MONASH University

ETC5521: Exploratory Data Analysis

Sculpting data using models, checking assumptions, co-dependency and performing diagnostics

Lecturer: *Di Cook*

ETC5521.Clayton-x@monash.edu

Week 11 - Session 2

Revisiting outliers

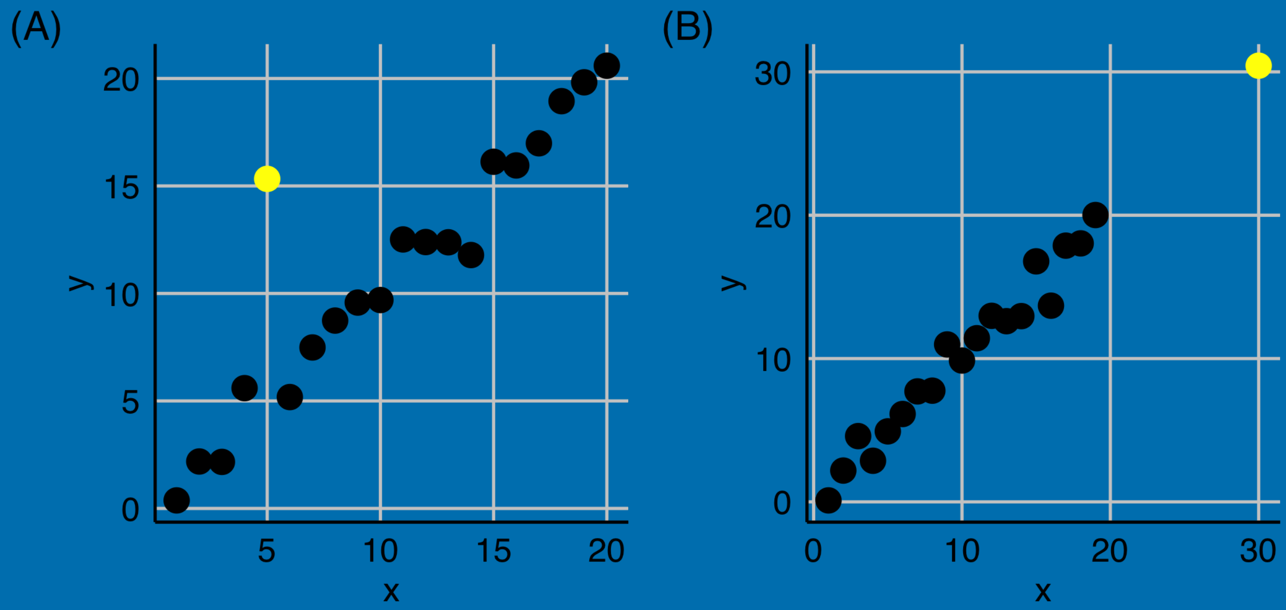
- We defined outliers in week 4 as "observations that are significantly different from the majority" when studying univariate variables.
- There is actually no hard and fast definition.

> We can also define an outlier as a data point that emanates from a different model than do the rest of the data.

- Notice that this makes this definition *dependent on the model* in question.

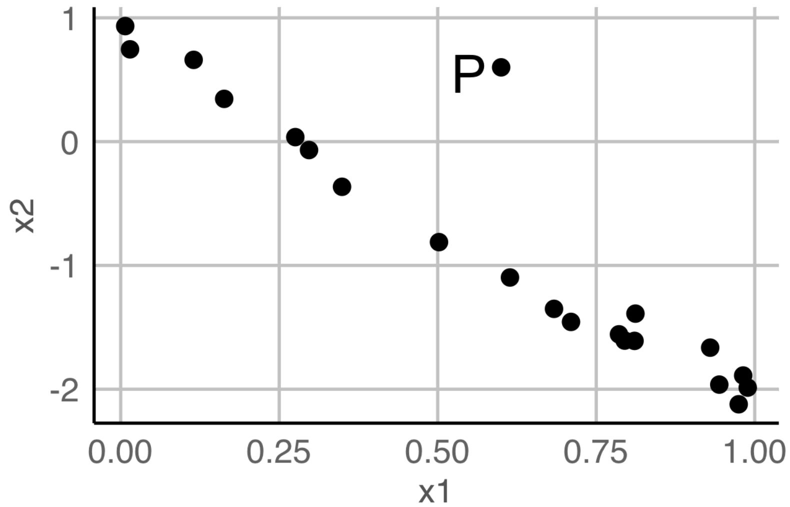
Pop Quiz

Would you consider the yellow points below as outliers?



Outlying values

- As with simple linear regression the fitted model should not be used to predict Y values for $\boldsymbol{x}$ combinations that are well away from the set of observed $\boldsymbol{x}_{\mathrm{i}}$ values.
- This is not always easy to detect!
- Here, a point labelled P has x_1 and x_2 coordinates well within their respective ranges but P is not close to the observed sample values in 2-dimensional space.
- In higher dimensions this type of behaviour is even harder to detect but we need to be on guard against extrapolating to extreme values.



Leverage

- The matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top$ is referred to as the **hat matrix**.
- The i-th diagonal element of $\mathbf{H}$, h_{ii}, is called the **leverage** of the i-th observation.
- Leverages are always between zero and one,

$$0 \leq h_{ii} \leq 1.$$

- Notice that leverages are not dependent on the response!
- Points with high leverage can exert a lot of influence on the parameter estimates

Leverage

On the data from the previous slide:

```
example_data
```

```
## # A tibble: 21 × 3
##       id    x1      x2
##    <int> <dbl>   <dbl>
##  1     1 0.982 -1.89
##  2     2 0.297 -0.0679
##  3     3 0.115  0.661
##  4     4 0.163  0.345
##  5     5 0.944 -1.96
##  6     6 0.795 -1.61
##  7     7 0.975 -2.12
##  8     8 0.349 -0.365
##  9     9 0.502 -0.812
## 10    10 0.810 -1.61
## # ℹ 11 more rows
```

Leverage

```
x <- as.matrix(example_data[2:3])
hat_matrix <- x %*% solve(t(x) %*% x) %*% t(x)
example_data %>%
  mutate(leverage = diag(hat_matrix)) %>%
  print(n = 21)
```

```
## # A tibble: 21 × 4
##        id     x1      x2 leverage
##     <int>  <dbl>   <dbl>    <dbl>
##  1      1 0.982  -1.89     0.105
##  2      2 0.297  -0.0679   0.0422
##  3      3 0.115   0.661    0.118
##  4      4 0.163   0.345    0.0656
##  5      5 0.944  -1.96     0.106
##  6      6 0.795  -1.61     0.0724
##  7      7 0.975  -2.12     0.123
##  8      8 0.349  -0.365    0.0230
##  9      9 0.502  -0.812    0.0275
```

Studentized residuals

- In order to obtain residuals with equal variance, many texts recommend using the **studentised residuals**

$$R_i^* = \frac{R_i}{\hat{\sigma}\sqrt{1 - h_{ii}}}$$

for diagnostic checks.

Cook's distance

- Cook's distance, D, is another measure of influence:

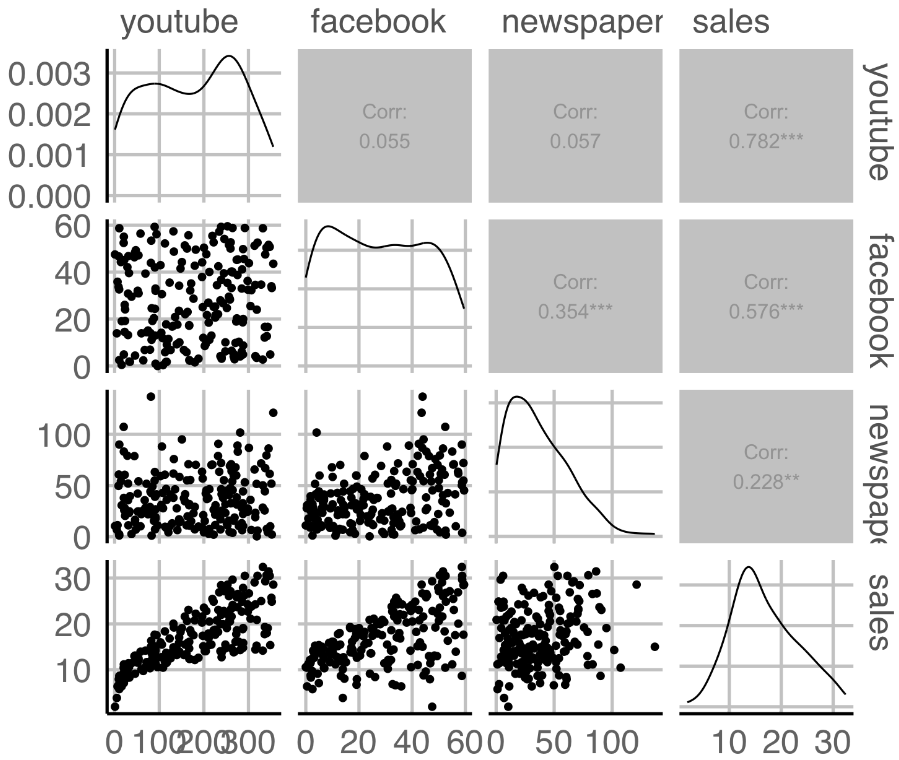
$$\begin{aligned} D_i &= \frac{(\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{[-i]})^\top \mathrm{Var}(\hat{\boldsymbol{\beta}})^{-1} (\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{[-i]})}{p} \\ &= \frac{R_i^2 h_{ii}}{(1 - h_{ii})^2 p \hat{\sigma}^2}, \end{aligned}$$

  where p is the number of elements in $\boldsymbol{\beta}$, $\hat{\boldsymbol{\beta}}_{[-i]}$ and $\hat{Y}_{j[-i]}$ are least squares estimates and the fitted value obtained by fitting the model ignoring the i-th data point $(\boldsymbol{x}_i, Y_i)$, respectively.

Case study 2 Social media marketing

Data collected from advertising experiment to study the impact of three advertising medias (youtube, facebook and newspaper) on sales.

data R



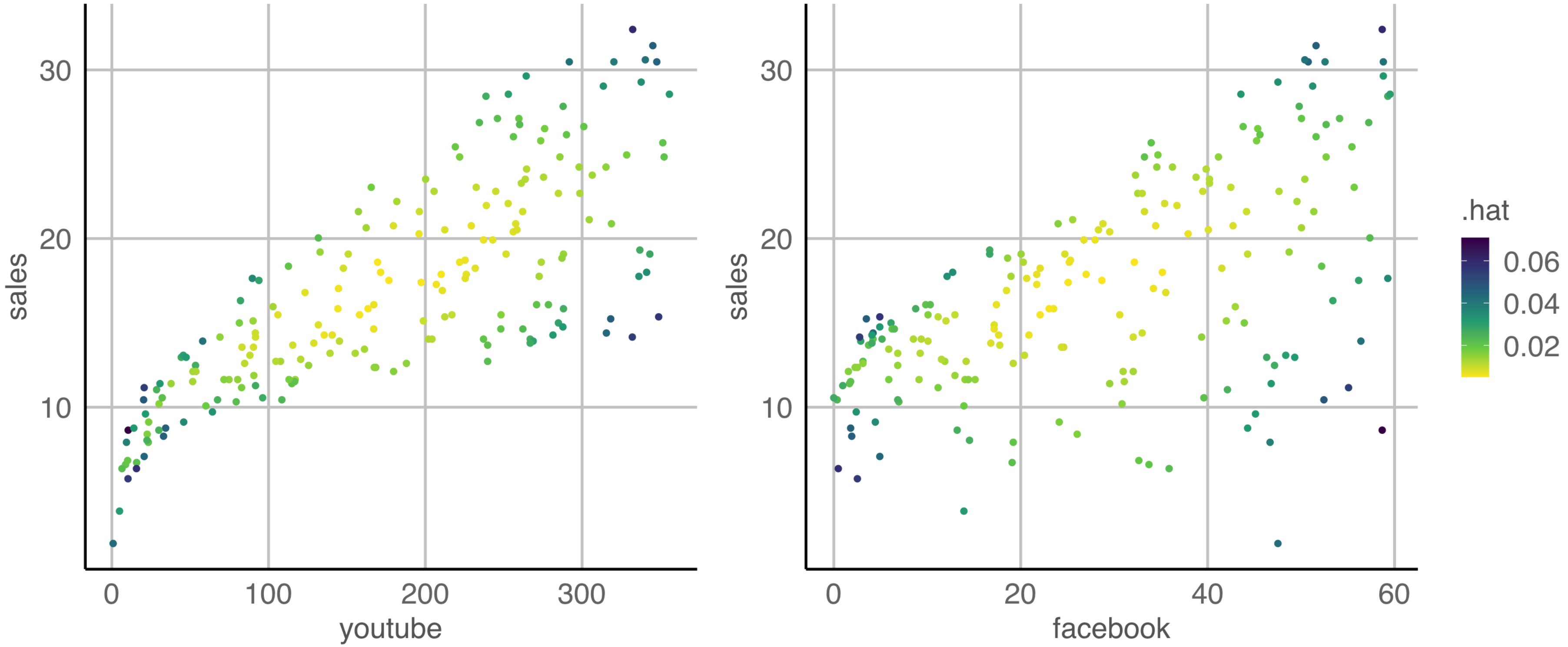
Extracting values from models in R

- The leverage value, studentised residual and Cook's distance can be easily extracted from a model object using `broom::augment`.
  - `.hat` is the leverage value
  - `.std.resid` is the studentised residual
  - `.cooksd` is the Cook's distance

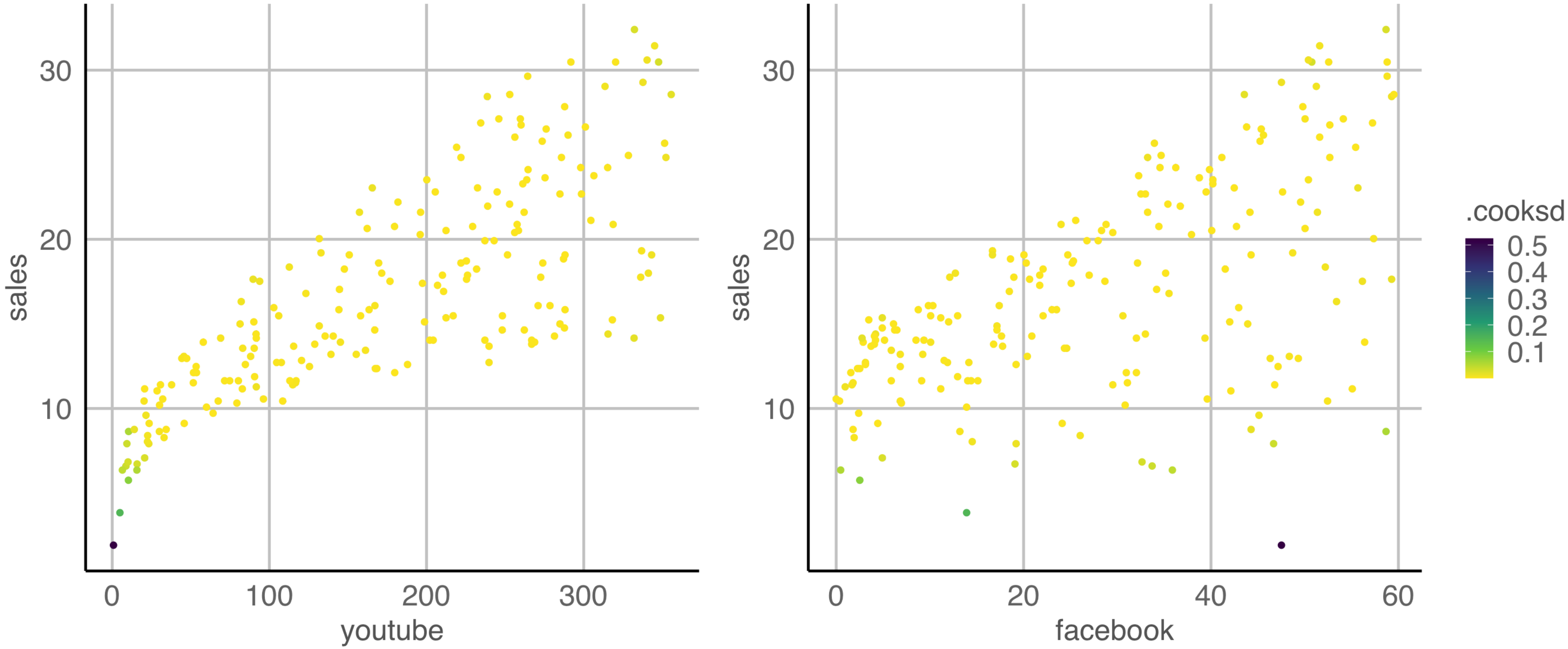
```
fit <- lm(sales ~ youtube * facebook, data = marketing)
(out <- broom::augment(fit))

## # A tibble: 200 × 9
##    sales youtube facebook .fitted  .resid    .hat .sigma   .cooksd .std.resid
##    <dbl>   <dbl>    <dbl>   <dbl>   <dbl>   <dbl>  <dbl>     <dbl>      <dbl>
##  1 26.5    276.     45.4    26.0   0.496  0.0174    1.13 0.000864      0.442
##  2 12.5     53.4    47.2    12.8  -0.281  0.0264    1.13 0.000431     -0.252
##  3 11.2     20.6    55.1    11.1   0.0465 0.0543    1.14 0.0000256     0.0423
##  4 22.2    182.     49.6    21.2   1.04   0.0124    1.13 0.00268       0.923
##  5 15.5    217.     13.0    15.2   0.316  0.0104    1.13 0.000207      0.280
```

Examining the leverage values



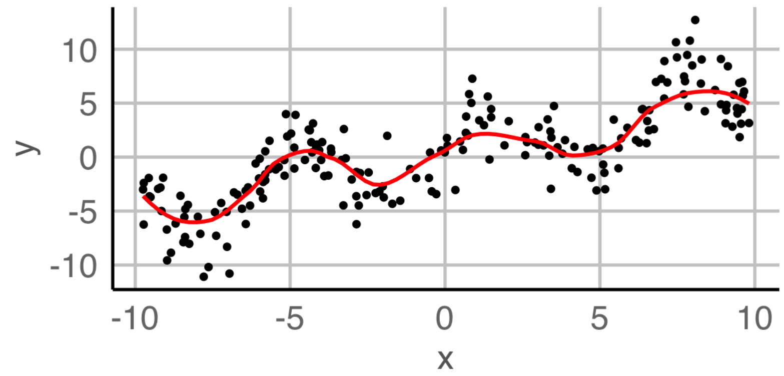
Examining the Cook's distance



Non-parametric regression

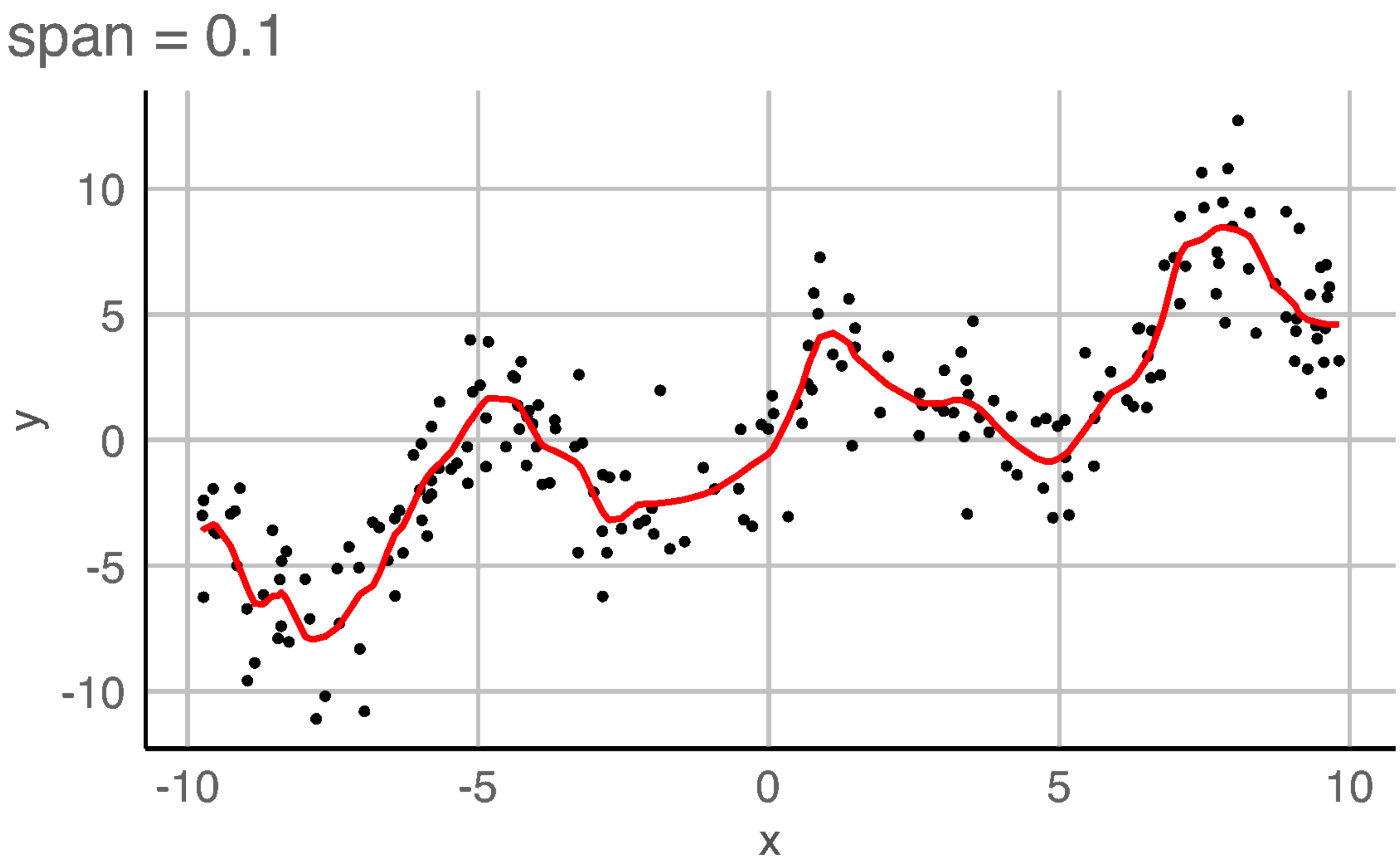
LOESS

- LOESS (LOcal regrESSion) and LOWESS (LOcally WEighted Scatterplot Smoothing) are **non-parametric regression** methods (LOESS is a generalisation of LOWESS)
- **LOESS fits a low order polynomial to a subset of neighbouring data** and can be fitted using `loess` function in `R`
- a user specified "bandwidth" or "smoothing parameter" α determines how much of the data is used to fit each local polynomial.



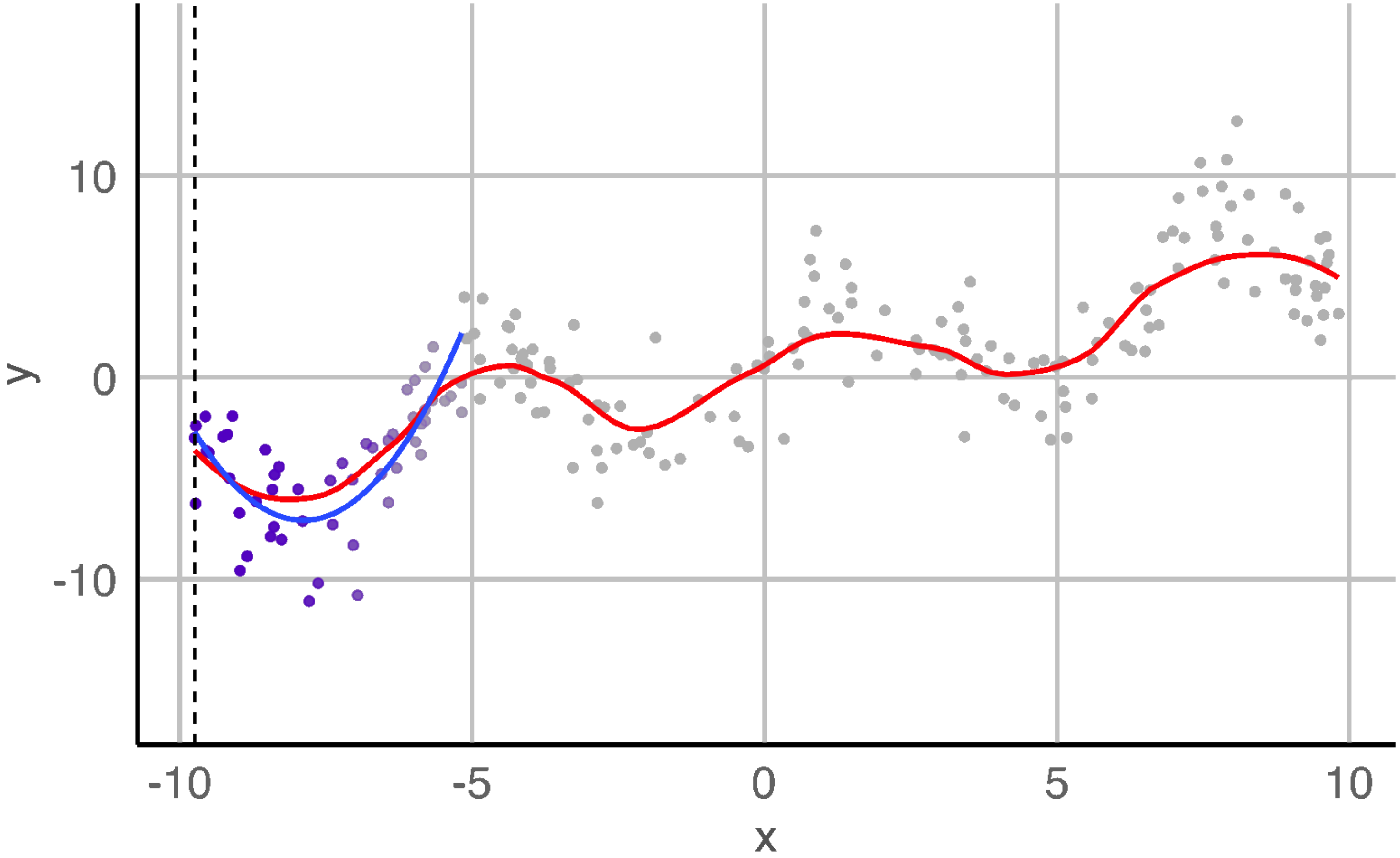
- $\alpha \in \left(\frac{\lambda+1}{n}, 1\right)$ (default `span=0.75`) where λ is the degree of the local polynomial (default `degree=2`) and n is the number of observations.
- Large α produce a smoother fit.
- Small α overfits the data with the fitted regression capturing the random error in the data.

How span changes the loess fit



Code inspired by http://varianceexplained.org/files/loess.html

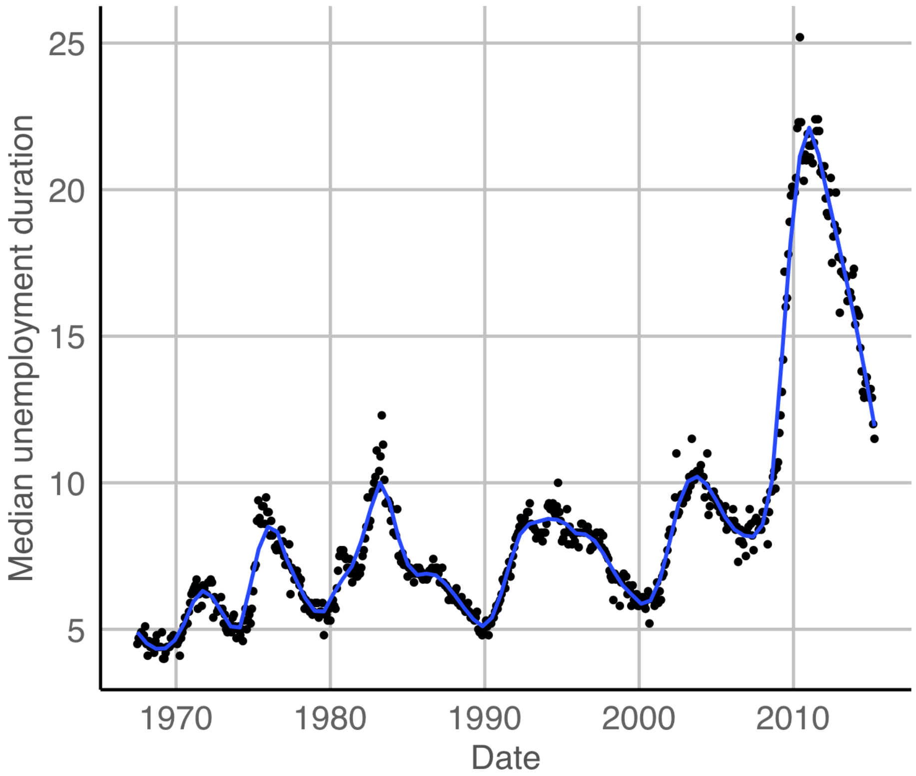
How `loess` works



Case study 3 US economic time series

This dataset was produced from US economic time series data available from http://research.stlouisfed.org/fred2.

data R



How to fit LOESS curves in R?

Model fitting

The model can be fitted using the `loess` function where

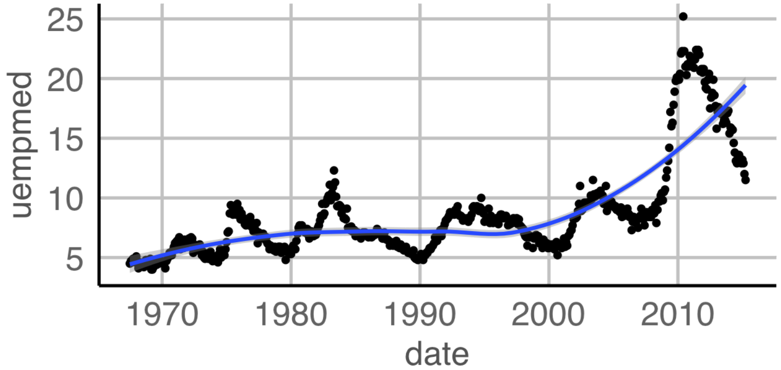
- the default span is 0.75 and
- the default local polynomial degree is 2.

```
fit <- economics %>%
          mutate(index = 1:n()) %>%
          loess(uempmed ~ index,
                data = .,
                span = 0.75,
                degree = 2)
```

Showing it on the plot

In `ggplot`, you can add the loess using `geom_smooth` with `method = loess` and method arguments passed as list:

```
ggplot(economics, aes(date, uempmed)) +
  geom_point() +
  geom_smooth(method = loess,
              method.args = list(span = 0.75,
                                 degree = 2))
```



Why non-parametric regression?

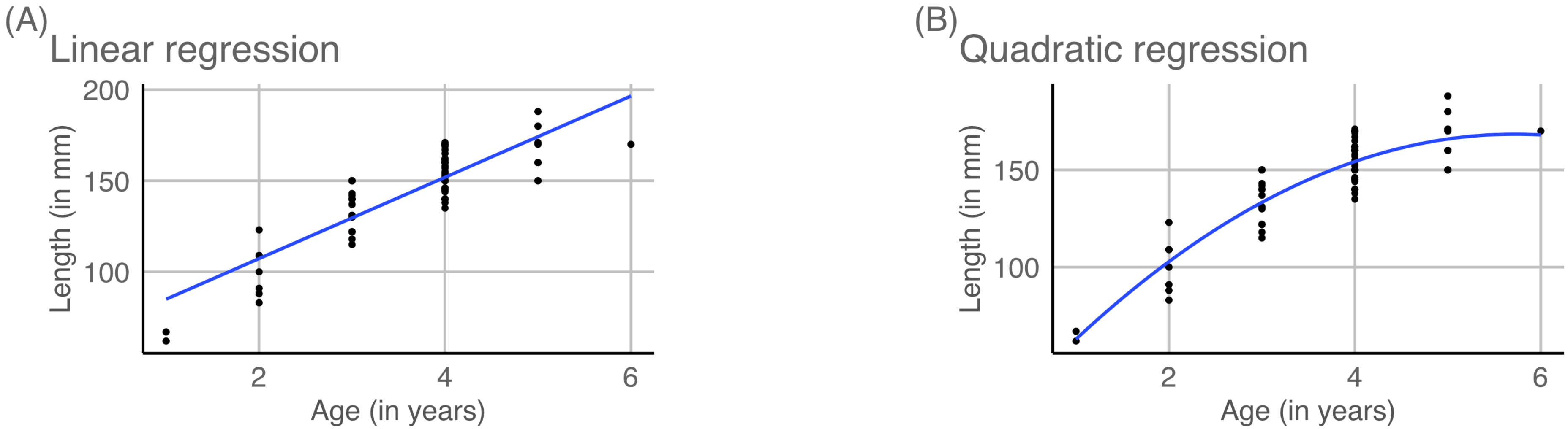
- Fitting a line to a scatter plot where noisy data values, sparse data points or weak inter-relationships interfere with your ability to see a line of best fit.
- Linear regression where least squares fitting doesn't create a line of good fit or is too labour intensive to use.
- Data exploration and analysis.
- Recall: In a parametric regression, some type of distribution is assumed in advance; therefore fitted model can lead to fitting a smooth curve that misrepresents the data.
- In those cases, non-parametric regression may be a better choice.
- *Can you think of where it might be useful?*

Case study 4 Bluegills Part 1/3

Data were collected on length (in mm) and the age (in years) of 78 bluegills captured from Lake Mary, Minnesota in 1981.

data R

Which fit looks better?

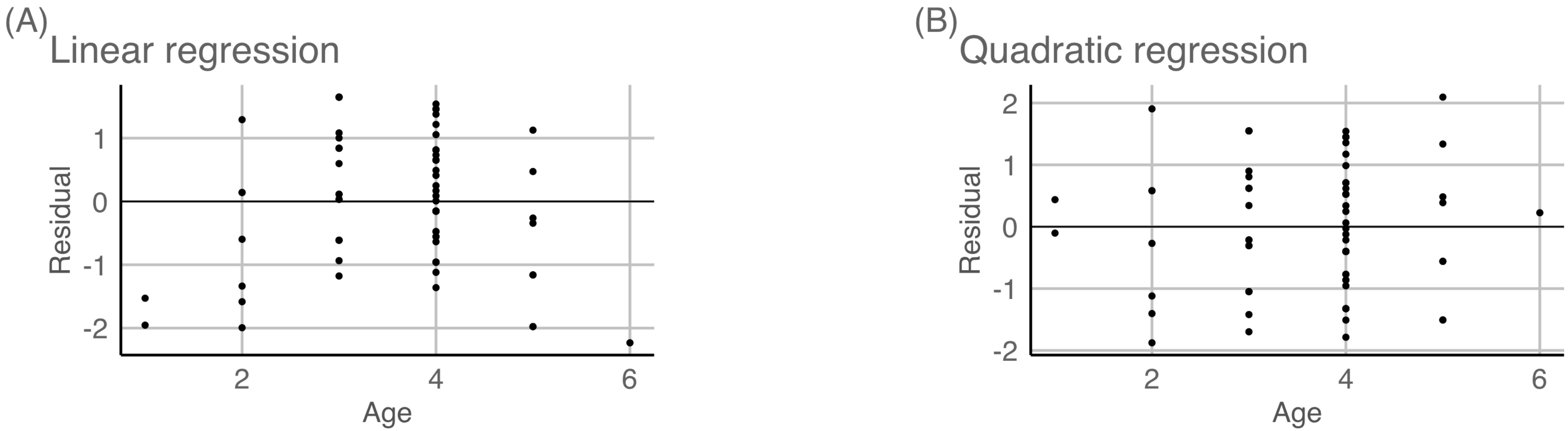


Weisberg (1986) A linear model approach to backcalculation of fish length, *Journal of the American Statistical Association* **81** (196) 922-929

Case study 4 Bluegills Part 2/3

- Let's have a look at the residual plots.
- Do you see any patterns on either residual plot?

data R

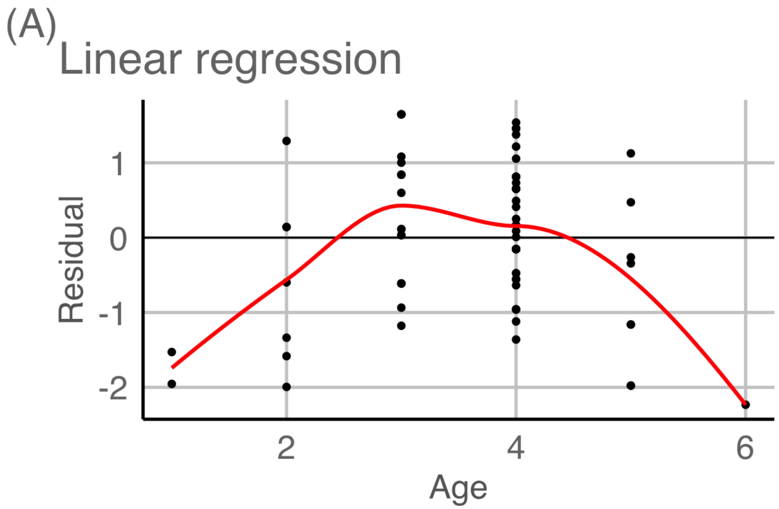


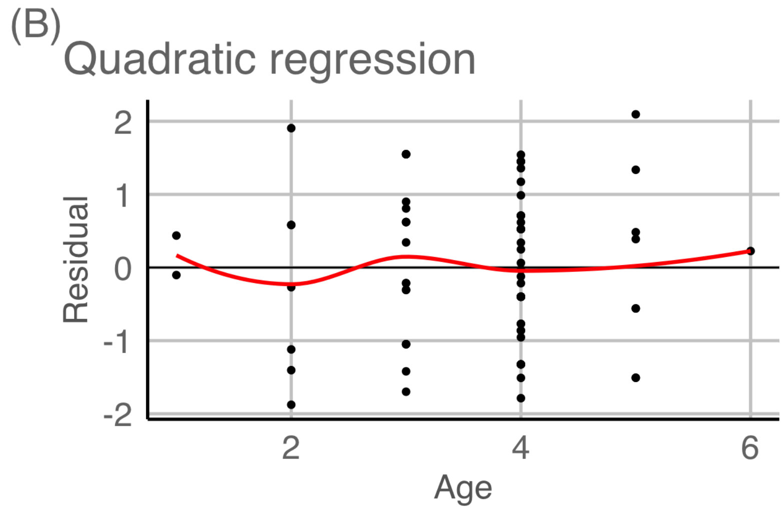
Weisberg (1986) A linear model approach to backcalculation of fish length, *Journal of the American Statistical Association* **81** (196) 922-929

Case study 4 Bluegills Part 3/3

The structure is easily visible with the LOESS curve:

data R



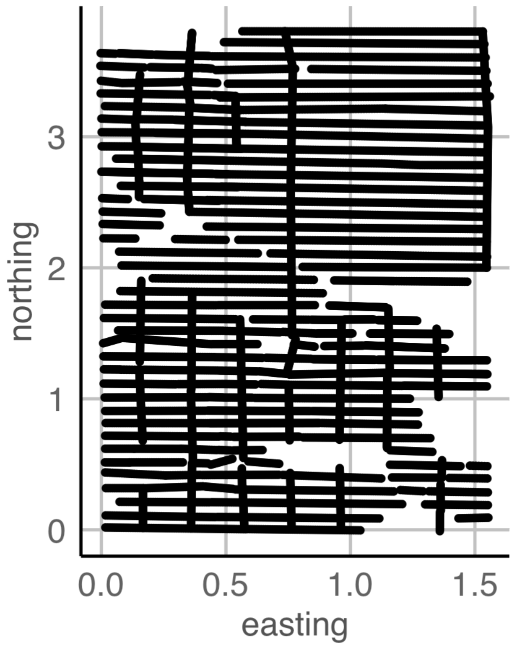


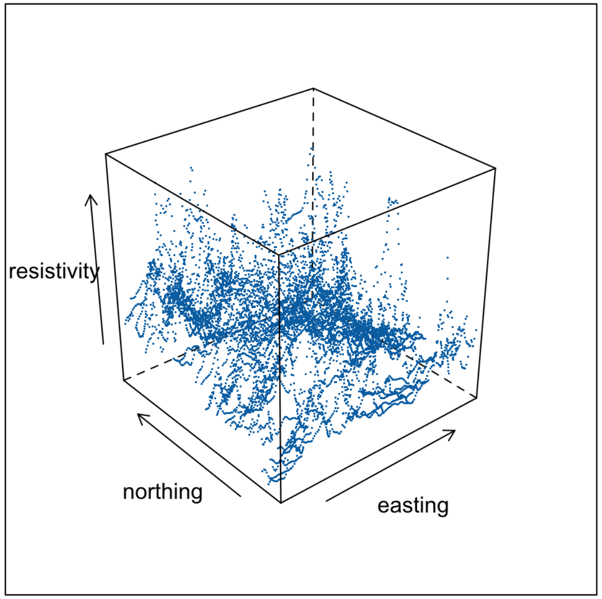
Weisberg (1986) A linear model approach to backcalculation of fish length, *Journal of the American Statistical Association* **81** (196) 922-929

Case study 5 Soil resistivity in a field

This data contains measurement of soil resistivity of an agricultural field.

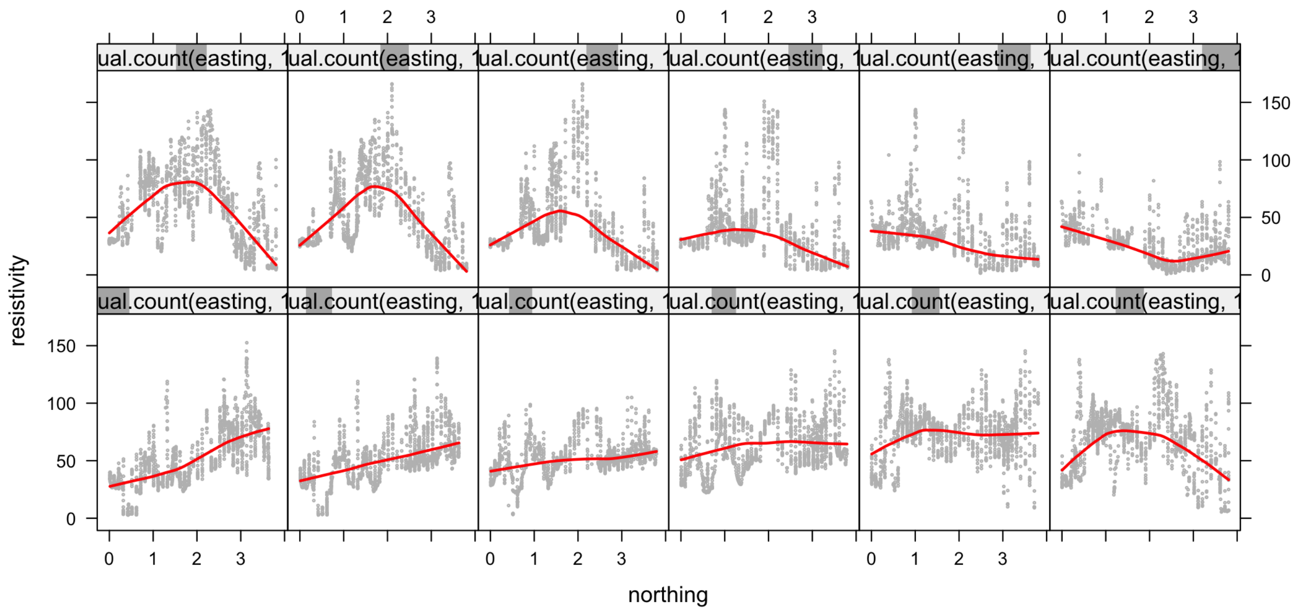
data R





Conditioning plots (Coplots)

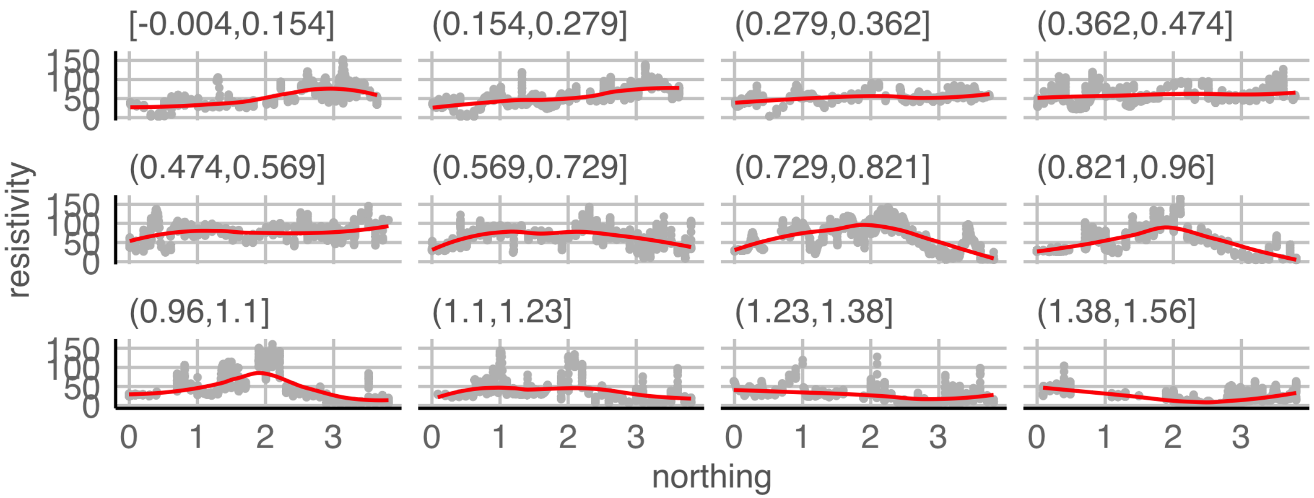
```
library(lattice)
xyplot(resistivity ~ northing | equal.count(easting, 12),
       data = cleveland.soil, cex = 0.2,
       type = c("p", "smooth"), col.line = "red",
       col = "gray", lwd = 2)
```



Coplots via `ggplot2`

- Coplots with `ggplot2` where the panels have overlapping observations is tricky.
- Below creates a plot for non-overlapping intervals of `easting`:

```
ggplot(cleveland.soil, aes(northing, resistivity)) +
  geom_point(color = "gray") +
  geom_smooth(method = "loess", color = "red", se = FALSE) +
  facet_wrap(~ cut_number(easting, 12))
```



Take away messages

- You can use leverage values and Cook's distance to query possible unusal values in the data
- Non-parametric regression, such as LOESS, can be useful in data exploration and analysis although parameters must be carefully chosen not to overfit the data
- Conditioning plots are useful in understanding the relationship between pairs of variables given at particular intervals of other variables

Resources and Acknowledgement

- These slides were originally created by Dr Emi Tanaka, and modified by Dr Michael Lydeamore.
- Cook & Weisberg (1994) "An Introduction to Regression Graphics"
- Data coding using `tidyverse` suite of R packages
- Slides constructed with `xaringan`, remark.js, `knitr`, and R Markdown.

Lecturer: *Di Cook*

ETC5521.Clayton-x@monash.edu

Week 11 - Session 2