UCLA Department of Statistics
Statistical Consulting Center

Introduction to Regression in R
Part II:
Multivariate Linear Regression

Denise Ferrari
denise@stat.ucla.edu

May 14, 2009

Outline

Denise Ferrari denise@stat.ucla.edu

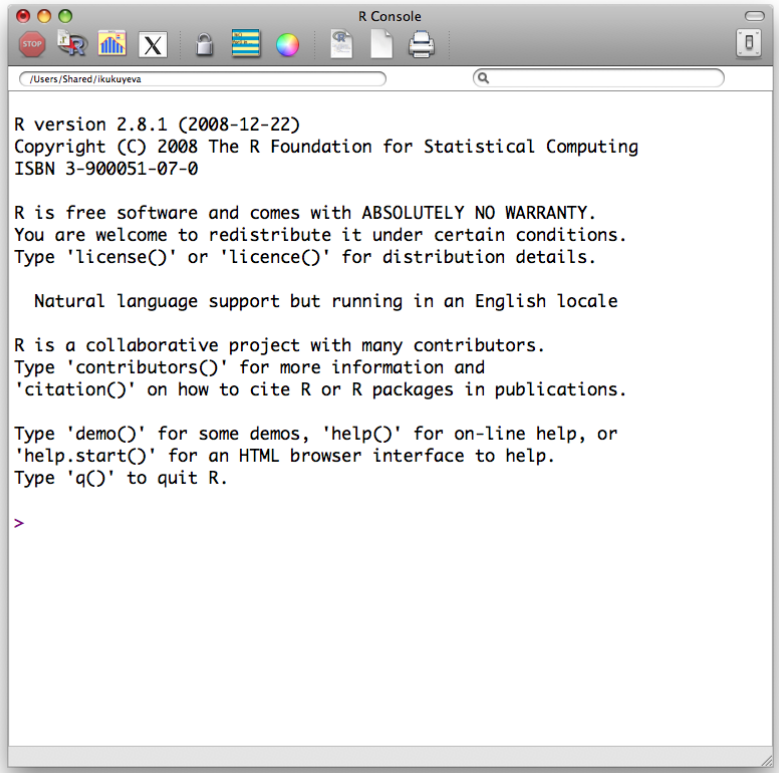
Objective

The main objective of this mini-course is to show how to perform Regression Analysis in R.
Prior knowledge of the basics of Linear Regression Models is assumed.

This second part deals with Multivariate Linear Regression.

Installing R on a Mac

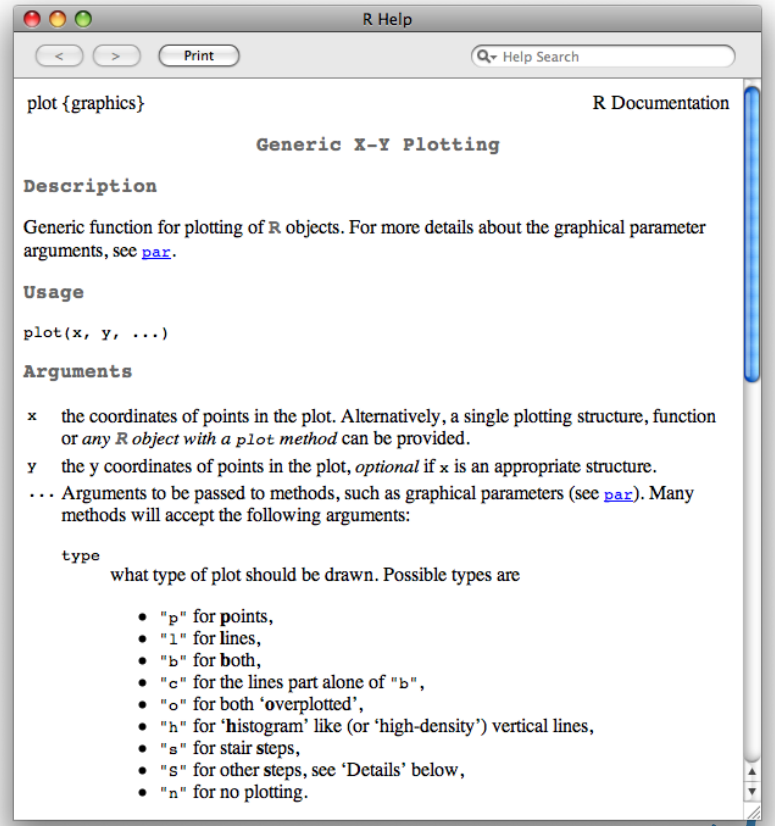
1. Go to `http://cran.r-project.org/` and select *MacOS X*
2. Select to download the latest version: 2.8.1 (2008-12-22)
3. Install and Open. The R window should look like this:



R Help

For help with any function in R, put a question mark before the function name to determine what arguments to use, examples and background information.

```
1  ?plot
```



Data from the Internet

When downloading data from the internet, use `read.table()`.
In the arguments of the function:

- `header:` if TRUE, tells R to include variables names when importing
- `sep:` tells R how the entires in the data set are separated
  - `sep=","`: when entries are separated by COMMAS
  - `sep="\t"`: when entries are separated by TAB
  - `sep=" "`: when entries are separated by SPACE

```
1 data <- read.table("http://www.stat.ucla.
      edu/data/moore/TAB1-2.DAT", header=
      FALSE, sep="")
```

Data from Your Computer

- Check the current R working folder:

```
1  getwd()
```

- Move to the folder where the data set is stored (if different from (1)). Suppose your data set is on your desktop:

```
1  setwd("~/Desktop")
```

- Now use `read.table()` command to read in the data:

```
1  data <- read.table(<name>, header=TRUE,
       sep="" )
```

R Data Sets

- To use a data set available in one of the R packages, install that package (if needed).
- Load the package into R, using the `library()` command:

```
1 library(MASS)
```

- Extract the data set you want from that package, using the `data()` command. Let's extract the data set called `Pima.tr`:

```
1 data(Boston)
```

When to Use Regression Analysis?

Regression analysis is used to describe the relationship between:

- A single response variable: Y; and
- One or more predictor variables: $X_1, X_2, \ldots, X_p$
  - $p = 1$: Simple Regression
  - $p > 1$: Multivariate Regression

The Variables

Response Variable

The response variable Y must be a continuous variable.

Predictor Variables

The predictors $X_1, \ldots, X_p$ can be continuous, discrete or categorical variables.

Initial Data Analysis I

Does the data look like as we expect?

Prior to any analysis, the data should always be inspected for:

- Data-entry errors
- Missing values
- Outliers
- Unusual (e.g. asymmetric) distributions
- Changes in variability
- Clustering
- Non-linear bivariate relatioships
- Unexpected patterns

Initial Data Analysis II

Does the data look like as we expect?

We can resort to:

- Numerical summaries:
  - 5-number summaries
  - correlations
  - etc.
- Graphical summaries:
  - boxplots
  - histograms
  - scatterplots
  - etc.

Loading the Data

Example: Diabetes in Pima Indian Women [1]

- Clean the workspace using the command: `rm(list=ls())`
- Download the data from the internet:

```
pima <- read.table("http://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-diabetes/pima-indians-diabetes.data", header=F, sep=",")
```

- Name the variables:

```
colnames(pima) <- c("npreg", "glucose", "bp", "triceps", "insulin", "bmi", "diabetes", "age", "class")
```

[1] Data from the UCI Machine Learning Repository

http://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-diabetes/pima-indians-diabetes.names

Having a peek at the Data

Example: Diabetes in Pima Indian Women

- For small data sets, simply type the name of the *data frame*
- For large data sets, do:

```
1    head(pima)
```

```
  npreg glucose bp triceps insulin  bmi diabetes age class
1     6     148 72      35       0 33.6    0.627  50     1
2     1      85 66      29       0 26.6    0.351  31     0
3     8     183 64       0       0 23.3    0.672  32     1
4     1      89 66      23      94 28.1    0.167  21     0
5     0     137 40      35     168 43.1    2.288  33     1
6     5     116 74       0       0 25.6    0.201  30     0
```

Numerical Summaries

Example: Diabetes in Pima Indian Women

- Univariate summary information:
  - Look for unusual features in the data (data-entry errors, outliers): check, for example, min, max of each variable

```
1    summary(pima)
```

```
    npreg           glucose             bp           triceps         insulin
Min.   : 0.000   Min.   :  0.0   Min.   :  0.0   Min.   : 0.00   Min.   :  0.0
1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.0   1st Qu.: 0.00   1st Qu.:  0.0
Median : 3.000   Median :117.0   Median : 72.0   Median :23.00   Median : 30.5
Mean   : 3.845   Mean   :120.9   Mean   : 69.1   Mean   :20.54   Mean   : 79.8
3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.0   3rd Qu.:32.00   3rd Qu.:127.2
Max.   :17.000   Max.   :199.0   Max.   :122.0   Max.   :99.00   Max.   :846.0
      bmi           diabetes           age           class
Min.   : 0.00   Min.   :0.0780   Min.   :21.00   Min.   :0.0000
1st Qu.:27.30   1st Qu.:0.2437   1st Qu.:24.00   1st Qu.:0.0000
Median :32.00   Median :0.3725   Median :29.00   Median :0.0000
Mean   :31.99   Mean   :0.4719   Mean   :33.24   Mean   :0.3490
3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00   3rd Qu.:1.0000
Max.   :67.10   Max.   :2.4200   Max.   :81.00   Max.   :1.0000
```

▸ Categorical

Coding Missing Data I

Example: Diabetes in Pima Indian Women

- Variable "npreg" has maximum value equal to 17
  - unusually large but not impossible
- Variables "glucose", "bp", "triceps", "insulin" and "bmi" have minimum value equal to zero
  - in this case, it seems that zero was used to code missing data

Coding Missing Data II

Example: Diabetes in Pima Indian Women

R code for missing data

- Zero should not be used to represent missing data
  - it's a valid value for some of the variables
  - can yield misleading results
- Set the missing values coded as zero to `NA`:

```
1  pima$glucose[pima$glucose==0] <- NA
2  pima$bp[pima$bp==0] <- NA
3  pima$triceps[pima$triceps==0] <- NA
4  pima$insulin[pima$insulin==0] <- NA
5  pima$bmi[pima$bmi==0] <- NA
```

Coding Categorical Variables

Example: Diabetes in Pima Indian Women

- Variable "class" is categorical, not quantitative Summary

R code for categorical variables

- Categorical should not be coded as numerical data
  - problem of "average zip code"
- Set categorical variables coded as numerical to `factor`:

```
1  pima$class <- factor (pima$class)
2  levels(pima$class) <- c("neg", "pos")
```

Final Coding

Example: Diabetes in Pima Indian Women

```
1  summary(pima)
```

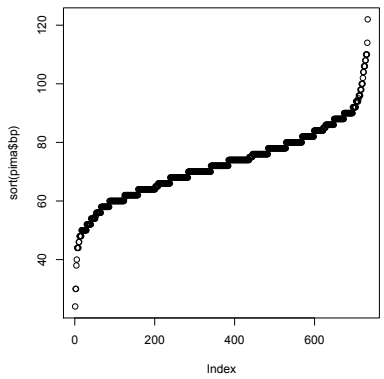
```
    npreg           glucose           bp           triceps          insulin
Min.   : 0.000   Min.   : 44.0   Min.   : 24.0   Min.   :  7.00   Min.   : 14.00
1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 64.0   1st Qu.: 22.00   1st Qu.: 76.25
Median : 3.000   Median :117.0   Median : 72.0   Median : 29.00   Median :125.00
Mean   : 3.845   Mean   :121.7   Mean   : 72.4   Mean   : 29.15   Mean   :155.55
3rd Qu.: 6.000   3rd Qu.:141.0   3rd Qu.: 80.0   3rd Qu.: 36.00   3rd Qu.:190.00
Max.   :17.000   Max.   :199.0   Max.   :122.0   Max.   : 99.00   Max.   :846.00
                 NA's   :  5.0   NA's   : 35.0   NA's   :227.00   NA's   :374.00
     bmi           diabetes          age          class
Min.   :18.20   Min.   :0.0780   Min.   :21.00   neg:500
1st Qu.:27.50   1st Qu.:0.2437   1st Qu.:24.00   pos:268
Median :32.30   Median :0.3725   Median :29.00
Mean   :32.46   Mean   :0.4719   Mean   :33.24
3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00
Max.   :67.10   Max.   :2.4200   Max.   :81.00
NA's   :11.00
```

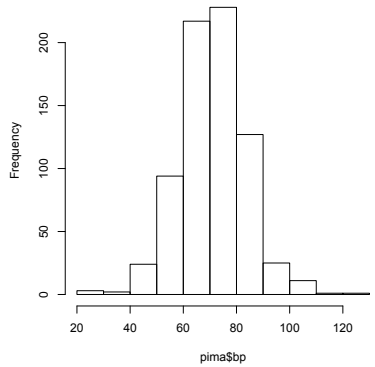
Graphical Summaries

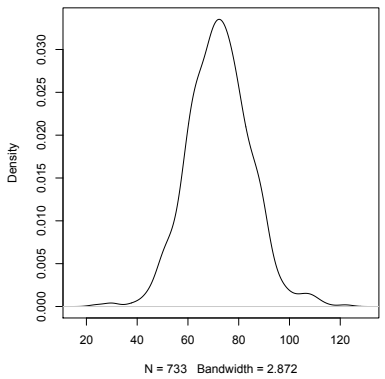
Example: Diabetes in Pima Indian Women

- Univariate

```
# simple data plot
plot(sort(pima$bp))
# histogram
hist(pima$bp)
# density plot
plot(density(pima$bp,na.rm=TRUE))
```





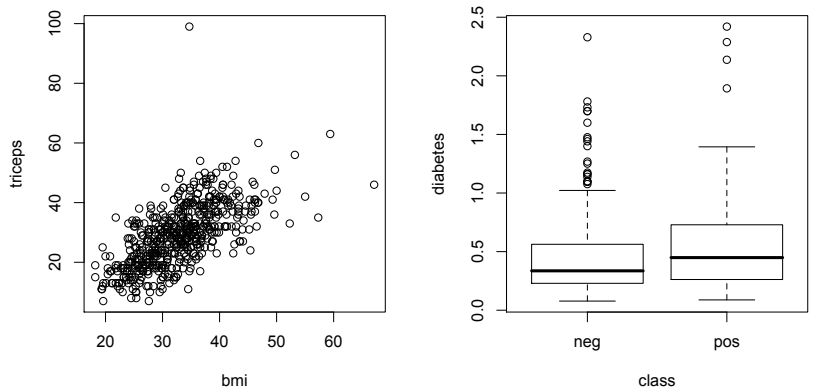


Graphical Summaries

Example: Diabetes in Pima Indian Women

- Bivariate

```
1  # scatterplot
2  plot(triceps~bmi, pima)
3  # boxplot
4  boxplot(diabetes~class, pima)
```



Linear regression with multiple predictors

Objective

Generalize the simple regression methodology in order to describe the relationship between a response variable Y and a set of predictors $X_1, X_2, \ldots, X_p$ in terms of a linear function.

The variables

$X_1, \ldots X_p$: explanatory variables
Y: response variable

After data collection, we have sets of observations:

$$(x_{11}, \ldots, x_{1p}, y_1), \ldots, (x_{n1}, \ldots, x_{np}, y_n)$$

Linear regression with multiple predictors

Example: Book Weight (Taken from Maindonald, 2007)

Loading the Data:

```
1    library(DAAG)
2    data(allbacks)
3    attach(allbacks)
4    # For the description of the data set:
5    ?allbacks
```

```
   volume area weight cover
1     885  382    800    hb
2    1016  468    950    hb
3    1125  387   1050    hb
...
14    595    0    425    pb
15   1034    0    725    pb
```

We want to be able to describe the weight of the book as a linear function of the volume and area (explanatory variables).

Linear regression with multiple predictors I

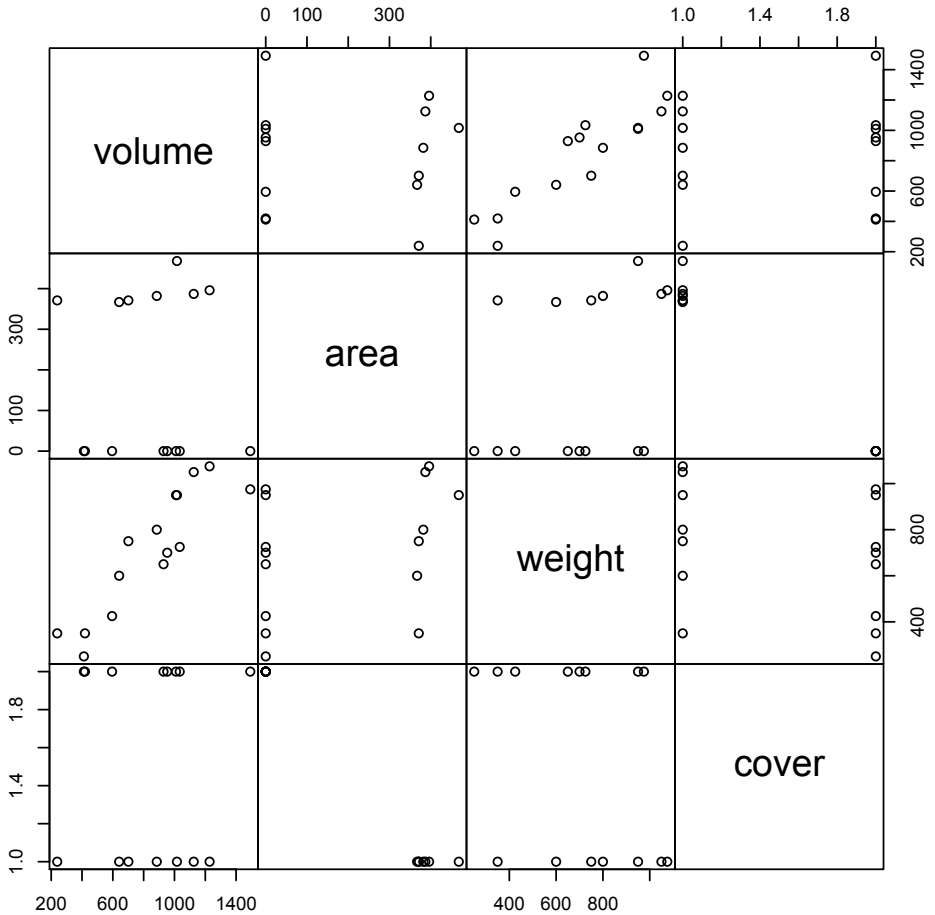
Example: Book Weight

The scatter plot allows one to obtain an overview of the relations between variables.

```
1    plot(allbacks, gap=0)
```

Linear regression with multiple predictors II

Example: Book Weight



Multivariate linear regression model [2]

The model is given by:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \epsilon_i \quad i = 1, \ldots, n$$

where:

- Random Error: $\epsilon_i \sim N(0, \sigma^2)$, independent
- Linear Function: $\beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p = E(Y|x_1, \ldots, x_p)$

Unknown parameters

- β_0: overall mean;
- $\beta_k, \quad k = 1, \ldots, p$: regression coefficients

[2] The linear in multivariate linear regression applies to the regression coefficients, not to the response or explanatory variables

Estimation of unknown parameters I

As in the case of simple linear regression, we want to find the equation of the line that "best" fits the data. In this case, it means finding $b_0, b_1, \ldots, b_p$ such that the fitted values of y_i, given by

$$\hat{y}_i = b_0 + b_1 x_{1i} + \ldots + b_p x_{pi},$$

are as "close" as possible to the observed values y_i.

Residuals

The difference between the observed value y_i and the fitted value $\hat{y}_i$ is called residual and is given by:

$$e_i = y_i - \hat{y}_i$$

Estimation of unknown parameters II

Least Squares Method

A usual way of calculating $b_0, b_1, \ldots, b_p$ is based on the minimization of the sum of the squared residuals, or residual sum of squares (RSS):

$$\begin{aligned} RSS &= \sum_i e_i^2 \\ &= \sum_i (y_i - \hat{y}_i)^2 \\ &= \sum_i (y_i - b_0 - b_1 x_{1i} - \ldots - b_p x_{pi})^2 \end{aligned}$$

Fitting a multivariate linear regression in R I

Example: Book Weight

The parameters $b_0, b_1, \ldots, b_p$ are estimated by using the function `lm()`:

```
1  # Fit the regression model using the
       function lm():
2  allbacks.lm <- lm(weight~volume + area,
       data= allbacks)
3  # Use the function summary() to get some
       results:
4  summary(allbacks.lm, corr=TRUE)
```

Fitting a multivariate linear regression in R II

Example: Book Weight

The output looks like this:

```
Call:
lm(formula = weight ~ volume + area, data = allbacks)

Residuals:
    Min      1Q  Median      3Q     Max
-104.06  -30.02  -15.46   16.76  212.30

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 22.41342   58.40247   0.384 0.707858
volume       0.70821    0.06107  11.597 7.07e-08 ***
area         0.46843    0.10195   4.595 0.000616 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 77.66 on 12 degrees of freedom
Multiple R-squared: 0.9285, Adjusted R-squared: 0.9166
F-statistic: 77.89 on 2 and 12 DF,  p-value: 1.339e-07

Correlation of Coefficients:
       (Intercept) volume
volume -0.88
area   -0.32        0.00
```

Fitted values and residuals

- Fitted values obtained using the function `fitted()`
- Residuals obtained using the function `resid()`

```
1 # Create a table with fitted values and
      residuals
2 data.frame(allbacks, fitted.value=fitted(
      allbacks.lm), residual=resid(allbacks.lm))
```

```
   volume area weight cover fitted.value    residual
1     885  382    800    hb     828.1183 -28.118305
2    1016  468    950    hb     961.1788 -11.178758
3    1125  387   1050    hb    1000.4300  49.569961
...
15   1034    0    725    pb     754.6989 -29.698938
```

Analysis of Variance (ANOVA) I

The ANOVA breaks the total variability observed in the sample into two parts:

Total sample variability (TSS)	=	Variability explained by the model (SSreg)	+	Unexplained (or error) variability (RSS)

Analysis of Variance (ANOVA) II

In R, we do:

```
1  anova(allbacks.lm)
```

```
Analysis of Variance Table

Response: weight
          Df Sum Sq Mean Sq F value    Pr(>F)
volume     1 812132  812132 134.659  7.02e-08 ***
area       1 127328  127328  21.112 0.0006165 ***
Residuals 12  72373    6031
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Models with no intercept term I

Example: Book Weight

To leave the intercept out, we do:

```
1 allbacks.lm0 <- lm(weight~-1+volume+area,
      data=allbacks)
2 summary(allbacks.lm0)
```

Models with no intercept term II

Example: Book Weight

The corresponding regression output is:

```
Call:
lm(formula = weight ~ -1 + volume + area, data = allbacks)

Residuals:
    Min      1Q  Median      3Q     Max
-112.53  -28.73  -10.52   24.62  213.80

Coefficients:
       Estimate Std. Error t value Pr(>|t|)
volume  0.72891    0.02767  26.344 1.15e-12 ***
area    0.48087    0.09344   5.146 0.000188 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 75.07 on 13 degrees of freedom
Multiple R-squared: 0.9914, Adjusted R-squared: 0.9901
F-statistic: 747.9 on 2 and 13 DF,  p-value: 3.799e-14
```

Comparing nested models I

Example: Savings Data (Taken from Faraway, 2002)

```
1 library(faraway)
2 data(savings)
3 attach(savings)
4 head(savings)
```

```
            sr pop15 pop75     dpi ddpi
Australia 11.43 29.35  2.87 2329.68 2.87
Austria   12.07 23.32  4.41 1507.99 3.93
Belgium   13.17 23.80  4.43 2108.47 3.82
Bolivia    5.75 41.89  1.67  189.13 0.22
Brazil    12.88 42.19  0.83  728.47 4.56
Canada     8.79 31.72  2.85 2982.88 2.43
```

Comparing nested models II

Example: Savings Data (Taken from Faraway, 2002)

```
1 # Fitting the model with all predictors
2 savings.lm <- lm(sr~pop15+pop75+dpi+ddpi,
      data=savings)
3 summary(savings.lm)
```

Comparing nested models III

Example: Savings Data (Taken from Faraway, 2002)

```
Call:
lm(formula = sr ~ pop15 + pop75 + dpi + ddpi, data = savings)

Residuals:
    Min      1Q  Median      3Q     Max
-8.2422 -2.6857 -0.2488  2.4280  9.7509

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 28.5660865  7.3545161   3.884 0.000334 ***
pop15       -0.4611931  0.1446422  -3.189 0.002603 **
pop75       -1.6914977  1.0835989  -1.561 0.125530
dpi         -0.0003369  0.0009311  -0.362 0.719173
ddpi         0.4096949  0.1961971   2.088 0.042471 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 3.803 on 45 degrees of freedom
Multiple R-squared: 0.3385, Adjusted R-squared: 0.2797
F-statistic: 5.756 on 4 and 45 DF,  p-value: 0.0007904
```

Comparing nested models IV

Example: Savings Data (Taken from Faraway, 2002)

```
1 # Fitting the model without the variable pop15
2 savings.lm15 <- lm(sr~pop75+dpi+ddpi, data=
      savings)
3 # Comparing the two models:
4 anova(savings.lm15, savings.lm)
```

```
Analysis of Variance Table

Model 1: sr ~ pop75 + dpi + ddpi
Model 2: sr ~ pop15 + pop75 + dpi + ddpi
  Res.Df    RSS Df Sum of Sq      F   Pr(>F)
1     46 797.72
2     45 650.71  1    147.01 10.167 0.002603 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Comparing nested models V

Example: Savings Data (Taken from Faraway, 2002)

An alternative way of fitting the nested model, by using the function `update()` :

```
1 savings.lm15alt <- update(savings.lm,
      formula=.~. - pop15)
2 summary(savings.lm15alt)
```

Comparing nested models VI

Example: Savings Data (Taken from Faraway, 2002)

```
Call:
lm(formula = sr ~ pop75 + dpi + ddpi, data = savings)

Residuals:
    Min      1Q  Median      3Q     Max
-8.0577 -3.2144  0.1687  2.4260 10.0763

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 5.4874944  1.4276619   3.844  0.00037 ***
pop75       0.9528574  0.7637455   1.248  0.21849
dpi         0.0001972  0.0010030   0.197  0.84499
ddpi        0.4737951  0.2137272   2.217  0.03162 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 4.164 on 46 degrees of freedom
Multiple R-squared: 0.189, Adjusted R-squared: 0.1361
F-statistic: 3.573 on 3 and 46 DF,  p-value: 0.02093
```

Measuring Goodness of Fit I

Coefficient of Determination, R^2

- R^2 represents the proportion of the total sample variability (sum of squares) explained by the regression model.
- indicates of how well the model fits the data.

Adjusted R^2

- R^2_{adj} represents the proportion of the mean sum of squares (variance) explained by the regression model.
- it takes into account the number of degrees of freedom and is preferable to R^2.

Measuring Goodness of Fit II

Attention

- Both R^2 and R^2_{adj} are given in the regression summary.
- Neither R^2 nor R^2_{adj} give direct indication on how well the model will perform in the prediction of a new observation.
- The use of these statistics is more legitimate in the case of comparing different models for the same data set.

Confidence and prediction bands I

Confidence Bands

Reflect the uncertainty about the regression line (how well the line is determined).

Prediction Bands

Include also the uncertainty about future observations.

Confidence and prediction bands II

Predicted values are obtained using the function `predict()` .

```
1 # Obtaining the confidence bands:
2 predict(savings.lm, interval="confidence")
```

```
                fit       lwr       upr
Australia  10.566420  8.573419 12.559422
Austria    11.453614  8.796229 14.110999
Belgium    10.951042  8.685716 13.216369
...
Malaysia    7.680869  5.724711  9.637027
```

Confidence and prediction bands III

```
1 # Obtaining the prediction bands:
2 predict(savings.lm, interval="prediction")
```

```
                fit         lwr      upr
Australia  10.566420  2.65239197 18.48045
Austria    11.453614  3.34673522 19.56049
Belgium    10.951042  2.96408447 18.93800
...
Malaysia    7.680869 -0.22396122 15.58570
```

Confidence and prediction bands IV

Attention

- these limits rely strongly on the assumption of independence and normally distributed errors with constant variance and should not be used if these assumptions are violated for the data being analyzed.
- the confidence and prediction bands only apply to the population from which the data were sampled.

Dummy Variables

In R, unordered factors are automatically treated as dummy variables, when included in a regression model.

Dummy Variable Regression I

Example: Duncan's Occupational Prestige Data

Loading the Data:

```
1 library(car)
2 data(Duncan)
3 attach(Duncan)
4 summary(Duncan)
```

```
  type         income          education         prestige
bc   :21   Min.   : 7.00   Min.   :  7.00   Min.   : 3.00
prof:18   1st Qu.:21.00   1st Qu.: 26.00   1st Qu.:16.00
wc   : 6   Median :42.00   Median : 45.00   Median :41.00
           Mean   :41.87   Mean   : 52.56   Mean   :47.69
           3rd Qu.:64.00   3rd Qu.: 84.00   3rd Qu.:81.00
           Max.   :81.00   Max.   :100.00   Max.   :97.00
```

Dummy Variable Regression II

Example: Duncan's Occupational Prestige Data

Fitting the linear regression:

```
1 # Fit the linear regression model
2 duncan.lm <- lm(prestige~income+education+type
      , data=Duncan)
3 # Extract the regression results
4 summary(duncan.lm)
```

Dummy Variable Regression III

Example: Duncan's Occupational Prestige Data

The output looks like this:

```
Call:
lm(formula = prestige ~ income + education + type, data = Duncan)

Residuals:
    Min      1Q  Median      3Q     Max
-14.890  -5.740  -1.754   5.442  28.972

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.18503    3.71377  -0.050  0.96051
income        0.59755    0.08936   6.687 5.12e-08 ***
education     0.34532    0.11361   3.040  0.00416 **
typeprof     16.65751    6.99301   2.382  0.02206 *
typewc      -14.66113    6.10877  -2.400  0.02114 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 9.744 on 40 degrees of freedom
Multiple R-squared: 0.9131, Adjusted R-squared: 0.9044
F-statistic:   105 on 4 and 40 DF,  p-value: < 2.2e-16
```

Dummy Variable Regression IV

Example: Duncan's Occupational Prestige Data

Note

- `R` internally creates a dummy variable for each level of the factor variable.
- overspecification is avoided by applying "contrasts" (constraints) on the parameters.
- by default, it uses dummy variables for all levels except the first one (in the example, `bc`), set as the reference level.
- Contrasts can be changed by using the function `contrasts()`.

Interactions in formulas

Let `a, b, c` be categorical variables and `x, y` be numerical variables. Interactions are specified in R as follows[3]:

Formula	Description
`y~a+x`	no interaction
`y~a:x`	interaction between variables `a` and `x`
`y~a*x`	the same and also includes the main effects
`y~a/x`	interaction between variables `a` and `x` (nested)
`y~(a+b+c)^2`	includes all two-way interactions
`y~a*b*c-a:b:c`	excludes the three-way interaction
`I()`	to use the original arithmetic operators

[3] Adapted from Kleiber and Zeileis, 2008.

Diagnostics

Assumptions

- Y relates to the predictor variables $X_1, \ldots, X_p$ by a linear regression model:

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \epsilon$$

- the errors are independent and identically normally distributed with mean zero and common variance

Diagnostics I

What can go wrong?

Violations:

- In the linear regression model:
  - linearity (e.g. higher order terms)
- In the residual assumptions:
  - non-normal distribution
  - non-constant variances
  - dependence
  - outliers

Diagnostics II

What can go wrong?

Checks:

- Residuals vs. each predictor variable
  - $\Rightarrow$ nonlinearity: higher-order terms in that variable
- Residuals vs. fitted values
  - $\Rightarrow$ variance increasing with the response: transformation
- Residuals Q-Q norm plot
  - $\Rightarrow$ deviation from a straight line: non-normality

Diagnostic Plots I

Checking assumptions graphically

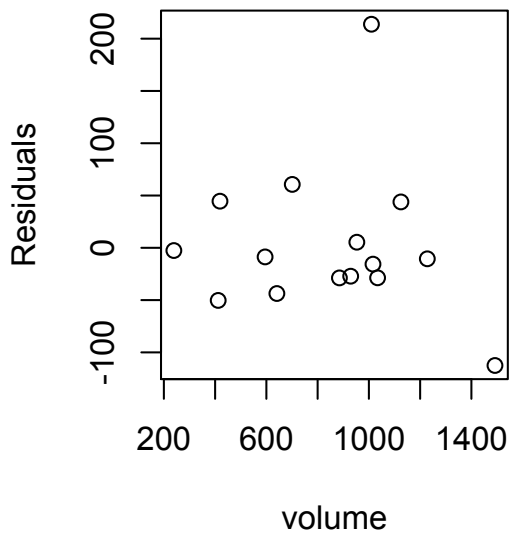
Book Weight example:

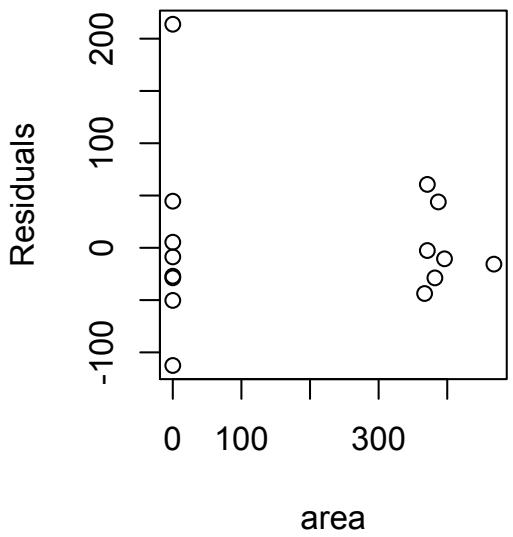
- Residuals vs. predictors

```
1 par(mfrow=c(1,2))
2 plot(resid(allbacks.lm)~volume,ylab="Residuals
      ")
3 plot(resid(allbacks.lm)~area,ylab="Residuals")
4 par(mfrow=c(1,1))
```

Diagnostic Plots II

Checking assumptions graphically





Leverage, influential points and outliers

Leverage points

Leverage points are those which have great influence on the fitted model.

- Bad leverage point: if it is also an outlier, that is, the y-value does not follow the pattern set by the other data points.
- Good leverage point: if it is not an outlier.

Standardized residuals I

Standardized residuals are obtained by dividing each residual by an estimate of its standard deviation:

$$r_i = \frac{e_i}{\hat{\sigma}(e_i)}$$

To obtain the standardized residuals in R, use the command `rstandard()` on the regression model.

Leverage Points

- Good leverage points have their standardized residuals within the interval $[-2, 2]$
- Outliers are leverage points whose standardized residuals fall outside the interval $[-2, 2]$

Cook's Distance

Cook's Distance: D

- the Cook's distance statistic combines the effects of leverage and the magnitude of the residual.
- it is used to evaluate the impact of a given observation on the estimated regression coefficients.

$\Rightarrow$ $D > 1$: undue influence

The Cook's distance plot is obtained by applying the function `plot()` to the linear model object.

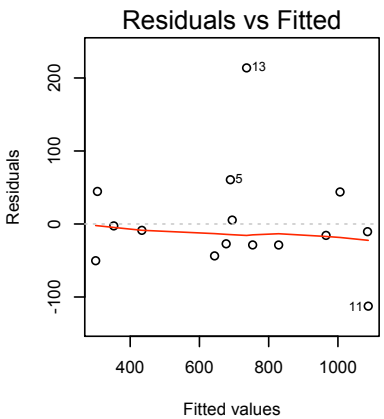
More diagnostic plots I

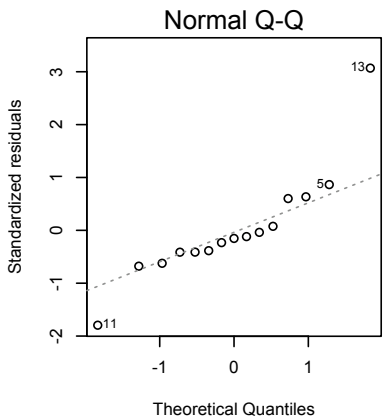
Checking assumptions graphically

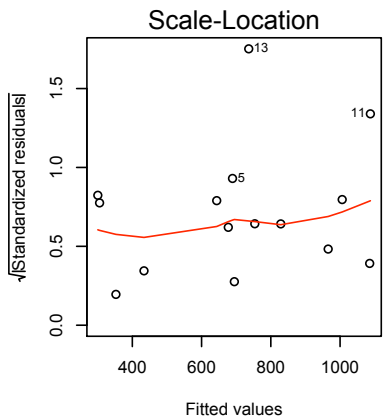
Book Weight example:

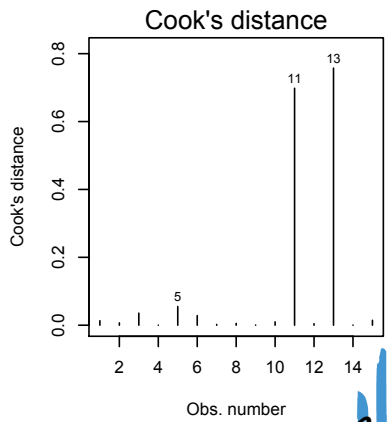
- Other Residual Plots

```
1 par(mfrow=c(1,4))
2 plot(allbacks.lm0, which=1:4)
3 par(mfrow=c(1,1))
```









How to deal with leverage, influential points and outliers I

- Remove invalid data points
  - $\Rightarrow$ if they look unusual or are different from the rest of the data
- Fit a different regression model
  - $\Rightarrow$ if the model is not valid for the data
    - higher-order terms
    - transformation

Removing points I

```
1 allbacks.lm13 <- lm(weight~-1+volume+area,
      data=allbacks[-13,])
2 summary(allbacks.lm13)
```

```
Call:
lm(formula = weight ~ -1 + volume + area, data = allbacks[-13,
    ])

Residuals:
    Min      1Q  Median      3Q     Max
-61.721 -25.291   3.429  31.244  58.856

Coefficients:
       Estimate Std. Error t value Pr(>|t|)
volume  0.69485    0.01629   42.65 1.79e-14 ***
area    0.55390    0.05269   10.51 2.08e-07 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 41.02 on 12 degrees of freedom
Multiple R-squared: 0.9973, Adjusted R-squared: 0.9969
F-statistic:  2252 on 2 and 12 DF,  p-value: 3.521e-16
```

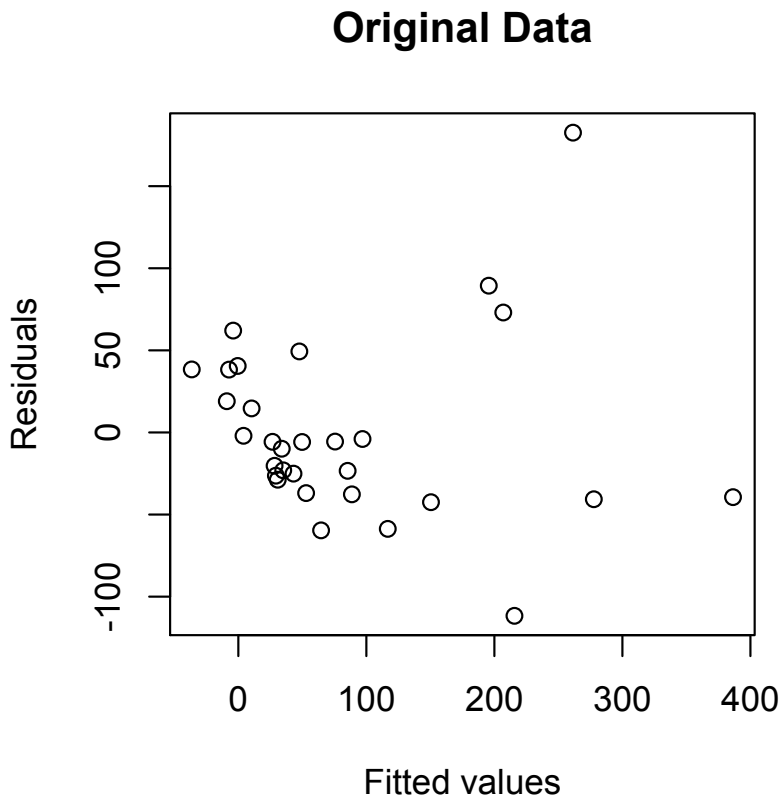
Non-constant variance I

Example: Galapagos Data (Taken from Faraway, 2002)

```
# Loading the data
library(faraway)
data(gala)
attach(gala)
# Fitting the model
gala.lm <- lm(Species~Area+Elevation+Scruz+
    Nearest+Adjacent, data=gala)
# Residuals vs. fitted values
plot(resid(gala.lm)~fitted(gala.lm), xlab="
    Fitted values", ylab="Residuals", main= "
    Original Data")
```

Non-constant variance II

Example: Galapagos Data (Taken from Faraway, 2002)



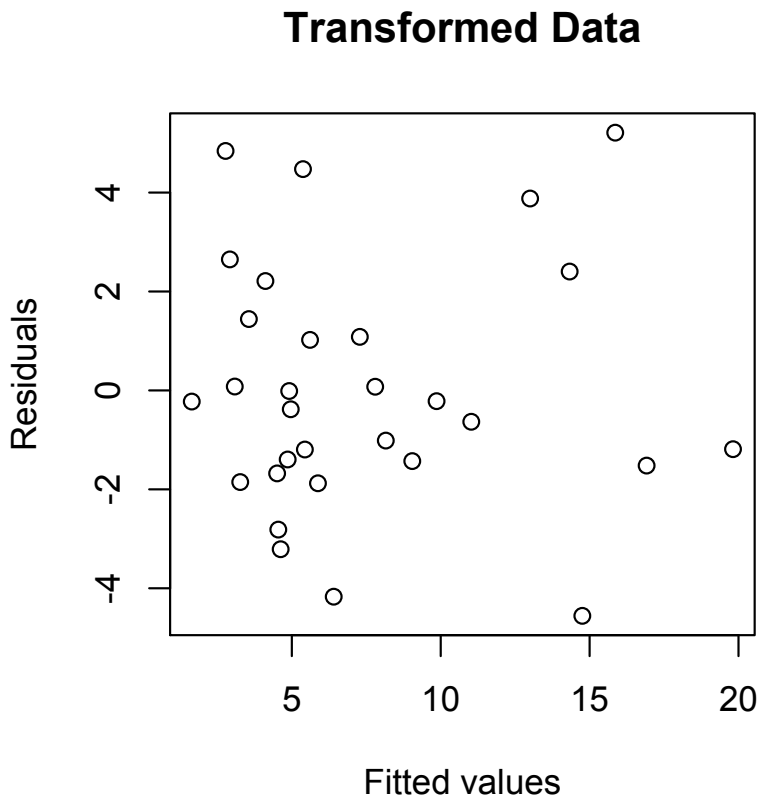
Non-constant variance III

Example: Galapagos Data (Taken from Faraway, 2002)

```
# Applying square root transformation:
sqrtSpecies <- sqrt(Species)
gala.sqrt <- lm(sqrtSpecies~Area+Elevation+
    Scruz+Nearest+Adjacent)
# Residuals vs. fitted values
plot(resid(gala.sqrt)~fitted(gala.sqrt), xlab=
    "Fitted values", ylab="Residuals", main= "
    Transformed Data")
```

Non-constant variance IV

Example: Galapagos Data (Taken from Faraway, 2002)



Generalized Linear Models

Objective

To model the realtionship between our predictors $X_1, X_2, \ldots, X_p$ and a binary or other count variable Y which cannot be modeled with standard regression.

Possible Applications

- Medical Data
- Internet Traffic
- Survival Data

Poisson Regression

Example: Randomized Controlled Trial (Taken from Dobson, 1990)

```
counts <- c(18,17,15,20,10,20,25,13,12)
outcome <- gl(3,1,9)
treatment <- gl(3,3)
print(d.AD <- data.frame(treatment, outcome,
    counts))
```

```
  treatment outcome counts
1         1       1     18
2         1       2     17
3         1       3     15
4         2       1     20
5         2       2     10
6         2       3     20
7         3       1     25
8         3       2     13
9         3       3     12
```

Poisson Regression

Example: Randomized Controlled Trial (Taken from Dobson, 1990)

```
1 glm.D93 <- glm(counts ~ outcome + treatment,
      family=poisson())
2 anova(glm.D93)
```

```
Analysis of Deviance Table

Model: poisson, link: log

Response: counts

Terms added sequentially (first to last)


          Df  Deviance Resid. Df Resid. Dev
NULL                          8    10.5814
outcome    2    5.4523         6     5.1291
treatment  2 8.882e-15         4     5.1291
```

Poisson Regression

Example: Randomized Controlled Trial (Taken from Dobson, 1990)

```
1 summary(glm.D93)
```

```
Call:
glm(formula = counts ~ outcome + treatment, family = poisson())

Deviance Residuals:
       1         2         3         4         5         6         7
-0.67125   0.96272  -0.16965  -0.21999  -0.95552   1.04939   0.84715
       8         9
-0.09167  -0.96656

Coefficients:
             Estimate Std. Error  z value Pr(>|z|)
(Intercept)  3.045e+00  1.709e-01   17.815   <2e-16 ***
outcome2    -4.543e-01  2.022e-01   -2.247   0.0246 *
outcome3    -2.930e-01  1.927e-01   -1.520   0.1285
treatment2   1.338e-15  2.000e-01 6.69e-15   1.0000
treatment3   1.421e-15  2.000e-01 7.11e-15   1.0000
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 10.5814  on 8  degrees of freedom
Residual deviance:  5.1291  on 4  degrees of freedom
AIC: 56.761

Number of Fisher Scoring iterations: 4
```

Poisson Regression

Example: Randomized Controlled Trial (Taken from Dobson, 1990)

```
1 anova(glm.D93, test = "Cp")
```

```
Analysis of Deviance Table

Model: poisson, link: log

Response: counts

Terms added sequentially (first to last)


          Df  Deviance Resid. Df Resid. Dev     Cp
NULL                          8    10.5814 12.581
outcome    2    5.4523          6     5.1291 11.129
treatment  2 8.882e-15          4     5.1291 15.129
```

Poisson Regression

Example: Randomized Controlled Trial (Taken from Dobson, 1990)

```
1 anova(glm.D93, test = "Chisq")
```

```
Analysis of Deviance Table

Model: poisson, link: log

Response: counts

Terms added sequentially (first to last)


          Df  Deviance Resid. Df Resid. Dev P(>|Chi|)
NULL                          8    10.5814
outcome    2    5.4523         6     5.1291    0.0655
treatment  2 8.882e-15         4     5.1291    1.0000
```

Logistic Regression

Example: Insecticide Concentration (Taken from Faraway, 2006)

```
1 library(faraway)
2 data(bliss)
3 bliss
```

```
  dead alive conc
1    2    28    0
2    8    22    1
3   15    15    2
4   23     7    3
5   27     3    4
```

Logistic Regression

Example: Insecticide Concentration (Taken from Faraway, 2006)

```
1 modl<-glm(cbind(dead,alive)~conc,family=
    binomial, data=bliss)
2 summary(modl)
```

```
Call:
glm(formula = cbind(dead, alive) ~ conc, family = binomial, data = bliss)

Deviance Residuals:
      1       2       3       4       5
-0.4510  0.3597  0.0000  0.0643 -0.2045

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.3238     0.4179  -5.561 2.69e-08 ***
conc          1.1619     0.1814   6.405 1.51e-10 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 64.76327  on 4  degrees of freedom
Residual deviance:  0.37875  on 3  degrees of freedom
AIC: 20.854

Number of Fisher Scoring iterations: 4
```

Other Link Functions

Example: Insecticide Concentration (Taken from Faraway, 2006)

```
1 modp<-glm(cbind(dead,alive)~conc,family=
    binomial(link=probit),data=bliss)
2 modc<-glm(cbind(dead,alive)~conc,family=
    binomial(link=cloglog),data=bliss)
```

Other Link Functions

Example: Insecticide Concentration (Taken from Faraway, 2006)

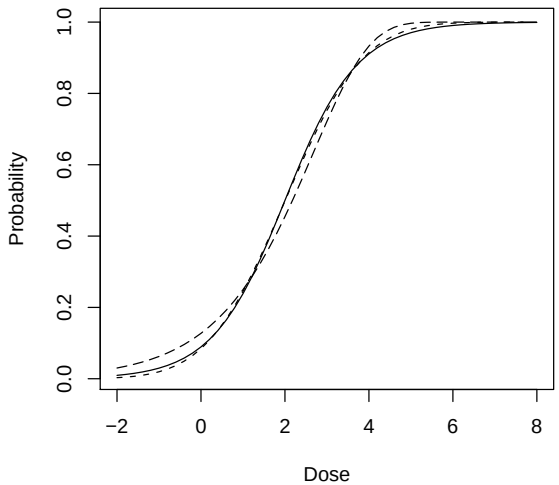
```
1  cbind(fitted(modl),fitted(modp),fitted(modc))
```

```
        [,1]       [,2]       [,3]
1 0.08917177 0.08424186 0.1272700
2 0.23832314 0.24487335 0.2496909
3 0.50000000 0.49827210 0.4545910
4 0.76167686 0.75239612 0.7217655
5 0.91082823 0.91441122 0.9327715
```

Other Link Functions

Example: Insecticide Concentration (Taken from Faraway, 2006)

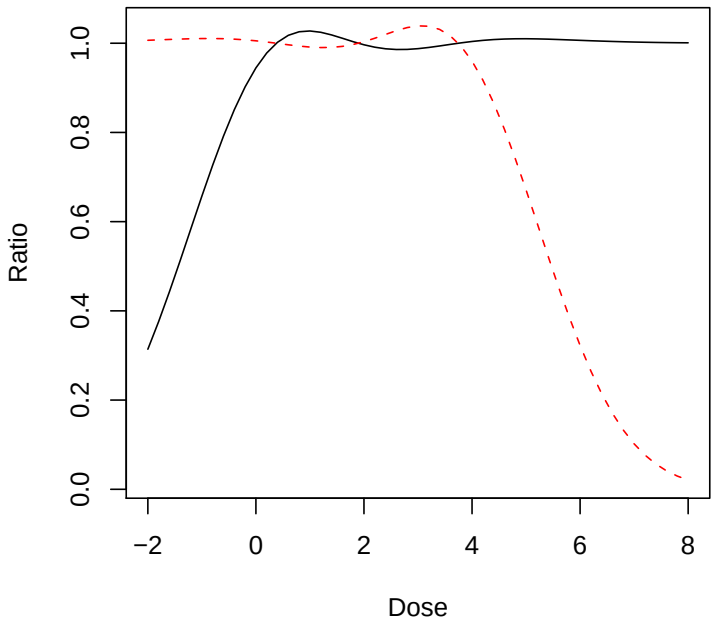
```
x<-seq(-2,8,.2)
pl<-ilogit(modl$coef[1]+modl$coef[2]*x)
pp<-pnorm(modp$coef[1]+modp$coef[2]*x)
pc<-1-exp(-exp((modc$coef[1]+modc$coef[2]*x)))
plot(x,pl,type="l",ylab="Probability",xlab="
    Dose")
lines(x,pp,lty=2)
lines(x,pc,lty=5)
```



But choose carefully!

Example: Insecticide Concentration (Taken from Faraway, 2006)

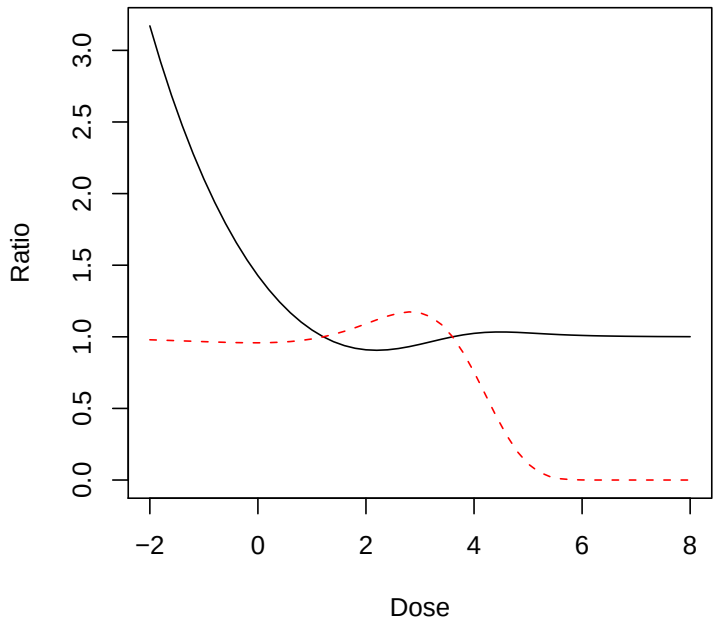
```
1  matplot(x,cbind(pp/pl,(1-pp)/(1-pl)),type="l",
       xlab="Dose",ylab="Ratio")
```



But choose carefully!

Example: Insecticide Concentration (Taken from Faraway, 2006)

```
1  matplot(x,cbind(pc/pl,(1-pc)/(1-pl)),type="l",
       xlab="Dose",ylab="Ratio")
```



Online Resources for R

Download R: `http://cran.stat.ucla.edu`

Search Engine for R: `rseek.org`

R Reference Card: `http://cran.r-project.org/doc/contrib/Short-refcard.pdf`

UCLA Statistics Information Portal: `http://info.stat.ucla.edu/grad/`

UCLA Statistical Consulting Center `http://scc.stat.ucla.edu`

References I

P. Daalgard
Introductory Statistics with R,
Statistics and Computing, Springer-Verlag, NY, 2002.

B.S. Everitt and T. Hothorn
A Handbook of Statistical Analysis using R,
Chapman & Hall/CRC, 2006.

J.J. Faraway
Practical Regression and Anova using R,
`www.stat.lsa.umich.edu/~faraway/book`

J.J. Faraway
Extending the Linear Model with R,
Chapman & Hall/CRC, 2006.

References II


J. Maindonald and J. Braun
Data Analysis and Graphics using R – An Example-Based Approach,
Second Edition, Cambridge University Press, 2007.


[Sheather, 2009] S.J. Sheather
A Modern Approach to Regression with R,
DOI: 10.1007/978-0-387-09608-7-3,
Springer Science + Business Media LCC 2009.

Upcoming Mini-Courses

- For a schedule of all mini-courses offered please visit: `http://scc.stat.ucla.edu/mini-courses`

Feedback Survey

PLEASE follow this link and take our brief survey:

`http://scc.stat.ucla.edu/survey`

It will help us improve this course. Thank you.

Thank you.

Any Questions?