Regression

Pavel Krivitsky
based on notes by Dr. Pam Davy
with minor edits by Andrew Zammit Mangion
Week 9, Autumn 2018

Learn about...

- Evaluating quantitative predictors.
- Linear and logistic regression.
- Specifying linear models and selecting parameters.
- Interpreting parameters.
- Regression and model trees.

The Regression Problem

- *Regression* (or numeric prediction) is the task of learning a target function f which maps each attribute set $\boldsymbol{x}$ to a numeric output (response) variable y.
- Consider a data set of n observations:

  $$\{(\boldsymbol{x}_i, y_i), i = 1, 2, \ldots, n\}$$

  Usually $\boldsymbol{x}_i$ consists of multiple attributes.
- Let $\hat{y}_i = f(\boldsymbol{x}_i)$ denote the predicted (fitted) value for observation i.

Performance Evaluation

Performance of a regression task can be evaluated by looking at the prediction error.

Mean Squared Error: $\text{MSE} = \frac{\text{SSE}}{n-p-1}$, where $\text{SSE} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

- Most common; easiest to work with.
- Very sensitive to outliers.
- Often reported as $\sqrt{\text{MSE}}$, the *Root Mean Squared Error (RMSE)* which is on the same scale as the data.
- Loosely, p is the number of explanatory variables in the model for $\hat{y}_i$, and $n - p - 1$ is the *error degrees of freedom*.

Mean Absolute Error: $\text{MAE} = \frac{1}{n-p-1} \sum_{i=1}^{n} |y_i - \hat{y}_i|$

- More resistant to outliers.
- Not differentiable (harder to optimise).

Degrees of Freedom

(a model's) degrees of freedom (df) is a measure of its complexity.

- Loosely, every free parameter you estimate, you spend 1 degree of freedom.
- Some models can have fractional degrees of freedom.

error degrees of freedom is the number of degrees of freedom left over after fitting the model.

- Typically, $n - p - 1$ for a model with p covariates.
- Once that runs out, you can't add any more.

Relative Performance Measures

- MSE and MAE have no natural upper bound.
- For most situations, the "first approximation" one might use is the overall sample mean: $\hat{y}_i = \bar{y}$.
  - One can't do much worse than that.
  - The model has 1 df (the mean).
  - Let Sum of Squared Total $\text{SST} = \sum_{i=1}^{n}(y_i - \bar{y})^2$.

$\implies$ We can evaluate error relative to this baseline, i.e., $\frac{\text{SSE}}{\text{SST}} = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$.

  - Similarly for MAE.

R^2 and Adjusted R^2

- High value of $R^2 = 1 - \frac{\text{SSE}}{\text{SST}}$ means the model explains a high proportion of total variation.
- For a simple linear model, R^2 really is a squared correlation and therefore non-negative.
- For a non-linear model, "R^2" can be negative!
- More complexity in a model almost always increases $R^2 \implies$ overfitting.

$\implies$ *Adjusted R^2*

$$R^2_{\text{adj}} = R^2 - (1 - R^2)\frac{p}{n - p - 1}$$

is more directly comparable across linear models of differing complexity: where p is number of explanatory variables.

- It's also a better estimator for out-of-sample prediction error.

Correlation

- Another commonly used performance measure of a numeric prediction model is the *correlation* R between observed and predicted response.

$$R = \frac{\sum_{i=1}^{n}(\hat{y}_i - \overline{\hat{y}})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \overline{\hat{y}})^2 \sum_i (y_i - \overline{y})^2}}$$

- Ideally, plot of predicted versus actual response should almost straight with positive slope.
- So R close to 1 means good prediction.

Linear Regression

- *Linear* regression fits a simple equation of the form

$$\hat{y}_i = \boldsymbol{x}_i\boldsymbol{\beta} \equiv \beta_0 + \beta_1 x_{i,1} + \cdots + \beta_p x_{i,p}$$

  where $\hat{y}_i$ denotes the predicted target variable and $\boldsymbol{x}_i = [1, x_{i,1}, \ldots, x_{i,p}]$ denote the explanatory attributes, with $\boldsymbol{\beta} = [\beta_0, \beta_1, \ldots, \beta_p]^\top$.
- Note the "special" element for the intercept.
- Every β_k including β_0 consumes 1 df: the whole model uses up $p+1$ df.
- For notational convenience, $\boldsymbol{x}_i$s and y_is are often "stacked":
  - $\boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$ and $\hat{\boldsymbol{y}} = \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{bmatrix}$
  - $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_n \end{bmatrix}$ so $\boldsymbol{x}\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{x}_1\boldsymbol{\beta} \\ \vdots \\ \boldsymbol{x}_n\boldsymbol{\beta} \end{bmatrix}$

$\Longrightarrow \quad \hat{\boldsymbol{y}} = \boldsymbol{x}\boldsymbol{\beta}$

Least Squares

- Usually fit via Least Squares:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \frac{1}{2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \arg\min_{\boldsymbol{\beta}} \frac{1}{2}\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i\boldsymbol{\beta})^2$$

$$\frac{\partial}{\partial\boldsymbol{\beta}}\left\{\frac{1}{2}\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i\boldsymbol{\beta})^2\right\} = -\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i\boldsymbol{\beta})\boldsymbol{x}_i$$

$$= \sum_{i=1}^{n}(\boldsymbol{x}_i\boldsymbol{\beta})\boldsymbol{x}_i - \sum_{i=1}^{n}y_i\boldsymbol{x}_i \overset{\text{set}}{=} \boldsymbol{0}^\top$$

$$\boldsymbol{0}^\top = \sum_{i=1}^{n}(\boldsymbol{x}_i\hat{\boldsymbol{\beta}})\boldsymbol{x}_i - \sum_{i=1}^{n}y_i\boldsymbol{x}_i = (\boldsymbol{x}\hat{\boldsymbol{\beta}})^\top\boldsymbol{x} - \boldsymbol{y}^\top\boldsymbol{x} = \hat{\boldsymbol{\beta}}^\top\boldsymbol{x}^\top\boldsymbol{x} - \boldsymbol{y}^\top\boldsymbol{x}$$

$$\hat{\boldsymbol{\beta}}^\top(\boldsymbol{x}^\top\boldsymbol{x}) = \boldsymbol{y}^\top\boldsymbol{x}$$

$$(\boldsymbol{x}^\top\boldsymbol{x})\hat{\boldsymbol{\beta}} = \boldsymbol{x}^\top\boldsymbol{y}$$

Least Squares (continued)

- $\boldsymbol{x}^\top \boldsymbol{x} = \sum_{i=1}^{n} \begin{bmatrix} 1\times 1 & \cdots & 1\times x_{i,p} \\ \vdots & \ddots & \vdots \\ x_{i,p}\times 1 & \cdots & x_{i,p}\times x_{i,p} \end{bmatrix}$
- $(\boldsymbol{x}^\top \boldsymbol{x})^{-1}$ is the *matrix inverse* of $\boldsymbol{x}^\top \boldsymbol{x}$: a matrix such that $(\boldsymbol{x}^\top \boldsymbol{x})(\boldsymbol{x}^\top \boldsymbol{x})^{-1} = \mathbf{I}_{p+1} = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}$.
  - Such an inverse might not exist, if any columns of $\boldsymbol{x}$ are "redundant", in that they are a linear function of other columns. Then, regression can't be fit.

$$(\boldsymbol{x}^\top \boldsymbol{x})^{-1}(\boldsymbol{x}^\top \boldsymbol{x})\hat{\boldsymbol{\beta}} = (\boldsymbol{x}^\top \boldsymbol{x})^{-1}\boldsymbol{x}^\top \boldsymbol{y}$$

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{x}^\top \boldsymbol{x})^{-1}\boldsymbol{x}^\top \boldsymbol{y}$$

- In practice, this is very fast.

Advantages and Disadvantages

+ Very fast (compared to other methods); has a unique solution (if columns of $\boldsymbol{x}$ are not "redundant").
+ Interpretable: β_k is the predicted effect of a unit change in $x_{i,k}$ on the predicted value of Y_i.
  - Can test for individual effects.
+ Parsimonious: less overfitting
  - Measures of fit like adj. R^2 automatically account for the fact that we are evaluating in-sample.
- Not as flexible.
- Not as "automatic": you have to specify the full form of the model, not just the predictors.

In R

- `lm()` is the workhorse function for fitting Linear Models.
- Same formula interface as before.
- Less "automatism" means that we have to be specific in how predictors should predict the response.

Transforming x

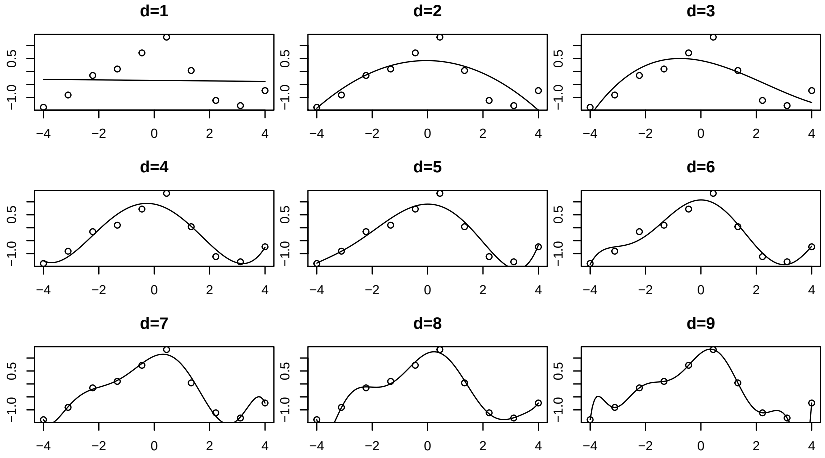
- Suppose the effect of x on $\hat{y}$ is believed to be nonlinear. Does it mean our model can't be linear?

  No! $\hat{y}$ only needs to be linear *in the parameters* $\boldsymbol{\beta}$!

- $\hat{y}_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2$ is still an LM
  - We don't have to stop at quadratic effects, but too high a power can be overfitting and unstable.
  - Each power costs 1 df.
  - In lm(), powers need to be enclosed in `I()` (e.g., `I(x^2)`), or R will process them differently.
  - We can also use `poly(x,degree)` to create *orthogonal polynomial contrasts*.
- Other nonlinear transformation of x also possible.
  - E.g., if x is right-skewed, taking $\sqrt{\ }$ or log can work better.

Polynomial regression

```
x <- seq(from = -4, to = 4, length.out = 10)
y <- cos(x) + rnorm(10, 0, 0.4)
lm(y ~ poly(x, d))  # for various d
```



Polynomial regression (summaries)

```r
summary(lm(y ~ poly(x, 4)))

##
## Call:
## lm(formula = y ~ poly(x, 4))
##
## Residuals:
##       1       2       3       4       5       6       7       8       9
## -0.1046  0.2118  0.1474 -0.4230 -0.2076  0.5795 -0.0137 -0.2742  0.0630
##      10
##  0.0215
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.3412     0.1196   -2.85   0.0357 *
## poly(x, 4)1  -0.0796     0.3782   -0.21   0.8415
## poly(x, 4)2  -2.1295     0.3782   -5.63   0.0024 **
## poly(x, 4)3   0.6593     0.3782    1.74   0.1417
## poly(x, 4)4   1.3004     0.3782    3.44   0.0185 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.378 on 5 degrees of freedom
## Multiple R-squared:  0.903,Adjusted R-squared:  0.826
## F-statistic: 11.7 on 4 and 5 DF,  p-value: 0.00951
```

Categorical Predictors

- Categorical predictors are represented using *dummy* or *inidcator* variables.
  1. One category (level) is set as the baseline.
  2. For each of the rest its corresponding $x_{i,k}$ is set to 1 if i is in the category and 0 otherwise.

$\Longrightarrow$ A factor with l levels needs $l - 1$ β_ks, so uses up $l - 1$ df.

- R automatically does this for factor variables.
  - Beware categorical variables coded as numbers! "Process" them with `factor()` to let R know.
  - An ordinal factor can be identified using the function `ordered()`.
- You can create a dummy variable explicitly via `I(var == "val")` to add

$$x_{i,k} = \mathbb{I}\{\texttt{var}[i] \text{ is "val"}\} = \begin{cases} 1 & \text{if value of } \texttt{var} \text{ for } i \text{ is "val"} \\ 0 & \text{otherwise} \end{cases}$$

Interaction

- Interaction occurs when the effect of one predictor variable depends on the level of another.
  - Trivial example: age vs. height for children.
  - Slope for boys will be higher than slope for girls.

  $\Longrightarrow$ Interaction between age and gender.
- Represented in LMs as a product between predictor variables (and indicators).
- This costs

  nominal(c_1)×nominal(c_2): $(c_1 - 1)(c_2 - 1)$ df
  quantitative×quantitative: 1 df
  nominal(c)×quantitative: $(c - 1)$ df
- In `lm()`, use `x1:x2` to add $\beta_{12}x_{i,1}x_{i,2}$.
- *Principle of marginality* says that if you include $x_{i,1}x_{i,2}$ in the model, you should also include $x_{i,1}$ and $x_{i,2}$.

$\Longrightarrow$ Use `x1*x2` to add $\beta_1 x_{i,1} + \beta_2 x_{i,2} + \beta_{12}x_{i,1}x_{i,2}$.

Transforming y

- LMs work best when residuals are symmetric and don't have extreme outliers.
  - $\sqrt{\ }$ transformations are often used for y a count
  - log transformations often used for other strictly positive measurements
- Transforming y changes interpretation:
  - $\log(\hat{y}) = \beta_0 + \beta_1 x \implies \hat{y} = e^{\beta_0+\beta_1 x} = e^{\beta_0}(e^{\beta_1})^x \implies$ a multiplicative effect
    - A unit increase in x will *multiply* the predicted y by e^{β_1}.
  - $\log(\hat{y}) = \beta_0 + \beta_1 \log x \implies \hat{y} = e^{\beta_0+\beta_1 \log x} = e^{\beta_0}e^{(\log x)\beta_1} = e^{\beta_0}x^{\beta_1} \implies$ a *power* effect

Example: Iris data

- What if we wanted to predict petal length from species?

$$\hat{y}_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2}$$

where

$$x_{i,1} = \begin{cases} 1 & \text{if species is versicolor} \\ 0 & \text{otherwise} \end{cases}, \; x_{i,2} = \begin{cases} 1 & \text{if species is virginica} \\ 0 & \text{otherwise} \end{cases}$$

- Then,

β_0 is the predicted mean for setosa

β_1 is how much higher the predicted mean for versicolor is than that for setosa

β_2 is how much higher the predicted mean for virginica is than that for setosa

Example: Iris data

```
data(iris)
summary(lm(Petal.Length ~ Species, data = iris))
```

```
## Call:
## lm(formula = Petal.Length ~ Species, data = iris)
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.260 -0.258  0.038  0.240  1.348
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1.4620     0.0609    24.0   <2e-16 ***
## Speciesversicolor   2.7980     0.0861    32.5   <2e-16 ***
## Speciesvirginica    4.0900     0.0861    47.5   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Residual standard error: 0.43 on 147 degrees of freedom
## Multiple R-squared:  0.941,Adjusted R-squared:  0.941
## F-statistic: 1.18e+03 on 2 and 147 DF,  p-value: <2e-16
```

Regression Output

- The standard errors ('Std. Error') show how accurately the regression coefficients have been estimated given the sample size; larger values indicate less accuracy.
- The asterisks (*) indicate which attributes have a statistically significant effect upon the response, for *fixed* values of the other attributes in the equation.
- The 'Multiple R-squared' (R^2) value shows the proportion of variation in the response which is collectively explained by the explanatory attributes.
- As R^2 continues to increase as more variables are included in a regression equation, 'Adjusted R-squared' involves a penalty for the number of predictors.

Interpretation

$$\hat{y}_i = 1.4620 + 2.7980\mathbb{I}\{i \text{ is versicolor}\} + 4.0900\mathbb{I}\{i \text{ is virginica}\}$$

*** all three β_ks are highly *statistically significant*:

β_0 there is enough evidence to believe that population mean petal length for setosa is different (higher) from 0 (a trivial statement); it's about 1.4620

β_1 there is enough evidence to believe that population mean petal length for versicolor is different (higher) from that of setosa (the baseline); it's about 4.2600

β_2 there is enough evidence to believe that population mean petal length for virginica is different (higher) from that of setosa (the baseline); it's about 5.5520

R^2: The model explains about 0.9414 of the squared variation in the data, or 0.9406 for predicting out of sample.

Model selection

- When there are many potential predictor variables and interaction terms, prediction performance for future data will often deteriorate if a very complex model is fitted.

Stepwise regression aims to select the most important terms for inclusion in the final model:

Forward selection: Start with the minimal model, and add one at a time. Stop when nothing can be added to improve the criterion.

Backwards elimination: Start with the maximal model, and remove one at a time. Stop when nothing can be removed to improve the criterion.

Bidirectional elimination: Start with some initial model, and try to add or remove one at a time. Stop when nothing can be changed to improve the criterion.

All-subsets regression: Try every single possible combination of terms. Takes a very long time!

Common criteria

R^2_{adj}: $1 - (1 - R^2) \times (n-1)/(n-p-1)$ (bigger is better)

Mallows C_P: $\text{SSE}\,/\hat{\sigma}^2_{\text{max}} - (n - 2p - 2)$, where $\hat{\sigma}^2_{\text{max}} = \text{SSE}_{\text{max}}\,/(n - p_{\text{max}} - 1)$ (smaller $=$ better)

Akaike Information Criterion (AIC): $-2l(\hat{\boldsymbol{\beta}}) + 2p + 2$ (smaller is better)

Bayesian Information Criterion (BIC): $-2l(\hat{\boldsymbol{\beta}}) + (p+1)\log n$ (smaller is better)

- Here, $l(\hat{\boldsymbol{\beta}})$ is the *maximum log-likelihood* of the model: the log of the observation density function $L(\boldsymbol{\beta})$ evaluated at the parameter $(\hat{\boldsymbol{\beta}})$ that maximises it.

Example: Swiss Fertility data

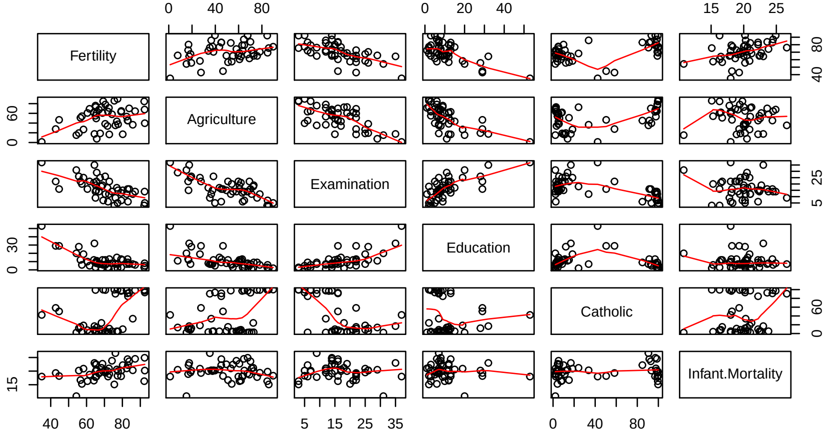
```r
library(datasets)
data(swiss)
```

- Data about 47 French-speaking provinces of Switzerland around 1888.

Fertility	standardised fertility measure
Agriculture	% of males involved in agriculture as occupation
Examination	% of draftees receiving highest mark on army exam
Education	% with education beyond primary school for draftees
Catholic	% Catholic (as opposed to Protestant)
Infant.Mortality	% live births who lives less than a year

Scatterplot matrix

```
pairs(swiss, panel = panel.smooth)
```



Regression Output

```r
summary(swiss.fit <- lm(Fertility ~ ., data = swiss))
```

```
## Call:
## lm(formula = Fertility ~ ., data = swiss)
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.274  -5.262   0.503   4.120  15.321
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       66.9152    10.7060    6.25  1.9e-07 ***
## Agriculture       -0.1721     0.0703   -2.45   0.0187 *
## Examination       -0.2580     0.2539   -1.02   0.3155
## Education         -0.8709     0.1830   -4.76  2.4e-05 ***
## Catholic           0.1041     0.0353    2.95   0.0052 **
## Infant.Mortality   1.0770     0.3817    2.82   0.0073 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Residual standard error: 7.17 on 41 degrees of freedom
## Multiple R-squared:  0.707,Adjusted R-squared:  0.671
## F-statistic: 19.8 on 5 and 41 DF,  p-value: 5.59e-10
```

Regression Equation

The fitted regression equation is

```
Fertility = 66.915 - 0.172 Agriculture
 - 0.258 Examination - 0.871 Education
 + 0.104 Catholic + 1.077 Infant.Mortality
```

- E.g., for every additional percentage point of draftees with education beyond primary school, the predicted fertility measure decreases by 0.8709 units.

Stepwise regression

```r
step(swiss.fit, data = swiss)
```

```
## Start:  AIC=190.7
## Fertility ~ Agriculture + Examination + Education + Catholic +
##     Infant.Mortality
##                    Df Sum of Sq  RSS AIC
## - Examination       1        53 2158 190
## <none>                          2105 191
## - Agriculture       1       308 2413 195
## - Infant.Mortality  1       409 2514 197
## - Catholic          1       448 2553 198
## - Education         1      1163 3268 209
## Step:  AIC=189.9
## Fertility ~ Agriculture + Education + Catholic + Infant.Mortality
##                    Df Sum of Sq  RSS AIC
## <none>                          2158 190
## - Agriculture       1       264 2422 193
## - Infant.Mortality  1       410 2568 196
## - Catholic          1       957 3115 205
## - Education         1      2250 4408 221
## Call:
## lm(formula = Fertility ~ Agriculture + Education + Catholic +
##     Infant.Mortality, data = swiss)
## Coefficients:
##      (Intercept)       Agriculture         Education          Catholic
##           62.101            -0.155            -0.980             0.125
## Infant.Mortality
##            1.078
```

All-Subsets regression: regsubsets() in leaps

- Gives best model for each predictor number.

```
library(leaps)
regsub <- regsubsets(Fertility ~ ., data = swiss)
summary(regsub)

## Subset selection object
## Call: regsubsets.formula(Fertility ~ ., data = swiss)
## 5 Variables  (and intercept)
##                  Forced in Forced out
## Agriculture          FALSE      FALSE
## Examination          FALSE      FALSE
## Education            FALSE      FALSE
## Catholic             FALSE      FALSE
## Infant.Mortality     FALSE      FALSE
## 1 subsets of each size up to 5
## Selection Algorithm: exhaustive
##          Agriculture Examination Education Catholic Infant.Mortality
## 1  ( 1 ) " "         " "         "*"       " "      " "
## 2  ( 1 ) " "         " "         "*"       "*"      " "
## 3  ( 1 ) " "         " "         "*"       "*"      "*"
## 4  ( 1 ) "*"         " "         "*"       "*"      "*"
## 5  ( 1 ) "*"         "*"         "*"       "*"      "*"
```

Selecting predictor number

```
summary(regsub)$cp  # Mallow's cp

## [1] 35.205 18.486  8.178  5.033  6.000

with(summary(regsub), which[which.min(cp), ])

##      (Intercept)      Agriculture      Examination        Education
##             TRUE             TRUE            FALSE             TRUE
##         Catholic Infant.Mortality
##             TRUE             TRUE
```

- I.e., best Mallows C_P measure is for 4 predictors, which are Agriculture, Education, Catholicism, and Infant Mortality.

Interactions

```
summary(swiss.fit <- lm(Fertility ~ (Agriculture +
    Examination + Education + Catholic + Infant.Mortality)^2,
    data = swiss))
```

```
## Call:
## lm(formula = Fertility ~ (Agriculture + Examination + Education +
##     Catholic + Infant.Mortality)^2, data = swiss)
## Residuals:
##    Min     1Q Median     3Q    Max
##  -8.76  -3.89  -0.68   3.14  14.10
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  253.97615   67.99721    3.74  0.00076 ***
## Agriculture                   -2.10867    0.70163   -3.01  0.00522 **
## Examination                   -5.58074    2.75010   -2.03  0.05109 .
## Education                     -3.47089    2.68377   -1.29  0.20547
## Catholic                      -0.17693    0.40653   -0.44  0.66642
## Infant.Mortality              -5.95748    3.08963   -1.93  0.06303 .
## Agriculture:Examination        0.02137    0.01377    1.55  0.13091
## Agriculture:Education          0.01906    0.01523    1.25  0.22009
## Agriculture:Catholic           0.00263    0.00285    0.92  0.36387
## Agriculture:Infant.Mortality   0.06370    0.02981    2.14  0.04060 *
## Examination:Education          0.07517    0.03634    2.07  0.04703 *
## Examination:Catholic          -0.00153    0.01079   -0.14  0.88791
## Examination:Infant.Mortality   0.17101    0.12907    1.33  0.19485
## Education:Catholic            -0.00713    0.01018   -0.70  0.48865
## Education:Infant.Mortality     0.03359    0.12420    0.27  0.78863
## Catholic:Infant.Mortality      0.00992    0.01617    0.61  0.54409
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Residual standard error: 6.47 on 31 degrees of freedom
## Multiple R-squared:  0.819,Adjusted R-squared:  0.731
## F-statistic: 9.35 on 15 and 31 DF,  p-value: 1.08e-07
```

Stepwise selection of interactions

```
swiss.fit2 <- lm(Fertility ~ (Agriculture + Examination +
    Education + Catholic + Infant.Mortality)^2, data = swiss)
step(swiss.fit2, data = swiss)
```

```
## Start:  AIC=188
## Fertility ~ (Agriculture + Examination + Education + Catholic +
##     Infant.Mortality)^2
##                                  Df Sum of Sq  RSS AIC
## - Examination:Catholic            1       0.8 1300 186
## - Education:Infant.Mortality      1       3.1 1302 186
## - Catholic:Infant.Mortality       1      15.8 1315 187
## - Education:Catholic              1      20.6 1320 187
## - Agriculture:Catholic            1      35.6 1335 187
## <none>                                         1299 188
## - Agriculture:Education           1      65.6 1365 188
## - Examination:Infant.Mortality    1      73.6 1373 189
## - Agriculture:Examination         1     100.9 1400 190
## - Examination:Education           1     179.3 1478 192
## - Agriculture:Infant.Mortality    1     191.4 1491 192
```

Note that removal of `Examination:Catholic` affects RSS the least (removed from model).

Stepwise selection of interactions (continued)

```
## Step:  AIC=186
## Fertility ~ Agriculture + Examination + Education + Catholic +
##     Infant.Mortality + Agriculture:Examination + Agriculture:Education +
##     Agriculture:Catholic + Agriculture:Infant.Mortality + Examination:Educa
## tion +
##     Examination:Infant.Mortality + Education:Catholic + Education:Infant.Mo
## rtality +
##     Catholic:Infant.Mortality
##                                Df Sum of Sq  RSS AIC
## - Education:Infant.Mortality    1       3.9 1304 184
## - Catholic:Infant.Mortality     1      17.3 1317 185
## - Agriculture:Catholic          1      37.1 1337 185
## <none>                                      1300 186
## - Education:Catholic            1      56.8 1357 186
## - Agriculture:Education         1      69.5 1369 186
## - Examination:Infant.Mortality  1      86.0 1386 187
## - Agriculture:Examination       1     114.3 1414 188
## - Examination:Education         1     178.4 1478 190
## - Agriculture:Infant.Mortality  1     205.3 1505 191
```

Note that removal of `Education:Infant.Mortality` affects RSS the least (removed from model).

Stepwise selection of interactions (continued 2)

```
## Step:  AIC=184.2
## Fertility ~ Agriculture + Examination + Education + Catholic +
##     Infant.Mortality + Agriculture:Examination + Agriculture:Education +
##     Agriculture:Catholic + Agriculture:Infant.Mortality + Examination:Educa
## tion +
##     Examination:Infant.Mortality + Education:Catholic + Catholic:Infant.Mor
## tality
##                               Df Sum of Sq  RSS AIC
## - Catholic:Infant.Mortality     1      25.8 1330 183
## - Agriculture:Catholic          1      36.4 1340 184
## <none>                                      1304 184
## - Agriculture:Education         1      79.2 1383 185
## - Education:Catholic            1      79.3 1383 185
## - Agriculture:Examination       1     116.3 1420 186
## - Examination:Education         1     185.9 1490 188
## - Agriculture:Infant.Mortality  1     219.8 1524 190
## - Examination:Infant.Mortality  1     230.5 1534 190
```

Note that removal of `Catholic:Infant.Mortality` affects RSS the least (removed from model).

Stepwise selection of interactions (continued 3)

```
## Step:  AIC=183.1
## Fertility ~ Agriculture + Examination + Education + Catholic +
##     Infant.Mortality + Agriculture:Examination + Agriculture:Education +
##     Agriculture:Catholic + Agriculture:Infant.Mortality + Examination:Educa
## tion +
##     Examination:Infant.Mortality + Education:Catholic
##                                Df Sum of Sq  RSS AIC
## - Agriculture:Catholic          1      26.7 1356 182
## <none>                                      1330 183
## - Education:Catholic            1      91.7 1421 184
## - Agriculture:Education         1      92.2 1422 184
## - Agriculture:Examination       1     121.2 1451 185
## - Examination:Education         1     197.2 1527 188
## - Examination:Infant.Mortality  1     210.7 1540 188
## - Agriculture:Infant.Mortality  1     220.4 1550 188
## Step:  AIC=182
## Fertility ~ Agriculture + Examination + Education + Catholic +
##     Infant.Mortality + Agriculture:Examination + Agriculture:Education +
##     Agriculture:Infant.Mortality + Examination:Education + Examination:Infa
## nt.Mortality +
##     Education:Catholic
##                                Df Sum of Sq  RSS AIC
## <none>                                      1356 182
## - Agriculture:Education         1      75.0 1431 183
## - Agriculture:Examination       1      99.7 1456 183
## - Examination:Education         1     174.6 1531 186
## - Education:Catholic            1     216.6 1573 187
## - Agriculture:Infant.Mortality  1     271.1 1627 189
## - Examination:Infant.Mortality  1     272.9 1629 189
```

Stepwise selection of interactions (continued 4)

```
## Call:
## lm(formula = Fertility ~ Agriculture + Examination + Education +
##     Catholic + Infant.Mortality + Agriculture:Examination + Agriculture:Edu
## cation +
##     Agriculture:Infant.Mortality + Examination:Education + Examination:Infa
## nt.Mortality +
##     Education:Catholic, data = swiss)
## Coefficients:
##                  (Intercept)                   Agriculture
##                     225.9101                       -1.9067
##                  Examination                     Education
##                      -5.1202                       -2.4735
##                     Catholic              Infant.Mortality
##                       0.2112                       -5.2693
##      Agriculture:Examination         Agriculture:Education
##                       0.0149                        0.0191
## Agriculture:Infant.Mortality         Examination:Education
##                       0.0635                        0.0639
## Examination:Infant.Mortality            Education:Catholic
##                       0.1722                       -0.0124
```

Final fit with interactions

```
summary(swiss.fit2.steps)


##
## Call:
## lm(formula = Fertility ~ Agriculture + Examination + Education +
##     Catholic + Infant.Mortality + Agriculture:Examination + Agriculture:Education +
##     Agriculture:Infant.Mortality + Examination:Education + Examination:Infant.Mortality +
##     Education:Catholic, data = swiss)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.608 -3.665 -0.564  2.922 13.736
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  225.91005   52.45757    4.31  0.00013 ***
## Agriculture                   -1.90665    0.56282   -3.39  0.00176 **
## Examination                   -5.12020    1.57831   -3.24  0.00259 **
## Education                     -2.47350    1.20277   -2.06  0.04725 *
## Catholic                       0.21116    0.05418    3.90  0.00042 ***
## Infant.Mortality              -5.26935    2.28727   -2.30  0.02729 *
## Agriculture:Examination        0.01488    0.00928    1.60  0.11771
## Agriculture:Education          0.01908    0.01372    1.39  0.17301
## Agriculture:Infant.Mortality   0.06353    0.02402    2.64  0.01216 *
## Examination:Education          0.06389    0.03010    2.12  0.04092 *
## Examination:Infant.Mortality   0.17219    0.06489    2.65  0.01189 *
## Education:Catholic            -0.01238    0.00524   -2.36  0.02374 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.23 on 35 degrees of freedom
## Multiple R-squared:  0.811,Adjusted R-squared:  0.752
## F-statistic: 13.7 on 11 and 35 DF,  p-value: 1.28e-09
```

Linear Regression for Big Data

- Computational issues arise for "big" data sets, either in the sense of many rows (instances) or many columns (attributes).
  - For many rows, the data may need to be read in chunks, and stored economically.
  - For many columns, solution may be slow and numerically unstable despite being non-iterative.
  - For many columns, attribute selection is particularly important but backwards stepwise regression may be infeasible.

Linear Regression for Big Data

- The R `biglm` package implements regression using p^2 memory for p explanatory variables.
- The `biglm` function extends the capabilities of `lm` for linear regression.
- Data can be read in chunks, and fitted models can be updated with additional data by the `update` function.
- The `bigglm` function extends the capabilities of `glm` for generalised linear models.

Linear Regression for Big Data

- Linear regression involves solving a potentially large system of simultaneous equations, or equivalently inverting a potentially large $(p+1) \times (p+1)$ matrix where $p =$ number of predictors.
  - If necessary, the basic R `solve` function used to invert a matrix can be replaced by `qr.solve` (QR decomposition) or `chol2inv` (Cholesky decomposition, suitable for inverting symmetric matrices).

Non-linear Regression

$$\hat{y}_i = f(\boldsymbol{x}_i; \boldsymbol{\beta})$$

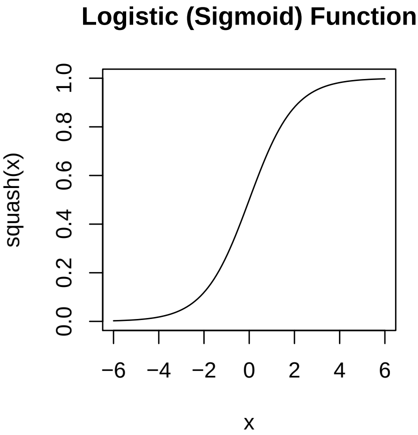
- Non-linear regression fits more complicated relationships, but the form of f must be specified.
- Non-linear least squares can be used. (R function: nls().)
  - A unique solution is not guaranteed.
- Neural networks can be regarded as a form of non-linear regression with many parameters, *e.g.*

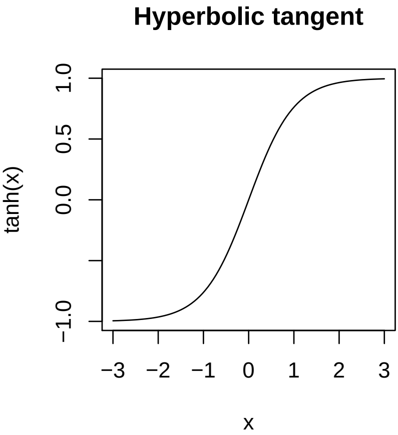
$$y = \text{squash}\left(v_0 + \sum_{j=1}^{m} v_j \text{squash}\left(w_{0j} + \sum_{i=1}^{k} w_{ij} x_i\right)\right)$$

where

$$\text{squash}(x) = \frac{1}{1 + \exp(-x)}$$

Logistic and tanh Functions





Logistic Regression

- Designed when you have binary response.
- Logistic regression involves fitting an equation of the form

$$\Pr(Y_i = 1) = \text{squash}(\beta_0 + \beta_1 x_{i,1} + \cdots + \beta_i x_{i,p})$$

  where $\text{squash}(x) = 1/(1 + e^{-x})$
- Statisticians call it the *logistic* function:

$$\text{logit}\{\Pr(Y_i = 1)\} = \beta_0 + \beta_1 x_{i,1} + \cdots + \beta_i x_{i,p},$$

  where $\text{logit}(q) = \log \frac{q}{1-q}$, the log of the odds associated with probability q.
    - The technique can be extended to more than 3 categories, either nominal (unordered) or ordinal (ordered categories).

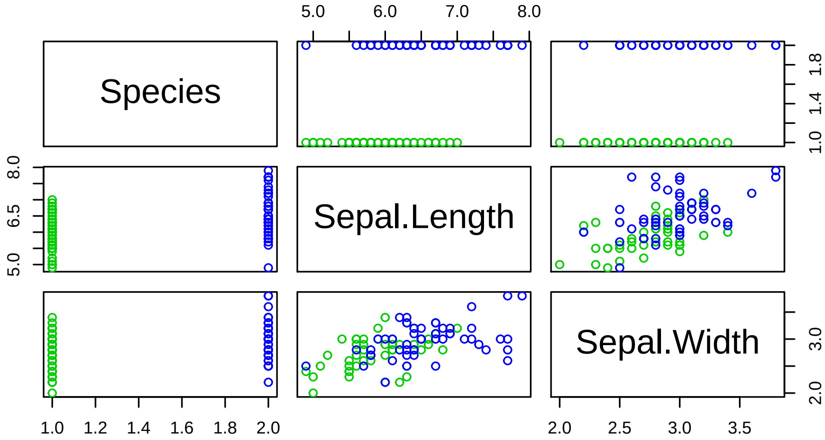
Generalised Linear Models

- Generalised linear models extend linear regression in a number of ways, and include logistic regression as a particular case.
- The fitted linear equation is linked to the mean response by a *link* function (not necessarily the identity).
- The underlying probability model for the response may differ from Normal.
- Implemented in R by `glm`.
  - To fit logistic regression, specify `family = binomial("logit")`

Example: Iris Data

- Recall the scenario from the SVM lecture.

```r
iris2 <- transform(subset(iris, Species != "setosa",
    c("Species", "Sepal.Length", "Sepal.Width")),
    Species = factor(Species))
```



Logistic Regression

```r
summary(glm(I(Species == "virginica") ~ ., data = iris2,
    family = binomial("logit")))
```

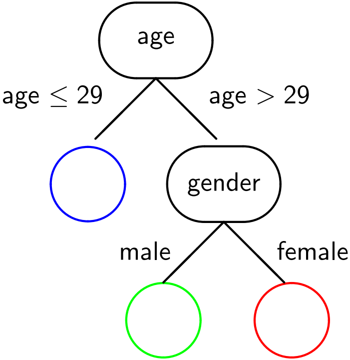
```
## Call:
## glm(formula = I(Species == "virginica") ~ ., family = binomial("logit"),
##     data = iris2)
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.874  -0.895  -0.055   0.961   2.357
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -13.046      3.097   -4.21  2.5e-05 ***
## Sepal.Length    1.902      0.517    3.68  0.00023 ***
## Sepal.Width     0.405      0.863    0.47  0.63908
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Dispersion parameter for binomial family taken to be 1)
##     Null deviance: 138.63  on 99  degrees of freedom
## Residual deviance: 110.33  on 97  degrees of freedom
## AIC: 116.3
## Number of Fisher Scoring iterations: 4
```

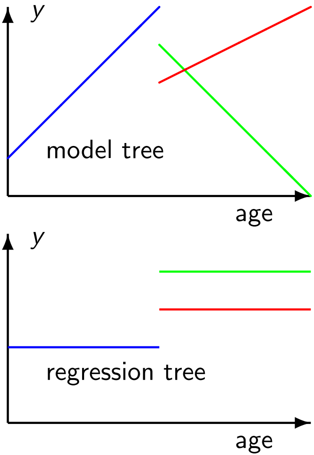
Interpretation

- Only sepal length is significant.
- *In the presence of sepal length*, sepal width is not.
- For every unit increase in sepal length, the predicted odds of it being a virginica are multiplied by $e^{1.9024} = 6.7018$.
- Standard functions (like `predict()`) are available. However, you must specify the type:
  - By default, predicts $\text{logit}\{\widehat{\Pr}(Y_{\text{new}} = 1)\}$.
  - Specify `type="response"` to predict $\widehat{\Pr}(Y_{\text{new}} = 1)$.

Regression Trees

- A regression tree is similar to a decision tree, except that the predicted value at a terminal leaf is given by the mean or median response variable of instances allocated to that leaf.
  - Branching is performed to reduce variation (as measured by standard deviation or mean absolute deviation) within the daughter nodes.
  - A *model tree* is a variation involving the fitting of linear regression models at each terminal leaf.
  - Quinlan's M5 algorithm fits both variants, implemented in `Weka` and the R `cubist` package.


age
age ≤ 29
age > 29
gender
male
female


y
model tree
age
y
regression tree
age

As a linear model

- A regression tree can be expressed as a linear regression model by the use of indicator variables, *e.g.*

$$\hat{y} = \beta_0 + \beta_1 \mathbb{I}_1(\text{age}) + \beta_2 \mathbb{I}_1(\text{age}) \mathbb{I}_2(\text{gender})$$

where

$$\mathbb{I}_1(\text{age}) = \begin{cases} 0 & \text{if age} \leq 29 \\ 1 & \text{otherwise} \end{cases}$$

and

$$\mathbb{I}_2(\text{gender}) = \begin{cases} 0 & \text{if gender} = \text{`female'} \\ 1 & \text{otherwise} \end{cases}$$

  - This is not very helpful unless the tree structure, branching attributes and split points are known in advance.

Example: Swiss fertility data

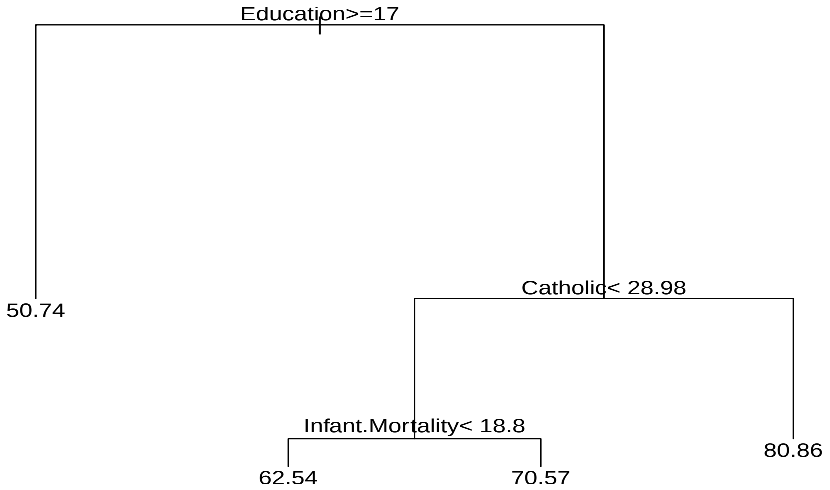
- When given a quantitative response variable, **rpart** and others automatically change to regression tree mode:

```
library(rpart)
(swiss.tree <- rpart(Fertility ~ ., data = swiss))

## n= 47
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 47 7178.0 70.14
##    2) Education>=17 7  628.3 50.74 *
##    3) Education< 17 40 3454.0 73.54
##      6) Catholic< 28.98 23  827.5 68.13
##       12) Infant.Mortality< 18.8 7  167.3 62.54 *
##       13) Infant.Mortality>=18.8 16  346.6 70.57 *
##      7) Catholic>=28.98 17 1042.0 80.86 *
```

Visualisation of regression trees

```
plot(swiss.tree)
text(swiss.tree, digits = 4)
```



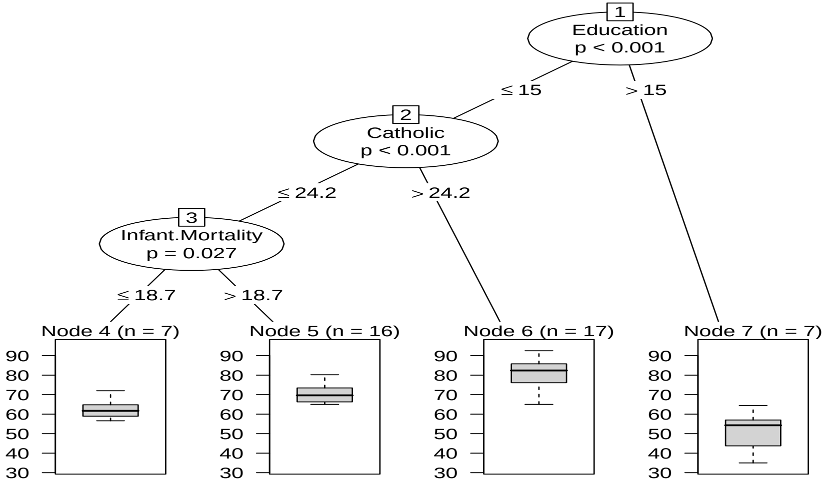
Example: Swiss fertility data with ctree

```
library(party)
(swiss.tree2 <- ctree(Fertility ~ ., data = swiss))

##
##   Conditional inference tree with 4 terminal nodes
##
## Response:  Fertility
## Inputs:  Agriculture, Examination, Education, Catholic, Infant.Mortality
## Number of observations:  47
##
## 1) Education <= 15; criterion = 1, statistic = 20.268
##   2) Catholic <= 24.2; criterion = 1, statistic = 15.218
##     3) Infant.Mortality <= 18.7; criterion = 0.973, statistic = 7.723
##       4)*  weights = 7
##     3) Infant.Mortality > 18.7
##       5)*  weights = 16
##   2) Catholic > 24.2
##     6)*  weights = 17
## 1) Education > 15
##   7)*  weights = 7
```

Visualisation

```
plot(swiss.tree2)
```



Smoothing

- Regression tree software involves extensive searching.
  - Drawback with regression trees: abrupt discontinuities (jumps) at boundaries between sibling nodes.
  - This happens for model trees as well as for simple regression trees.
- To reduce size of jumps at boundaries, raw predictions can be smoothed via

$$\frac{n_p \hat{y}_p + k \hat{y}_q}{n_p + k}$$

  at each step along path from terminal leaf back to top of tree (root node).
  - $\hat{y}_p =$ partially-smoothed prediction passed from lower in tree,
  - $\hat{y}_q =$ raw prediction at the current node,
  - $n_p =$ number of instances in lower node,
    $k =$ smoothing parameter.