# Linear Regression

Gerardo Ferrara

Master in Economics and Complexity, Collegio Carlo Alberto

Fall 2013

# Linear Regression

The regression coefficient can be found directly from the formulas, but R provides the **lm()** function which does it for us directly. The basic usage is of the form:

$$lm(model.formula, data = ..., subset = ...)$$

The **subset=** argument can be used to restrict the variable used by the modeling function.

# Linear Regression (cont.)

By default, the *lm* function will print out the estimates of the coefficients. We can get more information as needed by using extractor functions:

- **summary()** returns summary info
- **plot()** makes diagnostic plots
- **coef()** returns the coefficients
- **resid()** returns the residuals $e_i$
- **fitted()** returns fitted values $\hat{y}_i$
- **deviance()** returns RSS
- **predict()** performs prediction
- **anova()** returns sum of squares

# Estimation of Coefficients

Linear models are fit using R s model formulas. The basic format for a formula is:

$$response \sim predictor$$

where $\sim$ (tilde) is read ”is modeled by” and is used to separate the response variable from the predictor.

If the intercept term is not desired, it can be dropped by including the term -**1**:

$$response \sim predictor - 1$$

Once we store the result of the estimation in an object, we can

extract the estimates of the regression coefficients:

$$> trial <\!- lm(weight\ \ height)$$
$$> coeff\,(trial)$$

# Plot

We may add to the scatter plot the regression line by:

$$> plot(weight,\ height) \qquad \#scatterplot$$

$$> abline(trial) \qquad \#add\ the\ regression\ line$$

**abline(trial)** is used to the result of the linear regression to an existing plot. It also corresponds to:

$$> abline(a = coef(trial)[1], b = coef(trial)[2])$$

# Prediction and Residuals

Use the regression line to predict the response value for a given value of the predictor:

$$> predict(trial, data.frame(height = height[c(13, 50)]))$$

The residuals $e_i$s can be computed by subtraction or with *resid* command:

$$> weight - fitted(trial)$$
$$> resid(trial)$$

# Assessing the Linearity

The most common diagnostic tool is looking at the residuals. Even if a simple scatterplot of the data may reveal if the regression line is appropriate, a residual plot with the residuals $ei = y_i - \hat{y}_i$ against $\hat{y}_i$ can be of use.

$$> plot(resid(trial) \sim fitted(trial))$$

In case of linearity, the point are scattered around the horizontal line at zero. The residual plot is also the first of the diagnostic plots produced by:

$$> plot(trial,\ which = 1)$$

# Normality of the Residuals

The residual are used to asses whether the error terms in the model are normally distributed. An histogram and a density plot of $e_i$s can be used to investigate normality. The second diagnostic plot produced by plot is the normal Q-Q plot:

$$> plot(trial,\ which = 2)$$

Deviations from a straight line indicate non-normality.

# Common Variance

The error terms should have a common variance. A residual plot can show whether this is the case. When they are not scattered about an horizontal line, it may be that the variances follows some patterns related to the predictor. A common problem is when the variance increases for larger values of the predictors. The scale-location plot is the third of the diagnostic plot and aims to detect this situation and shows the square root of the absolute value of the standardized residuals against the fitted values $\hat{y}_i$:

$$> plot(trial,\ which = 3)$$

The graph should show points scattered along the y-axis, as we can scan across the x-axis, but the spread should not get larger or smaller.

# Influential Points

The regression line can be greatly influenced by one or few outliers. Intuitively, the difference in slopes between the regression line with all data and the regression line obtained by excluding the i-th observation should be small except for influential point. The Cook's distance measures this by looking at the difference at the fitted value $\hat{y}_i$ obtained by including or not including the i-th observation in the regression. The Cook's distance is computed as follow:

$$> cooks.distance(trial)$$

$$> plot(1:n,\ cooks.distance(trial),\ type = "h")$$

We may then read on the horizontal axis the index corresponding to the influential point.