure and then the mean structure
  - ▷ start with an elaborate mean model (i.e., in order to be more or less certain that we have removed all systematic trends)
  - ▷ build up the random-effects structure, starting from random intercepts, random intercepts and random slopes, etc. until you find a satisfying model
  - ▷ then return to the mean structure and simplify it if required

# 6.1 Practical 1: Lin. Mixed Models with R (cont'd)

- T6: Fit a linear mixed model with mean structure the same as the one you used in the simple linear model to calculate the OLS residuals in T5, and random intercepts – you will need to load package **nlme** first using `library("nlme")`

- T7: Continue on elaborating the random-effects structure and perform likelihood ratio tests (using function `anova()`) to see if the additional random effects are required
  - ▷ random intercepts & random slopes
  - ▷ random intercepts & splines for the time effect

# 6.1 Practical 1: Lin. Mixed Models with R (cont'd)

- Technical/Theorical Issue: Consider the hypothesis test between the random intercepts and the random intercepts & random slopes models
  - ▷ random intercepts model

$$y_{ij} = X\beta + b_{i0} + \varepsilon_{ij}, \qquad b_{i0} \sim \mathcal{N}(0, \sigma^2_{b1})$$

  - ▷ random intercepts & random slopes model

$$y_{ij} = X\beta + b_{i0} + b_{i1}t + \varepsilon_{ij}, \qquad b_{i0} \sim \mathcal{N}(0, D)$$

with

$$D = \begin{bmatrix} \sigma^2_{b1} & \sigma_{b12} \\ \sigma_{b12} & \sigma^2_{b2} \end{bmatrix}$$

# 6.1 Practical 1: Lin. Mixed Models with R (cont'd)

- Hence, the hypotheses to be tested are

$$H_0: \quad \sigma^2_{b2} = \sigma_{b12} = 0$$

$$H_a: \quad \sigma^2_{b2} \neq 0 \text{ or } \sigma_{b12} \neq 0$$

- What is the problem? The null hypothesis for $\sigma^2_{b2}$ is on the boundary of its corresponding parameter space
  - ▷ statistical tests derived from standard ML theory assume the $H_0$ is an interior point of the parameter space
  - ▷ the classical asymptotic $\chi^2$ distribution for the likelihood ratio test statistic does not apply

# 6.1 Practical 1: Lin. Mixed Models with R (cont'd)

- For simple settings (as the one above), it has been proposed to use a mixture of $\chi^2$ distributions
  - ▷ nonetheless, it has been suggested that this does not always work satisfactorily (e.g., see package **RLRsim** and the references therein)

- Here we will just use the $\chi^2$ distribution and be a bit conservative

- T8: Continue by relaxing the fixed-effects structure
  - ▷ start be checking if all interaction terms can be dropped using a likelihood ratio test
  - ▷ due to a numerical problem, fit first again the final model of T7 assuming a diagonal matrix for the random effects – this can be done by using function `pdDiag()` in the `random` argument of `lme()`

# 6.1 Practical 1: Lin. Mixed Models with R (cont'd)

- Technical/Theorical Issue: By default `lme()` fits linear mixed models using REML
  - ▷ REML estimation proceeds by transforming the response variable using the design matrix $X$
  - ▷ hence, by comparing linear mixed models with different fixed-effect structures, we are actually comparing models with different response variables $\Rightarrow$ LRT is not valid in models with different response variables

- T9: Re-fit the mixed model you ended up with in T8 using maximum likelihood instead of REML, and redo the LRT (check argument `method` of `lme()`)
  - ▷ continue by checking if any main effects may be dropped

# 6.1 Practical 1: Lin. Mixed Models with R (cont'd)

- T10: For the final model use function `summary()` to obtain a detailed output and interpret the results

# 6.2 Practical 2: Cox Models

- We will perform some basic survival analysis calculations and fit a series of Cox models for the AIDS dataset

- Start R and load package **survival**, using `library("survival")`

- Load the R workspace with the AIDS dataset

# 6.2 Practical 2: Cox Models (cont'd)

- We will need the following variables
  - * `Time`: observed event times in years
  - * `death`: the death indicator
  - * `drug`: the randomized treatment
  - * `gender`: the sex of the patients
  - * `AZT`: intolerance or failure
  - * `CD4`: the square root CD4 cell count at baseline

- T1: Calculate and plot the Kaplan-Meier estimator for the time to death
  - ▷ to compute the Kaplan-Meier estimator you will need function `survfit()`
  - ▷ to plot it, just use the `plot()` function on the resulting object

# 6.2 Practical 2: Cox Models (cont'd)

- T2: Calculate and plot the Kaplan-Meier estimator for the time to death, separately for the two treatment groups
  - ▷ what do you observe?

- T3: Calculate and plot the Kaplan-Meier estimator for the time to death, separately for males and females
  - ▷ what do you observe?

- T4: Calculate the log-rank tests for the two treatment groups and for males versus females
  - ▷ you will need function `survdiff()`, which has a very similar syntax as `survfit()`

# 6.2 Practical 2: Cox Models (cont'd)

- T5: We are interesting in studying the relationship between the hazard for death, and `drug`, `gender`, `AZT`, and `CD4`. Fit a Cox model that relaxes the linearity assumption for the effect of `CD4` using natural cubic splines (you need function `ns()`). In addition, assume that there is an effect `drug`, `gender` and `AZT` on the hazard for death, but the effect of these predictors is different for different levels of `CD4` cell count

  ▷ use the `summary()` method and try to interpret the results

- T6: Use a likelihood ratio test to test whether the model can be reduced by dropping all interaction terms

  ▷ use the `anova()` function

# 6.2 Practical 2: Cox Models (cont'd)

- T7: Use the `summary()` method to obtain a detailed summary of the second fitted model. What is the interpretation of the estimated coefficient for `drug`? In addition, in the output you have values for `exp(coef)` and `exp(-coef)`. What do these values represent?

- The main motivation to introduce the semiparametric Cox model was to avoid the impact of a possibly wrong assumption for the distribution of the event times

- However, all statistical models make assumptions – in the Cox model we make no assumption for the distribution of $T_i^*$ but we do make other assumptions:

  ▷ **proportional hazards (PH)**

# 6.2 Practical 2: Cox Models (cont'd)

- If PH is seriously violated, then the results we obtain from the Cox model may not be trustworthy!

- In practice, PH means that the effect of a covariate in the risk for an event is **constant over time**

- Some times the PH assumption may not be reasonable, e.g.,
  - ▷ the new treatment requires a time period to start working ⇒ at the beginning of follow-up the risk for the treatment group is the same as in the control group, however we expect that later the risk for the treatment group will decrease
  - ▷ ...

# 6.2 Practical 2: Cox Models (cont'd)

- To check the PH assumption we will (hypothetically) consider an extension of the Cox model, namely the Cox model with a *time-dependent coefficient*

$$h_i(t) = h_0(t) \exp\{X_i \beta(t)\}$$

where, the effect of $X$ on the hazard *varies* with time

- Grambsch and Therneau (Biometrika, 1994) have shown that, if $\widehat{\beta}$ is the estimated coefficient from the ordinary (time-independent) Cox model, then

$$\beta(t) \approx \widehat{\beta} + E\{s^*(t)\}$$

where $s^*(t)$ is the scaled Schoenfeld residual

# 6.2 Practical 2: Cox Models (cont'd)

- The formula and rationale behind the scaled Schoenfeld residuals is rather technical
  - ▷ we will not give them here (see Therneau & Grambsch (2000) for more info)

- Plotting scaled Schoenfeld residuals against time or suitable transformation of time, reveals violations of the PH assumption

- An additional advantage of the scaled Schoenfeld residuals is that they can be used to statistically test PH (though this is not advisable)

# 6.2 Practical 2: Cox Models (cont'd)

- T8: In R, plots of the Schoenfeld residuals are calculated by function `cox.zph()`
  - ▷ use this function on the final Cox model you fitted above
  - ▷ use the `plot()` function to produce the plots (before running `plot()`, run `par(mfrow = c(3, 3))`)
  - ▷ we will interpret together the results...

- T9: Check if conclusions change by using other transformations of the time variable (i.e., argument `transform` of `cox.zph()`)