UCLA Department of Statistics
Statistical Consulting Center

Regression in R
Part I :
Simple Linear Regression

Denise Ferrari & Tiffany Head
denise@stat.ucla.edu tiffany@stat.ucla.edu

Feb 10, 2010

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.

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.

```
?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

```
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:

```
getwd()
```

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

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

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

```
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:

```
library(MASS)
```

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

```
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:

```
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

```r
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`:

```
pima$glucose[pima$glucose==0] <- NA
pima$bp[pima$bp==0] <- NA
pima$triceps[pima$triceps==0] <- NA
pima$insulin[pima$insulin==0] <- NA
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`:

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

Final Coding

Example: Diabetes in Pima Indian Women

```
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

```
# scatterplot
plot(triceps~bmi, pima)
# boxplot
boxplot(diabetes~class, pima)
```

Linear regression with a single predictor

Objective

Describe the relationship between two variables, say X and Y as a straight line, that is, Y is modeled as a linear function of X.

The variables

X: explanatory variable (horizontal axis)
Y: response variable (vertical axis)

After data collection, we have pairs of observations:

$$(x_1, y_1), \ldots, (x_n, y_n)$$

Linear regression with a single predictor

Example: Production Runs (Taken from Sheather, 2009)

Loading the Data:

```
production <- read.table("http://www.stat.tamu.edu/~sheather/book/docs/datasets/production.txt", header=T, sep="")
```

Case	RunTime	RunSize
1	195	175
2	215	189
3	243	344
4	162	88
5	185	114
...		
18	230	337
19	208	146
20	172	68

Variables:

RunTime (Y): time taken (in minutes) for a production run

RunSize (X): number of items produced in each run

We want to be able to describe the production run time as a linear function of the number of items in the run

Linear regression with a single predictor I

Example: Production Runs

The scatter plot allows one to check if the linear relationship is supported by the data.

```
attach(production)
plot(RunTime
     ~RunSize)
```

Simple linear regression model

The regression of variable Y on variable X is given by:

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

where:

- Random Error: $\epsilon_i \sim N(0, \sigma^2)$, independent
- Linear Function: $\beta_0 + \beta_1 x_i = E(Y|X = x_i)$

Unknown parameters

- β_0 (Intercept): point in which the line intercepts the y-axis;
- β_1 (Slope): increase in Y per unit change in X.

Estimation of unknown parameters I

We want to find the equation of the line that "best" fits the data. It means finding b_0 and b_1 such that the fitted values of y_i, given by

$$\hat{y}_i = b_0 + b_1 x_i,$$

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 and b_1 is based on the minimization of the sum of the squared residuals, or residual sum of squares (RSS):

$$\begin{aligned} RSS &= \sum_i e_i^2 \\ &= \sum_i (y_i - \hat{y}_i)^2 \\ &= \sum_i (y_i - b_0 - b_1 x_i)^2 \end{aligned}$$

Fitting a simple linear regression in R I

Example: Production Runs

The parameters b_0 and b_1 are estimated by using the function `lm()`:

```
# Fit the regression model using the function lm():
production.lm <- lm(RunTime~RunSize, data=production)
# Use the function summary() to get some results:
summary(production.lm)
```

Fitting a simple linear regression in R II

Example: Production Runs

The output looks like this:

```
Call:
lm(formula = RunTime ~ RunSize, data = production)

Residuals:
    Min      1Q  Median      3Q     Max
-28.597 -11.079   3.329   8.302  29.627

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 149.74770    8.32815   17.98 6.00e-13 ***
RunSize       0.25924    0.03714    6.98 1.61e-06 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 16.25 on 18 degrees of freedom
Multiple R-squared: 0.7302, Adjusted R-squared: 0.7152
F-statistic: 48.72 on 1 and 18 DF,  p-value: 1.615e-06
```

Fitted values and residuals

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

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

Case	RunTime	RunSize	fitted.value	residual
1	195	175	195.1152	-0.1152469
2	215	189	198.7447	16.2553496
3	243	344	238.9273	4.0726679
...				
20	172	68	167.3762	4.6237657

$$\hat{y}_1 = 149.75 + 0.26 * 175 = 195.115$$
$$e_1 = 195 - 195.115 = -0.115$$

Fitted values and residuals I

When there are missing data

Missing data need to be handled carefully. Using the `na.exclude` method:

```
# Load the package that contains the data
library(ISwR)
data(thuesen); attach(thuesen)
# Option for dealing with missing data
options(na.action=na.exclude)
# Now fit the regression model as before
velocity.lm <- lm(short.velocity~blood.glucose
    )
# Create a table with fitted values and
    residuals
data.frame(thuesen, fitted.value=fitted(
    velocity.lm), residual=resid(velocity.lm))
```

Fitted values and residuals II

When there are missing data

```
   blood.glucose short.velocity fitted.value     residual
1           15.3           1.76     1.433841  0.326158532
2           10.8           1.34     1.335010  0.004989882
...
16           8.6             NA           NA           NA
...
23           8.8           1.12     1.291085 -0.171085074
24           9.5           1.70     1.306459  0.393541161
```

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:

```
anova(production.lm)
```

```
Analysis of Variance Table

Response: RunTime
          Df  Sum Sq Mean Sq F value    Pr(>F)
RunSize    1 12868.4 12868.4  48.717 1.615e-06
Residuals 18  4754.6   264.1
```

Measuring Goodness of Fit I

Coefficient of Determination, R^2

- represents the proportion of the total sample variability explained by the regression model.
- for simple linear regression, the R^2 statistic corresponds to the square of the correlation between Y and X.
- indicates of how well the model fits the data.

From the ANOVA table:

$$R^2 = \frac{12868.4}{(12868.4 + 4754.6)} = 0.7302$$

which we can also find in the regression summary.

Measuring Goodness of Fit II

Adjusted R^2

The adjusted R^2 takes into account the number of degrees of freedom and is preferable to R^2.

From the ANOVA table:

$$R^2_{adj} = 1 - \frac{4754.6/18}{(12868.4 + 4754.6)/(18+1)} = 0.7152$$

also found in the regression summary.

Attention

Neither R^2 nor R^2_{adj} give direct indication on how well the model will perform in the prediction of a new observation.

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.

Attention

These limits rely strongly on the assumption of normally distributed errors with constant variance and should not be used if this assumption is violated for the data being analyzed.

Confidence and prediction bands II

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

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

```
        fit      lwr      upr
1  195.1152 187.2000 203.0305
2  198.7447 191.0450 206.4443
3  238.9273 225.4549 252.3998
...
20 167.3762 154.4448 180.3077
```

Confidence and prediction bands III

```r
# Obtaining the prediction bands:
predict(production.lm, interval="prediction")
```

```
       fit      lwr      upr
1  195.1152 160.0646 230.1659
2  198.7447 163.7421 233.7472
3  238.9273 202.2204 275.6343
...
20 167.3762 130.8644 203.8881
```

Confidence and prediction bands IV

For plotting:

```
# Create a new data frame containing the values of X at which we want the predictions to be made
pred.frame <- data.frame(RunSize= seq(55, 345, by=10))
# Confidence bands
pc <- predict(production.lm, int="c", newdata= pred.frame)
# Prediction bands
pp <- predict(production.lm, int="p", newdata= pred.frame)
```

Confidence and prediction bands V

```
# Plot
require(graphics)
# Standard scatterplot with extended limits
plot(RunSize, RunTime, ylim=range(RunSize, pp,
    na.rm=T))
pred.Size <- pred.frame$RunSize
# Add curves
matlines(pred.Size, pc, lty=c(1,2,2), lwd=1.5,
    col=1)
matlines(pred.Size, pp, lty=c(1,3,3), lwd=1.5,
    col=1)
```

Confidence and prediction bands VI

Dummy Variable Regression

The simple dummy variable regression is used when the predictor variable is not quantitative but categorical and assumes only two values.

Dummy Variable Regression I

Example: Change over time (Taken from Sheather, 2009)

Loading the Data:

```
changeover <- read.table("http://www.stat.tamu.edu/~sheather/book/docs/datasets/changeover_times.txt", header=T, sep="")
```

```
      Method Changeover New
1   Existing         19   0
2   Existing         24   0
3   Existing         39   0
    ...

118      New         14   1
119      New         40   1
120      New         35   1
```

Variables:

Change-over(Y): time (in minutes) required to change the line of food production

New (X): 1 for the new method, 0 for the existing method

We want to be able to test whether the change-over time is different for the two methods.

Dummy Variable Regression II

Example: Change over time (Taken from Sheather, 2009)

```
attach(changeover)
# Summary:
summary(changeover)
```

```
     Method     Changeover          New
 Existing:72   Min.   : 5.00   Min.   :0.0
 New     :48   1st Qu.:11.00   1st Qu.:0.0
               Median :15.00   Median :0.0
               Mean   :16.59   Mean   :0.4
               3rd Qu.:21.00   3rd Qu.:1.0
               Max.   :40.00   Max.   :1.0
```

We need to recode the X variable (New) to `factor` :

Dummy Variable Regression III

Example: Change over time (Taken from Sheather, 2009)

```
changeover$New <- factor(changeover$New)
summary(changeover)
```

```
     Method      Changeover     New
Existing:72   Min.   : 5.00   0:72
New     :48   1st Qu.:11.00   1:48
              Median :15.00
              Mean   :16.59
              3rd Qu.:21.00
              Max.   :40.00
```

Dummy Variable Regression IV

Example: Change over time (Taken from Sheather, 2009)

Plotting the data:

```
plot(Changeover~New)
```

Dummy Variable Regression V

Example: Change over time (Taken from Sheather, 2009)

Fitting the linear regression:

```
# Fit the linear regression model
changeover.lm <- lm(Changeover~New, data=
    changeover)
# Extract the regression results
summary(changeover.lm)
```

Dummy Variable Regression VI

Example: Change over time (Taken from Sheather, 2009)

The output looks like this:

```
Call:
lm(formula = Changeover ~ New, data = changeover)

Residuals:
    Min      1Q  Median      3Q     Max
-10.861  -5.861  -1.861   4.312  25.312

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  17.8611     0.8905  20.058   <2e-16 ***
New1         -3.1736     1.4080  -2.254   0.0260 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 7.556 on 118 degrees of freedom
Multiple R-squared: 0.04128, Adjusted R-squared: 0.03315
F-statistic: 5.081 on 1 and 118 DF,  p-value: 0.02604
```

Dummy Variable Regression VII

Example: Change over time (Taken from Sheather, 2009)

Analysis of the results:

- There's significant evidence of a reduction in the mean change-over time for the new method.
- The estimated mean change-over time for the new method ($X = 1$) is:

$$\hat{y}_1 = 17.8611 + (-3.1736) * 1 = 14.7 \text{ minutes}$$

- The estimated mean change-over time for the existing method ($X = 0$) is:

$$\hat{y}_0 = 17.8611 + (-3.1736) * 0 = 17.9 \text{ minutes}$$

Diagnostics

Assumptions

The assumptions for simple linear regression are:

- Y relates to X by a linear regression model:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

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

Diagnostics

What can go wrong?

Violations:

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

Checks:

⇒ look at plot of residuals vs. X

⇒ look at plot of residuals vs. fitted values

⇒ look at residuals Q-Q norm plot

Validity of the regression model I

Example: The Anscombe's data sets (Taken from Sheather, 2009)

```
# Loading the data:
anscombe <- read.table("http://www.stat.tamu.edu/~sheather/book/docs/datasets/anscombe.txt", h=T, sep="" )
attach(anscombe)
# Looking at the data:
anscombe
```

```
case x1 x2 x3 x4    y1   y2    y3    y4
1    10 10 10  8  8.04 9.14  7.46  6.58
2     8  8  8  8  6.95 8.14  6.77  5.76
...
10    7  7  7  8  4.82 7.26  6.42  7.91
11    5  5  5  8  5.68 4.74  5.73  6.89
```

Validity of the regression model II

Example: The Anscombe's data sets (Taken from Sheather, 2009)

```
# Fitting the regressions
a1.lm <- lm(y1~x1, data=anscombe)
a2.lm <- lm(y2~x2, data=anscombe)
a3.lm <- lm(y3~x3, data=anscombe)
a4.lm <- lm(y4~x4, data=anscombe)

#Plotting
# For the first data set
plot(y1~x1, data=anscombe)
abline(a1.lm, col=2)
```

Validity of the regression model III

Example: The Anscombe's data sets (Taken from Sheather, 2009)

For all data sets, the fitted regression is the same:

$$\hat{y} = 3.0 + 0.5x$$

All models have $R^2 = 0.67$, $\hat{\sigma} = 1.24$ and the slope coefficients are significant at $< 1\%$ level. To check that, use the `summary()` function on the regression models.

Residual Plots I

Checking assumptions graphically

- Residuals vs. X

```
# For the first data set
plot(resid(a1.lm)~x1)
```

Residual Plots II

Checking assumptions graphically

- Residuals vs. fitted values

```
# For the first data set
plot(resid(a1.lm)~fitted(a1.lm))
```

Leverage (or influential) points and outliers I

Leverage points

Leverage points are those which have great influence on the fitted model, that is, those whose x-value is distant from the other x-values.

- 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.

Leverage (or influential) points and outliers II

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 $[-2, 2]$

Leverage (or influential) points and outliers I

How to deal with them

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

Leverage (or influential) points and outliers II

How to deal with them

Data set containing outliers:

Leverage (or influential) points and outliers III

How to deal with them

After their removal:

Normality and constant variance of errors

Normality and Constant Variance Assumptions

These assumptions are necessary for inference:

- hypothesis testing
- confidence intervals
- prediction intervals

$\Rightarrow$ Check the Normal Q-Q plot of the standardized residuals.

$\Rightarrow$ Check the Standardized Residuals vs. X plot.

Note

When these assumptions do not hold, we can try to correct the problem using data transformations.

Normality and constant variance checks

Example: Production Runs

```
# Regression model
production.lm <- lm(RunTime~RunSize, data=
    production)
# Residual plots
plot(production.lm)
```

When to use transformations?

Transformations can be used to correct for:

- non-constant variance
- non-linearity
- non-normality

The most common transformations are:

- Square root
- Log
- Power transformation

Example of correction: non-constant variance I

Example: Cleaning Data (Taken from Sheather, 2009)

Variables:

Rooms (Y): number of rooms cleaned
Crews (X): number crews

We want to be able to model the relationship between the number of rooms cleaned and the number of crews.

```
# Load the data
cleaning <- read.table("http://www.stat.tamu.edu/~sheather/book/docs/datasets/cleaning.txt", h=T, sep="")
attach(cleaning)
```

Example of correction: non-constant variance II

Example: Cleaning Data (Taken from Sheather, 2009)

```
# Regression model
cleaning.lm <- lm(Rooms ~Crews, data=cleaning)
# Plotting data and regression line
plot(Rooms~Crews)
abline(cleaning.lm, col=2)
```

Example of correction: non-constant variance III

Example: Cleaning Data (Taken from Sheather, 2009)

```
# Diagnostic plots
plot(cleaning.lm)
```

Example of correction: non-constant variance IV

Example: Cleaning Data (Taken from Sheather, 2009)

```
# Applying square root transformation (counts)
sqrtRooms <- sqrt(Rooms)
sqrtCrews <- sqrt(Crews)
# Regression model on the transformed data
sqrt.lm <- lm(sqrtRooms ~sqrtCrews)
# Plotting data and regression line
plot(sqrtRooms~sqrtCrews)
abline(sqrt.lm, col=2)
```

Example of correction: non-constant variance V

Example: Cleaning Data (Taken from Sheather, 2009)

```
# Diagnostic plots
plot(sqrt.lm)
```

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.umic.edu/~faraway/book

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.

Online Resources for R

- Next Week:
  - Nonlinear Regression (Feb 15, Monday)
- Week After Next:
  - Survival Analysis in R(February 22, Monday)
  - Spatial Statistics in R (February 24, Wednesday)
- For a schedule of all mini-courses offered please visit: `http://scc.stat.ucla.edu/mini-courses`

Thank you.

Any Questions?

Exercise in R I

Airfares Data (Taken from Sheather, 2009)

The data set for this exercise can be found at:

`http://www.stat.tamu.edu/~sheather/book/docs/datasets/airfares.txt`

It contains information on one-way airfare (in US$) and distance (in miles) from city A to 17 other cities in the US.

Exercise in R II

Airfares Data (Taken from Sheather, 2009)

1. Fit the regression model given by:

$$\text{Fare} = \beta_0 + \beta_1 \text{Distance} + \epsilon$$

2. Critique the following statement:

> *The regression coefficient of the predictor variable (Distance) is highly statistically significant and the model explains 99.4% of the variability in the Y-variable (Fare).*
>
> *Thus this model is highly effective for both understanding the effects of Distance on Fare and for predicting future values of Fare given the value of the predictor variable.*

3. Does the regression model above seem to fit the data well? If not, describe how the model can be improved.