Applied Econometrics with R

Chapter 3

Linear Regression

Linear Regression

Overview

Linear regression model

Workhorse of applied econometrics: linear regression model, typically estimated by ordinary least squares (OLS).

$$y_i = x_i^\top \beta + \varepsilon_i, \quad i = 1, \dots, n.$$

In matrix form:

$$y = X\beta + \varepsilon.$$

- y: dependent variable, $n \times 1$ vector.
- x_i: regressors (or covariates) for observation i, $k \times 1$ vector.
- $X = (x_1, \dots, x_n)^\top$: regressor (or model) matrix, $n \times k$ matrix.
- β: regression coefficients, $k \times 1$ vector.
- ε: disturbances (or error terms), $n \times 1$ vector.

Assumptions

Assumptions on the error terms depend on the context. Typical sets of assumptions are:

For cross sections:

- $\mathsf{E}(\varepsilon|X) = 0$ (exogeneity)
- $\mathsf{Var}(\varepsilon|X) = \sigma^2 I$ (conditional homoskedasticity and lack of correlation)

For time series: Exogeneity too strong, commonly replaced by

- $\mathsf{E}(\varepsilon_j|x_i) = 0$, $i \le j$ (predeterminedness).

Methods for checking these assumptions are discussed in Chapter 4: “Diagnostics and Alternative Methods of Regression”.

Notation

OLS estimator of β:

$$\hat{\beta} \quad = \quad (X^\top X)^{-1} X^\top y.$$

Fitted values:

$$\hat{y} \quad = \quad X\hat{\beta}.$$

Residuals:

$$\hat{\varepsilon} \quad = \quad y - \hat{y}.$$

Residual sum of squares (RSS):

$$\sum_{i=1}^{n} \hat{\varepsilon}_i^2 \quad = \quad \hat{\varepsilon}^\top \hat{\varepsilon}.$$

Background: Baltagi (2002) or Greene (2003).

R tools

In R, models are typically fitted by calls of type

```
fm <- lm(formula, data, ...)
```

- `lm()`: model-fitting function for linear models.
- `formula`: symbolic description of the model.
- `data`: data set containing the variables from the formula.
- `...`: further arguments, e.g., control parameters for the fitting algorithm, further model details, etc.
- `fm`: fitted-model object of class "`lm`".

Many other modeling functions in R have analogous interfaces (e.g., `glm()`, `rq()`). The fitted-model objects can typically be queried using methods to generic functions such as `summary()`, `residuals()`, or `predict()`, etc.

Linear Regression

Simple Linear Regression

Demand for economics journals

Data set from Stock & Watson (2007), originally collected by T. Bergstrom, on subscriptions to 180 economics journals at US libraries, for the year 2000.

Bergstrom (2001) argues that commercial publishers are charging excessive prices for academic journals and also suggests ways that economists can deal with this problem. See
`http://www.econ.ucsb.edu/~tedb/Journals/jpricing.html`

10 variables are provided including:

- `subs` – number of library subscriptions,
- `price` – library subscription price,
- `citations` – total number of citations,

and other information such as number of pages, founding year, characters per page, etc.

Demand for economics journals

For compactness: Preprocessing yielding smaller data frame with transformed variables.

```
R> data("Journals", package = "AER")
R> journals <- Journals[, c("subs", "price")]
R> journals$citeprice <- Journals$price/Journals$citations
R> summary(journals)
```

```
      subs            price          citeprice     
 Min.   :   2   Min.   :  20   Min.   : 0.005  
 1st Qu.:  52   1st Qu.: 134   1st Qu.: 0.464  
 Median : 122   Median : 282   Median : 1.321  
 Mean   : 197   Mean   : 418   Mean   : 2.548  
 3rd Qu.: 268   3rd Qu.: 541   3rd Qu.: 3.440  
 Max.   :1098   Max.   :2120   Max.   :24.459  
```


Demand for economics journals

Goal: Estimate the effect of the price per citation on the number of library subscriptions.

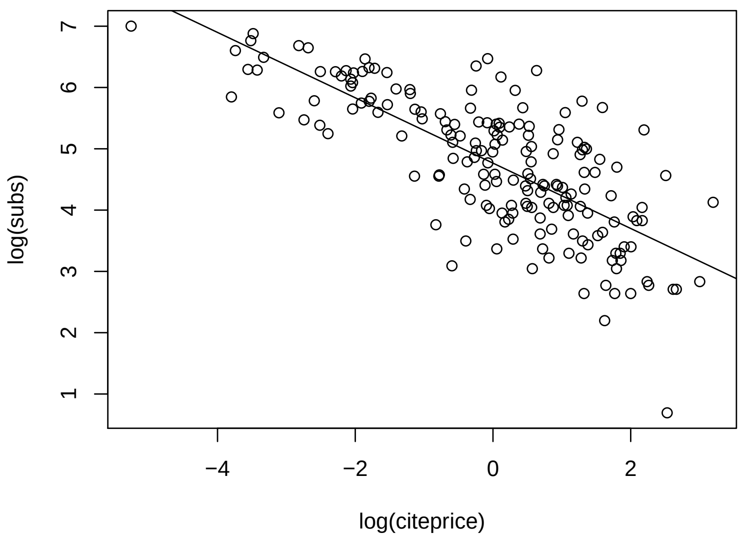
Regression equation:

$$\texttt{log(subs)}_i = \beta_1 + \beta_2 \, \texttt{log(citeprice)}_i + \varepsilon_i.$$

R formula: `log(subs) ~ log(citeprice)`
i.e., `log(subs)` explained by `log(citeprice)`. This can be used both for plotting and for model fitting:

```
R> plot(log(subs) ~ log(citeprice), data = journals)
R> jour_lm <- lm(log(subs) ~ log(citeprice), data = journals)
R> abline(jour_lm)
```


Demand for economics journals



Fitted-model objects

Inspect fitted-model object:

```
R> class(jour_lm)
```

```
[1] "lm"
```

```
R> names(jour_lm)
```

```
 [1] "coefficients"  "residuals"     "effects"      
 [4] "rank"          "fitted.values" "assign"       
 [7] "qr"            "df.residual"   "xlevels"      
[10] "call"          "terms"         "model"        
```

```
R> jour_lm$rank
```

```
[1] 2
```

More details: `str(jour_lm)`

For most tasks, do not compute on internal structure. Use methods for generic extractor functions instead.

Generic functions

`print()`	simple printed display
`summary()`	standard regression output
`coef()`	(or `coefficients()`) extract regression coefficients
`residuals()`	(or `resid()`) extract residuals
`fitted()`	(or `fitted.values()`) extract fitted values
`anova()`	comparison of nested models
`predict()`	predictions for new data
`plot()`	diagnostic plots
`confint()`	confidence intervals for the regression coefficients
`deviance()`	residual sum of squares
`vcov()`	(estimated) variance-covariance matrix
`logLik()`	log-likelihood (assuming normally distributed errors)
`AIC()`	information criteria including AIC, BIC/SBC

Summary of fitted-model objects

```
R> summary(jour_lm)

Call:
lm(formula = log(subs) ~ log(citeprice), data = journals)

Residuals:
    Min      1Q  Median      3Q     Max
-2.7248 -0.5361  0.0372  0.4662  1.8481

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)      4.7662     0.0559    85.2   <2e-16
log(citeprice)  -0.5331     0.0356   -15.0   <2e-16

Residual standard error: 0.75 on 178 degrees of freedom
Multiple R-squared:  0.557,	Adjusted R-squared:  0.555
F-statistic:  224 on 1 and 178 DF,  p-value: <2e-16
```


Summary of fitted-model objects

```
R> jour_slm <- summary(jour_lm)
R> class(jour_slm)

[1] "summary.lm"

R> names(jour_slm)

 [1] "call"          "terms"         "residuals"
 [4] "coefficients"  "aliased"       "sigma"
 [7] "df"            "r.squared"     "adj.r.squared"
[10] "fstatistic"    "cov.unscaled"

R> jour_slm$coefficients

               Estimate Std. Error t value   Pr(>|t|)
(Intercept)      4.7662    0.05591   85.25 2.954e-146
log(citeprice)  -0.5331    0.03561  -14.97  2.564e-33
```


Analysis of variance

```
R> anova(jour_lm)

Analysis of Variance Table

Response: log(subs)
                Df Sum Sq Mean Sq F value Pr(>F)
log(citeprice)   1    126   125.9     224 <2e-16
Residuals      178    100     0.6
```

ANOVA breaks the sum of squares about the mean of `log(subs)` into two parts:

- part accounted for by linear function of `log(citeprice)`,
- part attributed to residual variation.

`anova()` produces

- ANOVA table for a single "`lm`" object, and also
- comparisons of several nested "`lm`" models using *F* tests.

Point and interval estimates

Extract the estimated regression coefficients $\hat{\beta}$:

```
R> coef(jour_lm)

   (Intercept) log(citeprice)
        4.7662        -0.5331
```

Confidence intervals:

```
R> confint(jour_lm, level = 0.95)

                 2.5 %  97.5 %
(Intercept)     4.6559  4.8765
log(citeprice) -0.6033 -0.4628
```

Here based on the t distribution with 178 degrees of freedom (residual df), exact under the assumption of (conditionally) Gaussian disturbances.

Prediction

Two types of predictions:

- points on the regression line,
- new data values (two sources of errors: uncertainty in regression line, and variation of individual points about line).

Expected subscriptions for `citeprice` = 2.11 (≈ *Journal of Applied Econometrics*, fairly expensive, owned by commercial publisher):

```
R> predict(jour_lm, newdata = data.frame(citeprice = 2.11),
+    interval = "confidence")

    fit   lwr   upr
1 4.368 4.247 4.489

R> predict(jour_lm, newdata = data.frame(citeprice = 2.11),
+    interval = "prediction")

    fit   lwr   upr
1 4.368 2.884 5.853
```

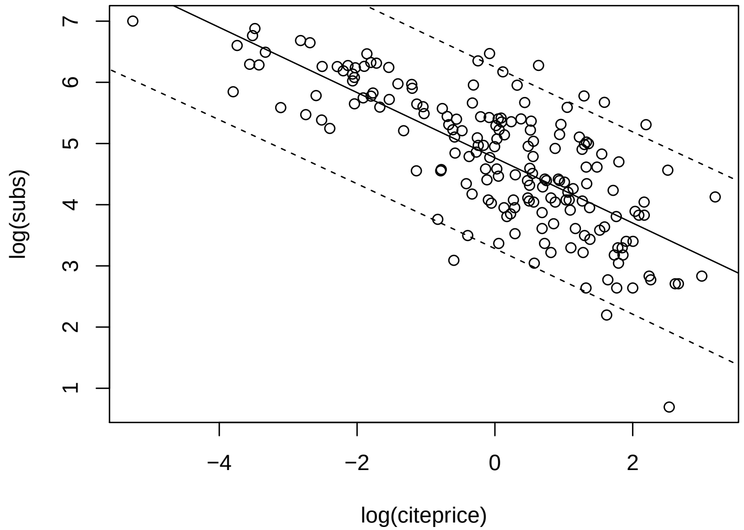

Prediction

By default: no intervals, and `newdata` the same as observed data (used for fitting), i.e., `predict(jour_lm)` computes $\hat{y}$ just as `fitted(jour_lm)`.

Visualization: data, fitted regression line, and prediction interval confidence bands.

```
R> lciteprice <- seq(from = -6, to = 4, by = 0.25)
R> jour_pred <- predict(jour_lm, interval = "prediction",
+    newdata = data.frame(citeprice = exp(lciteprice)))
R> plot(log(subs) ~ log(citeprice), data = journals)
R> lines(jour_pred[, 1] ~ lciteprice, col = 1)
R> lines(jour_pred[, 2] ~ lciteprice, col = 1, lty = 2)
R> lines(jour_pred[, 3] ~ lciteprice, col = 1, lty = 2)
```


Prediction



Diagnostic plots

The `plot()` method for class `lm()` provides six types of diagnostic plots, four of which are shown by default.

```
R> plot(jour_lm)
```

produces

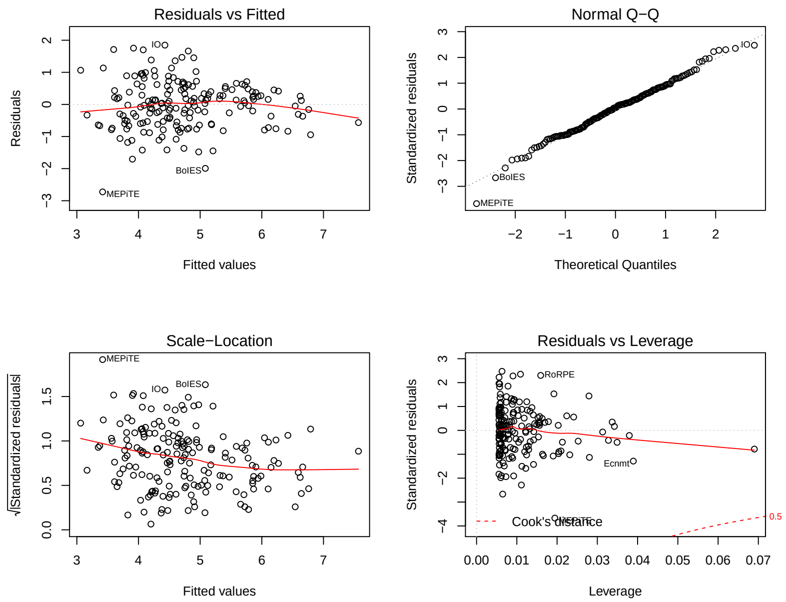
- residuals versus fitted values,
- QQ plot for normality,
- scale-location plot,
- standardized residuals versus leverages.

Plots are also accessible individually, e.g.,

```
R> plot(jour_lm, which = 2)
```

for QQ plot.

Diagnostic plots



Diagnostic plots

Interpretation: singled-out observations

- “MEPiTE” (*MOCT-MOST: Economic Policy in Transitional Economics*),
- “RoRPE” (*Review of Radical Political Economics*),
- “IO” (*International Organization*),
- “BoIES” (*Bulletin of Indonesian Economic Studies*),
- “Ecnmt” (*Econometrica*).

All these journals are not overly expensive: either heavily cited (*Econometrica*), resulting in a low price per citation, or with few citations, resulting in a rather high price per citation.

Testing a linear hypothesis

Standard `summary()` only indicates individual significance of each regressor and joint significance of all regressors (t and F statistics, respectively).

Often it is necessary to test more general hypotheses of type

$$R\beta = r,$$

where R is a $q \times k$ matrix of restrictions, and r is a $q \times 1$ vector.

In R: `linearHypothesis()` from **car** package, automatically loaded with **AER**.

Testing a linear hypothesis

Example: Test linear hypothesis $H_0 : \beta_2 = -0.5$ (price elasticity of library subscriptions equals -0.5).

```
R> linearHypothesis(jour_lm, "log(citeprice) = -0.5")
```

```
Linear hypothesis test

Hypothesis:
log(citeprice) = - 0.5

Model 1: restricted model
Model 2: log(subs) ~ log(citeprice)

  Res.Df RSS Df Sum of Sq    F Pr(>F)
1    179 100
2    178 100  1     0.484 0.86   0.35
```

Equivalently, specify `hypothesis.matrix` R and `rhs` vector r:

```
R> linearHypothesis(jour_lm, hypothesis.matrix = c(0, 1), rhs = -0.5)
```


Linear Regression

Multiple Linear Regression

Wage equation

Example: estimation of wage equation in semilogarithmic form.

Use `CPS1988` data

- March 1988 Current Population Survey (CPS) collected by the US Census Bureau,
- analyzed by Bierens and Ginther (*Empirical Economics* 2001),
- “industry-strength” example with 28,155 observations,
- cross-section data on males aged 18 to 70 with positive annual income greater than US$ 50 in 1992 who are not self-employed or working without pay,
- wages are deflated by the deflator of personal consumption expenditures for 1992.

Wage equation

```
R> data("CPS1988", package = "AER")
R> dim(CPS1988")
```

```
[1] 28155     7
```

```
R> summary(CPS1988)
```

```
      wage          education      experience    ethnicity
 Min.   :   50   Min.   : 0.0   Min.   :-4.0   cauc:25923
 1st Qu.:  309   1st Qu.:12.0   1st Qu.: 8.0   afam: 2232
 Median :  522   Median :12.0   Median :16.0
 Mean   :  604   Mean   :13.1   Mean   :18.2
 3rd Qu.:  783   3rd Qu.:15.0   3rd Qu.:27.0
 Max.   :18777   Max.   :18.0   Max.   :63.0
  smsa              region      parttime
 no : 7223   northeast:6441   no :25631
 yes:20932   midwest  :6863   yes: 2524
             south    :8760
             west     :6091
```

Wage equation

- `wage` – wage in dollars per week.
- `education` and `experience` – measured in years.
- `ethnicity` – factor with levels Caucasian (`"cauc"`) and African-American (`"afam"`).
- `smsa` – factor indicating residence in standard metropolitan statistical area (SMSA).
- `region` – factor indicating region within USA.
- `parttime` – factor indicating whether individual works part-time.

CPS does not provide actual work experience.
Standard solution: compute "potential" experience

`age - education - 6`

. . . which may become negative.

Wage equation

Model:

$$\begin{aligned}\texttt{log(wage)} &= \beta_1 + \beta_2\,\texttt{experience} + \beta_3\,\texttt{experience}^2\\ &\quad + \beta_4\,\texttt{education} + \beta_5\,\texttt{ethnicity} + \varepsilon\end{aligned}$$

In R:

```
R> cps_lm <- lm(log(wage) ~ experience + I(experience^2) +
+    education + ethnicity, data = CPS1988)
```

where log-wage and squared experience can be computed on the fly (the latter using `I()` to ensure the arithmetic meaning (rather than the formula meaning) of the `^` operator.

For the factor `ethnicity` an indicator variable (or dummy variable) is automatically created.

Wage equation

```
R> summary(cps_lm)

Call:
lm(formula = log(wage) ~ experience + I(experience^2) +
  education + ethnicity, data = CPS1988)

Residuals:
   Min     1Q Median     3Q    Max
-2.943 -0.316  0.058  0.376  4.383

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      4.321395   0.019174   225.4   <2e-16
experience       0.077473   0.000880    88.0   <2e-16
I(experience^2) -0.001316   0.000019   -69.3   <2e-16
education        0.085673   0.001272    67.3   <2e-16
ethnicityafam   -0.243364   0.012918   -18.8   <2e-16

Residual standard error: 0.584 on 28150 degrees of freedom
Multiple R-squared:  0.335,       Adjusted R-squared:  0.335
F-statistic: 3.54e+03 on 4 and 28150 DF,  p-value: <2e-16
```


Dummy variables and contrast codings

Factor `ethnicity`:

- Only a single coefficient for level `"afam"`.
- No coefficient for level `"cauc"` which is the "reference category".
- `"afam"` coefficient codes the difference in intercepts between the `"afam"` and the `"cauc"` groups.
- In statistical terminology: "treatment contrast".
- In econometric jargon: "dummy variable".

Dummy variables and contrast codings

Internally:

- R produces a dummy variable for each level.
- Resulting overspecifications are resolved by applying "contrasts", i.e., a constraint on the underlying parameter vector.
- Contrasts can be attributed to factors (or queried and changed) by `contrasts()`.
- Default for unordered factors: use all dummy variables except for reference category.
- This is typically what is required for fitting econometric regression models.

The function `I()`

Wilkinson-Rogers type formulas:

- The arithmetic operator + has a different meaning: it is employed to add regressors (main effects).
- Operators `:`, `*`, `/`, `^` also have special meanings, all related to the specification of interaction effects.
- To ensure arithmetic meaning, protect by insulation in a function, e.g., `log(x1 * x2)`.
- If no other transformation is required: `I()` can be used, it returns its argument "as is".

Comparison of models

Question: Is there a difference in the average log-wage (controlling for experience and education) between Caucasian and African-American men?

Answer: Test for the relevance of the variable `ethnicity`.

As treatment contrasts are used: significance is already indicated by *t* test in the model summary.

More generally: Test for the relevance of subsets of regressors by applying `anova()` to the corresponding nested models.

For a single coefficient, both lead to equivalent results, i.e., identical *p* values.

Comparison of models

```
R> cps_noeth <- lm(log(wage) ~ experience + I(experience^2) +
+    education, data = CPS1988)
R> anova(cps_noeth, cps_lm)

Analysis of Variance Table

Model 1: log(wage) ~ experience + I(experience^2) + education
Model 2: log(wage) ~ experience + I(experience^2) +
  education + ethnicity
  Res.Df  RSS Df Sum of Sq   F Pr(>F)
1  28151 9720
2  28150 9599  1       121 355 <2e-16
```

Thus, if several fitted models are supplied to `anova()`, the associated RSS are compared (in the order in which the models are entered) based on the usual *F* statistic

$$F = \frac{(RSS_0 - RSS_1)/q}{RSS_1/(n-k)}.$$

Comparison of models

If only a single model is passed to `anova()`: terms are added sequentially in the order specified by the formula .

```
R> anova(cps_lm)
```

```
Analysis of Variance Table

Response: log(wage)
                   Df Sum Sq Mean Sq F value Pr(>F)
experience          1    840     840    2462 <2e-16
I(experience^2)     1   2249    2249    6597 <2e-16
education           1   1620    1620    4750 <2e-16
ethnicity           1    121     121     355 <2e-16
Residuals       28150   9599       0
```

The next to last line in ANOVA table is equivalent to direct comparison of `cps_lm` and `cps_noeth`.

Comparison of models

More elegantly: Use `update()` specifying the model only relative to the original.

```
R> cps_noeth <- update(cps_lm, formula = . ~ . - ethnicity)
```

yielding the same fitted-model object as before.

The expression `. ~ . - ethnicity` specifies to take the LHS and RHS in the formula (signaled by the "`.`"), only removing `ethnicity` on the RHS.

Alternative interface: `waldtest()` from package **lmtest**, loaded automatically by **AER**.

`waldtest()` by default computes the same *F* tests, but can also perform quasi-*F* tests in situations where errors are potentially heteroskedastic. See Chapter 4.

Comparison of models

Equivalent outputs can be obtained via

```
R> waldtest(cps_lm, cps_noeth)
```

or by using the update formula directly

```
R> waldtest(cps_lm, . ~ . - ethnicity)
```

```
Wald test

Model 1: log(wage) ~ experience + I(experience^2) +
  education + ethnicity
Model 2: log(wage) ~ experience + I(experience^2) + education
  Res.Df Df   F Pr(>F)
1  28150
2  28151 -1 355 <2e-16
```


Linear Regression

Partially Linear Models

Partially linear models

Motivation: More flexible specification of influence of `experience` in wage equation (instead of the usual quadratic specification).

Idea: Semiparametric model using regression splines for (unknown) function g:

$$\texttt{log(wage)} = \beta_1 + g(\texttt{experience}) + \beta_2\,\texttt{education} + \beta_3\,\texttt{ethnicity} + \varepsilon$$

In R: available in the package **splines** (part of base R and automatically loaded with **AER**).

Many types of splines available. B splines are computationally convenient and provided by `bs()`. It can directly be used in `lm()`:

```
R> library("splines")
R> cps_plm <- lm(log(wage) ~ bs(experience, df = 5) +
+    education + ethnicity, data = CPS1988)
R> summary(cps_plm)
```


Partially linear models

```
Call:
lm(formula = log(wage) ~ bs(experience, df = 5) +
  education + ethnicity, data = CPS1988)

Residuals:
   Min     1Q Median     3Q    Max
-2.931 -0.308  0.057  0.367  3.994

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)              2.77558    0.05608    49.5   <2e-16
bs(experience, df = 5)1  1.89167    0.07581    24.9   <2e-16
bs(experience, df = 5)2  2.25947    0.04647    48.6   <2e-16
bs(experience, df = 5)3  2.82458    0.07077    39.9   <2e-16
bs(experience, df = 5)4  2.37308    0.06520    36.4   <2e-16
bs(experience, df = 5)5  1.73934    0.11969    14.5   <2e-16
education                0.08818    0.00126    70.1   <2e-16
ethnicityafam           -0.24820    0.01273   -19.5   <2e-16

Residual standard error: 0.575 on 28147 degrees of freedom
Multiple R-squared:  0.356,       Adjusted R-squared:  0.356
F-statistic: 2.22e+03 on 7 and 28147 DF,  p-value: <2e-16
```


Partially linear models

Specification of `bs()`: Either supply

- `degree` of piecewise polynomial (defaulting to 3) and `knots` by hand,
- parameter `df`, which selects the remaining ones.

The expression `bs(experience, df = 5)` internally generates piecewise cubic polynomials evaluated at the observations pertaining to `experience`: $5 - 3 = 2$ interior knots, evenly spaced (i.e., located at the 33.33% and 66.67% quantiles of `experience`).

Partially linear models

Model selection: `df` is chosen based on Schwarz criterion (BIC).

```
R> cps_bs <- lapply(3:10, function(i) lm(log(wage) ~
+    bs(experience, df = i) + education + ethnicity,
+    data = CPS1988))
R> structure(sapply(cps_bs, AIC, k = log(nrow(CPS1988))),
+    .Names = 3:10)

    3     4     5     6     7     8     9    10
49205 48836 48794 48795 48801 48797 48799 48802
```

Details:

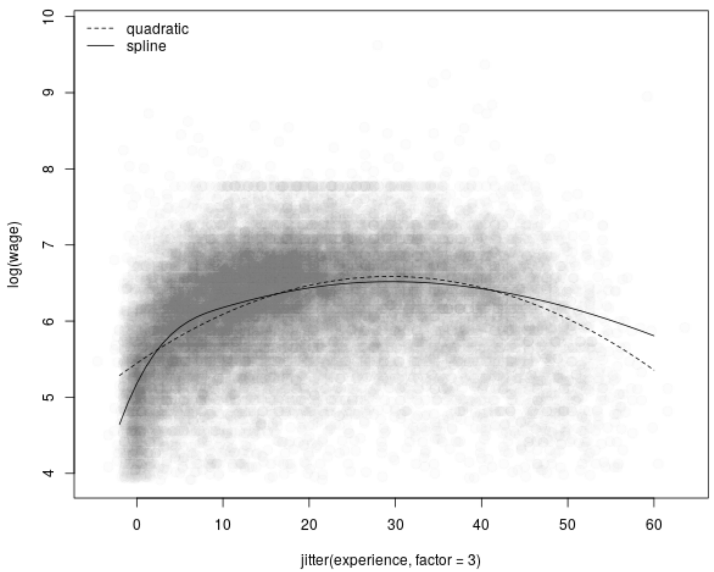
- Construct a list `cps_bs` of fitted linear models via `lapply()`.
- Apply extractor functions, e.g., `sapply(cps_bs, AIC)`.
- The call above additionally sets the penalty term to $\log(n)$ (yielding BIC instead of the default AIC), and assigns names via `structure()`.

Partially linear models

Comparison: Cubic spline and classical fit are best compared graphically, e.g., regression function for log-wage by experience (for Caucasian workers with average years of education).

```
R> cps <- data.frame(experience = -2:60, education =
+    with(CPS1988, mean(education[ethnicity == "cauc"])),
+    ethnicity = "cauc")
R> cps$yhat1 <- predict(cps_lm, newdata = cps)
R> cps$yhat2 <- predict(cps_plm, newdata = cps)
R> plot(log(wage) ~ jitter(experience, factor = 3), pch = 19,
+    cex = 1.5, col = rgb(0.5, 0.5, 0.5, alpha = 0.02),
+    data = CPS1988)
R> lines(yhat1 ~ experience, data = cps, lty = 2)
R> lines(yhat2 ~ experience, data = cps)
R> legend("topleft", c("quadratic", "spline"),
+    lty = c(2, 1), bty = "n")
```


Partially linear models



Partially linear models

Challenge: large number of observations and numerous ties in `experience`.

Solution:

- Add some amount of "jitter" to `experience`.
- Set the color to "semi-transparent" gray yielding darker shades of gray for areas with more data points and conveying a sense of density ("alpha blending").

In R:

- Set `alpha` for color (0: fully transparent, 1: opaque).
- Argument `alpha` available in various color functions, e.g., `rgb()`.
- `rgb()` implements RGB (red, green, blue) color model.
- Selecting equal RGB intensities yields a shade of gray.

Partially linear models

Alpha transparency is only available for selected plotting devices in R including

- `windows()` (typically used on Microsoft Windows),
- `quartz()` (typically used on Mac OS X),
- `pdf()` (on all platforms for `version = "1.4"` or greater).

See `?rgb` for further details.

Alternatives:

- Visualization: Employ tiny plotting character such as `pch = "."`.
- Model specification: Use penalized splines with package **mgcv** – or kernels instead of splines with package **np**.

Linear Regression

Factors, Interactions, and Weights

Factors and Interactions

Motivation: Investigate discrimination (e.g., by gender or ethnicity) in labor economics.

Illustration: Interactions of `ethnicity` with other variables in wage equation for `CPS1988`.

R formula operators:

- `:` specifies an interaction effect (i.e., in the default contrast coding, the product of a dummy variable and another variable, possibly also a dummy).
- `*` does the same but also includes the corresponding main effects.
- `/` does the same but uses a nested coding (instead of the interaction coding).
- `^` can be used to include all interactions up to a certain order.

Interactions

Formula	Description
`y ~ a + x`	Model without interaction: identical slopes with respect to `x` but different intercepts with respect to `a`.
`y ~ a * x` `y ~ a + x + a:x`	Model with interaction: the term `a:x` gives the difference in slopes compared with the reference category.
`y ~ a / x` `y ~ a + x %in% a`	Model with interaction: produces the same fitted values as the model above but using a nested coefficient coding. An explicit slope estimate is computed for each category in `a`.
`y ~ (a + b + c)^2` `y ~ a*b*c - a:b:c`	Model with all two-way interactions (excluding the three-way interaction).

Interactions

Consider an interaction between `ethnicity` and `education`:

```
R> cps_int <- lm(log(wage) ~ experience + I(experience^2) +
+    education * ethnicity, data = CPS1988)
R> coeftest(cps_int)

t test of coefficients:

                         Estimate Std. Error t value Pr(>|t|)
(Intercept)              4.313059   0.019590  220.17   <2e-16
experience               0.077520   0.000880   88.06   <2e-16
I(experience^2)         -0.001318   0.000019  -69.34   <2e-16
education                0.086312   0.001309   65.94   <2e-16
ethnicityafam           -0.123887   0.059026   -2.10    0.036
education:ethnicityafam -0.009648   0.004651   -2.07    0.038
```

Interactions

Interpretation: Coefficients correspond to

- intercept for Caucasians,
- quadratic polynomial in `experience` for all men,
- the slope for `education` for Caucasians,
- the difference in intercepts,
- the difference in slopes.

Equivalently:

```
R> cps_int <- lm(log(wage) ~ experience + I(experience^2) +
+    education + ethnicity + education:ethnicity,
+    data = CPS1988)
```

`coeftest()` (instead of `summary()`) can be used for a more compact display of the coefficient table, see Chapter 4 for further details.

Separate regressions for each level

Task: Fit separate regressions for African-Americans and Caucasians.

First solution: Compute two separate "`lm`" objects using the `subset` argument to `lm()` (e.g., `lm(formula, data, subset = ethnicity=="afam", ...)`).

More convenient: Nested coding

```
R> cps_sep <- lm(log(wage) ~ ethnicity /
+    (experience + I(experience^2) + education) - 1,
+    data = CPS1988)
```

All terms within parentheses are nested within `ethnicity`. Single intercept is replaced by two separate intercepts for the two levels of `ethnicity`.
Note that in this case the R^2 is computed differently in the `summary()`; see `?summary.lm` for details.

Separate regressions for each level

Comparison:

```
R> cps_sep_cf <- matrix(coef(cps_sep), nrow = 2)
R> rownames(cps_sep_cf) <- levels(CPS1988$ethnicity)
R> colnames(cps_sep_cf) <- names(coef(cps_lm))[1:4]
R> cps_sep_cf

     (Intercept) experience I(experience^2) education
cauc       4.310    0.07923      -0.0013597   0.08575
afam       4.159    0.06190      -0.0009415   0.08654

R> anova(cps_sep, cps_lm)

Analysis of Variance Table

Model 1: log(wage) ~ ethnicity/(experience +
  I(experience^2) + education) - 1
Model 2: log(wage) ~ experience + I(experience^2) +
  education + ethnicity
  Res.Df  RSS Df Sum of Sq    F  Pr(>F)
1  28147 9582
2  28150 9599 -3     -16.8 16.5 1.1e-10
```


Change of the reference category

In R: For unordered factors, the first level is used by default as the reference category (whose coefficient is fixed at zero).

For `CPS1988`: `"cauc"` for `ethnicity` and `"northeast"` for `region`.

Bierens and Ginther (2001) employ `"south"` as the reference category for `region`. One way of achieving this in R is to use `relevel()`.

```
R> CPS1988$region <- relevel(CPS1988$region, ref = "south")
R> cps_region <- lm(log(wage) ~ ethnicity + education +
+    experience + I(experience^2) + region, data = CPS1988)
R> coef(cps_region)

    (Intercept)   ethnicityafam       education      experience
       4.283606       -0.225679        0.084672        0.077656
I(experience^2) regionnortheast   regionmidwest      regionwest
      -0.001323        0.131920        0.043789        0.040327
```

Weighted least squares

Problem: Heteroskedasticity in many cross-section regressions. (Diagnostic tests in Chapter 4.)

Illustration: Journals data.

One remedy: Weighted least squares (WLS).

Model: Conditional heteroskedasticity with nonlinear skedastic function.

$$\mathsf{E}(\varepsilon_i^2 | x_i, z_i) = g(z_i^\top \gamma),$$

z_i is ℓ-vector of observations on exogenous or predetermined variables, and γ is ℓ-vector of parameters.

Weighted least squares

Background: For $\mathsf{E}(\varepsilon_i^2 | x_i, z_i) = \sigma^2 z_i^2$

- have regression of y_i/z_i on $1/z_i$ and x_i/z_i.
- fitting criterion changes to

$$\sum_{i=1}^n z_i^{-2} (y_i - \beta_1 - \beta_2 x_i)^2,$$

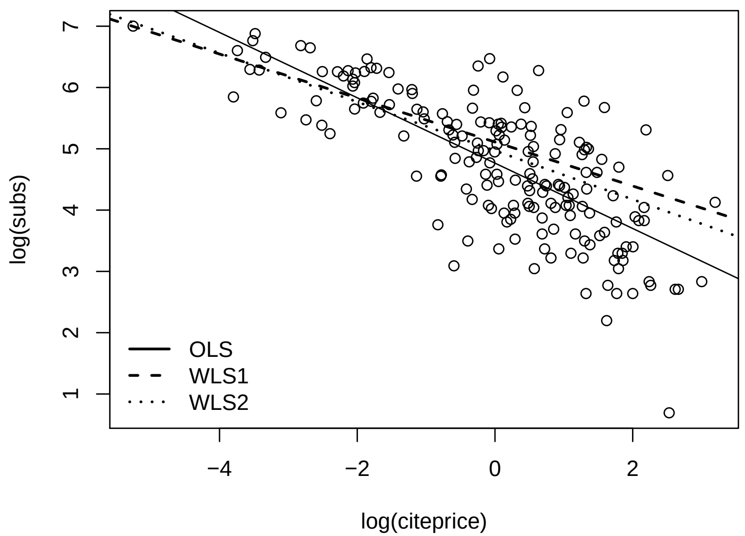
thus each term is now weighted by z_i^{-2}.

Solutions $\hat\beta_1, \hat\beta_2$ of new minimization problem are called WLS estimates, a special case of generalized least squares (GLS).

In R: Weights are entered as in fitting criterion.

```
R> jour_wls1 <- lm(log(subs) ~ log(citeprice), data = journals,
+    weights = 1/citeprice^2)
R> jour_wls2 <- lm(log(subs) ~ log(citeprice), data = journals,
+    weights = 1/citeprice)
```


Weighted least squares



Feasible generalized least squares

Problem: Skedastic function often unknown and must be estimated.

Solution: Feasible generalized least squares (FGLS). Starting point is

$$\mathsf{E}(\varepsilon_i^2 | x_i) = \sigma^2 x_i^{\gamma_2} = \exp(\gamma_1 + \gamma_2 \log x_i),$$

which can be estimated by an auxiliary regression for the logarithm of the squared OLS residuals on the logarithm of `citeprice` and a constant.

```
R> auxreg <- lm(log(residuals(jour_lm)^2) ~ log(citeprice),
+    data = journals)
R> jour_fgls1 <- lm(log(subs) ~ log(citeprice),
+    weights = 1/exp(fitted(auxreg)), data = journals)
```

Feasible generalized least squares

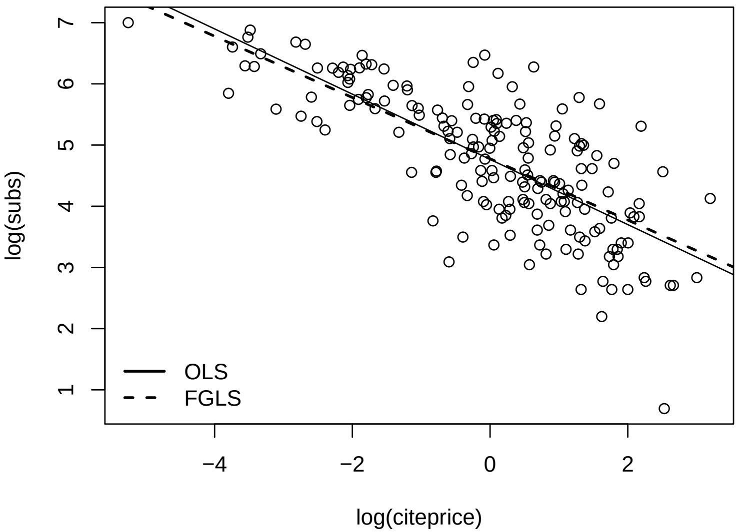
Iterate further:

```
R> gamma2i <- coef(auxreg)[2]
R> gamma2 <- 0
R> while(abs((gamma2i - gamma2)/gamma2) > 1e-7) {
+   gamma2 <- gamma2i
+   fglsi <- lm(log(subs) ~ log(citeprice), data = journals,
+     weights = 1/citeprice^gamma2)
+   gamma2i <- coef(lm(log(residuals(fglsi)^2) ~
+     log(citeprice), data = journals))[2]
+ }
R> gamma2
```

```
log(citeprice)
        0.2538
```

```
R> jour_fgls2 <- lm(log(subs) ~ log(citeprice), data = journals,
+   weights = 1/citeprice^gamma2)
```


Feasible generalized least squares



Linear Regression

Linear Regression with Time Series Data

Linear regression with time series data

In econometrics, time series regressions are often fitted by OLS:

- `lm()` can be used for fitting if data held in “`data.frame`”.
- Time series data more conveniently stored in one of R’s time series classes.
- Basic time series class is “`ts`”: a data matrix (or vector) plus time series attributes (start, end, frequency).

Problem: “`ts`” objects can be passed to `lm()`, but:

- Time series properties are by default not preserved for fitted values or residuals.
- Lags or differences cannot directly be specified in the model formula.

Linear regression with time series data

Two solutions:

- Data preprocessing (e.g., lags and differences) "by hand" before calling `lm()`. (See also Chapter 6.)
- Use `dynlm()` from package **dynlm**.

Example: Autoregressive distributed lag (ADL) model.

- First differences of a variable `y` are regressed its first difference lagged by one period and on the fourth lag of a variable `x`.
- Equation: $y_i - y_{i-1} = \beta_1 + \beta_2 \, (y_{i-1} - y_{i-2}) + \beta_3 \, x_{i-4} + \varepsilon_i$.
- Formula for `dynlm()`: `d(y) ~ L(d(y)) + L(x, 4)`.

Linear regression with time series data

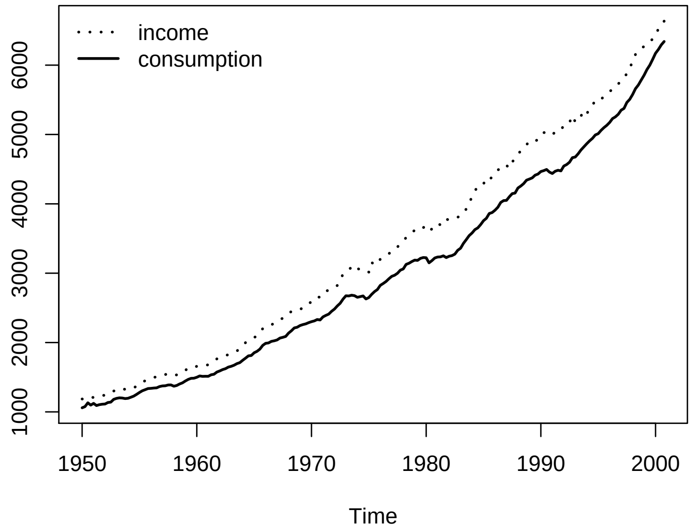
Illustration: Different specifications of consumption function taken from Greene (2003).

Data: Quarterly US macroeconomic data from 1950(1) – 2000(4) provided by `USMacroG`, a "`ts`" time series. Contains disposable income `dpi` and `consumption` (in billion USD).

Visualization: Employ corresponding `plot()` method.

```
R> data("USMacroG", package = "AER")
R> plot(USMacroG[, c("dpi", "consumption")], lty = c(3, 1),
+    lwd = 2, plot.type = "single", ylab = "")
R> legend("topleft", legend = c("income", "consumption"),
+    lwd = 2, lty = c(3, 1), bty = "n")
```


Linear regression with time series data



Linear regression with time series data

Models: Greene (2003) considers

$$\begin{aligned}\texttt{consumption}_i &= \beta_1 + \beta_2 \, \texttt{dpi}_i + \beta_3 \, \texttt{dpi}_{i-1} + \varepsilon_i \\ \texttt{consumption}_i &= \beta_1 + \beta_2 \, \texttt{dpi}_i + \beta_3 \, \texttt{consumption}_{i-1} + \varepsilon_i.\end{aligned}$$

Interpretation:

- Distributed lag model: consumption responds to changes in income only over two periods.
- Autoregressive distributed lag: effects of income changes persist.

In R:

```
R> library("dynlm")
R> cons_lm1 <- dynlm(consumption ~ dpi + L(dpi), data = USMacroG)
R> cons_lm2 <- dynlm(consumption ~ dpi + L(consumption),
+    data = USMacroG)
```

Linear regression with time series data

```
R> summary(cons_lm1)
Time series regression with "ts" data:
Start = 1950(2), End = 2000(4)

Call:
dynlm(formula = consumption ~ dpi + L(dpi),
  data = USMacroG)

Residuals:
   Min      1Q Median     3Q    Max
-190.0  -56.7    1.6   49.9  323.9

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -81.0796    14.5081   -5.59  7.4e-08
dpi           0.8912     0.2063    4.32  2.4e-05
L(dpi)        0.0309     0.2075    0.15     0.88

Residual standard error: 87.6 on 200 degrees of freedom
Multiple R-squared:  0.996,        Adjusted R-squared:  0.996
F-statistic: 2.79e+04 on 2 and 200 DF,  p-value: <2e-16
```


Linear regression with time series data

```
R> summary(cons_lm2)
Time series regression with "ts" data:
Start = 1950(2), End = 2000(4)

Call:
dynlm(formula = consumption ~ dpi + L(consumption),
  data = USMacroG)

Residuals:
    Min      1Q  Median      3Q     Max
-101.30   -9.67    1.14   12.69   45.32

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     0.53522    3.84517    0.14     0.89
dpi            -0.00406    0.01663   -0.24     0.81
L(consumption)  1.01311    0.01816   55.79   <2e-16

Residual standard error: 21.5 on 200 degrees of freedom
Multiple R-squared:     1,        Adjusted R-squared:     1
F-statistic: 4.63e+05 on 2 and 200 DF,  p-value: <2e-16
```


Linear regression with time series data

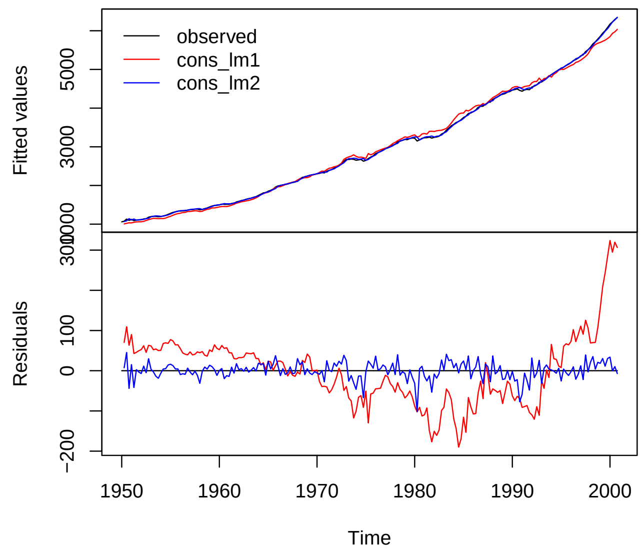
Model comparison: In terms of RSS

```
R> deviance(cons_lm1)
[1] 1534001
R> deviance(cons_lm2)
[1] 92644
```

Graphically:

```
R> plot(merge(as.zoo(USMacroG[,"consumption"]), fitted(cons_lm1),
+   fitted(cons_lm2), 0, residuals(cons_lm1),
+   residuals(cons_lm2)), screens = rep(1:2, c(3, 3)),
+   col = rep(c(1, 2, 4), 2), xlab = "Time",
+   ylab = c("Fitted values", "Residuals"), main = "")
R> legend(0.05, 0.95, c("observed", "cons_lm1", "cons_lm2"),
+   col = c(1, 2, 4), lty = 1, bty = "n")
```


Linear regression with time series data



Linear regression with time series data

Details:

- `merge()` original series with fitted values from both models, a zero line and residuals of both models.
- merged series is plotted on two screens with different colors and some more annotation.
- Before merging, original "`ts`" series is coerced to class "`zoo`" (from package **zoo**) via `as.zoo()`.
- "`zoo`" generalizes "`ts`" with slightly more flexible `plot()` method.

More details on "`ts`" and "`zoo`" classes in Chapter 6.

Encompassing test

Task: Discriminate between competing models of consumption.

Problem: Models are not nested.

Solutions:

- `encomptest()` (encompassing test).
- `jtest()` (*J* test).
- `coxtest()` (Cox test).

Illustration: Use encompassing test.

Encompassing test

Idea:

- Transform nonnested model comparison into nested model comparison.
- Fit the encompassing model comprising all regressors from both competing models.
- Compare each of the two nonnested models with the encompassing model.
- If one model is not significantly worse than the encompassing model while the other is, this test would favor the former model over the latter.

Encompassing test

By hand: Fit encompassing model

```
R> cons_lmE <- dynlm(consumption ~ dpi + L(dpi) +
+    L(consumption), data = USMacroG)
```

and compute `anova()`.

```
R> anova(cons_lm1, cons_lmE, cons_lm2)
```

```
Analysis of Variance Table

Model 1: consumption ~ dpi + L(dpi)
Model 2: consumption ~ dpi + L(dpi) + L(consumption)
Model 3: consumption ~ dpi + L(consumption)
  Res.Df     RSS Df Sum of Sq      F  Pr(>F)
1    200 1534001
2    199   73550  1   1460451 3951.4 < 2e-16
3    200   92644 -1    -19094   51.7 1.3e-11
```

Encompassing test

More conveniently: `encomptest()` from **lmtest**.

```
R> encomptest(cons_lm1, cons_lm2)
```

```
Encompassing test

Model 1: consumption ~ dpi + L(dpi)
Model 2: consumption ~ dpi + L(consumption)
Model E: consumption ~ dpi + L(dpi) + L(consumption)
          Res.Df Df      F  Pr(>F)
M1 vs. ME    199 -1 3951.4 < 2e-16
M2 vs. ME    199 -1   51.7 1.3e-11
```

Interpretation: Both models perform significantly worse compared with the encompassing model, although F statistic is much smaller for `cons_lm2`.

Linear Regression

Linear Regression with Panel Data

Static linear models

Example: Data from Grunfeld (1958).

- 20 annual observations (1935–1954).
- 11 large US firms.
- 3 variables: real gross investment (`invest`), real value of the firm (`value`), and real value of the capital stock (`capital`).
- Popular textbook example.
- Various published versions (some including errors, see `?Grunfeld`).

Data structure:

- Two-dimensional index.
- Cross-sectional objects are called “individuals”.
- Time identifier is called “time”.

Static linear models

Data handling: Select subset of three firms for illustration and declare individuals (`"firm"`) and time identifier (`"year"`).

```
R> data("Grunfeld", package = "AER")
R> library("plm")
R> gr <- subset(Grunfeld, firm %in% c("General Electric",
+    "General Motors", "IBM"))
R> pgr <- plm.data(gr, index = c("firm", "year"))
```

Alternatively: Instead of computing `pgr` in advance, specify `index = c("firm", "year")` in each `plm()` call.

For later use: Fit plain OLS on pooled data.

```
R> gr_pool <- plm(invest ~ value + capital, data = pgr,
+    model = "pooling")
```

Static linear models

Basic model:

$$\texttt{invest}_{it} = \beta_1 \texttt{value}_{it} + \beta_2 \texttt{capital}_{it} + \alpha_i + \nu_{it},$$

i.e., one-way panel regression with indexes $i = 1, \dots, n$, $t = 1, \dots, T$ and individual-specific effects α_i.

Fixed effects: Run OLS on within-transformed model.

```
R> gr_fe <- plm(invest ~ value + capital, data = pgr,
+    model = "within")
```

Remarks:

- two-way model upon setting `effect = "twoways"`,
- fixed effects via `fixef()` method and associated `summary()` method.

Static linear models

```
R> summary(gr_fe)
Oneway (individual) effect Within Model

Call:
plm(formula = invest ~ value + capital, data = pgr,
  model = "within")

Balanced Panel: n=3, T=20, N=60

Residuals :
   Min. 1st Qu.  Median 3rd Qu.    Max.
-167.00  -26.10    2.09   26.80  202.00

Coefficients :
        Estimate Std. Error t-value Pr(>|t|)
value     0.1049     0.0163    6.42  3.3e-08
capital   0.3453     0.0244   14.16  < 2e-16

Total Sum of Squares:    1890000
Residual Sum of Squares: 244000
```


Static linear models

```
R-Squared:      0.871
Adj. R-Squared: 0.861
F-statistic: 185.407 on 2 and 55 DF, p-value: <2e-16
```

Question: Are the fixed effects really needed?

Answer: Compare fixed effects and pooled OLS fits via `pFtest()`.

```
R> pFtest(gr_fe, gr_pool)

        F test for individual effects

data:  invest ~ value + capital
F = 57, df1 = 2, df2 = 55, p-value = 4e-14
alternative hypothesis: significant effects
```

This indicates substantial inter-firm variation.

Static linear models

Random effects:

- Specify `model = "random"` in `plm()` call.
- Select method for estimating the variance components.
- Recall: Random-effects estimator is essentially FGLS estimator, utilizing OLS after “quasi-demeaning” all variables.
- Precise form of quasi-demeaning depends on `random.method` selected.
- Four methods available: Swamy-Arora (default), Amemiya, Wallace-Hussain, and Nerlove.

In plm: Using Wallace-Hussain for Grunfeld data.

```
R> gr_re <- plm(invest ~ value + capital, data = pgr,
+    model = "random", random.method = "walhus")
```

Static linear models

```
R> summary(gr_re)
```

```
Oneway (individual) effect Random Effect Model
   (Wallace-Hussain's transformation)

Call:
plm(formula = invest ~ value + capital, data = pgr,
  model = "random", random.method = "walhus")

Balanced Panel: n=3, T=20, N=60

Effects:
                 var std.dev share
idiosyncratic 4389.3    66.3  0.35
individual    8079.7    89.9  0.65
theta:  0.837

Residuals :
   Min. 1st Qu.  Median 3rd Qu.    Max.
-187.00  -32.90    6.96   31.40  210.00
```


Static linear models

```
Coefficients :
             Estimate Std. Error t-value Pr(>|t|)
(Intercept) -109.9766    61.7014   -1.78     0.08
value          0.1043     0.0150    6.95  3.8e-09
capital        0.3448     0.0245   14.06  < 2e-16

Total Sum of Squares:    1990000
Residual Sum of Squares: 258000
R-Squared:      0.87
Adj. R-Squared: 0.866
F-statistic: 191.545 on 2 and 57 DF, p-value: <2e-16
```

Comparison of regression coefficients shows that fixed- and random-effects methods yield rather similar results for these data.

Static linear models

Question: Are the random effects really needed?

Answer: Use Lagrange multiplier test. Several versions available in `plmtest()`.

```
R> plmtest(gr_pool)
```

```
        Lagrange Multiplier Test - (Honda) for balanced panels

data:  invest ~ value + capital
normal = 15, p-value <2e-16
alternative hypothesis: significant effects
```

Test also suggests that some form of parameter heterogeneity must be taken into account.

Static linear models

Random-effects methods more efficient than fixed-effects estimator under more restrictive assumptions, namely exogeneity of the individual effects.

Use Hausman test to test for endogeneity:

```
R> phtest(gr_re, gr_fe)
```

```
        Hausman Test

data:  invest ~ value + capital
chisq = 0.04, df = 2, p-value = 1
alternative hypothesis: one model is inconsistent
```

In line with estimates presented above, endogeneity does not appear to be a problem here.

Dynamic linear models

Dynamic panel data model:

$$y_{it} = \sum_{j=1}^{p} \varrho_j y_{i,t-j} + x_{it}^\top \beta + u_{it}, \quad u_{it} = \alpha_i + \beta_t + \nu_{it},$$

Estimator: Generalized method of moments (GMM) estimator suggested by Arellano and Bond (1991), utilizing lagged endogenous regressors after a first-differences transformation.

Illustration: Determinants of employment in UK (`EmplUK`).

- Unbalanced panel: 7–9 annual observations (1976–1984) for 140 UK firms.
- 4 variables: employment (`emp`), average annual wage per employee (`wage`), book value of gross fixed assets (`capital`), index of value-added output at constant factor cost (`output`).
- Original example from Arellano and Bond (1991).

Dynamic linear models

Data and basic static formula:

```
R> data("EmplUK", package = "plm")
R> form <- log(emp) ~ log(wage) + log(capital) + log(output)
```

Arellano-Bond estimator is provided by `pgmm()`. Dynamic formula derived from static formula via list of lags.

```
R> empl_ab <- pgmm(dynformula(form, list(2, 1, 0, 1)),
+    data = EmplUK, index = c("firm", "year"),
+    effect = "twoways", model = "twosteps",
+    gmm.inst = ~ log(emp), lag.gmm = list(c(2, 99)))
```

Details: Dynamic model with

- $p = 2$ lagged endogenous terms,
- log(`wage`) and log(`output`) occur up to lag 1,
- log(`capital`) contemporaneous term only,
- time- and firm-specific effects,
- instruments are lagged terms of the dependent variable (all lags beyond lag 1 are to be used).

Dynamic linear models

```
R> summary(empl_ab)

Twoways effects Two steps model

Call:
pgmm(formula = dynformula(form, list(2, 1, 0, 1)),
  data = EmplUK, effect = "twoways", model = "twosteps",
  index = c("firm", "year"), gmm.inst = ~log(emp),
  lag.gmm = list(c(2, 99)))

Unbalanced Panel: n=140, T=7-9, N=1031

Number of Observations Used:  611

Residuals
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.6190 -0.0256  0.0000 -0.0001  0.0332  0.6410

Coefficients
```


Dynamic linear models

```
                         Estimate Std. Error z-value Pr(>|z|)
lag(log(emp), c(1, 2))1   0.4742     0.1854    2.56  0.01054
lag(log(emp), c(1, 2))2  -0.0530     0.0517   -1.02  0.30605
log(wage)                -0.5132     0.1456   -3.53  0.00042
lag(log(wage), 1)         0.2246     0.1419    1.58  0.11353
log(capital)              0.2927     0.0626    4.67  3.0e-06
log(output)               0.6098     0.1563    3.90  9.5e-05
lag(log(output), 1)      -0.4464     0.2173   -2.05  0.03996

Sargan Test: chisq(25) = 30.11 (p.value=0.22)
Autocorrelation test (1): normal = -1.538 (p.value=0.124)
Autocorrelation test (2): normal = -0.2797 (p.value=0.78)
Wald test for coefficients: chisq(7) = 142 (p.value=<2e-16)
Wald test for time dummies: chisq(6) = 16.97 (p.value=0.00939)
```

Dynamic linear models

Interpretation: Autoregressive dynamics important for these data.

Diagnostics: Tests at the bottom of summary indicate that model could be improved. Arellano and Bond (1991) address this by additionally treating wages and capital as endogenous.

Note: Due to constructing lags and taking first differences, three cross sections are lost. Hence, estimation period is 1979–1984 and only 611 observations effectively available for estimation.

Linear Regression

Systems of Linear Equations

Systems of linear equations

Systems of regression equations have been a hallmark of econometrics for several decades.

Examples: Seemingly unrelated regressions (SUR) and various macroeconomic simultaneous equation models.

In R: Package **systemfit** provides various multiple-equation models.

Illustration: SUR model for Grunfeld data. Unlike panel data models considered above, SUR model allows for individual-specific slopes (in addition to individual-specific intercepts).

Terminology: “Individuals” now referred to as “equations”.

Assumption: Contemporaneous correlation across equations. Thus joint estimation of all parameters more efficient than OLS on each equation.

Systems of linear equations

SUR model in **systemfit**:

- Fitting function is `systemfit()`.
- Data should be supplied in a “`plm.data`” object.

Use only two firms (to save space):

```
R> gr2 <- subset(Grunfeld, firm %in% c("Chrysler", "IBM"))
R> pgr2 <- plm.data(gr2, c("firm", "year"))
```

Fit model:

```
R> library("systemfit")
R> gr_sur <- systemfit(invest ~ value + capital,
+    method = "SUR", data = pgr2)
```

Systems of linear equations

```
R> summary(gr_sur, residCov = FALSE, equations = FALSE)
```

```
systemfit results
method: SUR

        N DF  SSR detRCov OLS-R2 McElroy-R2
system 40 34 4114   11022  0.929      0.927

          N DF  SSR   MSE  RMSE    R2 Adj R2
Chrysler 20 17 3002 176.6 13.29 0.913  0.903
IBM      20 17 1112  65.4  8.09 0.952  0.946


Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
Chrysler_(Intercept)  -5.7031    13.2774   -0.43  0.67293
Chrysler_value         0.0780     0.0196    3.98  0.00096
Chrysler_capital       0.3115     0.0287   10.85  4.6e-09
IBM_(Intercept)       -8.0908     4.5216   -1.79  0.09139
IBM_value              0.1272     0.0306    4.16  0.00066
IBM_capital            0.0966     0.0983    0.98  0.33951
```


Systems of linear equations

Details:

- `summary()` provides standard regression results for each equation in compact layout plus some measures of overall fit.
- More detailed output (between-equation correlations, etc.) available, but was suppressed here.
- Output indicates again that there is substantial variation among firms.

Further features: systemfit can estimate linear simultaneous-equations models by several methods (two-stage least squares, three-stage least squares, and variants thereof), as well as certain nonlinear specifications.