Econometrics with Python

Manish Agarwal

January 4, 2021

These notes are based on the classic book by Wooldridge. We will focus on numerical examples and computations here.

Contents

1 The Nature of Econometrics and Economic Data

There are two purposes - inference and prediction. The structure of economic data is of following four kinds

- Cross-Sectional Data (sec. 2-9): we ignore any minor timing difference and are assumed to be obtained by random sampling. Ordering does not matter in this case. In case of dependence of data econometric methods do work but need some refinement.
- Time-Series Data (sec. 10-13): Time is important and so is the order. The observations are not independent across time.
- Panel or Longitudinal Data (sec. 14-18): Pooled cross sections which has both cross-sectional and time series features. Random sampling across time and case with correlation over time because of same entity followed over time.

Finally, we also intend to infer the **causal effect** variables have on each other. Association does not imply causation. The notion of **ceteris paribus** plays an important role in causal analysis. In most collected experimental data ceteris paribus is violated and accounting for unobservable factors is problematic.

2 The Simple Regression Model

We model the data using simple linear regression model as

$$y = \beta_0 + \beta_1 x + u.$$

As long as the intercept is included in the equation we can make the assumption $E(u) = 0$. We also assume that u is mean independent of x, i.e. $E(u|x) = E(u)$. Combining it with the previous assumption gives us the zero conditional mean assumption of $E(u|x) = 0$. The OLS solution is

$$\hat{\beta}_1 = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2},$$

and

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}.$$

The residual for ith observation is $\hat{u}_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)$. The above is derived by minimizing the sum of squared residual $\sum \hat{u}_i^2$.

Example 2.1. For the population of chief executive officers, let y be annual salary in thousand of dollars and let x be the average return on equity for the CEO's firm for the previous three years. We postulate a simple model

$$salary = \alpha + \beta roe + u$$

```
import statsmodels.formula.api as smf
import wooldridge as woo

df = woo.dataWoo('ceosal1')
model = smf.ols(formula='salary~roe', data=df)
results = model.fit()
results.summary()
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 salary   R-squared:                       0.013
Model:                            OLS   Adj. R-squared:                  0.008
Method:                 Least Squares   F-statistic:                     2.767
Date:                Wed, 23 Sep 2020   Prob (F-statistic):             0.0978
Time:                        09:40:22   Log-Likelihood:                -1804.5
No. Observations:                 209   AIC:                             3613.
Df Residuals:                     207   BIC:                             3620.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    963.1913    213.240      4.517      0.000     542.790    1383.592
roe           18.5012     11.123      1.663      0.098      -3.428      40.431
==============================================================================
Omnibus:                      311.096   Durbin-Watson:                   2.105
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            31120.902
Skew:                           6.915   Prob(JB):                         0.00
Kurtosis:                      61.158   Cond. No.                         43.3
==============================================================================
```

The fitting gives

$$\widehat{salary} = \underset{(213.240)}{963.191} + \underset{(11.123)}{18.501} roe.$$

If the roe of the company is 0, the estimated salary is \$963,191 and on a 1% in crease in roe the salary increases by \$18,501. □

Example 2.2. For the population of people in the workforce in 1976, let $y = wage$, where $wage$ is measured in dollars per hour. Let $x = educ$ denote years of schooling.

```
df = woo.dataWoo('wage1')

model = smf.ols(formula='wage~educ', data=df)
results = model.fit()
results.summary()
```

```
==============================================================================
Dep. Variable:                   wage   R-squared:                       0.165
Model:                            OLS   Adj. R-squared:                  0.163
Method:                 Least Squares   F-statistic:                     103.4
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.9049      0.685     -1.321      0.187      -2.250       0.441
educ           0.5414      0.053     10.167      0.000       0.437       0.646
==============================================================================
```

We get a fit of

$$\widehat{wage} = \underset{(0.685)}{-0.9049} + \underset{(0.053)}{0.5414} educ.$$

The intercept of -0.90 literally means the person with no education has -90c wage, which is silly. It is not surprising that regression line does poorly at beyond the extent of observed data. Every extra year of education adds 54c to the hourly estimated wages. □

Example 2.3. We look at the election outcomes and campaign expenditures for 173 two-party races for the US house of Representatives in 1988. There are two candidates in each race, A and B. Let $voteA$ be the percentage of the vote received by Candidate A and $shareA$ be the percentage of total campaign expenditures accounted for by Candidate A.

```
df = woo.dataWoo('vote1')

model = smf.ols(formula='voteA~shareA', data=df)
results = model.fit()
results.summary()
```

```
==============================================================================
Dep. Variable:                  voteA   R-squared:                       0.856
Model:                            OLS   Adj. R-squared:                  0.855
Method:                 Least Squares   F-statistic:                     1018.
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     26.8122      0.887     30.221      0.000      25.061      28.564
shareA         0.4638      0.015     31.901      0.000       0.435       0.493
==============================================================================
```

The estimated equation is

$$\widehat{voteA} = \underset{(0.887)}{26.81} + \underset{(0.015)}{0.464} shareA$$

This means that if Candidate A's share of spending increases by one percentage point, Candidate A receives almost one-half a percentage point (0.464) more of the total vote. The causal effect is unclear. □

2.1 Properties of OLS regression

1. $\sum_i \hat{u}_i = 0$.
2. $\sum_i x_i \hat{u}_i = 0$.
3. $(\bar{x}, \bar{y})$ is always on the regression line.
4. $\hat{y}_i$ and $\hat{u}_i$ are uncorrelated, and $\bar{\hat{y}} = \bar{y}$, with $y_i = \hat{y}_i + \hat{u}_i$.
5. OLS decomposes y_i into two orthogonal parts $\hat{y}_i$ and $\hat{u}_i$.
6. The decomposition of variance can be written as

$$TSS = ESS + RSS$$
$$\sum (y_i - \bar{y})^2 = \sum (\hat{y}_i - \bar{y})^2 + \sum \hat{u}_i^2.$$

The goodness of fit is generally measured using R^2, the fraction of the sample variation in y that is explained by x.

$$R^2 = \frac{ESS}{TSS} = 1 - \frac{RSS}{TSS}.$$

The R^2 for the CEO salary against Return on Equity was only 1.3%, so a lot of left unexplained. But low R^2 does not necessarily mean that the OLS is not good. Sometimes, the explanatory variables explains a substantial part of the sample variation, e.g. in the Voting outcomes and Campaign Expenditure data the R^2 was 85.6%.

Taking log of y gives a constant percentage effect model, or equivalently and increasing return to x. For the Wage equation we take the log of the wages to obtain

$$\widehat{\log(wage)} = \underset{(0.097)}{0.5838} + \underset{(0.008)}{0.0827} educ.$$

```
df = woo.dataWoo('wage1')

model = smf.ols(formula='np.log(wage)~educ', data=df)
results = model.fit()
results.summary()
```

```
==============================================================================
Dep. Variable:           np.log(wage)   R-squared:                       0.186
Model:                            OLS   Adj. R-squared:                  0.184
Method:                 Least Squares   F-statistic:                     119.6
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.5838      0.097      5.998      0.000       0.393       0.775
educ           0.0827      0.008     10.935      0.000       0.068       0.098
==============================================================================
```

The R^2 has increased from 16.5% to 18.6%. The equation means that the wages have 8.3% increase for every additional year of education. This relationship can be expressed as $\%\Delta wage \approx 100\beta_1 \Delta educ$. The Diploma effect can be investigated using dummy variables as we will see later.

Another important use of natural log is in obtaining a constant elasticity model. A constant elasticity model relating CEO slary to firm sales is

$$log(salary) = \beta_0 + \beta_1 log(sales) + u$$

```
df = woo.dataWoo('ceosal1')

model = smf.ols(formula='np.log(salary)~np.log(sales)', data=df)
results = model.fit()
results.summary()
```

```
==============================================================================
Dep. Variable:         np.log(salary)   R-squared:                       0.211
Model:                            OLS   Adj. R-squared:                  0.207
Method:                 Least Squares   F-statistic:                     55.30
==============================================================================
                    coef    std err          t      P>|t|      [0.025    0.975]
------------------------------------------------------------------------------
Intercept          4.8220      0.288     16.723      0.000       4.254    5.390
np.log(sales)      0.2567      0.035      7.436      0.000       0.189    0.325
==============================================================================
```

The OLS fit gives,

$$\widehat{log(salary)} = \underset{(0.288)}{4.8220} + \underset{(0.035)}{0.2567} educ$$

with R^2 of 21.1%, where $\beta_1 = 0.2567$ which is the elasticity of *salary* with respect to *sales*. It implies that a 1% increase in the firm sales increases CEO salary by about 0.2567%. When in log form a change in unit does not change the coefficient.

2.2 Gauss-Markov assumptions for Simple linear regression

- SLR.1: Linear in parameters.
- SLR.2: Random Sampling.
- SLR.3: Sample variation in the explanatory variable, i.e. $\sigma_x \neq 0$.
- SLR.4: Zero conditional mean of error given the explanatory variable, i.e. $E(u|x) = 0$.

Theorem 2.1 (Unbiasedness of OLS)**.** $E(\hat{\beta}_0) = \beta_0$ *and* $E(\hat{\beta}_1) = \beta_1$.

This can be easily seen via the expression

$$\hat{\beta}_1 = \beta_1 + \frac{1}{SST_x}\sum_{i=1}^{n}(x_i - \bar{x})u_i.$$

Unbiasedness is a feature of the sampling distribution of $\hat{\beta}_1$ and $\hat{\beta}_0$. Assumption 4 is the most critical to check. Spurious correlations can occur if x is correlated with u, the primary reason being omitted variables.

- SLR.5 The error u is homoskedastic, i.e. $Var(u|x) = \sigma^2$.

Variance of OLS estimators can be calculated based on the first 4 assumptions only, but with the fifth assumption the estimator is bestowed with some efficiency properties. The assumption of u begin independent of x will result in unbiasedness and constant variance, but it is sometimes too strong. In fact, σ^2 is also the unconditional variance of u. It is often useful to write SLR.4 and SLR.5 in terms of conditional mean and variance of y as $E(Y|x) = \beta_0 + \beta_1 x$, and $Var(y|x) = \sigma^2$. Heteroskedasticity occurs when $Var(u|x)$ depends on x.

Theorem 2.2 (Sampling variances of the OLS estimators)**.** *Under the assumptions SLR.1 to SLR.5 we have*

$$Var(\hat{\beta}_1|x) = \frac{\sigma^2}{\sum(x_i - \bar{x})^2}$$

and

$$Var(\hat{\beta}_0|x) = \frac{\sigma^2}{\sum(x_i - \bar{x})^2}\frac{1}{n}\sum x_i^2.$$

These expressions are invalid in presence of heteroscedasticity. This shows that if we are interested in β_1 we should choose x_i to be as spread out as possible.

Generally σ^2 is unknown and is to be estimated as well. Differentiating error u_i with residual $\hat{u}_i$ is crucial to do that. We can write $\hat{u}_i = u_i - (\hat{\beta}_0 - \beta_0) - (\hat{\beta}_1 - \beta_1)x_i$. There are n degrees of freedom in errors while only $n-2$ degrees of freedom in OLS residuals because of the two conditions $\sum \hat{u}_i = 0$, and $\sum x_i\hat{u}_i = 0$. Hence, the unbiased estimator of σ^2 is

$$s^2 = \hat{\sigma}^2 = \frac{1}{n-2}\sum \hat{u}_i^2.$$

Theorem 2.3 (unbiased estimation of σ^2)**.** $E[\hat{\sigma}^2] = \sigma^2$.

The estimate of standard errors of the regression $\hat{\sigma} = \sqrt{\hat{\sigma}^2}$ is not unbiased by consistent. THis is primarily used to calculate the standard error of the estimates of the OLS coefficients.

$$se(\hat{\beta}_1) = \frac{\hat{\sigma}}{\sqrt{SST_x}} = \frac{\hat{\sigma}}{\sqrt{\sum(x_i - \bar{x})^2}}.$$

Generally regression without intercept gives biased estimate of β_1, unless $\bar{x} = 0$. For a regression without intercept, $R^2 = 1 - \frac{\sum \hat{u}_i^2}{\sum(y_i - \bar{y})^2}$ can be negative, suggesting $\bar{y}$ is a better predictor of y than x_i.

3 Multiple Regression Estimation

The key assumption, SLR.4 - that all other factors affecting y are uncorrelated with x - is often unrealistic. Multiple regression analysis is more amenable to account for simultaneous effects. Taking the wage question again

$$wage = \beta_0 + \beta_1 educ + \beta_2 exper + u,$$

where are are still interested in the effect of educ on wage. Here we will be able to measure the effect of education on wage, holding experience fixed (the ceteris paribus effect). In a simple regression analysis - which put *exper* in the error term - we would have to assume that experience is uncorrelated with education, a tenuous assumption. Correlation of x with u would create biased estimates. The required assumption $E(u|educ, exper) = 0$ means that other factors affecting *wage* are, **on average**, unrelated to *educ* and *exper*. A general model with k variables is given by

$$y = \beta_0 + \sum_i \beta_i x_i + u.$$

No matter how many explanatory variables we include in our model, there will always be factors we cannot include, and these are collectively contained in u, until the key condition holds $E(u|x_1, \ldots, x_k) = 0$. The estimated regression line is written as $\hat{y} = \hat{\beta}_0 + \sum_i \hat{\beta}_i x_i$, which in its **partial effect or ceteris paribus** form is $\Delta\hat{y} = \sum_i \hat{\beta}_i \Delta x_i$.

Example 3.1. For 141 students we relate the college grade point average to the high school GPA and achievement test score.

```
import statsmodels.formula.api as smf
import wooldridge as woo

df = woo.dataWoo('gpa1')
model = smf.ols(formula='colGPA~hsGPA+ACT', data=df)
results = model.fit()
results.summary()
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 colGPA   R-squared:                       0.176
Model:                            OLS   Adj. R-squared:                  0.164
Method:                 Least Squares   F-statistic:                     14.78
Date:                Sun, 04 Oct 2020   Prob (F-statistic):           1.53e-06
Time:                        17:57:57   Log-Likelihood:                -46.573
No. Observations:                 141   AIC:                             99.15
Df Residuals:                     138   BIC:                             108.0
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      1.2863      0.341      3.774      0.000       0.612       1.960
```

```
hsGPA          0.4535      0.096      4.733      0.000       0.264       0.643
ACT            0.0094      0.011      0.875      0.383      -0.012       0.031
==============================================================================
Omnibus:                        3.056   Durbin-Watson:                   1.885
Prob(Omnibus):                  0.217   Jarque-Bera (JB):                2.469
Skew:                           0.199   Prob(JB):                        0.291
Kurtosis:                       2.488   Cond. No.                         298.
==============================================================================
```

We get the regression equation as

$$\widehat{colGPA} = \underset{(0.341)}{1.2863} + \underset{(0.096)}{0.4535} hsGPA + \underset{(0.011)}{0.0094} ACT.$$

□

Like the simple regression multiple regression have similar properties:

- The sample average of residual is 0 and $\bar{y} = \bar{\hat{y}}$.
- The sample covariance between each independent variable and residual is 0. Consequently, the sample covariance between the OLS fitted values and the OLS residuals is zero.
- The point $(\bar{x_1}, \ldots, \bar{x}_k, \bar{y})$ is always on the OLS regression line.

Multiple regression has a partialling out interpretation. We can write $\hat{\beta}_1 = \frac{\sum \hat{r}_{i1} y_i}{\sum \hat{r}_{i1}^2}$, where $\hat{r}_{i1}$ is the residual from regressing x_1 on $x_1, \ldots, x_k$. Thus $\hat{\beta}_1$ measures the effect of x_1 on y after $x_2, \ldots, x_k$ have been partialled or netted out.

If we do the simple regression $\tilde{y} = \tilde{\beta}_0 + \tilde{\beta}_1 x_1$ and the multiple regression as $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2$ on the same data, we can relate the coefficients as $\tilde{\beta}_1 = \hat{\beta}_1 + \hat{\beta}_1 \tilde{\delta}_1$, where $\tilde{\delta}_1$ is the slope coefficient from the simple regression of x_2 on x_1. Further, $\tilde{\beta}_1 = \hat{\beta}_1$ if either $\hat{\beta}_2 = 0$ i.e. the partial effect of x_2 on $\hat{y}$ is zero, or $\tilde{\delta}_1 = 0$ i.e. x_1 and x_2 are uncorrelated.

Example 3.2. To estimate the effect of match rate ($mrate$) on the participation rate ($prate$) in 401k plans we run the regression and include age of the plan to get

$$\widehat{prate} = \underset{(0.779)}{80.1190} + \underset{(0.526)}{5.5213} mrate + \underset{(0.045)}{0.2431} age.$$

Both the variables seem to have an effect. If we don't control for age, and drop it we get

$$\widehat{prate} = \underset{(0.563)}{83.0755} + \underset{(0.527)}{5.8611} mrate.$$

This small difference can be explained by the fact that the sample correlation between $mrate$ and age is only 11.88%. □

R^2 is nothing but the squared correlation between y and $\hat{y}$ and it never decreases, and usually increases when a new independent variable is added to a regression. It is a poor tool for deciding whether one variable or several variables should be added to a model. What matters is whether the explanatory variable has as nonzero partial effect on y. A low R^2 means predictability is difficult, it is still possible that the OLS estimates are reliable estimates of the ceteris paribus effects.

3.1 Gauss Markov assumptions

- MLR.1 **Linearity**: linear in parameters.
- MLR.2 **Independence**: random sampling.
- MLR.3 **No perfect collinearity** between dependent variables.
- MLR.4 **Zero conditional mean**: $E(u|x_1, \ldots, x_k) = 0$.

It should be emphasized that we do expect correlation between the independent variable, just that they should not be perfectly correlated. Omitting an important factor that is correlated with any of $x_1, \ldots, x_k$ causes MLR.4 to fail and will induce bias in the estimated coefficients.

Theorem 3.1 (Unbiasedness of OLS)**.** *$E(\hat{\beta}_j) = \beta_j$, for $j = 0, 1, \ldots, k$, for any values of the population parameter β_j.*

Unbiasedness means that there is assurance that we have no reason to believe that our estimates are more likely to be too big or more likely to be too small. Including irrelevant variables does not effect the expected value but can increase the variance of the OLS estimators. On the other hand, excluding a relevant variable causes OLS estimators to be biased. The correlation between a single explanatory variable and the error generally results in all OLS estimators being biased.

- MLR.5 **Homoskedasticity**: the error u has the same variance given any values of the explanatory variables, $Var(u|x_1, \ldots, x_k) = \sigma^2$.

MLR.5 simplifies the formulas and gives efficiency property to the OLS estimates. It also means that the variance does not depend on the independent variables.

Theorem 3.2. *Under MLR.1 to MLR.5, conditional on the sample values of the independent variables,*

$$Var(\hat{\beta}_j) = \frac{\sigma^2}{(1 - R_j^2)\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2}, \quad j = 1, \ldots, k$$

R_j^2 is the R-squared from regressing x_j on all other independent variables with intercept.

The problem of multi-collinearity occurs when $R_j^2 \to 1$ causing $Var(\hat{\beta}_j) \to \infty$. Micronumerosity, small sample size, has a similar effect on the variance of the estimates. High correlations between unimportant control variables is not an issue and hence omnibus multicollinearity statistic like condition number might not be vary useful. **Variance inflation**

factor (VIF) for individual coefficients $VIF_j = \frac{1}{1-R_j^2}$ might be slightly more useful. Though the variance of $\tilde{\beta}_1$ may be less than that of $\hat{\beta}_1$ when comparing single variable versus multi-variable regression, it is misleading since $Var(\tilde{\beta}_1)$ does not account for the effect of x_2, and hence leaving out a relevant factor causes both bias as well as under specification of variance.

Theorem 3.3. *Under the Gauss-Markov assumptions* $E(\hat{\sigma}^2) = \sigma^2$, *where* $\hat{\sigma}^2 = \frac{\sum_{i=1}^n \hat{u}_i^2}{n-k-1}$

The term $n-k-1$ is the degree of freedom. $\hat{\sigma}$ is called the **standard error** of the regression or root mean squared error. Though SSR must fall when another explanatory variable is added, the degrees of freedom also falls by one, so $\hat{\sigma}$ may fall or rise. Under homoskedasticity, estimate of $\hat{\beta}_j$ remains unbiased but leads to bias in $Var(\hat{\beta}_j)$, which then invalidates the standard errors.

Theorem 3.4 (**Gauss-Markov Theorem**)**.** *Under assumptions MLR.1 through MLR.5,* $\hat{\beta}_j$, $j = 0, 1, \ldots, k$ *are the best linear unbiased estimators (BLUE) of* β_j, *respectively.*

An estimator is linear if we can write $\hat{\beta}_j = \sum_{i=1}^n w_{ij} y_i$, i.e. it can be written as a linear combinations of dependent variable realization, where the weights depend on the independent variable realizations. By best we mean having the smallest variance.

4 Multiple Regression Inference

In order to perform statistical inference, we need to know more than just the first two moments of $\hat{\beta}_j$; we need to know the full sampling distribution of the $\hat{\beta}_j$. To make the sampling distributions of the $\hat{\beta}_j$ tractable, we assume that the unobserved error is normally distributed in the population.

- MLR.6 **Normality**: the population error u is independent of the explanatory variables $x_1, \ldots, x_k$ and is normally distributed with zero mean and variance $\sigma^2 : u \sim \mathcal{N}(0, \sigma^2)$.

The assumptions MLR.1 to MLR.6 are called the classical linear model assumptions. Under this additional assumption the OLS estimators are the **minimum variance unbaised estimators** (BUE); we no longer have to restrict our comparison to estimators that are linear in the y_i. A succinct way to summarize the population assumptions is:

$$y|\boldsymbol{x} \overset{ind.}{\sim} \mathcal{N}(\beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k, \sigma^2)$$

The argument that u is a sum of many different unobserved factors affecting y and hence CLT can be used to approximate normal distribution has some weaknesses. The normal approximation can be poor depending on how many factors appear in u and how different their distributions are. Further the assumption that each factors affect y in a separate, additive fashion which is unlikely. Normality of u, then, becomes an empirical matter. Sometime log transformations are needed to get empirical closeness to normality. However, non-normality of the errors is not a serious problem with large sample sizes.

Theorem 4.1. *Under the CLM assumptions, conditional on sample values of the independent variables,* $\frac{\hat{\beta}_j - \beta_j}{\sqrt{Var(\hat{\beta}_j)}} \sim \mathcal{N}(0,1)$. *Further,* $\frac{\hat{\beta}_j - \beta_j}{se(\hat{\beta}_j)} \sim t_{n-k-1}$, *where* $k+1$ *is the number of unknown parameters.*

The proof relies on the fact that $(n-k-1)\frac{\hat{\sigma}^2}{\sigma^2} \sim \chi^2_{n-k-1}$.

4.1 t test: Testing single hypothesis

We are primarily interested in testing the **null hypothesis $H_O : \beta_j = 0$**. The statistic we use to test this hypothesis is called the **t statistic** of $\hat{\beta}_j$ and is defined as $t_{\hat{\beta}_j} = \frac{\hat{\beta}_j}{se(\hat{\beta}_j)}$. It is important to remember that we are testing hypothesis about the population parameters (β_j) and not about the estimates from a particular sample ($\hat{\beta}_j$). For practical purposes t-distribution with degree of freedom > 100 is very close to normal distribution.

Example 4.1. The wage data gives us the following estimated equation

$$\widehat{\log(wage)} = \underset{(0.104)}{0.2844} + \underset{(0.007)}{0.0920}educ + \underset{(0.002)}{0.0041}exper + \underset{(0.003)}{0.0221}tenure.$$

This has $n = 526$ observations with R^2 of 0.316, with 3 parameters giving 522 degrees of freedom for the residual. Our null hypothesis is to test whether the return *exper*, controlling for *educ* and *tenure*, is zero in the population, against the alternative that it is positive.

$$H_0 : \beta_{exper} = 0 \quad \text{versus} \quad H_1 : \beta_{exper} > 0.$$

```
df = woo.dataWoo('wage1')
model = smf.ols(formula='np.log(wage)~educ+exper+tenure', data=df)
results = model.fit()
results.summary()
```

```
==============================================================================
Dep. Variable:           np.log(wage)   R-squared:                       0.316
Model:                            OLS   Adj. R-squared:                  0.312
Method:                 Least Squares   F-statistic:                     80.39
Date:                Fri, 09 Oct 2020   Prob (F-statistic):           9.13e-43
Time:                        08:16:08   Log-Likelihood:                -313.55
No. Observations:                 526   AIC:                             635.1
Df Residuals:                     522   BIC:                             652.2
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.2844      0.104      2.729      0.007       0.080       0.489
educ           0.0920      0.007     12.555      0.000       0.078       0.106
exper          0.0041      0.002      2.391      0.017       0.001       0.008
tenure         0.0221      0.003      7.133      0.000       0.016       0.028
==============================================================================
Omnibus:                       11.534   Durbin-Watson:                   1.769
Prob(Omnibus):                  0.003   Jarque-Bera (JB):               20.941
Skew:                           0.021   Prob(JB):                     2.84e-05
Kurtosis:                       3.977   Cond. No.                         135.
==============================================================================
```

Since we have 522 degrees of freedom, we can use the standard normal critical values of 1.645 for 5% and 2.326 for 1%. The t-statistic for $\hat{\beta}_{exper}$ is $t_{exper} = \frac{0.0041}{0.002} \approx 2.41$, and so $\hat{\beta}_{exper}$ is statistically significant even at 1% level. Adding three more years of experience increases $\log(wage)$ by $3 \times 0.0041 = 0.0123$, i.e. 1.2% higher. The estimated return for another year of experience, holding tenure and education fixed, is not especially large. Nevertheless, we have persuasively shown that the partial effect of experience is positive in the population. □

Example 4.2. To understand the effect of school size on student performance we look at standardized tenth-grade math test ($math10$) regressed on school size measured by student enrollment ($enroll$). The null hypothesis is $H_0 : \beta_{enroll} = 0$, and the alternative is $H_1 : \beta_{enroll} < 0$. We control for two other factors, teacher quality proxied by teacher compensation ($totcomp$), and attention students receive proxied by staff size ($staff$). The estimated equation is

$$\widehat{math10} = \underset{(6.114)}{2.2740} + \underset{(0.000)}{0.0005}totcomp + \underset{(0.040)}{0.0479}staff - \underset{(0.000)}{0.0002}enroll,$$

with 404 degrees of freedom and $R^2 = 0.054$. We use standard normal critical value. The coefficient on $enroll$ is negative according to the conjecture but the t-static is only -0.918, which is greater than the 5% critical level of -1.65 and 15% critical level of -1.04, making use fail to reject H_0 in favor or H_1 at those significance levels. We conclude $enroll$ is not statistically significant at 15% level. The variable $totcomp$ is statistically significant even at 1% significance level because its t-statistic is 4.6. On the other hand, the t-statistic for $staff$ is 1.2, and se we cannot reject $H_0 : \beta_{staff} = 0$ against $H_1 : \beta_{staff} > 0$ even at 10% significance level with critical value 1.28 for standard normal distribution.

```
df = woo.dataWoo('meap93')
model = smf.ols(formula='math10~totcomp+staff+enroll', data=df)
results = model.fit()
results.summary()
```

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	2.2740	6.114	0.372	0.710	-9.745	14.293
totcomp	0.0005	0.000	4.570	0.000	0.000	0.001
staff	0.0479	0.040	1.204	0.229	-0.030	0.126
enroll	-0.0002	0.000	-0.918	0.359	-0.001	0.000

```
model = smf.ols(formula='math10~np.log(totcomp)+np.log(staff)+np.log(enroll)', data=df)
results = model.fit()
results.summary()
```

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-207.6649	48.703	-4.264	0.000	-303.408	-111.922
np.log(totcomp)	21.1550	4.056	5.216	0.000	13.182	29.128
np.log(staff)	3.9800	4.190	0.950	0.343	-4.256	12.216
np.log(enroll)	-1.2680	0.693	-1.829	0.068	-2.631	0.095

We now use the logarithmic form for the independent variables to get

$$\widehat{math10} = \underset{(48.703)}{-207.6649} + \underset{(4.056)}{21.1550}\log(totcomp) + \underset{(4.190)}{3.9800}\log(staff) - \underset{(0.693)}{1.268}\log(enroll),$$

with 404 degrees of freedom and$R^2 = 0.065$. The t-statistic on $\log(enroll)$ is about -1.83; since this is below 5% critical value of -1.65, we rject $H_0 : \beta_{\log(enroll)}$ in favour of $H_1 : \beta_{\log(enroll)} < 0$ at the 5% level.

which of the two models: level-level or level-log do we prefer? The later due to higher R^2 for the same number of parameters. Higher explanatory power of level-log model of 6.5% against 5.4% of the level-level model makes it more suitable. In fact, this becomes much clear if we look a the histogram of *enroll* and $\log(enroll)$. □

When the sign of the coefficient is not well determined by theory we use the two-sided alternative $H_1 : \beta_j \neq 0$. Since we should never form a hypothesis after looking at the regression estimates, using a two sided hypothesis is often more prudent. The rejection rule for $H_0 : \beta_j = 0$ is $|t_{\hat{\beta}_j}| > c$, where c is an chosen critical value. For example, for a two-tailed test at 5% significance level, we want 2.5% in each tail equally. Hence, c is the 97.5th percentile in the t distribution with $n - k - 1$ degrees of freedom. For a normal distribution the 5% critical value for a two-sided test is $c \approx 2$. When a specific alternative is not stated, it is usually considered to be two-sided.

Example 4.3. We estimate a model explaining collage GPA ($colGPA$), with the average number of lectures missed per week ($skipped$), controlling for high school GPA ($hsGPA$) and ACT score (ACT). The estimated model is

$$\widehat{colGPA} = \underset{(0.332)}{1.3896} + \underset{(0.094)}{0.4118}hsGPA + \underset{(0.011)}{0.0147}ACT - \underset{(0.026)}{0.0831}skipped,$$

with 137 degrees of freedom and $R^2 = 0.234$.

```
df = woo.dataWoo('gpa1')
model = smf.ols(formula='colGPA~hsGPA+ACT+skipped', data=df)
results = model.fit()
results.summary()
```

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	1.3896	0.332	4.191	0.000	0.734	2.045
hsGPA	0.4118	0.094	4.396	0.000	0.227	0.597
ACT	0.0147	0.011	1.393	0.166	-0.006	0.036
skipped	-0.0831	0.026	-3.197	0.002	-0.135	-0.032

Using standard normal approximation the 1% critical value is 2.58. The t-statistic show that $hsGPA$ is statistically significant and ACT is not statistically significant at 1% levels for a

two-tailed test. The t-statistic of *skipped* is -3.197, so it is statistically significant at 1% significance level, meaning another lecture missed per week lower predicted $colGPA$ by about 0.083, holding $hsGPA$ and ACT fixed, as an average across a sub-population of students.

We could argue that a one-sided alternative is appropriate. The variables $hsGPA$ and *skipped* are very significant using a two-tailed test and have the signs that we expect, so there is no reason to do a one-tailed test. If we did do that, against a one-sided alternative $\beta_3 > 0$, ACT is significant at the 10% level but not at the 5% level. This does not change the fact that the coefficient on ACT is pretty small. □

For a null $H_0 : \beta_j = a_j$, where a_j is our hypothesized value of β_j, then the appropriate t-statistic is $t = \frac{\hat{\beta}_j - a_j}{se(\hat{\beta}_j)}$, which is distributed as t_{n-k-1}. Generally in log-log models we test for $a_j = 1$ or $a_j = -1$.

Example 4.4. We want to test the annual number of crimes on college campuses (*crime*) relation to student enrollment (*enroll*). This is a constant elasticity model and here we expect the total number of crimes to increase as the size of the campus increases. A more interesting hypothesis to test would be that the elasticity of crime with respect to enrollment is one $H_0 : \beta_1 = 1$. This means that a 1% increase in enrollment leads to, on an average, 1% increase in crime. A noteworthy alternative is $H_1 : \beta_1 > 1$. If $\beta_1 > 1$ then, in a relative sense, and not just relative sense, crime is more of a problem on larger campuses. The fitted equation is

$$\widehat{\log(crime)} = \underset{(1.034)}{-6.6314} + \underset{(0.110)}{1.2698}\log(enroll),$$

with degrees of freedom 95 and $R^2 = 0.585$.

```
df = woo.dataWoo('campus')
model = smf.ols(formula='np.log(crime)~np.log(enroll)', data=df)
results = model.fit()
results.summary()
```

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-6.6314	1.034	-6.416	0.000	-8.683	-4.580
np.log(enroll)	1.2698	0.110	11.567	0.000	1.052	1.488

To see if there is evidence to conclude $\beta_1 > 1$ we calculate the t-statistic as $t = (1.27 - 1)/0.11 \approx 2.45$. The one sided 5% critical value for a t distribution with 95 df is about 1.66, so we clearly reject $\beta_1 = 1$ in favor of $\beta_1 > 1$ at 5% level. This analysis holds no other factor constant, so the elasticity of 1.27 is not necessarily a good estimate of ceteris paribus effect. It could be that large enrollments are correlated with other factors that cause higher crime: large schools might be located in higher crime areas. We could control for this by collecting data on crime rates in the local city. □

For a two sided alternative, for example $H_0 : \beta_j = -1$, $H_1 : \beta_j \neq -1$, we still compute the t-statistic as $t = \frac{\hat{\beta}_j + 1}{se(\hat{\beta}_j)}$. The rejection rule is the usual one for a two-sided test: reject H_0 if

$|t| > c$, where c is a two-tailed critical value.

Rather than testing at different significance levels, it is more informative to answer - given the observed value of the t-statistic, what is the smallest significance level at which the null hypothesis would be rejected. This level is known as the **p-value** for the test. The p-value summarizes the strength or weakness of the empirical evidence against the null hypothesis. The p-value is the probability of observing a t-statistic as extreme as we did if the null hypothesis is true. This means that small p-values are evidence against the null; large p-values provide little evidence against H_0. If α denotes the significance level of the test, then H_0 is rejected if p-value $< \alpha$; otherwise, H_0 is not rejected at the 100 α % level. To calculate a single-tail p-value the two-tail p-value can be divided by 2. The magnitude and sign of the coefficients determine the economic or practical significance while t-statistic determine the statistical significance. With lower number of observations one can be more lenient with the significance levels. Remember that large standard errors can also be because of multicollinearity. Constructing confidence intervals is also straightforward giving $\hat{\beta}_j \pm c \times se(\hat{\beta}_j)$, where c is a critical value at a given significance level in a t_{n-k-1} distribution.

For a hypothesis of interest $H_0 : \beta_1 = \beta_2$ versus one sided alternative $H_1 : \beta_1 < \beta_2$ the main work is to estimate $se(\hat{\beta}_1 - \hat{\beta}_2)$, in the t-statistic $t = \frac{\hat{\beta}_1 - \hat{\beta}_2}{se(\hat{\beta}_1 - \hat{\beta}_2)}$. We can use $Var(\hat{\beta}_1 - \hat{\beta}_2) = Var(\hat{\beta}_1) + Var(\hat{\beta}_2) - 2Cov(\hat{\beta}_1 \hat{\beta}_2)$. But an easier way would be to estimate a different model that estimates $se(\hat{\beta}_1 - \hat{\beta}_2)$ directly where we define a new parameter $\theta = \beta_1 - \beta_2$.

Example 4.5. We want to estimate the difference of effects due to two year (jc) and four year ($univ$) of education on $\log(wage)$, controlling for the effect of work experience ($exper$).

$$\widehat{\log(wage)} = \underset{(0.021)}{1.4723} + \underset{(0.007)}{0.0667} jc + \underset{(0.002)}{0.0769} univ + \underset{(0.0002)}{0.0049} exper$$

with 6759 degrees of freedom and $R^2 = 0.222$. The t-statistic show that all the variables are significant but we are more concerned about testing whether the estimated difference in the coefficients is statistically significant. This gives $\hat{\beta}_{jc} - \hat{\beta}_{univ} = -0.0102$

```
df = woo.dataWoo('twoyear')
model = smf.ols(formula='lwage~jc+univ+exper', data=df)
results = model.fit()
results.summary()
```

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	1.4723	0.021	69.910	0.000	1.431	1.514
jc	0.0667	0.007	9.767	0.000	0.053	0.080
univ	0.0769	0.002	33.298	0.000	0.072	0.081
exper	0.0049	0.000	31.397	0.000	0.005	0.005

Now defining $\theta = \beta_1 - \beta_2$, we define $totcoll = jc + univ$ to get $\log(wage) = \beta_0 + \theta jc + \beta_2 totcoll + \beta_3 exper + u$. We estimate it to be

$$\widehat{\log(wage)} = \underset{(0.021)}{1.4723} - \underset{(0.007)}{0.0102} jc + \underset{(0.002)}{0.0769} totcoll + \underset{(0.0002)}{0.0049} exper$$

with 6759 degrees of freedom and $R^2 = 0.222$.

```
df = woo.dataWoo('twoyear')
model = smf.ols(formula='lwage~jc+totcoll+exper', data=df)
results = model.fit()
results.summary()
```

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	1.4723	0.021	69.910	0.000	1.431	1.514
jc	-0.0102	0.007	-1.468	0.142	-0.024	0.003
totcoll	0.0769	0.002	33.298	0.000	0.072	0.081
exper	0.0049	0.000	31.397	0.000	0.005	0.005

We see the t-statistic for θ is -1.668. Against the one-sided alternative, the p-value is about 0.071, so there is some, but not strong, evidence against the null. □

4.2 F test: Testing multiple linear restrictions

Frequently we wish to test multiple hypotheses about the underlying parameters. Using separate t-statistic to test a multiple hypothesis can be very misleading. We need to jointly test the restrictions. The unrestricted model with k independent variables is $y = \beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k + u$. Suppose we have q exclusion restrictions to test. The null hypothesis is stated as $H_0 : \beta_{k-q+1} = \ldots = \beta_k = 0$, with the alternative $H_1 : H_0$ is not true, i.e. one of the parameters is different from zero. The restricted model is $y = \beta_0 + \beta_1 x_1 + \ldots + \beta_{k-q} x_{k-q} + u$. The residual sum of squares will be higher for restricted model as it has lower number of parameters, but the relative increase of SSR can be used to test our hypothesis. The **F statistic** is defined by

$$F = \frac{(SSR_r - SSR_u)/q}{SSR_u/(n-k-1)} = \frac{(R_u^2 - R_r^2)/q}{(1-R_u^2)/(n-k-1)},$$

where SSR_r is the sum of squared residuals from the restricted model and SSR_u is the sum of squared residuals from the unrestricted model, and we use the fact $SSR = SST(1-R^2)$. The F statistic is always non-negative. The denominator of F is just the unbiased estimator of $\sigma^2 = Var(u)$ in the unrestricted model. F statistic is distributed as an F random variable with $(q, n-k-1)$ degrees of freedom ($F \sim F_{q,n-k-1}$). We will reject H_0 in favor of H_1 when F is sufficiently large. If the null is not rejected, then the variables are jointly insignificant and can be dropped. The F statistic is often useful for testing exclusion of a group of variables when the variables in the group are highly correlated, as F statistic is robust to

multi-collinearity.

F-statistic for testing exclusion of a single variable is equal to the square of the corresponding t-statistic. Since t^2_{n-k-1} has an $F_{1,n-k-1}$ distribution, the two approaches lead to exactly the same outcome, provided that the alternative is two-sided. The t-statistic is more flexible for testing a single hypothesis because it can be directly used to test against one sided alternatives. Two or more variables that each have insignificant t-statistic can be jointly very significant. It is also possible that in a group of variables jointly insignificant on variable has a significant t-statistic. For this reason we should not 'accept' null hypothesis but only fail to reject them. The t test is best suited for testing a single hypothesis, while F test is to detect whether a set of coefficients is different from zero. Often, when a variable is very statistically significant and it is tested jointly with other set of variables, the set will be jointly significant removing logical inconsistencies.

Example 4.6. To explain the child birth weight ($bwght$) we use the factors average number of cigarettes the mother smoked per day during pregnancy ($cigs$), the birth order of this child ($parity$), annual family income ($faminc$), years of schooling for the mother ($motheduc$), and years of schooling for the father ($fatheduc$). This gives the fitted model as

$$\begin{aligned} bwght = &\underset{(3.728)}{114.5243} - \underset{(0.110)}{0.5959}cigs + \underset{(0.659)}{1.7876}parity + \underset{(0.037)}{0.0560}faminc \\ &- \underset{(0.320)}{0.3705}motheduc + \underset{(0.283)}{0.4724}fatheduc, \end{aligned}$$

with degrees of freedom 1185 and $R^2 = 0.039$.

```
df = woo.dataWoo('bwght')
df = df[['bwght', 'cigs', 'parity', 'faminc', 'motheduc', 'fatheduc']].dropna()
model0 = smf.ols(formula='bwght~cigs+parity+faminc', data=df)
results0 = model0.fit()
model1 = smf.ols(formula='bwght~cigs+parity+faminc+motheduc+fatheduc', data=df)
results1 = model1.fit()
results1.summary()
```

```
==============================================================================
Dep. Variable:                  bwght   R-squared:                       0.039
Model:                            OLS   Adj. R-squared:                  0.035
Method:                 Least Squares   F-statistic:                     9.553
Date:                Sun, 11 Oct 2020   Prob (F-statistic):           5.99e-09
Time:                        08:16:13   Log-Likelihood:                -5242.2
No. Observations:                1191   AIC:                         1.050e+04
Df Residuals:                    1185   BIC:                         1.053e+04
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    114.5243      3.728     30.716      0.000     107.209     121.839
cigs          -0.5959      0.110     -5.401      0.000      -0.812      -0.379
parity         1.7876      0.659      2.711      0.007       0.494       3.081
faminc         0.0560      0.037      1.533      0.126      -0.016       0.128
```

```
motheduc      -0.3705      0.320     -1.158      0.247      -0.998       0.257
fatheduc       0.4724      0.283      1.671      0.095      -0.082       1.027
==============================================================================
# manual calc
import scipy.stats as stats
F = ((results1.rsquared-results0.rsquared)/(results0.df_resid-results1.df_resid))/
     ((1-results1.rsquared)/results1.df_resid)
c = stats.f.ppf(1-0.05, 2, 1185)
pval = 1-stats.f.cdf(F, 2, 1185)
print(F, c, pval)
>> 1.4372686389751963 3.003318396872088 0.23798962194786832
# automatic test
ftest = results.f_test(['motheduc=0', 'fatheduc=0'])
print(ftest.statistic[0][0], ftest.pvalue)
>> 1.4372686389751665 0.23798962194786966
```

We want to test the null hypothesis that, after controlling for *cigs*, *parity*, $famic$, parents' education has no effect on birth weight. That is $H_0 : \beta_{motheduc} = \beta_{fatheduc} = 0$. So there are $q = 2$ exclusion restrictions to be tested and the degrees of freedom of unrestricted model is $n - 6 = 1185$. When there is missing data, like here, we must the same observations to estimate both the restricted and unrestricted model. The 5% critical value for the F distribution with 2 and 1185 df is 3.003, while the calculated F-statistic is 1.437, making us fail to reject H_0, i.e. *motheduc* and $fatheduc$ are jointly insignificant in the birth weight equation at 5% critical level. □

For F tests, p-values ($P(\mathcal{F} > F)$) are especially useful, where $\mathcal{F}$ is the F random variable with $(q, n - k - 1)$ degrees of freedom and F is the actual value of the test statistic. A small p-value is evidence against H_0. The p-value in the previous example was 0.238, and so the null hypothesis that both $\beta_{motheduc}$ and $\beta_{fatheduc}$ are zero is not rejected at even 20% significance level.

To test the overall significance of the regression, i.e $H_0 : \beta_1 = \ldots = \beta_k = 0$ versus the alternative that at least one of the β_j is different form zero the F statistic reduces to

$$F = \frac{R/k}{(1 - R^2)/(n - k - 1)}$$

because the R^2 of the model $y = \beta_0 + u$ is zero, as there are no variables to explain the variance of y. In the previous example the F-statistic was reported to be 9.553 for the overall significance of the regression, with $k = 5$ and $n - k - 1 = 1185$ degrees of freedom. The p-value is extremely small making us reject the null hypothesis and concluding that at least one of the variables do explain some variation in *bwght*. The amount explained is not large and only 3.87%, but seemingly small R^2 results in highly significant F-statistic. That is why we need to look at the joint test and not just look at the R^2.

Example 4.7. We want to estimate the house price based on some factors including the assessed housing value (*assess*). Other factors are *lotsize*, $sqrft$, and *bdrms*. An unrestricted

model gives

$$\log(price) = \underset{(0.570)}{0.2637} + \underset{(0.151)}{1.0431}\log(assess) + \underset{(0.039)}{0.0074}\log(lotsize) - \underset{(0.138)}{0.1032}\log(sqrft) + \underset{(0.022)}{0.0338}bdrms,$$

with degrees of freedom 83 and $R^2 = 0.773$. We would like to test whether the assessed housing price is a rational valuation. This can be assessed by $H_0 : \beta_{lassess} = 1$, $\beta_{llotsize} = \beta_{lsqrft} = \beta_{bdrma} = 0$. To get the restricted model we run $\log price - \log(assess) = \beta_0 + u$ giving a fit

$$\log(price) - \log(assess) = \underset{(0.016)}{-0.0848},$$

with degrees of freedom 87 and $R^2 = 0$ as there are no variables to explain the variance. We can't use the R^2 form of the F-statistic here because the dependent variable is different in the two regressions. We, hence, resort to using SSR form of the F-statistic here. The calculated F-statistic is 0.667 which is below the 5% ciritcal value of $F(4, 83) = 2.48$, and so we fail to reject H_0. The p-value if 0.6162 suggesting there is essentially no evidence against the hypothesis that the assessed values are rational.

```
df = woo.dataWoo('hprice1')
df = df[['lprice', 'lassess', 'llotsize', 'lsqrft', 'bdrms']].dropna()
model1 = smf.ols(formula='lprice~lassess+llotsize+lsqrft+bdrms', data=df)
results1 = model1.fit()
results1.summary()
```

Dep. Variable:	lprice	R-squared:	0.773
Model:	OLS	Adj. R-squared:	0.762
Method:	Least Squares	F-statistic:	70.58
Date:	Sun, 11 Oct 2020	Prob (F-statistic):	6.45e-26
Time:	09:33:18	Log-Likelihood:	45.750
No. Observations:	88	AIC:	-81.50
Df Residuals:	83	BIC:	-69.11
Df Model:	4		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.2637	0.570	0.463	0.645	-0.869	1.397
lassess	1.0431	0.151	6.887	0.000	0.742	1.344
llotsize	0.0074	0.039	0.193	0.848	-0.069	0.084
lsqrft	-0.1032	0.138	-0.746	0.458	-0.379	0.172
bdrms	0.0338	0.022	1.531	0.129	-0.010	0.078

```
df['lpriceMlassess'] = df.lprice - df.lassess
model0 = smf.ols(formula='lpriceMlassess~1', data=df)
results0 = model0.fit()
results0.summary()
```

Dep. Variable:	lpriceMlassess	R-squared:	0.000
Model:	OLS	Adj. R-squared:	0.000
Method:	Least Squares	F-statistic:	nan

```
Date:                Sun, 11 Oct 2020   Prob (F-statistic):                nan
Time:                        09:36:04   Log-Likelihood:                  44.357
No. Observations:                  88   AIC:                             -86.71
Df Residuals:                      87   BIC:                             -84.24
Df Model:                           0
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0848      0.016     -5.412      0.000      -0.116      -0.054
==============================================================================
SSR1 = results1.mse_resid*results1.df_resid
SSR0 = results0.mse_resid*results0.df_resid
F = ((SSR0-SSR1)/(results0.df_resid-results1.df_resid))/(SSR1/results1.df_resid)
print(SSR1, SSR0, F)
>> 1.82152879331078 1.8801488540043438 0.6677721833760327
print(stats.f.ppf(1-0.05,4,83), 1-stats.f.cdf(F, 4, 83))
>> 2.4816614292470063 0.6161596292005588
```

□

5 OLS Asymptotics

Till now we have looked at the finite sample properties of OLS estimators. Asymptotic properties work when $n \to \infty$. One particularly important finding is that even without the normality assumption, MLR.6, t and F statistics have approximately t and F distributions, at least in large sample sizes.

5.1 Consistency

Theorem 5.1 (consistency of OLS)**.** *Under assumptions MLR.1 through MLR.4, the OLS estimator $\hat{\beta}_j$ is consistent for β_j, for all $j = 0, 1, \ldots, k$.*

For each n, $\hat{\beta}_j$ has a probability distribution. If this estimator is consistent, then this distribution becomes more and more tightly distributed around β_j as $n \to \infty$, where it collapses to the single point β_j. We can write the estimated coefficients as

$$\begin{aligned}
\hat{\boldsymbol{\beta}} &= \left(\sum_{t=1}^{n} \boldsymbol{x}_t^T \boldsymbol{x}_t\right)^{-1} \left(\sum_{t=1}^{n} \boldsymbol{x}_t^T y_t\right) \\
&= \left(\sum_{t=1}^{n} \boldsymbol{x}_t^T \boldsymbol{x}_t\right)^{-1} \left(\sum_{t=1}^{n} \boldsymbol{x}_t^T (\boldsymbol{\beta}\boldsymbol{x}_t + u_t)\right) \quad (\because y_t = \boldsymbol{\beta}\boldsymbol{x}_t + u_t) \\
&= \boldsymbol{\beta} + \left(\frac{1}{n}\sum_{t=1}^{n} \boldsymbol{x}_t^T \boldsymbol{x}_t\right)^{-1} \left(\frac{1}{n}\sum_{t=1}^{n} \boldsymbol{x}_t^T u_t\right)
\end{aligned}$$

By law of large numbers we see the following convergence in probability,

$$\frac{1}{n}\sum_{t=1}^{n}\boldsymbol{x}_t^T\boldsymbol{x}_t \xrightarrow{P} \boldsymbol{A} \quad \text{and} \quad \frac{1}{n}\sum_{t=1}^{n}\boldsymbol{x}_t^T u_t \xrightarrow{P} \boldsymbol{0},$$

where $\boldsymbol{A} = E(\boldsymbol{x}_t^T\boldsymbol{x}_t)$ is the nonsingular matrix, and $E(\boldsymbol{x}_t^T u_t) = 0$. Hence,

$$plim(\hat{\boldsymbol{\beta}}) = \boldsymbol{\beta} + \boldsymbol{A}^{-1}.\boldsymbol{0} = \boldsymbol{\beta}.$$

- MLR.4' Zero mean and correlation: $E(u) = 0$ and $Cov(x_j, u) = 0$, for $j = 1, 2, \ldots, k$.

Assumption MLR.4' is weaker than MLR.4 in the sense that the latter implies the former. OLS turns out to be biased but consistent under assumption MLR.4' if $E(u|x_1, \ldots, x_k)$ depends on any of the x_j. Finite sample estimators need the stronger zero conditional mean assumption to be unbaised. Further, MLR.4 means that we have properly modeled the population regression function (PRF) as $E(y|x_1, \ldots, x_k) = \beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k$, so we can obtain partial effects of the explanatory variables on the average value of y. If we instead assume MLR.4', $\beta_0+\beta_1 x_1+\ldots+\beta_k x_k$ need not represent the PRF, and we face the possibility that some nonlinear function of x_j could be correlated with the error u. Most of the times we hope to get a good estimate of the PRF, and so the zero conditional mean assumption is natural. Nevertheless, the weaker zero correlation assumption turns out to be useful in interpreting OLS estimation of a linear model as providing the best linear approximation to the PRF, and in the situation where we have no interest in modeling a PRF.

If the error is correlated with any of the independent variables, then OLS is biased and inconsistent, i.e. asymptotic bias $= plim\hat{\beta}_1 - \beta_1 = \frac{Cov(x_1,u)}{Var(x_1)}$. For an omitted variable case were the true model $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \nu$, satisfies the four Gauss-Markov assumptions, ν has zero mean and is uncorrelated with x_1 and x_2. $\hat{\beta}_0, \hat{\beta}_1$, and $\hat{\beta}_2$ are consistent. If we omit x_2 and fit $y = \tilde{\beta}_0 + \tilde{\beta}_1 x_1 + u$, then

$$plim\tilde{\beta}_1 = \beta_1 + \beta_2\delta_1, \quad \text{where } \delta_1 = \frac{Cov(x_1, x_2)}{Var(x_1)}.$$

This inconsistency is expressed in terms of the population variance of x_1 and the population covariance between x_1 and x_2, while the bias is based on their sample counterparts. If x_1 and x_2 are uncorrelated, then $\delta_1 = 0$, and $\tilde{\beta}_1$ is a consistent estimator of β_1, although not necessarily unbiased. This consistency only gets worse by adding more observations to the sample. If x_1 is correlated with u but the other independent variables are uncorrelated with u, all the OLS estimators are generally inconsistent. If x_1 is correlated with u, but x_1 and u are uncorrelated with the other independent variables, then only $\hat{\beta}_1$ is inconsistent.

5.2 Asymptotic Normality and Large Sample Inference

Under finite sample, the exact normality of the OLS estimators hinge crucially on the normality of the distribution of the error, u in the population. Assumption MLR.6 is equivalent to saying that the distribution of y given $x_1, x_2, \ldots, x_k$ is normal, which clearly is violated in many cases. Normality does not play any role in unbiasedness of OLS, not does it affect

the conclusion that OLS is BLUE under Gauss-Markov assumptions. But exact inference based on t and F statistics require MLR.6. Fortunately, even if y_i are not from a normal distribution, we can use the central limit theorem to conclude that OLS estimators satisfy asymptotic normality.

Theorem 5.2 (**Asymptotic normality of OLS**). *Under the Gauss-Markov assumptions MLR.1 to MLR.5*

1. $\sqrt{n}(\hat{\beta}_j - \beta_j) \overset{a}{\sim} \mathcal{N}(0, \sigma^2/a_j^2)$, *where* $a_j^2 = plim\left(\frac{1}{n}\sum_{i=1}^{n}\hat{r}_{ij}^2\right)$, *where* $\hat{r}_{ij}$ *are the residuals from regressing* x_j *on the other independent variables. We say that* $\hat{\beta}_j$ *is asymptotically normally distributed.*

2. $\hat{\sigma}^2$ *is a consistent estimator of* $\sigma^2 = Var(u)$.

3. *For each* j, $\frac{\hat{\beta}_j - \beta_j}{sd(\hat{\beta}_j)} \overset{a}{\sim} \mathcal{N}(0,1)$ *and* $\frac{\hat{\beta}_j - \beta_j}{se(\hat{\beta}_j)} \overset{a}{\sim} \mathcal{N}(0,1)$, *where* $se(\hat{\beta}_j)$ *is the usual OLS standard error.*

This theorem essentially drops the normality assumption of MLR.6, with the only restriction on the distribution of error is that it has finite variance. We have also assumed zero conditional mean (MLR.4) and homoskedasticity (MLR.5). Regardless of the population distribution of u, the OLS estimators, when properly standardized, have approximately standard normal distribution asymptotically, by virtue of the fact that OLS estimators involve the use of sample averages and application of central limit theorem. From an asymptotic point of view it does not matter if we use σ or $\hat{\sigma}$. Hence, we can use standard normal distribution for inference rather than t distribution. But from a practical perspective it is legitimate to use t distribution because it approaches normal distribution as df gets large. The estimate variance of $\hat{\beta}_j$ is

$$\widehat{Var(\hat{\beta}_j)} = \frac{\hat{\sigma}^2}{SST_j(1 - R_j^2)},$$

where SST_j is the total sum of squares of x_j in the sample, and R_j^2 is the R-squared from regressing x_j on all of the other independent variables. As sample size grows $\hat{\sigma}^2$ converges to the constant σ^2. The sample variance of x_j is SST_j/n, hence SST_j converges to $n\sigma_j^2$, where σ_j^2 is the population variance of x_j. Hence, $\widehat{Var(\hat{\beta}_j)}$ shrinks to zero at the rate $1/n$. The square root of this is called the asymptotic standard error

$$se(\hat{\beta}_j) \approx \frac{1}{\sqrt{n}}\frac{\sigma}{\sigma_j\sqrt{1 - \rho_j^2}},$$

where $\sigma = sd(u)$, $\sigma_j = sd(x_j)$, and ρ_j^2 is the population R^2 from regression x_j on the other explanatory variables. The asymptotic normality of OLS estimator also implies that the F statistic have approximate F distributions in large sample sizes. Thus, for testing exclusion restrictions of other multiple hypotheses, we an continue to use the same recipe as before.

5.3 Large sample Lagrange Multiplier statistic

Sometimes it is useful to have other ways to test multiple exclusion restrictions. **Lagrange multiplier statistic** (LM) is a popular one; also called score statistic. Consider the usual multiple regression model with k independent variables $y = \beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k + u$. We would like to test if the last q of these variables all have zero population parameters with the null hypothesis $H_0 : \beta_{k-q+1} = \ldots = \beta_k = 0$, which puts q exclusion restrictions on the model. The alternative is that at least one of the parameters is different form zero. The LM statistic first runs the restricted model as $y = \tilde{\beta}_0 + \tilde{\beta}_1 x_1 + \ldots + \tilde{\beta}_{k-q} x_{k-q} + \tilde{u}$. If the omitted variables x_{k-q+1} through x_k truly have zero population coefficients, then approximately, $\tilde{u}$ should be uncorrelated with each of these variables in the sample. Thus, we run an auxiliary regression of $\tilde{u}$ on $x_1, \ldots, x_k$. If Null is true, R^2 from the auxiliary regression should be close to zero, subject to sampling error. Under the null hypothesis, the sample size multiplied by the usual R^2 from the auxiliary regression is distributed asymptotically as a chi-squared random variable with q degrees of freedom.

Example 5.1. We want to explain the number of times a man was arrested ($narr86$) based on the factors the proportion of prior arrests leading to conviction ($pcnv$), average sentence served from past convictions ($avgsen$), total time the man has spent in prison prior to 1986 since reaching the age of 18 ($tottime$), months spent in prison in 1986 ($ptime86$), and number of quarters in 1986 during which the man was legally employed ($qemp86$). We use LM statistic to test the null hypothesis that $avgsen$ and $tottime$ have no effect on $narr86$ once the other factors have been controlled for.

```
# restricted regression
df = woo.dataWoo('crime1')
model_r = smf.ols(formula='narr86~pcnv+ptime86+qemp86', data=df)
results_r = model_r.fit()
r2_r = results_r.rsquared
df['utilde'] = results_r.resid

# auxiliary regression
model_a = smf.ols(formula='utilde~pcnv+ptime86+qemp86+avgsen+tottime', data=df)
results_a = model_a.fit()
r2_a = results_a.rsquared

# LM test
n = results_a.nobs
LM = n * r2_a
c = stats.chi2.ppf(1-0.1, 2)
pval = 1-stats.chi2.cdf(LM, 2)

print(r2_r, r2_a, n, LM, c, pval)
>> 0.041323 0.001494 2725.0 4.070729 4.605170 0.130633

# F test
model_f = smf.ols(formula='narr86~pcnv+ptime86+qemp86+avgsen+tottime', data=df)
results_f = model_f.fit()
ftest = results_f.f_test(['avgsen=0', 'tottime=0'])
print(ftest.statistic[0][0], ftest.pvalue)
```

```
>> 2.033921 0.131020
```

We obtain $\tilde{u}$ from the restricted regression with 2725 observations. We then run the auxiliary regression and obtain $R_a^2 = 0.001494$, giving the LM statistic of $LM = nR_a^2 = 4.07$. The 10% critical value in a chi-squared distribution with two degree of freedom is about 4.61. Thus, we fail to reject the null hypothsis that $\beta_{avgsen} = \beta_{tottime} = 0$ at the 10% level. The p-value $P(\chi_2^2 > 4.07) \approx 0.131$, so we would reject H_0 at the 15% level. As a comparison, the F test for joint significance of *avgsen* and *tottime* yields a p-value of about 0.131, which is pretty close to that obtained using the LM statistic. □

5.4 Efficiency

OLS is also asymptotically efficient among a certain class of estimators under the Gauss-Markov assumptions. For a simple regression case $y = \beta_0 + \beta_1 x + u$, u has a zero conditional mean under MLR.4: $E(u|x) = 0$. Let $g(x)$ be any function of x, then u is uncorrelated with $g(x)$. Let $z_i = g(x_i)$ for all observations i. Then the estimator

$$\tilde{\beta}_1 = \frac{\sum_{i=1}^{n}(z_i - \bar{z})y_i}{\sum_{i=1}^{n}(z_i - \bar{z})x_i}$$

is consistent for β_1, provided $g(x)$ and x are correlated, i.e. $plim\tilde{\beta}_1 = \beta_1$. Further, $\sqrt{n}(\tilde{\beta}_1 - \beta_1)$ is asymptotically normal with mean zero and asymptotic variance $\sigma^2 Var(z)/(Cov(z,x))^2$. The asymptotic variance of the OLS estimator is obtained when $z = x$, in which case $\sqrt{n}(\hat{\beta}_1 - \beta_1)$ is asymptotically normal with variance $\sigma^2/Var(x)$. Using Cauchy-Schwartz inequality we get $(Cov(z,x))^2 \leq Var(z)Var(x)$, which implies that the asymptotic variance of $\sqrt{n}(\hat{\beta}_1 - \beta_1)$ is no larger than that of $\sqrt{n}(\tilde{\beta}_1 - \beta_1)$. Hence, for simple regression, under the Gauss-Markov assumptions, OLS has smaller asymptotic variance than any of the **instrumental variables estimator**. If homoskedasticity assumption fails, then there are estimators of this form that have a smaller asymptotic variance than OLS. In the k regressor case, the class of consistent estimators is obtained by generalizing the OLS first order conditions.

Theorem 5.3. *Asymptotic efficiency of OLS Under the Gauss-Markov assumptions, let $\tilde{\beta}_j$ denote estimators that solve equations*

$$\sum_{i=1}^{n} g_j(\boldsymbol{x}_i)(y_i - \tilde{\beta}_0 - \tilde{\beta}_1 x_{i1} - \ldots - \tilde{\beta}_k x_{ik}) = 0, \quad j = 0, 1, \ldots, k$$

where $g_j(\boldsymbol{x}_i)$ denotes any function of all explanatory variables for observation i, and let $\hat{\beta}_j$ denote the OLS estimators. Then for $j = 0, 1, 2, , k$, the OLS estimators have the smallest asymptotic variances: $aVar\sqrt{n}(\hat{\beta}_j - \beta_j) \leq aVar\sqrt{n}(\tilde{\beta}_j - \beta_j)$.

6 Further Issues

Sometimes, it is useful to obtain regression results when all variables involved have been standardized. The regression equation $y_i = \hat{\beta}_0 + \beta_1 x_{i1} + \ldots + \hat{\beta}_k x_{ik} + \hat{u}_i$ can be transformed to

$$z_y = \hat{b}_1 z_1 + \hat{b}_2 z_2 + \ldots + \hat{b}_k z_k + \hat{v}_i,$$

where $z_j = (x_{ij} - \bar{x}_j)/\hat{\sigma}_j$, $z_y = (y_i - \bar{y})/\hat{\sigma}_y$, and $\hat{v}_i = \hat{u}_i/\hat{\sigma}_y$. The new coefficients called **standardized** or **beta coefficients**, are

$$\hat{b}_j = \frac{\hat{\sigma}_j}{\hat{\sigma}_y}\hat{\beta}_j \quad \text{for } j = 1, \ldots, k$$

Notice that this transformed equation has no intercept. If x_1 increases by one standard deviation, then $\hat{y}$ changes by $\hat{b}_1$ standard deviations, making the scale of regressors irrelevant and putting explanatory variables on equal footing. Comparing the magnitudes of the resulting beta coefficients is more compelling. For a single variable case the beta coefficient is simply the correlation coefficient between y and x_1. The t-statistics are not affected.

Example 6.1. We are interested in how air pollution (nox) and other neighborhood characteristics affect the value of a house ($price$). We first do a standard regression to get

$$\begin{aligned} price = & \underset{(5054.599)}{20870} - \underset{(354.087)}{2706.4326}\, nox - \underset{(32.929)}{153.601}\, crime \\ & + \underset{(393.604)}{6735.4983}\, rooms - \underset{(188.108)}{1026.8063}\, dist - \underset{(127.429)}{1149.2038}\, stratio. \end{aligned}$$

with degrees of freedom 500 and $R^2 = 0.632$

```
df = woo.dataWoo('hprice2')
model = smf.ols(formula='price~nox+crime+rooms+dist+stratio', data=df)
results = model.fit()
results.summary()
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept   2.087e+04   5054.599      4.129      0.000    1.09e+04    3.08e+04
nox        -2706.4326    354.087     -7.643      0.000   -3402.114   -2010.751
crime       -153.6010     32.929     -4.665      0.000    -218.297     -88.905
rooms       6735.4983    393.604     17.112      0.000    5962.177    7508.819
dist       -1026.8063    188.108     -5.459      0.000   -1396.386    -657.227
stratio    -1149.2038    127.429     -9.018      0.000   -1399.566    -898.842
==============================================================================
zdf = (df-df.mean())/df.std()
model = smf.ols(formula='price~nox+crime+rooms+dist+stratio', data=zdf)
results = model.fit()
results.summary()
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
```

Intercept	8.066e-17	0.027	2.99e-15	1.000	-0.053	0.053
nox	-0.3404	0.045	-7.643	0.000	-0.428	-0.253
crime	-0.1433	0.031	-4.665	0.000	-0.204	-0.083
rooms	0.5139	0.030	17.112	0.000	0.455	0.573
dist	-0.2348	0.043	-5.459	0.000	-0.319	-0.150
stratio	-0.2703	0.030	-9.018	0.000	-0.329	-0.211

We then do the standardized regression to get

$$\begin{aligned} Z_{price} = &\underset{(7.643)}{-0.3404}\ Z_{nox} - \underset{(0.031)}{0.1433}\ Z_{crime} \\ &+ \underset{(0.030)}{0.5139}\ Z_{rooms} - \underset{(0.043)}{0.2348}\ Z_{dist} - \underset{(0.030)}{0.2703}\ Z_{stratio}. \end{aligned}$$

with degrees of freedom 500 and $R^2 = 0.632$. The table shows that the t-statistics are the same and the intercept in the standardized regression is 0. It is much easier to interpret the second regression coefficients. □

6.1 Functional form

When we take a natural log of an independent variable x_j and dependent variable y, its coefficient β_j is the **elasticity** of y with respect to x_j; while when we take log only of the dependent variable y leaving the independent variable as it is, the coefficient is called the **semi-elasticity** of y with respect to x_j. For the model $\widehat{\log(y)} = \hat{\beta}_0 + \hat{\beta}_1 \log(x_1) + \hat{\beta}_2 x_2$, we have for constant x_2, $\Delta\hat{y} = \Delta x_1$. While for constant x_1, we have $\Delta\hat{y} = e^{\hat{\beta}_2 \Delta x_2} - 1$, which is a biased estimator due to nonlinear function, though consistent. Logarithms make the coefficient independent of the scale. Moreover, if $y > 0$, models using $\log(y)$ convert the often heteroskedastic or skewed distributions in more CLM compliant distributions. Taking the log also narrows the range of the variable making OLS estimates less sensitive to outliers.

Quadratic functions are often used to capture decreasing or increasing marginal effects. For the equation $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\beta}_2 x^2$, we have $\Delta\hat{y} = (\hat{\beta}_1 + 2\hat{\beta}_2 x)\Delta x$, meaning the slope depends on the value or x. When $\hat{\beta}_1$ is positive and $\hat{\beta}_2$ is negative we have a case of diminishing marginal effect of x on y. When $\hat{\beta}_1$ is negative and $\hat{\beta}_2$ is positive we have an increasing effect of x on y.

Example 6.2. We look at the effect of quadratic form in *rooms* on log *prices*.

$$\begin{aligned} \log(price) = &\underset{(0.57)}{13.39} - \underset{(0.12)}{0.91}\ \log(nox) - \underset{(0.043)}{0.087}\ \log(dist) \\ &- \underset{(0.165)}{0.545}\ rooms + \underset{(0.013)}{0.0623}\ rooms^2 - \underset{(0.006)}{0.0476}\ stratio \end{aligned}$$

with 500 degrees of freedoma and $R^2 = 0.603$.

```
df = woo.dataWoo('hprice2')
model = smf.ols(formula='lprice~lnox+np.log(dist)+rooms+I(rooms**2)+stratio', data=df)
results = model.fit()
results.summary()
```

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	13.3855	0.566	23.630	0.000	12.273	14.498
lnox	-0.9017	0.115	-7.862	0.000	-1.127	-0.676
np.log(dist)	-0.0868	0.043	-2.005	0.045	-0.172	-0.002
rooms	-0.5451	0.165	-3.295	0.001	-0.870	-0.220
I(rooms ** 2)	0.0623	0.013	4.862	0.000	0.037	0.087
stratio	-0.0476	0.006	-8.129	0.000	-0.059	-0.036

```
(df.rooms<4.4).mean()
>> 0.009881422924901186
```

The t-statistic for the term $rooms^2$ is 4.862, and so is very significant.The coefficient of the room being negative suggest that the curve is U-shaped reaching a minimum at $x^* = -\hat{\beta}_1/(2\hat{\beta}_2) \approx 4.4$. We notice that less than 1% of the data is below this value and hence can be ignored. To the right of 4.4 we see that adding another room has an increasing effect on the percentage change in price $\Delta\widehat{price} \approx (-0.545 + 2 \times 0.0623\ rooms)\Delta\ rooms$. Thus, there is a marginally increasing effect of *rooms* on *price*. □

Sometimes, it is natural to include **interaction effect**, where the partial effect, elasticity, or semi-elasticity of the dependent variable with respect to an explanatory variable depends on the magnitude of yet another explanatory variable. Interpretation has to be carefully done in this case and often reparameterization by centring to some central tendency can help.

Example 6.3. To explain the standardized outcome on a final exam (*stndfnl*) in terms of percentage of classes attended (*atndrte*), prior college grade point average (*priGPA*), and ACT score (*ACT*), we use the standardized exam scores and in addtion to the quadratics in *priGPA* and *ACT*, we include an interaction between *priGPA* and *atndrte*. The idea is that class attendance might have a different effect for students who have performed differently in the past. We fit the model

$$\begin{aligned}\widehat{stndfnl} = &\underset{(1.36)}{2.05} - \underset{(0.01)}{0.0067}\ atndrte - \underset{(0.48)}{1.63}\ priGPA \\ &- \underset{(0.098)}{0.128}\ ACT + \underset{(0.101)}{0.2959}\ priGPA^2 + \underset{(0.002)}{0.0045}\ ACT^2 \\ &+ \underset{(0.004)}{0.0056}\ priGPA \times atndrte\end{aligned}$$

with degrees of freedom 673 and $R^2 = 0.229$.

```
df = woo.dataWoo('attend')
model = smf.ols(formula='stndfnl~atndrte+priGPA+ACT+I(priGPA**2)'+
```

```
                        'I(ACT**2)+I(priGPA * atndrte)', data=df)
results = model.fit()
results.summary()
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      2.0503      1.360      1.507      0.132      -0.621       4.721
atndrte       -0.0067      0.010     -0.656      0.512      -0.027       0.013
priGPA        -1.6285      0.481     -3.386      0.001      -2.573      -0.684
ACT           -0.1280      0.098     -1.300      0.194      -0.321       0.065
I(priGPA ** 2)  0.2959      0.101      2.928      0.004       0.097       0.494
I(ACT ** 2)     0.0045      0.002      2.083      0.038       0.000       0.009
I(priGPA * atndrte)  0.0056      0.004      1.294      0.196      -0.003       0.014
==============================================================================
ftest = results.f_test(['atndrte=0','I(priGPA * atndrte)=0'])
print(ftest.statistic[0][0], ftest.pvalue)
>> 4.319021763197381 0.013683848683999768
df.priGPA.mean()
>> 2.586774999078582
```

Looking at the coefficient for $atndre$, which is negative, we may wrongly conclude that attendance has a negative effect on final exam score. But this coefficient supposedly measures the effect when $priGPA = 0$, which is not interesting, as the smallest $priGPA$ in this sample is 0.86. We must also take care not to look separately at the estimates of the coefficients of $atndrte$ and $priGPA \times atndrte$ and conclude that, because each t statistic is insignificant, we cannot reject $H_0 : \beta_{atndrte} = \beta_{priGPA \times atndrte} = 0$. We calculate the p-value for the F test of this joint hypothesis to 0.0137, so we certainly reject H_0 at 5% level. The effect of $atndrte$ on $stndfnl$ can be represented by $\Delta\widehat{stndfnl} = (-0.0067 + 0.0056\ priGPA)\Delta atndrte$. To get the average effect we use the average value of $priGPA$ which is 2.59, giving the average effect as $\Delta\widehat{stndfnl} \approx 0.0078\Delta atndrte$. A 10 percentage point increase in $atndrte$ increases $\widehat{stndfnl}$ by 0.078 standard deviations from the mean final exam score.

To figure out if 0.0078 is statistically significant we replace $pirGPA \times atndrte$ with $(priGPA - 2.59) \times atndrte$ and get the t-statistic on $atndrte$ as 2.938, showing at the average $priGPA$, attendance has a statistically significant positive effect on final exam score.

```
df['DpriGPA'] = df.priGPA - df.priGPA.mean()
model = smf.ols(formula='stndfnl~atndrte+priGPA+ACT+I(priGPA**2)'+
                        'I(ACT**2)+I(DpriGPA * atndrte)', data=df)
results = model.fit()
results.summary()
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      2.0503      1.360      1.507      0.132      -0.621       4.721
atndrte        0.0077      0.003      2.938      0.003       0.003       0.013
priGPA        -1.6285      0.481     -3.386      0.001      -2.573      -0.684
ACT           -0.1280      0.098     -1.300      0.194      -0.321       0.065
I(priGPA ** 2)  0.2959      0.101      2.928      0.004       0.097       0.494
```

```
I(ACT ** 2)              0.0045      0.002      2.083      0.038       0.000    0.009
I(DpriGPA * atndrte)     0.0056      0.004      1.294      0.196      -0.003    0.014
==============================================================================
```

The effect of $priGPA$ on $stndfnl$ can be written as $\Delta\widehat{stndfnl} = (-1.63 + 0.2959\ priGPA + 0.0056\ atndrte)\Delta priGPA$. We can calculate the effect at the average values of $priGPA = 2.59$ and $attndrte = 82$.

```
df['Datndrte'] = df.atndrte - df.atndrte.mean()
model = smf.ols(formula='stndfnl~atndrte+priGPA+ACT+I(DpriGPA**2)'+
                        'I(ACT**2)+I(priGPA * Datndrte)', data=df)
results = model.fit()
results.summary()
==============================================================================
                          coef    std err          t      P>|t|      [0.025    0.975]
------------------------------------------------------------------------------
Intercept                0.0703      1.445      0.049      0.961      -2.768    2.908
atndrte                 -0.0067      0.010     -0.656      0.512      -0.027    0.013
priGPA                   0.3588      0.078      4.612      0.000       0.206    0.511
ACT                     -0.1280      0.098     -1.300      0.194      -0.321    0.065
I(DpriGPA ** 2)          0.2959      0.101      2.928      0.004       0.097    0.494
I(ACT ** 2)              0.0045      0.002      2.083      0.038       0.000    0.009
I(priGPA * Datndrte)     0.0056      0.004      1.294      0.196      -0.003    0.014
==============================================================================
```

We see a t-stats of 4.612 showing high statistical significance of the effect of $priGPA$ on $stndfnl$ at average values of $priGPA$ and $atndrte$. □

6.2 Goodness of fit and selection of regressors

Choosing set of explanatory variables based on the size of the R-squared can lead to nonsensical models. R^2 is simply an estimate of how much variation in y is explained by $x_1, \ldots, x_k$ in the population. A small R-squared does imply that the error variance is large relative to the variance of y. The large error variance can, however, be offset by a large sample size to precisely estimate the partial effects even though we have not controlled for many unobserved factors. In other words poor explanatory power has nothing to do with unbiased estimation of β_j. On the other hand, the relative change in the R-squared when variables are added to an equation is very useful: the F statistic for testing the joint significance crucially depends on the difference in R-squareds between the unrestricted and restricted models. Also, low R-squared also means prediction is difficult.

The population R-squared is defined as $\rho^2 = 1 - \sigma_u^2/\sigma_y^2$, which we estimate using $R^2 = 1 - SSR/SST$, which is biased. The adjusted R-squared uses the unbiased estimators of the variance to give

$$\overline{R}^2 = 1 - \frac{SSR/(n-k-1)}{SST/(n-1)}.$$

However, it is not unbiased because the ratio of two unbiased estimators is not an unbiased estimator. We know that R^2 can never fall when a new independent variable is added to a regression equation because SSR never goes up as more independent variables are added. $\overline{R}^2$, on the other hand, imposes a penalty for adding additional independent variables to a model, hence it can go up or down when a new independent variable is added to a regression. Algebraically, if we add a new independent variable to a regression equation, $\overline{R}^2$ increases only if t-statistic on the new variable is greater than one in absolute value. Similarly, $\overline{R}^2$ increases when a group of variables is added to a regression only if the F statistic for join significance of the new variables is greater than unity. $\overline{R}^2$ can be negative and indicates a very poor model fit relative to the number of degrees of freedom. It is important to remember that it is R^2 that is used in F statistic and not $\overline{R}^2$.

F statistic only allow to compare nested models. $\overline{R}^2$ can be used to compare non-nested models and also models when the variables represent different functional forms. Everything else being equal, simpler models are better. We cannot use $\overline{R}^2$ (or R^2)to choose between different functional forms for the dependent variable.

Example 6.4. Consider two estimated models relating CEO compensation to firm performance:

$$\widehat{salary} = \underset{(223.905)}{830.6313} + \underset{(0.009)}{0.0163}\ sales + \underset{(11.077)}{19.6310}\ roe$$

with 206 degrees of freedom, $R^2 = 0.029$, and $\overline{R}^2 = 0.02$.

$$\widehat{\log(salary)} = \underset{(0.294)}{4.3622} + \underset{(0.033)}{0.2751}\ \log(sales) + \underset{(0.004)}{0.0179}\ roe$$

with 206 degrees of freedom, $R^2 = 0.282$, and $\overline{R}^2 = 0.275$.

```
df = woo.dataWoo('ceosal1')
model = smf.ols(formula='salary~sales+roe', data=df)
results = model.fit()
results.summary()
```

```
==============================================================================
Dep. Variable:                 salary   R-squared:                       0.029
Model:                            OLS   Adj. R-squared:                  0.020
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    830.6313    223.905      3.710      0.000     389.192    1272.070
sales          0.0163      0.009      1.842      0.067      -0.001       0.034
roe           19.6310     11.077      1.772      0.078      -2.207      41.469
==============================================================================
```

```
model = smf.ols(formula='lsalary~lsales+roe',data=df)
results = model.fit()
results.summary()
```

```
==============================================================================
Dep. Variable:                lsalary   R-squared:                       0.282
Model:                            OLS   Adj. R-squared:                  0.275
==============================================================================
```

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	4.3622	0.294	14.843	0.000	3.783	4.942
lsales	0.2751	0.033	8.272	0.000	0.210	0.341
roe	0.0179	0.004	4.519	0.000	0.010	0.026

Can we say which model is better than the other? The R^2 of the first model is 2.9% while of the second is 28.2%. The adjusted R-squareds are 2.0% and 27.5% respectively. But we can't use them to compare the model since the total sum of squares for the two moles are different. We do note that the second model has much more statistically significant coefficients than the first, and that makes it the right criteria to choose the second model as the better one. □

In many cases we are worried about omitting important factors from a model that might be correlated with the independent variables, in hope to avoid biases that might arise by leaving out an important explanatory variable. If we overemphasize goodness-of-fit, we might control for factors that should not be controlled for. If we remember that different models serve different purposes, and we focus on the ceteris paribus interpretation of regression, then we will not include the wrong factors in a regression model.

- To assess the impact of state beer taxes on traffic fatalities we can model fatalities as a function of several factors, including the beer tax, like total miles driven, percentage of the state population that is male, percentage of the population between ages 16 and 21, etc. The idea is that a higher tax on beer will reduce alcohol consumption, and likewise drunk driving, resulting in fewer traffic fatalities. We should not include the variable measuring per capita beer consumption unless we want to test for some sort of indirect effect of beer taxes.

- We want to estimate the effects of pesticide usage among farmers on family health expenditures. We should include pesticide usage amounts, but not the number of doctor visits as an explanatory variable. If we include it we are only measuring the effects of pesticide use on health expenditures than than doctor visits, which makes little sense and leads to over controlling.

- To study the effect of high school quality on subsequent earnings, if better school quality results in more education, then controlling for education in the regression along with measures of quality will underestimate the returns to quality.

- If our goal is to estimate a hedonic price model, which allows us to obtain the marginal values of various housing attributes, then we should not include assessed house price, *assess*, to explaining the real house price. However, if want to predict the real price, it makes sense to include the assessed house price as a factor.

Adding a new independent variable to a regression can exacerbate the multicollinearity problem. But it also reduces the error variance. In the case where independent variables that affect y are uncorrelated with all of the independent variables of interest should always

be included. This is because, this variable does not induce any multicollinearity in the population, but will reduce the error variance at least in large sample case.

- Estimating the individual demand for beer as a function of the average county beer price is reasonable. It may be also reasonable to assume that individual characteristics like age and amount of education may be uncorrelated with county-level prices and may be indicator of demand, and hence if included will reduce the standard error of the price coefficient in large samples.

- In order to understand the effect of random computer grant on college GPA, measures like high school grade point average and rank, SAT and ACT scores and family background variables are good candidates. Because the grant amount are randomly assigned, all additional control variables are uncorrelated with the grant amount; avoiding multicollinearity. But adding the extra controls might significantly reduce the error variance, leading to a more precise estimate of the grant effect. We get unbiased estimates even without the extra controls, it is that including them gives smaller sampling variance.

6.3 Prediction and Residual analysis

We now look at the confidence intervals for a prediction from the OLS regression line. For the estimated equation $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \ldots + \hat{\beta}_k$, we want to estimate the expected value of y given the particular values for the explanatory variables, i.e. $\theta_0 = E(y|x_1 = c_1, \ldots, x_k = c_k)$. Hence, the estimator for θ_0 is $\hat{\theta}_0 = \hat{\beta}_0 + \hat{\beta}_1 c_1 + \ldots + \hat{\beta}_k c_k$. To get the standard deviation for $\hat{\theta}_0$, we write $\beta_0 = \theta_0 - \beta_1 c_1 - \ldots - \beta_k c_k$ and plug this into the equation $y = \beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k + u$ to obtain $y = \theta_0 + \beta_1(x_1 - c_1) + \ldots + \beta_k(x_k - c_k) + u$. This regression can be done and the standard deviation of the intercept will give us the required value. The variance of this prediction is smallest at the mean values of x_j, i.e. as values of c_j get farther away from $\bar{x}_j$, $Var(\hat{y})$ gets larger and larger.

Example 6.5. To estimate the collage GPA we use some relevant factors in the following estimation

$$\widehat{colgpa} = \underset{(0.075)}{1.4927} + \underset{(0.00007)}{0.00149}\ sat - \underset{(0.00056)}{0.01386}\ hsperc - \underset{(0.0165)}{0.06088}\ hsize + \underset{(0.00227)}{0.00546}\ hsize^2$$

with 4132 degrees of freedom, $R^2 = 0.278$ and $\overline{R}^2 = 0.277$. For $sat = 1200$, $hsperc = 30$ and $hsize = 5$ we can easily do the mean prediction as $\widehat{colgpa} = 2.70$.

```
df = woo.dataWoo('gpa2')
model = smf.ols(formula='colgpa~sat+hsperc+hsize+I(hsize**2)', data=df)
results = model.fit()
results.summary()
==============================================================================
Dep. Variable:                 colgpa   R-squared:                       0.278
Model:                            OLS   Adj. R-squared:                  0.277
==============================================================================
                    coef    std err          t      P>|t|      [0.025  0.975]
```

```
------------------------------------------------------------------------------
Intercept         1.4927      0.075     19.812      0.000       1.345   1.640
sat               0.0015   6.52e-05     22.886      0.000       0.001   0.002
hsperc           -0.0139      0.001    -24.698      0.000      -0.015  -0.013
hsize            -0.0609      0.017     -3.690      0.000      -0.093  -0.029
I(hsize ** 2)     0.0055      0.002      2.406      0.016       0.001   0.010
==============================================================================
pred_x = pd.Series([1200, 30, 5, 25],index=['sat', 'hsperc', 'hsize', 'I(hsize**2)'])
results.get_prediction(pred_x).summary_frame()
>>    mean   mean_se  mean_ci_lower  mean_ci_upper  obs_ci_lower  obs_ci_upper
0  2.700075  0.019878       2.661104       2.739047      1.601749      3.798402
np.sqrt(results.mse_resid)
>> 0.5598638443489726

df['Dsat'] = df.sat - 1200
df['Dhsperc'] = df.hsperc - 30
df['Dhsize'] = df.hsize - 5
model = smf.ols(formula='colgpa~Dsat+Dhsperc+Dhsize+I(hsize**2-25)', data=df)
results = model.fit()
results.summary()
==============================================================================
Dep. Variable:                 colgpa   R-squared:                       0.278
Model:                            OLS   Adj. R-squared:                  0.277
==============================================================================
                          coef    std err          t      P>|t|      [0.025  0.975]
------------------------------------------------------------------------------
Intercept              2.7001      0.020    135.833      0.000       2.661   2.739
Dsat                   0.0015   6.52e-05     22.886      0.000       0.001   0.002
Dhsperc               -0.0139      0.001    -24.698      0.000      -0.015  -0.013
Dhsize                -0.0609      0.017     -3.690      0.000      -0.093  -0.029
I(hsize ** 2 - 25)     0.0055      0.002      2.406      0.016       0.001   0.010
==============================================================================
```

To calculate the standard deviation of this estimate we do the following regression with the mean of variables removed to get

$$\widehat{colgpa} = \underset{(0.020)}{2.7} + \underset{(0.00007)}{0.00149}\ (sat - 1200) - \underset{(0.00056)}{0.01386}\ (hsperc - 30)$$
$$- \underset{(0.0165)}{0.06088}\ (hsize - 5) + \underset{(0.00227)}{0.00546}\ (hsize^2 - 25)$$

with 4132 degrees of freedom, $R^2 = 0.277$ and $\overline{R}^2 = 0.277$. The only difference between this regression and the previous one is the intercept, which is the prediction we want, along with its standard error, 0.020. A 95% confidence interval can be given as $[2.66, 2.739]$ as shown in the table. This has very high statistical significance. □

The previous method allows us to put a confidence interval around the OLS estimate of $E(y|x_1, \ldots, x_k)$. A confidence interval for the average person in the sub-population is not the same as a confidence interval for a particular unit from the population. For a prediction of y^0 the expected value of the prediction error $\hat{e}^0 = y^0 - \hat{y}^0$, is zero, while the variance of the prediction error is $Var(\hat{e}^0) = Var(\hat{y}^0) + Var(u^0) = Var(\hat{y}^0) + \sigma^2$. There are two sources of variation in $\hat{e}^0$:

- Sampling error in $\hat{y}^0$ which arises because we have estimated β_j. Each $\hat{\beta}_j$ has a variance proportional to $1/n$, $Var(\hat{y}^0)$ is proportional to $1/n$.
- Variance of error in the population σ^2, and it does not change with the sample size, and generally is the dominant term.

Under the classical linear model assumptions, the $\hat{\beta}_j$ and u^0 are normally distributed, and so $\hat{e}^0$ is also normally distributed. Using the unbaised estimators of the two parts, we can calculate $se(\hat{e}^0) = \sqrt{se(\hat{y}^0)^2 + \hat{\sigma}^2}$. For large samples this gives a confidence interval of $\hat{y}^0 \pm 2se(\hat{e}^0)$.

Example 6.6. In the previous example we found the confidence intervals for the average college GPA among all students with the particular characteristics $sat = 1200$, $hsperc = 30$, and $hsize = 5$. Now we want a 95% confidence interval for any particular student with these characteristics. We have $\hat{y}^0 = 2.70$, and $se(\hat{y}^0) = 0.020$. We also have $\hat{\sigma} = 0.560$. And so we have $se(\hat{e}^0) = \sqrt{(0.020)^2 + (0.560)^2} \approx 0.560$. Virtually all of the variation in $\hat{e}^0$ comes from $\hat{\sigma}$. The 95% confidence interval is $2.70 \pm 2 \times 0.560 = [1.60, 3.80]$, which is the same as shown in the printouts. Evidently, the unobserved characteristics that affect college GPA vary widely among individuals with the same observed SAT score and high school rank. □

Residuals $\hat{u}_i = y_i - \hat{y}_i$ can tell us about the deviation from the expected value due to idiosyncratic content of that observation.

When we want to predict y in the regression $\log(y) = \beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k + u$, we can use the predicted log value $\widehat{\log(y)} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \ldots + \hat{\beta}_k x_k$. But using $\hat{y} = \exp(\widehat{log(y)})$ does not work and will systematically underestimate the expected value of y. In fact, $y = \exp(\beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k + u)$, hence $E(y|\boldsymbol{x}) = \exp(u)\exp(\beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k)$. Now, if $u \sim \mathcal{N}(0, \sigma^2)$, $\exp(u) = \exp(\sigma^2/2)$. Hence for a predicted value of $\widehat{\log(y)}$ the predicted value of y is, $\hat{y} = exp(\hat{\sigma}^2/2)\exp(\widehat{\log(y)})$. This prediction is not unbiased, but is consistent.

This, however, relies on the normality of the error terms. It is useful to have a prediction that does not rely on normality. If we just assume that u is independent of the explanatory variables, then we have $E(y|\boldsymbol{x}) = \alpha_0 \exp(\beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k)$, where $\alpha_0 = E(e^u)$. Given an estimate $\hat{\alpha}_0$ we can predict y as $\hat{y} = \hat{\alpha}_0 \exp(\widehat{\log(y)})$. There are two approaches for estimating α_0 without the normality assumption.

1. Smearing estimate: Using the method of moments estimator we calculate $\hat{\alpha}_0 = \frac{1}{n}\sum_{i=1}^{n} \exp(\hat{u}_i)$. This again is consistent but not unbiased because we have replaced u_i with $\hat{u}_i$ inside a nonlinear function. Since OLS residuals have a zero sample average, we necessarily have $\hat{\alpha}_0 > 1$.
2. OLS estimate: We do a simple regression through the origin by defining $m_i = \exp(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_k x_{ik})$ and hence $E(y_i|m_i) = \alpha_0 m_i$. We replace the β_j with their estimates and obtain $\hat{m}_i = \exp(\widehat{\log(y_i)})$, where $\widehat{\log(y_i)}$ are the fitted values from the regression $log(y_i)$ on $x_{i1}, \ldots, x_{ik}$ with an intercept. Then the OLS slope estimate from the simple

regression of y_i on $\hat{m}_i$ give $\check{\alpha}_0$. Again, it is consistent but not unbiased, but is not guaranteed to be greater than one. If $\check{\alpha}_0$ is much less than one, it is likely that the assumption of independence between u and the x_j is violated.

Example 6.7. The estimated model of interest is

$$\log(salary) = \underset{(0.257)}{4.5038} + \underset{(0.039)}{0.1629}\ \log(sales) + \underset{(0.050)}{0.1092}\ \log(mktval) + \underset{(0.005)}{0.0117}\ ceoten$$

with 173 degrees of freedom, $R^{=}0.318$, and $\overline{R}^2 = 0.306$.

```
df = woo.dataWoo('ceosal2')
model = smf.ols(formula='lsalary~lsales+lmktval+ceoten', data=df)
results = model.fit()
print(results.summary())
==============================================================================
Dep. Variable:                lsalary   R-squared:                       0.318
Df Residuals:                     173   Adj. R-squared:                  0.306
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      4.5038      0.257     17.509      0.000       3.996       5.012
lsales         0.1629      0.039      4.150      0.000       0.085       0.240
lmktval        0.1092      0.050      2.203      0.029       0.011       0.207
ceoten         0.0117      0.005      2.198      0.029       0.001       0.022
==============================================================================
pred_x = pd.Series([np.log(5000), np.log(10000), 10],
                   index=['lsales', 'lmktval', 'ceoten'])
pred_logy = results.get_prediction(pred_x).summary_frame()
print(pred_logy)
>>      mean   mean_se  mean_ci_lower  mean_ci_upper  obs_ci_lower  obs_ci_upper
0  7.014077  0.074556       6.866921       7.161233      6.006969      8.021185
```

we want to find the prediction of *salary* for given value of $sales = 5000$, $nktval = 10000$, and $ceoten = 10$. The expected value of $\log(slary)$ can be calculated to be 7.0141. The naive, biased, value of $\hat{y}_{naive} = \exp(\widehat{\log(y)})$ is 1,110.983. We now calculate the α_0 using the normality assumption, smearing estimate and the OLS estimate as 1.136, 1.136 and 1.117 respectively. We use this to estimate the predicted value for $\hat{y}$ as 1263.2877, 1263.0595, and 1242.1453 respectively.

```
alpha_norm = np.exp(results.mse_resid/2)
alpha_smearing = np.exp(results.resid).mean()
adf = pd.DataFrame({'y': df.salary, 'm': np.exp(df.lsalary-results.resid)})
alpha_ols = smf.ols(formula='y~m-1', data=adf).fit().params.m
print(alpha_norm, alpha_smearing, alpha_ols)
>> 1.1358665799771173 1.1356613266630744 1.1168566605074874
pred_y = np.array([alpha_norm, alpha_smearing, alpha_ols])*
                    np.exp(pred_logy.loc[0, 'mean'])
print(pred_y)
>> [1263.28773937 1263.0594608  1242.14529305]
```

□

We can use the previous method of obtaining predictions to determine how well the model with $\log(y)$ as the dependent variable does against the one with y as the dependent variable. We need to find goodness-of-fit measure in $\log(y)$ model that can be compared against. Recall that the usual R-squared is simply the square of the correlation between y_i and $\hat{y}_i$. It then makes sense to use the fitted values from $\hat{y}_i = \hat{\alpha}_0 m_i$ and calculate the squared correlation between y_i and $\hat{m}_i$. We can compare this directly with the R-squared from the estimated equation $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \ldots + \hat{\beta}_k x_k$. This is independent of $\hat{\alpha}_0$.

Example 6.8. Following the previous example the suqare of the correlation between $salary_i$ and $\hat{m}_i$ is 0.2431 and this is the measure of how well log model explains the variation in $salary$.

```
adf.corr().loc['y', 'm']**2
>> 0.24308077958679802
```

As a competing linear model, we fit the following equation

$$\underset{}{salary} = \underset{(65.237)}{613.4361} + \underset{(0.010)}{0.019}\ sales + \underset{(0.009)}{0.0234}\ mktval + \underset{(5.618)}{12.7034}\ ceoten$$

with 173 degrees of freedom, $R^2 = 0.201$ and $\overline{R}^2 = 0.187$. Thus, the log model explains more of the variation in $salary$, ans so we prefer it to this fitted model. The log model is also preferred because it seems more realistic and its parameter are easier to interpret. We now explore how to estimate the standard deviation of this predicted $\hat{y}$ and to construct its confidence intervals.

```
==============================================================================
Dep. Variable:                 salary   R-squared:                       0.201
Df Residuals:                     173   Adj. R-squared:                  0.187
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    613.4361     65.237      9.403      0.000     484.673     742.199
sales          0.0190      0.010      1.891      0.060      -0.001       0.039
mktval         0.0234      0.009      2.468      0.015       0.005       0.042
ceoten        12.7034      5.618      2.261      0.025       1.615      23.792
==============================================================================
```

The confidence interval on the prediction $\log(y^0)$ is simply $\widehat{\log(y^0)} \pm 2se(\hat{e}^0)$. We can then write $P[\widehat{\log(y^0)} - 2se(\hat{e}^0) \leq \log(y^0) \leq \widehat{\log(y^0)} + 2se(\hat{e}^0)] = 0.95$. Since exponential is a strictly increasing function, it is also true that $P[\exp(\widehat{\log(y^0)} - 2se(\hat{e}^0)) \leq y^0 \leq \exp(\widehat{\log(y^0)} + 2se(\hat{e}^0))] = 0.95$. For our CEO salary example, this can be calculated to $P[400.85 \leq y^0 \leq 3085.80] = 0.95$, with the expected value of 1112.18 which is clearly not symmetric, as mentioned before.

```
se = np.sqrt(results.mse_resid + pred_logy.loc[0, 'mean_se']**2)
print(se)
>> 0.5102453035512231
ci = [np.exp(pred_logy.loc[0, 'mean'])*np.exp(-2*se),
      np.exp(pred_logy.loc[0, 'mean']),
      np.exp(pred_logy.loc[0, 'mean'])*np.exp(2*se)]
print(ci)
>> [400.8496603826342, 1112.1796887381379, 3085.8044356603095]
```

□

7 Dummy Variables

Quantitative variables also play an important role in multiple regression. They often come in form of binary information, encapsulated as dummy variables coded as 1 or 0. This leads to regression models where the parameters have very natural interpretations.

7.1 Single and multiple dummy variables

For the simple model of hourly wage determination

$$wage = \beta_0 + \delta_0 female + \beta_1 educ + u,$$

we use δ_0 as the parameter on *female* in order to highlight the interpretation of the parameters multiplying dummy variables. Here only two observed factors affect *wage*: gender and education. Because $female = 1$ when the person is female, and $female = 0$ when the person is male, δ_0 is the difference in hourly wage between females and males, given the same amount of education. Hence, if $\delta_0 < 0$, then for the same level of other factors, women earn less than men on average. Using $E(u|female, educ) = 0$ we see

$$\delta_0 = E(wage|female, educ) - E(wage|male, educ).$$

In this equation the intercept for the *male* is β_0, while for the $female$ is $\beta_0 + \delta_0$. This graphically results in intercept shift between males and females. To avoid perfect collinearity or **dummy variable trap**, we remove one of the variables, e.g. *male* here, which serve as the **base group**. One could also include *male* instead of $female$, or drop the intercept and include both. But dropping intercept creates problems in interpreting R^2.

Nothing much changes when more exploratory variables are involved. Taking males as the base group, a model that controls for experience and tenure in addition to education is

$$wage = \beta_0 + \delta_0 female + \beta_1 educ + \beta_2 exper + \beta_3 tenure + u.$$

The null hypothesis $H_0 : \delta_0 = 0$ is unbiased wages. The alternative is discrimination against women $H_1 : \delta_0 < 0$.

Example 7.1. The 1974 data produces the following model

$$wage = \underset{(0.725)}{-1.568} - \underset{(0.265)}{1.8109} female + \underset{(0.049)}{0.5715} educ + \underset{(0.012)}{0.0254} exper + \underset{(0.021)}{0.141} tenure.$$

The $\bar{R}^2$ for the regression is 0.359, with very significant F-statistics.

```
df = woo.dataWoo('wage1')
reg = smf.ols(formula='wage ~ female + educ + exper + tenure', data=df)
results = reg.fit()
results.summary()
```

Dep. Variable:		wage	R-squared:			0.364
Df Residuals:		521	Adj. R-squared:			0.359
	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-1.5679	0.725	-2.164	0.031	-2.991	-0.145
female	-1.8109	0.265	-6.838	0.000	-2.331	-1.291
educ	0.5715	0.049	11.584	0.000	0.475	0.668
exper	0.0254	0.012	2.195	0.029	0.003	0.048
tenure	0.1410	0.021	6.663	0.000	0.099	0.183

The coefficient of $female$ suggest women earning \$1.811 less per hour than a man, given everything else equal. Because we performed multiple regression and controlled for *educ*, *exper*, and *tenure*, the \$1.81 wage differential cannot be explained by different average levels of education, experience, or tenure between men and women. We can conclude that the differential of \$1.81 is due to gender or factors associated with gender that we have not controlled for in the regression.

If we simply run the regression of $wage$ against $female$ we get

$$wage = \underset{(0.21)}{7.10} - \underset{(0.30)}{2.51} female.$$

with $R^2 = 0.116$. The intercept here is simply the average wage for men, and the coefficient on $female$ is the difference in the average wage between women and men, showing women earning \$2.51 less than men per hour. This is a simple comparison-of-means test. This wage difference is larger because we did not control for other relevant factors. For the t test to be valid, we must assume that the homoskedasticity assumption holds, which means that that population variance in wages for men is the same as that for women. However, nothing in the model implies causality. □

Example 7.2. When we use $\log(y)$ as dependent variable in a model, the coefficient on a dummy variable is interpreted as the percentage difference in y, holding all other factors

fixed. If we refit the model with $\log(wage)$ and quadratic terms for $exper$ and $tenure$ added, we get

$$\log(wage) = \underset{(0.099)}{0.417} - \underset{(0.036)}{0.297} female + \underset{(0.007)}{0.080} educ + \underset{(0.005)}{0.029} exper - \underset{(0.0001)}{0.00058} exper^2 + \underset{(0.007)}{0.032} tenure - \underset{(0.00023)}{0.00059} tenure^2.$$

The $\bar{R}^2$ for the regression is 0.434, with very significant F-statistics.

```
import numpy as np
import statsmodels.formula.api as smf
import wooldridge as woo

df = woo.dataWoo('wage1')
reg = smf.ols(formula='np.log(wage) ~ female + educ + exper'
                      + 'I(exper**2)+ tenure + I(tenure**2)', data=df)
results = reg.fit()
results.summary()
```

Dep. Variable:	np.log(wage)	R-squared:	0.441
Df Residuals:	519	Adj. R-squared:	0.434

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.4167	0.099	4.212	0.000	0.222	0.611
female	-0.2965	0.036	-8.281	0.000	-0.367	-0.226
educ	0.0802	0.007	11.868	0.000	0.067	0.093
exper	0.0294	0.005	5.916	0.000	0.020	0.039
I(exper ** 2)	-0.0006	0.000	-5.431	0.000	-0.001	-0.000
tenure	0.0317	0.007	4.633	0.000	0.018	0.045
I(tenure ** 2)	-0.0006	0.000	-2.493	0.013	-0.001	-0.000

We can use this result to write

$$\log(wage_F) - \log(wage_M) = -0.297$$

giving, $wage_f/wage_M - 1 = \exp(-0.297) - 1 \approx -0.257$, showing women being underpaid by 25.6% below a comparable man's wage. This however is a bit erroneous as if we change the estimation to calculating the percentage by which a man's wage exceeds a comparable woman's wage. This estimate is $\exp(-\beta_1) - 1 = \exp(0.297) - 1 \approx 0.346$. The approximation based on $\beta_1 = 0.297$ sits in between and might be more sensible. □

Example 7.3. We now introduce a new dummy variable $marr$, for marital status and interact with the dummy variable $female$. We choose single males as the base case represented by the intercept and do the multiple regression.

```
reg = smf.ols(formula='np.log(wage) ~ marrmale + marrfem + singfem +
      educ + exper + I(exper**2)+ tenure + I(tenure**2)', data=df)
```

```
results = reg.fit()
results.summary()
==============================================================================
Dep. Variable:           np.log(wage)   R-squared:                       0.461
Df Residuals:                     517   Adj. R-squared:                  0.453
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept          0.3214      0.100      3.213      0.001       0.125       0.518
marrmale           0.2127      0.055      3.842      0.000       0.104       0.321
marrfem           -0.1983      0.058     -3.428      0.001      -0.312      -0.085
singfem           -0.1104      0.056     -1.980      0.048      -0.220      -0.001
educ               0.0789      0.007     11.787      0.000       0.066       0.092
exper              0.0268      0.005      5.112      0.000       0.017       0.037
I(exper ** 2)     -0.0005      0.000     -4.847      0.000      -0.001      -0.000
tenure             0.0291      0.007      4.302      0.000       0.016       0.042
I(tenure ** 2)    -0.0005      0.000     -2.306      0.022      -0.001   -7.89e-05
==============================================================================
```

$$\begin{aligned}\log(wage) = \underset{(0.100)}{0.321} - \underset{(0.055)}{0.213}marrmale - \underset{(0.058)}{0.198}marrfem \\ - \underset{(0.056)}{0.110}singfem + \underset{(0.007)}{0.079}educ + \underset{(0.007)}{0.079}exper - \underset{(0.00011)}{0.00054}exper^2 \\ + \underset{(0.007)}{0.029}tenure - \underset{(0.00023)}{0.00053}tenure^2 + \varepsilon.\end{aligned}$$

with R^2 of 0.461. All the coefficients seem to be significant at 5% significance. Against the base group of single males, married men are estimated to earn about 21.3% more, holding all other factors equal. A married woman, on the other hand, earns a predicted 19.8% less than a single man with the same levels of the other variables. To estimate the proportionate difference between single and married women we can use $-0.110-(-0.198) = 0.088$. To get the significance we can simply make the married woman as base and rerun the regression, to get a t-stats of 1.679 for $singfem$ showing only marginal evidence against the hypothesis that married women are discriminated against single women. □

Including all dummy variables with the intercept will result in dummy variable trap. An alternative is to include all the dummy variables and exclude an overall intercept. It has two practical drawbacks. First, it makes it more cumbersome to test for differences relative to a base group. Second, regression packages usually change the way R-squared is computed when an overall intercept is not included. Usually SST is replaced with total sum of squared that does not center y_i about its mean. The resulting R-squared is sometimes called the uncentered R-squared. It is rarely suitable as a goodness of fit measure, and is always greater than the correct R-squared with equality only if $\bar{y} = 0$.

An ordinal variable can be directly used in the regression if marginal changes have the same uniform effect. Otherwise, if the variable takes relatively few values, we can use one-hot-encoding to create several dummy variables in include them, and as usual dropping out one of them.

Example 7.4. We want to measure the effect of looks on log wages. The data at hand gives 5 categories of looks which we could convert into 5 dummy variables but the extreme categories have very few entries so we convert them into three categories - average, below average and above average. We create the dummy variables with the base group as *average*. We control for education, experience, marital status and race as well. We fit it separately for men and women.

```
df = woo.dataWoo('beauty')
model = smf.ols(formula='lwage~belavg+abvavg+educ'
        +'exper+expersq+married+black', data=df[df.female == 0])
results = model.fit()
results.summary()
```

Dep. Variable:	lwage	R-squared:				0.235
Df Residuals:	816	Adj. R-squared:				0.228
	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.5755	0.102	5.631	0.000	0.375	0.776
belavg	-0.1802	0.053	-3.377	0.001	-0.285	-0.075
abvavg	-0.0332	0.038	-0.873	0.383	-0.108	0.042
educ	0.0580	0.007	8.866	0.000	0.045	0.071
exper	0.0466	0.006	7.973	0.000	0.035	0.058
expersq	-0.0007	0.000	-5.843	0.000	-0.001	-0.000
married	0.0618	0.045	1.382	0.167	-0.026	0.150
black	-0.2479	0.076	-3.246	0.001	-0.398	-0.098

We see that below average men are estimate to earn 18.02% less than an average looking man who is the same in other respects. The effect is statistically different from zero, with $t = -3.377$. Men above average don't show statistically different wage than average men as the t-statistic is only -0.873.

```
model = smf.ols(formula='lwage~belavg+abvavg+educ+'
        +'exper+expersq+married+black', data=df[df.female == 1])
results = model.fit()
results.summary()
```

Dep. Variable:	lwage	R-squared:				0.224
Df Residuals:	428	Adj. R-squared:				0.212
	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.0204	0.130	0.156	0.876	-0.236	0.277
belavg	-0.1184	0.068	-1.733	0.084	-0.253	0.016
abvavg	0.0450	0.050	0.892	0.373	-0.054	0.144
educ	0.0800	0.009	8.787	0.000	0.062	0.098
exper	0.0320	0.007	4.350	0.000	0.018	0.047
expersq	-0.0005	0.000	-2.997	0.003	-0.001	-0.000
married	-0.0588	0.046	-1.290	0.198	-0.148	0.031

```
black          0.1477       0.071       2.077       0.038       0.008       0.288
==============================================================================
```

A women with below average looks earns about 11.84% less than an otherwise comparable average-looking woman, with $t = -1.733$. As was the case for men, the estimate on above average is not statistically different from zero. □

In some cases, the ordinal variable takes on too many values so that a dummy variable cannot be included for each value. In these cases we can break the variable into categories.

Example 7.5. We want to predict the log median starting salary for law school graduates based on the ranking of the law school, which ranges from 1 to 175. We break the rank into 6 categories and choose the schools ranked below 100 to be the base group. We also control for the factors like LSAT score, GPA, log(*libvol*) and log(*cost*).

```
df = woo.dataWoo('lawsch85')
cutpts = [0,10,25,40,60,100,df['rank'].max()]
df['rc'] = pd.cut(df['rank'], bins=cutpts,
           labels=['top10','11_25','26_40','41_60','61_100','base'])
model = smf.ols(formula='lsalary~C(rc, Treatment("base"))+LSAT+GPA+llibvol+lcost',
                data=df)
results = model.fit()
```

We immediately see that all the dummy variables defining the different ranks are very statistically significant. The different between a top 10 school and a below 100 school is quite large, with the predicted salary $e^{0.6996} - 1 \approx 1.0129$, i.e. 100% higher.

```
==============================================================================
Dep. Variable:                lsalary   R-squared:                       0.911
Df Residuals:                     126   Adj. R-squared:                  0.905
==============================================================================
               coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     9.1653      0.411     22.277      0.000       8.351       9.979
T.top10       0.6996      0.053     13.078      0.000       0.594       0.805
T.11_25       0.5935      0.039     15.049      0.000       0.515       0.672
T.26_40       0.3751      0.034     11.005      0.000       0.308       0.443
T.41_60       0.2628      0.028      9.399      0.000       0.207       0.318
T.61_100      0.1316      0.021      6.254      0.000       0.090       0.173
LSAT          0.0057      0.003      1.858      0.066      -0.000       0.012
GPA           0.0137      0.074      0.185      0.854      -0.133       0.161
llibvol       0.0364      0.026      1.398      0.165      -0.015       0.088
lcost         0.0008      0.025      0.033      0.973      -0.049       0.051
==============================================================================
```

As an indicator of whether breaking the rank into different groups is an improvement, we can compare the adjusted R-squared between the previous regression and one with including

rank as a single variable. We see that the adjusted R-squared decreases from 0.905 to 0.836 and hence categorization has clearly helped.

```
model = smf.ols(formula='lsalary~rank+LSAT+GPA+llibvol+lcost', data=df)
results = model.fit()
results.summary()
```

Dep. Variable:	lsalary	R-squared:	0.842
Df Residuals:	130	Adj. R-squared:	0.836

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	8.3432	0.533	15.667	0.000	7.290	9.397
rank	-0.0033	0.000	-9.541	0.000	-0.004	-0.003
LSAT	0.0047	0.004	1.171	0.244	-0.003	0.013
GPA	0.2475	0.090	2.749	0.007	0.069	0.426
llibvol	0.0950	0.033	2.857	0.005	0.029	0.161
lcost	0.0376	0.032	1.170	0.244	-0.026	0.101

Interestingly, once the rank is put into the given categories, all of the other variables become insignificant. We can do a joint F-test on the remaining variables to get a p-value of 0.055 making it barely significant.

```
ftest = results.f_test(['LSAT=0','GPA=0', 'llibvol=0', 'lcost=0'])
print(ftest.statistic[0][0], ftest.pvalue)
>> 2.385316132354486 0.05470437645684762
```

As a final note, in deriving the properties of ordinary least squares we assumed that we had a random sample. The current application violates that assumption because the way rank is defined: a school's rank necessarily depends on the rank of the other schools in the sample, and so the data cannot represent independent draws from the population of all law schools. This does not cause any serious problems provided the error term is uncorrelated with the explanatory variables. □

7.2 Interactions involving dummy variables

Interacting dummy variables with explanatory variables allow for a difference in slopes.

Example 7.6. We estimate the effect interaction of gender and education on log wages.

```
df = woo.data('wage1')
model = smf.ols(formula='lwage~female+educ+I(female*educ)+'
                        +'exper+I(exper**2)+tenure+I(tenure**2)', data=df)
results = model.fit()
results.summary2()
```

```
=================================================================
Df Residuals:       518              Adj. R-squared:      0.433
Dependent Variable: lwage            R-squared:           0.441
-----------------------------------------------------------------
                   Coef.  Std.Err.    t    P>|t|   [0.025  0.975]
-----------------------------------------------------------------
Intercept          0.3888   0.1187  3.2759 0.0011  0.1556  0.6220
female            -0.2268   0.1675 -1.3536 0.1764 -0.5559  0.1024
educ               0.0824   0.0085  9.7249 0.0000  0.0657  0.0990
I(female * educ)  -0.0056   0.0131 -0.4260 0.6703 -0.0312  0.0201
exper              0.0293   0.0050  5.8860 0.0000  0.0195  0.0391
I(exper ** 2)     -0.0006   0.0001 -5.3978 0.0000 -0.0008 -0.0004
tenure             0.0319   0.0069  4.6470 0.0000  0.0184  0.0454
I(tenure ** 2)    -0.0006   0.0002 -2.5089 0.0124 -0.0011 -0.0001
-----------------------------------------------------------------
```

The estimate return to education for men in this equation is 8.2%, while for women it is $0.0824 - 0.0056$ =7.6%. The difference of 0.56% is not economically large or significant: the t-statistic is -0.43. Thus, we conclude that there is no evidence against the hypothesis that returns to education is the same for men and women.

The coefficient on $female = -0.2268$ corresponds to differential against men when $educ = 0$ and all other factors are controlled for. This has low t-stats (-1.35), though economically large coefficient. This is misleading due to the presence of the interaction term $female{*}educ$. The coefficient on $female$ is now estimated much less precisely, than without the interaction term, because it is highly correlated to the interaction term, causing multicollinearity. Additionally, $educ = 0$ is a much less represented regime in the dataset, increasing the variance or the coefficient of $female$. More interesting would be to estimate the gender differential at, say, the average level in the sample. To do this, we would replace $female * educ$ with $female * (educ - 12.6)$ and rerun the regression. This only changes the coefficient on $female$ and its standard error.

```
print(df.educ.mean())
>> 12.562737642585551
model = smf.ols(formula='lwage~female+educ+I(female*(educ-12.6))+'+
                        'exper+I(exper**2)+tenure+I(tenure**2)', data=df)
results = model.fit()
results.summary2()
=========================================================================
Model:                 OLS               Adj. R-squared:       0.433
Dependent Variable:    lwage             R-squared:            0.441
-------------------------------------------------------------------------
                          Coef.  Std.Err.    t    P>|t|   [0.025  0.975]
-------------------------------------------------------------------------
Intercept                 0.3888   0.1187  3.2759 0.0011  0.1556  0.6220
female                   -0.2969   0.0358 -8.2828 0.0000 -0.3673 -0.2265
```

```
educ                          0.0824   0.0085  9.7249 0.0000  0.0657  0.0990
I(female * (educ - 12.6)) -0.0056   0.0131 -0.4260 0.6703 -0.0312  0.0201
exper                         0.0293   0.0050  5.8860 0.0000  0.0195  0.0391
I(exper ** 2)                -0.0006   0.0001 -5.3978 0.0000 -0.0008 -0.0004
tenure                        0.0319   0.0069  4.6470 0.0000  0.0184  0.0454
I(tenure ** 2)               -0.0006   0.0002 -2.5089 0.0124 -0.0011 -0.0001
------------------------------------------------------------------------------
```

We now see the t-statistic of the coefficient β_{female} restored to -8.28. To do joint testing for $H_0 : \beta_{female} = 0, \beta_{female*educ} = 0$, we use the F-test. The F-statistic is 34.33 and p-value is almost zero, giving strong evidence to reject the null hypothesis.

```
ftest = results.f_test(['female=0', 'I(female * educ)=0'])
print(ftest.statistic[0][0], ftest.pvalue)
34.32554911447195 1.002343957231108e-14
```

We compare this to the example 7.2 which has no interaction term. We prefer the model in example 7.2 which allows for constant wage differential between women and men, to the current one because it gives the same adjusted R-squared with a simpler model. □

Again, these coefficients and statistics suggest that there might be some relationships, but can't ascertain the causal links. To allow for any intercept difference between groups, we can include a dummy variable. If we want any of the slopes to depend on the group, we simply interact the appropriate variable with the group dummy and include it in the equation. If we are interested in testing whether there is any difference between the groups, then we must allow a model where the intercept and all slopes can be different across the two groups. The null hypothesis we then test is to seek all the coefficients with the relevant dummy variable, including the stand alone and interaction terms, to have a value of 0. If any one of them is non-zero, then the model is different for the two groups.

Example 7.7. Suppose we want to test whether the same regression model describes college grade point averages for male and female college athletes. The equation is

$$cumgpa = \beta_0 + \beta_1\ sat + \beta_2\ hsperc + \beta_3\ tothrs + u,$$

where *sat* is SAT score, *hsperc* is high school rank percentile, and *tothrs* is total hours of college courses. To account for any difference between men and women we want to fit

$$\begin{aligned} cumgpa = \beta_0 + \delta_0\ female + \beta_1\ sat + \delta_1\ female * sat + \beta_2\ hsperc \\ + \delta_2\ female * hsperc + \beta_3\ tothrs + \delta_3\ female * tothrs + u \end{aligned}$$

We fit a model to get $R^2 = 0.254$

```
df = woo.data('gpa3').dropna()
model = smf.ols(formula='cumgpa~female+sat+I(female * sat)+hsperc+'+
```

```
        'I(female * hsperc)+tothrs+I(female * tothrs)', data=df)
results = model.fit()
results.summary2()
-------------------------------------------------------------------
                    Coef.  Std.Err.    t    P>|t|   [0.025  0.975]
-------------------------------------------------------------------
Intercept           1.4808   0.2073  7.1422 0.0000  1.0731  1.8886
female             -0.3535   0.4105 -0.8610 0.3898 -1.1608  0.4539
sat                 0.0011   0.0002  5.8073 0.0000  0.0007  0.0014
I(female * sat)     0.0008   0.0004  1.9488 0.0521 -0.0000  0.0015
hsperc             -0.0085   0.0014 -6.1674 0.0000 -0.0111 -0.0058
I(female * hsperc) -0.0005   0.0032 -0.1739 0.8621 -0.0068  0.0057
tothrs              0.0023   0.0009  2.7182 0.0069  0.0006  0.0040
I(female * tothrs) -0.0001   0.0016 -0.0712 0.9433 -0.0033  0.0031
-------------------------------------------------------------------
```

None of the four terms involving the variable $female$ is very statistically significant, but we know better than to rely on the individual t-statistics. We need to do a F-test with hypothesis $H_0 : \delta_0 = \delta_1 = \delta_2 = \delta_3 = 0$. This gives a very small p-value and the null can be rejected soundly. Thus men and women do follow different GPA models, even though each term that allows women and men to be different is individually insignificant at the 5% level.

```
f = (results.rsquared-results0.rsquared)/(1-results.rsquared) *
    (results.df_resid)/(results0.df_resid - results.df_resid)
ftest = results.f_test(['female=0', 'I(female * sat)=0',
                        'I(female * hsperc)=0', 'I(female * tothrs)=0'])
print(f, ftest.statistic[0][0], ftest.pvalue)
>> 8.179111637046125 8.179111637045061 2.544637191827682e-06
```

If we only look at the $female$ variable, we would wrongly conclude that $cumgpa$ is about 0.3535 less for women than men, holding other factors constant. This is the estimated difference only when sat, $hsperc$, $tothrs$ are all set to zero, which is not close to being a possible scenario represented by the sample data. At the median value of $sat = 900$, $hsperc = 31$, and $tothrs = 47$, we find $\Delta\widehat{cumgpa} = (-0.3535 + 0.000751 * sat - 0.000550 * hsperc - 0.000116 * tothrs)\Delta female = 0.300\Delta female$. That is, female athlete is predicted to have a GPA 0.3 points higher than comparable male athlete. □

In the general model with k explanatory variables and an intercept, suppose we have two groups $g = 1, 2$ We would like to test whether the intercept and all slopes are the same across the two groups. We write the model as

$$y = \beta_{g,0} + \beta_{g,1}x_1 + \ldots + \beta_{g,k}x_k + u, \quad \text{for } g = 1, 2.$$

The hypothesis that each beta is the same across the two groups involves $k + 1$ restrictions. The unrestricted model has k interaction terms in addition to the intercept and the variables

themselves, thus has a degree of freedom of $n - 2(k+1)$, where $n = n_1 + n_2$. *The sum of squared residuals from the unrestricted model can be obtained from two separate regressions, one for each group.* Let SSR_g be the sum of squared residuals and n_g be the number of observations for $g = 1, 2$. Now, the sum of squared residuals for the unrestricted model is simply $SSR_u = SSR_1 + SSR_2$. The restricted sum of squared residuals is just the SSR from pooling the groups and estimating a single equation, say SSR_p. The F-statistic is

$$F = \frac{SSR_p - (SSR_1 + SSR_2)}{SSR_1 + SSR_2} \left(\frac{n - 2(k+1)}{k+1} \right).$$

This is called **Chow statistic**. Chow test is same as the F-test, it is only valid under homoskedasticity. In particular, under null hypothesis, the error variances for the two groups must be equal. As usual, normality is not needed for asymptotic analysis.

This setup of null hypothesis allows for no differences at all between the groups. In many cases it is more interesting to allow for an intercept differences between the groups and then to test for slope differences. There are two ways to allow for intercept to differ under null hypothesis. One is to include the group dummy and all interaction terms, but then test joint significance of the interaction terms only. The second approach, which produces an identical statistic, is to form a F-statistic based on SSR'_p obtained using a regression that contains only an intercept shift. The F statistic then changes to

$$F = \frac{SSR'_p - (SSR_1 + SSR_2)}{SSR_1 + SSR_2} \left(\frac{n - 2(k+1)}{k} \right).$$

Example 7.8. In our last example if $g = 1$ corresponds to $female$ then $n_1 = 90$ and $n_2 = 276$. We can calculate the Chow statistic as 8.18 which is highly significant. We also note that the two groups have very similar variance as required for homoskedasticity assumption.

```
df = woo.data('gpa3').dropna()
print((df.female==1).sum(), (df.female==0).sum())
>> 90 276
r1 = smf.ols(formula='cumgpa~sat+hsperc+tothrs', data=df[df.female==1]).fit()
r2 = smf.ols(formula='cumgpa~sat+hsperc+tothrs', data=df[df.female==0]).fit()
rp = smf.ols(formula='cumgpa~sat+hsperc+tothrs', data=df).fit()
ssr_1 = r1.mse_resid * r1.df_resid
ssr_2 = r2.mse_resid * r2.df_resid
ssr_p = rp.mse_resid * rp.df_resid
n, k = rp.nobs, r1.df_model
print(r1.mse_resid, r2.mse_resid)
>> 0.22793937177780546 0.21599896774430155
print(ssr_1, ssr_2, ssr_p, n, k)
>> 19.60278597289127 58.75171922645002 85.5150665992399 366 3.0
f = (ssr_p-ssr_1-ssr_2)/(ssr_1+ssr_2) * (n-2*(k+1))/(k+1)
pvalue = 1-stats.f.cdf(f, k+1, n-2*(k+1))
```

```
print(f, pvalue)
>> 8.179111637046153 2.5446371918480537e-06
```

Using the second approach to allow for an intercept difference between $female$ and $male$. We can now test $H_0 : \delta_1 = \delta_2 = \delta_3 = 0$, with δ_0 unrestricted under the null. We get a F statistic of 1.53 with a p-value 0.205, and so we do not reject the null hypothesis at even the 20% significance level.

```
rp = smf.ols(formula='cumgpa~female+sat+hsperc+tothrs', data=df).fit()
print(ssr_1, ssr_2, ssr_p, n, k)
>> 19.60278597289127 58.75171922645002 79.36166556802716 366 3.0
f = (ssr_p-ssr_1-ssr_2)/(ssr_1+ssr_2) * (n-2*(k+1))/(k)
print(f, 1-stats.f.cdf(f, k, n-2*(k+1)))
>> 1.5338978108628865 0.20537335628108755
```

It is important here to get a bird's eye view of what we have done and see what complexity of model makes sense in terms of our final choice. The fact that we were not able to reject $H_0 : \delta_0 = \delta_1 = \delta_2 = \delta_3 = 0$, but were able to reject $H_0 : \delta_1 = \delta_2 = \delta_3 = 0$ and the fact that β_{female} is statistically significant (t-statistic of 5.29) in the following model, tells us that the suitable model to select is

$$\underset{}{cumgpa} = \underset{(0.1798)}{1.3285} + \underset{(0.0586)}{0.3101}\ female + \underset{(0.0002)}{0.0012}\ sat - \underset{(0.0012)}{0.0084}\ hsperc + \underset{(0.0007)}{0.0025}.$$

with R^2 of 39.8%. □

7.3 Linear Probability model

What happens if we want to use multiple regression to explain a qualitative event, e.g. a binary outcome? By the assumption MLR.4 we have $E(y|\boldsymbol{x}) = \beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k$. If y is a binary variable taking value zero and one, it is always true that $P(y = 1|\boldsymbol{x}) = E(y|\boldsymbol{x})$. Hence, we can then model the probability of the event as a linear function

$$P(y = 1|\boldsymbol{x}) = \beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k,$$

which says that the probability of success, say $p(\boldsymbol{x}) = P(y = 1|\boldsymbol{x})$, is a linear function of the x_j, also called the response probability. This is called the **linear probability model** (LPM). In LPM β_j measures the change in the probability of success when x_j changes, holding other factors fixed: $\Delta P(y = 1|\boldsymbol{x}) = \beta_j \Delta x_j$. $\hat{y}$ now predicts the probability of success.

The shortcomings of LPM are apparent. It is easy to see that we can get value below 0 or above 1. A related problem is that a probability cannot be linearly related to the independent variables for all their possible values. Even with these problems, the LMP is useful and often applied in economics. It usually works well for values of the independent variables that are near the averages in the sample. There are ways to use the estimated probabilities

to predict a zero-one outcome. A predicted value defined as $\tilde{y}_i = 1$ if $\hat{y}_i \geq 0.5$ and $\tilde{y}_i = 0$ if $\hat{y}_i < 0.5$. The proportion of overall correct predictions is a widely used goodness-of-fit measure called **percent correctly predicted**.

Due to the binary nature of y, the linear probability model does violate one of the Gauss-Markov assumptions. When y is a binary variable, its variance, conditional on $\boldsymbol{x}$, is $Var(y|\boldsymbol{x}) = p(\boldsymbol{x})(1 - p(\boldsymbol{x}))$, i.e. in general it depends on the independent variables; there must be heteroskedasticity in a linear probability model. This does not cause bias in the OLS estimator of the β_j, but is crucial for justifying the usual t and F statistics, even in large samples. This means the standard errors are not generally valid, and we should use them with caution. It turns out that, in many applications, the usual OLS statistics are not far off.

Example 7.9. *arr*86 is a binary variable equal to 1 if a man was arrested during 1986, and 0 otherwise. A linear probability model fore describing *arr*86 based on factors: *pcnv* proportion of prior arrests that led to a conviction, *avgsen* average months served from prior convictions, *tottime* months spent in prison since age 18 prior to 1986, *ptime*86 months spent in prison in 1986 and *qemp*86 number of quarters that the man was legally employed in 1986. The estimated equation is

$$\widehat{P(arr86 = 1|\boldsymbol{x})} = \underset{(0.0172)}{0.4406} - \underset{(0.0212)}{0.1624}\ pcnv + \underset{(0.0065)}{0.0061}\ avgsen - \underset{(0.005)}{0.0023}\ totime - \underset{(0.0046)}{0.022}\ ptime86 - \underset{(0.0054)}{0.0428}\ qemp86,$$

with 2719 degrees of freedom and $R^2 = 0.047$ and $\overline{R}^2 = 0.046$.

```
df = woo.data('crime1')
df['arr86'] = (df.narr86>0).astype(int)
model = smf.ols(formula='arr86~pcnv+avgsen+tottime+ptime86+qemp86', data=df)
results = model.fit()
results.summary2()
```

```
                                                                 
Df Residuals:       2719              Adj. R-squared:      0.046
Dependent Variable: arr86             R-squared:           0.047
-----------------------------------------------------------------
               Coef.   Std.Err.     t     P>|t|    [0.025   0.975]
-----------------------------------------------------------------
Intercept       0.4406   0.0172  25.5683  0.0000   0.4068   0.4744
pcnv           -0.1624   0.0212  -7.6492  0.0000  -0.2041  -0.1208
avgsen          0.0061   0.0065   0.9474  0.3435  -0.0065   0.0188
tottime        -0.0023   0.0050  -0.4543  0.6496  -0.0120   0.0075
ptime86        -0.0220   0.0046  -4.7393  0.0000  -0.0311  -0.0129
qemp86         -0.0428   0.0054  -7.9247  0.0000  -0.0534  -0.0322
-----------------------------------------------------------------
```

```
ftest = results.f_test(['avgsen=0','tottime=0'])
print(ftest.statistic[0][0], ftest.pvalue)
>> 1.059700444036301 0.346702695391494
```

The intercept, 0.4406, is the predicted probability of arrest for all other factors 0. The variables *avgsen* and *tottime* are statistically insignificant both individually and jointly (F test gives p-value = 0.347).

Increasing the probability of conviction does lower the probability of arrest. The incarcerative effect is given by the coefficient on *ptime*86. If a man is in prison, he cannot be arrested, as shown by negative coefficient on *ptime*86. Finally, employment reduces the probability of arrest in a significant way. It has to be remembered that LPM model can't be true over all ranges of the independent variables and work mainly around the mean values. □

One has to be careful when evaluating programs because in most examples the control and treatment groups are not randomly assigned. We must also be careful to include factors that might be systematically related to the binary independent variable of interest. The problem of **self-selection** is common, where individuals self-select into certain behaviors or programs: participation is not randomly determined. In these cases a binary indicator of participation might be systematically related to unobserved factors, i.e. the residual term u. Thus *the self-selection problem is another way that an explanatory variables can be endogenous*. When this causes the coefficients to be biased, more advanced methods should be used.

When the dependent variables have more than two discrete possible values, the coefficients should be interpreted as related to the average of y, i.e. under assumptions MLR.1 and MLR.4 we have

$$E(y|\boldsymbol{x}) = \beta_0 + |beta_1 x_1 = \ldots + \beta_k x_k.$$

8 Heteroskedasticity

Under the Gauss-Markov assumptions, MLR.1 to MLR.4 the OLS estimates are unbiased and consistent. The homoskedasticity assumption MLR.5, states as $Var(u|\boldsymbol{x}) = \sigma^2$, played no role in showing whether OLS was unbiased or consistent. It is important to remember that heteroskedasticity does not cause bias or inconsistency in the OLS estimators of the β_j, whereas something like omitting an important variable would have this effect.

The interpretation of R^2 and $\overline{R}^2$ is also unaffected by presence of heteroskedasticity. This is because they depend on unconditional variance of population error and dependent variable which is unaffected by the presence of heteroskedasticity in $Var(u|\boldsymbol{x})$. Further, SSR/n and SST/n consistently estimate σ_u^2 and σ_y^2 respectively, whether or not $Var(u|\boldsymbol{x})$ is constant.

However, estimators of variances, $Var(\hat{\beta}_j)$ are biased without the homoskedasticity assumption. Biased variance are no longer suitable to construct t and F statistics. The usual OLS t statistic do not have t distributions in presence of heteroscedasticity, and the problem is not resolved by using large sample sizes. Similarly, F statistics are no longer F distributed, and the LM statistic is no longer asymptotic chi-square distribution.

Homoskedasticity fails whenever the variance of the unobserved factors changes across different segments of the population, where the segments are determined by the different values of

the explanatory variables. If $Var(u|\boldsymbol{x})$ is not constant, OLS is no longer BLUE. In addition, OLS is no longer asymptotically efficient in the class of instrumental variables estimators. Fortunately, adjustments can be made so that these statistics are valid in presence of heteroskedasticity of unknown form - at least in large sample case.

8.1 Heteroskedasticity-robust Inference

Consider the multiple regression model $\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\varepsilon}$. Along with the first four Gauss-Markov assumptions we now assume that $Var(u_i|\boldsymbol{x}_i) = \sigma_i^2$, no longer constant. Since the standard error of $\hat{\beta}_i$ is based directly on estimating $Var(\hat{\beta}_j)$ we need an estimate for it when heteroskedasticity is present. White (1980) showed how it is done. **White estimator** is efficient using heteroskedasticity-robust standard errors.

$$\widehat{Var}(\hat{\beta}_j) = \frac{\sum_{i=1}^{n} \hat{r}_{ij}^2 \hat{u}_i^2}{SSR_j^2},$$

where $\hat{r}_{ij}$ is the ith residual from regression x_j on all other independent variables, and SSR_j is the sum of squared residuals from this regression. $\hat{u}_i =$ is the OLS residuals. These are called **robust standard errors** for $\hat{\beta}_j$. Sometimes, software packages adjust for degrees of freedom before taking the square root, but they matter only asymptotically and are equivalent. The term SSR_j can be replaced by $SST_j(1 - R_j^2)$, where SST_j is the total sum of squares of x_j and R_j^2 is the usual R-squared from regressing x_j on all other explanatory variables. Consequently, little sample variation in x_j for strong multicollinearity can cause the heteroskedasticity-robust standard errors to be large.

Example 8.1. We apply this on the wage data and compare the heteroskedasticity-robust standard error against usual OLS standard errors.

```
df = woo.dataWoo('wage1')
df['marrmale'] = (1-df.female)*df.married
df['marrfem'] = df.female*df.married
df['singfem'] = df.female*(1-df.married)
reg = smf.ols(formula='lwage ~ marrmale + marrfem + singfem + educ +' +
        'exper + I(exper**2)+ tenure + I(tenure**2)', data=df)
results = reg.fit(cov_type='HC3')
```

```
       ols  robust
r2   0.461   0.461
ar2  0.453   0.453
```

```
                coeff          pval        stderr           tval
                  ols robust    ols robust    ols robust     ols  robust
I(exper ** 2)  -0.001 -0.001  0.000  0.000  0.000  0.000  -4.847  -4.948
I(tenure ** 2) -0.001 -0.001  0.022  0.050  0.000  0.000  -2.306  -1.959
Intercept       0.321  0.321  0.001  0.004  0.100  0.112   3.213   2.882
educ            0.079  0.079  0.000  0.000  0.007  0.008  11.787  10.422
exper           0.027  0.027  0.000  0.000  0.005  0.005   5.112   5.149
marrfem        -0.198 -0.198  0.001  0.001  0.058  0.060  -3.428  -3.321
marrmale        0.213  0.213  0.000  0.000  0.055  0.058   3.842   3.656
```

```
singfem        -0.110 -0.110  0.048  0.056  0.056  0.058  -1.980  -1.907
tenure          0.029  0.029  0.000  0.000  0.007  0.007   4.302   3.942
```

In this particular case the significance does not change much when using robust covariance matrix. As an empirical matter the robust standard errors are usually larger than the usual standard errors, but occasionally can be smaller too. We must emphasize that we do not know, at this point, whether heteroskedasticity is even present in the population model under consideration. All we have done is to report the two kinds of standard errors. We can see that no important conclusions are overturned by using the robust standard errors. □

In large samples, we can make a case for always reporting only the heteroskedasticity-robust standard errors in cross-sectional applications. In finite samples, the robust t statistics can have distributions that are not very close to the t distribution, and that could throw off our inference. If homoskedasticity assumption holds and the errors are normally distributed, then the usual t statistic have exact t distributions, regardless of the sample size. On the same lines, it is possible to obtain F and LM statistics that are robust to heteroskedasticity of an unknown arbitrary form.The heteroskedasticity-robust F statistic is also called heteroskedasticity-robust Wald statistic.

Example 8.2. We use spring semester gpa data to apply wald test on. We include the usual OLS numbers for comparison. Again, the differences are not very big. Joint significance tests are not much affected either.

```
df = woo.data('gpa3').dropna()
reg = smf.ols(formula='cumgpa~sat+hsperc+tothrs+female+black+white', data=df)
results_ols = reg.fit()
results_robust = reg.fit(cov_type='HC0')
results_robust.summary2()
print_compare({'ols': results_ols, 'robust': results_robust})
       ols  robust
r2   0.401   0.401
ar2  0.391   0.391
           coeff          pval        stderr          tval
             ols robust    ols robust    ols robust    ols robust
Intercept  1.470  1.470  0.000  0.000  0.230  0.219  6.397  6.726
black     -0.128 -0.128  0.385  0.277  0.147  0.118 -0.870 -1.086
female     0.303  0.303  0.000  0.000  0.059  0.059  5.141  5.181
hsperc    -0.009 -0.009  0.000  0.000  0.001  0.001 -6.906 -6.100
sat        0.001  0.001  0.000  0.000  0.000  0.000  6.389  6.014
tothrs     0.003  0.003  0.001  0.001  0.001  0.001  3.426  3.414
white     -0.059 -0.059  0.677  0.595  0.141  0.110 -0.416 -0.532
ftest = results_ols.f_test(['black=0', 'white=0'])
print(ftest.statistic[0][0], ftest.pvalue)
>> 0.6796041956073351 0.5074683622584049
ftest = results_robust.f_test(['black=0', 'white=0'])
#ftest = results_robust.wald_test(['black=0', 'white=0'], use_f=True) # alternative
print(ftest.statistic[0][0], ftest.pvalue)
>> 0.747796981803617 0.4741442714738484
```

Suppose we wish to test the null hypothesis that, after the other factors are controlled for, there are no differences in *cumgpa* by race, i.e. $H_0 : \beta_{black} = 0, \beta_{white} = 0$. The usual F test p-value is 0.51, while the heteroskedasticity-robust F test p-value is 0.47, both of which are not close to standard significance levels. □

When heteroskedasticity-robust F statistics are not available, sometimes it is convenient to have a alternate way of obtaining a test of multiple exclusion restrictions that is robust to heteroskedasticity. Heteroskedasticity-robust LM statistic is easily obtained. Say we want to test $H_0 : \beta_{k-q+1} = \ldots = \beta_k = 0$, which puts q exclusion restrictions on the model $y = \beta_0\beta_1 x_1 + \ldots + \beta_k x_k + u$.

1. Obtain the residuals $\tilde{u}$ from the restricted model, $y = \tilde{\beta}_0 + \tilde{\beta}_1 x_1 + \ldots + \tilde{\beta}_{k-q} x_{k-q} + \tilde{u}$.
2. Regress each of the excluded independent variables on all the included independent variables, leading to q sets of residuals $(\tilde{r}_1, \ldots, \tilde{r}_q)$.
3. Find the product between each $\tilde{r}_j$ and $\tilde{u}$.
4. Run the regression of 1 on $\tilde{r}_1\tilde{u}, \ldots, \tilde{r}_q\tilde{u}$ without an intercept.
5. The heteroskedasticity-robust LM statistic is $n - SSR$, where SSR is the usual sum of squared residuals from the final regression. Under H_0, LM is distributed approximately as χ^2_q.

Example 8.3. For the crime data we fit the usual and the heteroskedastic-robust model.

```
df = woo.data('crime1')
reg = smf.ols(formula='narr86~pcnv+avgsen+I(avgsen ** 2)+'+
              'ptime86+qemp86+inc86+black+hispan', data=df)
results_ols = reg.fit()
results_robust = reg.fit(cov_type='HC0')
print_compare({'ols':results_ols, 'robust': results_robust})
                coeff          pval        stderr           tval
                  ols robust    ols robust    ols robust     ols  robust
I(avgsen ** 2) -0.001 -0.001  0.082  0.013  0.000  0.000  -1.738  -2.490
Intercept       0.567  0.567  0.000  0.000  0.036  0.040  15.725  14.102
avgsen          0.018  0.018  0.066  0.078  0.010  0.010   1.840   1.765
black           0.325  0.325  0.000  0.000  0.045  0.058   7.147   5.557
hispan          0.193  0.193  0.000  0.000  0.040  0.040   4.871   4.807
inc86          -0.001 -0.001  0.000  0.000  0.000  0.000  -4.345  -6.458
pcnv           -0.136 -0.136  0.001  0.000  0.040  0.034  -3.359  -4.040
ptime86        -0.039 -0.039  0.000  0.000  0.009  0.006  -4.528  -6.335
qemp86         -0.051 -0.051  0.000  0.000  0.014  0.014  -3.499  -3.562
```

To see whether average sentence length has a statistically significant effect on *nar*86, we must test the joint hypothesis $H_0 : \beta_{avgsen} = 0, \beta_{avgsen^2} = 0$. Using the usual and heteroskedasticity-robust LM statistic we obtain the following:

```
# restricted regression
reg_r = smf.ols(formula='narr86~pcnv+ptime86+qemp86+inc86+black+hispan', data=df)
results_r_ols = reg_r.fit()
results_r_robust = reg_r.fit(cov_type='HC0')

# LM OLS
r2_r_ols = results_r_ols.rsquared
r2_u_ols = results_ols.rsquared

df['utilde_ols'] = results_r_ols.resid
m = smf.ols(formula='utilde_ols~pcnv+avgsen+I(avgsen ** 2)+'+
            'ptime86+qemp86+inc86+black+hispan', data=df).fit()
r2_utilde_ols = m.rsquared
n = m.nobs
n_df = results_ols.df_model - results_r_ols.df_model
LM_ols = n * r2_utilde_ols
pval_ols = 1-stats.chi2.cdf(LM_ols, n_df)
data_ols = pd.Series({'r2_r': r2_r_ols, 'r2_u': r2_u_ols,
    'r2_utilde': r2_utilde_ols, 'LM': LM_ols, 'pval': pval_ols})

# LM robust
r2_r_robust = results_r_robust.rsquared
r2_u_robust = results_robust.rsquared

df['utilde_ols'] = results_r_ols.resid
m1 = smf.ols(formula='avgsen~pcnv+ptime86+qemp86+inc86+black+hispan', data=df).fit()
m2 = smf.ols(formula='I(avgsen ** 2)~pcnv+ptime86+'+
             'qemp86+inc86+black+hispan', data=df).fit()

dff = pd.DataFrame({'ru1': m1.resid*df.utilde_ols,
                    'ru2': m2.resid*df.utilde_ols})

m = smf.ols(formula='1~ru1+ru2-1', data=dff).fit()

LM_robust = n - m.ssr
pval_robust = 1-stats.chi2.cdf(LM_robust, n_df)
data_robust = pd.Series({'r2_r': r2_r_robust, 'r2_u': r2_u_robust,
       'r2_utilde': m.rsquared, 'LM': LM_robust, 'pval': pval_robust})
print(pd.DataFrame({'ols': data_ols, 'robust': data_robust}))
                ols    robust
r2_r       0.071618  0.071618
r2_u       0.072798  0.072798
r2_utilde  0.001271  0.001467
LM         3.462601  3.997085
pval       0.177054  0.135533
```

Using the usual LM statistics we obtain LM=3.46; in a chi-square distribution with two df, this yields a p-value = 0.177. Thus we do not reject H_0 even at 15% significance level. The heteroskedasticity-robust LM statistic is LM=4.00, with a p-value = 0.136. Clearly the two have very similar implication, i.e. this is not a very strong evidence against H_0; $avgsen$ does

not appear to have a strong effect on $nar86$. □

8.2 Testing for heteroskedasticity

We could use the usual OLS standard errors and test statistics, unless there is evidence of heteroskedasticity. We will restrict ourselves to tests that detect the kind of heteroskedasticity that invalidates the usual OLS statistics. For the equation $y = \beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k + u$, we work with the assumptions MLR.1 through MLR.4 and assume the null hypothesis is $H_0 : Var(u|\boldsymbol{x}) = \sigma^2$. This is same as $H_0 : E(u^2|\boldsymbol{x}) = \sigma^2$. We thus want to test if u^2 is related to one ore more of the explanatory variables. If H_0 is false, the expected value of u^2, should be a function of independent variables. A simple approach is to assume a linear function, i.e $u^2 = \delta_0 + \delta_1 x_1 + \ldots + \delta_k x_k + \nu$, with the null hypothesis for homoskedasticity being $H_0 : \delta_1 = \delta_2 = \ldots = \delta_k = 0$. If H_0 is true either the F or LM statistics could be used for overall significance. Both statistics would have asymptotic justification. Since u^2 is not available we replace it by its estimate $\hat{u}^2$ and it turns out that it does not effect the large sample distribution of the F or LM statistics. The F statistic under null hypothesis is

$$F = \frac{R^2_{\hat{u}^2}/k}{(1 - R^2_{\hat{u}^2})/(n-k-1)} \sim F_{k,n-k-1},$$

where $R^2_{\hat{u}^2}$ is the R-squared from the regression $\hat{u}^2 = \hat{\delta}_0 + \hat{\delta}_1 x_1 + \ldots + \hat{\delta}_k x_k$. The LM statistic for heteroskedasticity is just the sample size times the R-squared, i.e. under null hypothesis,

$$LM = nR^2_{\hat{u}^2} \sim \chi^2_k.$$

The LM version of the test is typically called the **Breusch-Pagan test** for heteroskedasticity (BP test).

Example 8.4. We use housing prices regression to test if heteroskedasticity is present.

```
df = woo.data('hprice1')
m = smf.ols(formula='price~lotsize+sqrft+bdrms', data=df).fit()

# manual test
df['resid2'] = m.resid**2
u2 = smf.ols(formula='resid2~lotsize+sqrft+bdrms', data=df).fit()
print(u2.rsquared, u2.nobs, u2.df_model)
>> 0.16014074436761616 88.0 3.0
print(u2.fvalue, u2.f_pvalue)
>> 5.338919363241411 0.002047744420936089
LM = u2.nobs * u2.rsquared
pval = 1-stats.chi2.cdf(LM, u2.df_model)
print(LM, pval)
>> 14.092385504350222 0.0027820595556891092

# automatic test
from statsmodels.stats.diagnostic import het_breuschpagan
import patsy as pt
y, X = pt.dmatrices('price~lotsize+sqrft+bdrms', data=df, return_type='dataframe')
```

```
bp = het_breuschpagan(m.resid, X)
print(bp)
>> (14.092385504350222, 0.00278205955568911, 5.338919363241411, 0.002047744420936089)
```

We have a strong evidence against null both from the F and LM p-values. This means that the usual standard errors reported by the OLS is not reliable. Using logarithmic functional form often reduces heteroskedasticity. Let us check it out by using log of *prices*, *lotsize* and $sqrft$ instead.

```
m = smf.ols(formula='np.log(price)~np.log(lotsize)+np.log(sqrft)+bdrms', data=df).fit()
y, X = pt.dmatrices('np.log(price)~np.log(lotsize)+np.log(sqrft)+bdrms', data=df,
                    return_type='dataframe')
bp = het_breuschpagan(m.resid, X)
print(bp)
>> (4.2232457418052665, 0.23834482631493092, 1.4114999061208027, 0.2451456613048952)
```

The p-values now make us fail to reject the null hypothesis of homoskedasticity with the dependent variables in logarithmic functional form. □

To test heteroskedasticity being dependent upon certain independent variables, we simply regression $\hat{u}^2$ on the suitable candidate independent variables. If the squared residual is regressed on only a single independent variable, the test for heteroskedasticity is just the usual t statistic on the variable. A significant t statistic suggests that heteroskedasticity is a problem.

Instead of simple linear form White (1980) added quadratic terms to the equation $\hat{u}^2 = \delta_0 + \delta_1^{(1)} x_1 + \ldots + \delta_1^{(2)} x_1^2 + \ldots + \delta_{12}^{(11)} x_1 x_2 + \ldots + \varepsilon$. Using the LM statistic with this is called the **White test** for heteroskedasticity. We can also use F test; both tests are asymptotically justified. But there can be lot of variables in this test making it weak. Conservation on degrees of freedom can be achieved by instead looking at $\hat{u}^2 = \delta_0 + \delta_1 \hat{y} + \delta_2 \hat{y}^2 + \varepsilon$, where $\hat{y}$ stand for the fitted values. We use these fitted values because they are functions of the independent variables; using y does not produce a valid test for heteroskedasticity. We can use the F or LM statistic for the null hypothesis $H_0 : \delta_1 = \delta_2 = 0$, comparing against $F_{2,n-3}$ and χ^2_2 distributions respectively. This test is especially useful in cases where the variance is thought to change with the level of the expected value $E(y|\boldsymbol{x})$. This is because $\hat{y}$ is an estimate of $E(y|\boldsymbol{x})$. This test can be viewed as a special case of the White test, as this is simply imposing restrictions on the parameters in original White test equation.

Example 8.5. We apply the White original test for the log price prediction in the previous example.

```
df = woo.data('hprice1')
m = smf.ols(formula='np.log(price)~np.log(lotsize)+np.log(sqrft)+bdrms', data=df).fit()

# manual
```

```
df['resid2'] = m.resid**2
u2 = smf.ols(formula='resid2~np.log(lotsize)+np.log(sqrft)+bdrms+'+
                     'I(np.log(lotsize)**2)+I(np.log(sqrft)**2)+I(bdrms**2)+'+
                     'I(np.log(lotsize) * np.log(sqrft)) +' +
                     'I(np.log(lotsize) * bdrms) + I(np.log(sqrft) * bdrms)',
                     data=df).fit()
print(u2.rsquared, u2.nobs, u2.df_model)
>> 0.1085165048432657 88.0 9.0
print(u2.fvalue, u2.f_pvalue)
>> 1.0549565756603887 0.4053123705653041
LM = u2.nobs * u2.rsquared
pval = 1-stats.chi2.cdf(LM, u2.df_model)
print(LM, pval)
>>  9.549452426207381 0.38817399191075075

# automatic
from statsmodels.stats.diagnostic import het_white
y, X = pt.dmatrices('np.log(price)~np.log(lotsize)+np.log(sqrft)+bdrms',
                    data=df, return_type='dataframe')
wt = het_white(m.resid, X)
print(wt)
>> (9.549452426207303, 0.38817399191075785, 1.0549565756603783, 0.40531237056531105)
```

We see here too that we fail to reject the null hypothesis as the p-values for are quite high. We now conduct the White special test as described to conserve on the degrees of freedom.

```
# speical White test
df = woo.data('hprice1')
m = smf.ols(formula='np.log(price)~np.log(lotsize)+np.log(sqrft)+bdrms', data=df).fit()
```

```
# manual
df['one'] = 1
df['resid2'] = m.resid**2
df['yhat'] = df.price.apply(np.log) - m.resid
df['yhat2'] = df.yhat**2
u2 = smf.ols(formula='resid2~yhat+yhat2', data=df).fit()
print(u2.rsquared, u2.nobs, u2.df_model)
>> 0.03917371075994047 88.0 2.0
print(u2.fvalue, u2.f_pvalue)
>> 1.732761401245865 0.18298154328305286
LM = u2.nobs * u2.rsquared
pval = 1-stats.chi2.cdf(LM, u2.df_model)
print(LM, pval)
>> 3.4472865468747615 0.17841494794135526

# automatic
bp = het_breuschpagan(m.resid, df[['one', 'yhat', 'yhat2']])
print(bp)
>> (3.447286546874742, 0.178414947941357, 1.7327614012458543, 0.18298154328305544)
```

We get a p-value of 0.178 from the LM test now. This is stronger evidence of heteroskedasticity than is provided by the Breusch-Pagan or the original White test, but we still fail to reject homoskedasticity at even the 15% level. □

A rejection using a heteroskedasticity test should be an evidence of heteroscedasticity only if we maintain assumptions MLR.1 through MLR.4. If MLR.4 is violated - in particular, *if the functional form of* $E(y|\boldsymbol{x})$ *is misspecified - then a test for heteroskedasticity can reject* H_0*, even if* $Var(y|\boldsymbol{x})$ *is constant.* For example if we omit a quadratic term or use level instead of log, a test for heteroskedasticity can be significant. It is better to use explicit tests for functional form misspecification first, and once satisfied, test for heteroskedasticity. Functional form misspecification is more important than heteroskedasticity.

8.3 Weighted least squares estimation

If heteroskedasticity is detected using one of the tests, one possible response is to use the heteroskedasticity-robust statistics after estimation by OLS. However, if the form of heteroskedasticity is known use of weighted least squares method is more efficient than OLS, and WLS leads to new t and F statistics that have t and F distributions.

8.3.1 Heteroskedasticity is known up to a multiplicative constant: GLS

Let us assume that $Var(u|\boldsymbol{x}) = \sigma^2 h(\boldsymbol{x})$, where $h(\boldsymbol{x})$ is known and positive, while σ^2 is unknown. Essentially, we take the original equation $y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_k x_{ik} + u_i$, which contains heteroskedastic errors, and transform it into an equation that has homoskedastic errors. Since $h_i(\boldsymbol{x}_i)$ is just a function of $\boldsymbol{x}_i$, $u_i/\sqrt{h_i(\boldsymbol{x}_i)}$ has a zero expected value conditional on $\boldsymbol{x}_i$. Further the $Var(u_i/\sqrt{h_i(\boldsymbol{x}_i)}|\boldsymbol{x}_i) = E((u_i/\sqrt{h_i(\boldsymbol{x}_i)})^2|\boldsymbol{x}_i) = E(u_i^2|\boldsymbol{x}_i)/h_i(\boldsymbol{x}_i) = \sigma^2$. Hence, we divide the original linear equation by $\sqrt{h_i(\boldsymbol{x}_i)} = \sqrt{h_i}$ to get

$$\frac{y_i}{\sqrt{h_i}} = \beta_0 \frac{1}{\sqrt{h_i}} + \beta_1 \frac{x_{i1}}{\sqrt{h_i}} + \ldots + \beta_k \frac{x_{ik}}{\sqrt{h_i}} + \frac{u_i}{\sqrt{h_i}}$$

or

$$y_i^* = \beta_0 x_{i0}^* + \beta_1 x_{i1}^* + \ldots + \beta_k x_{ik}^* + u_i^*,$$

where $x_{i0}^* = 1/\sqrt{h_i}$ and $x_{ij}^* = \beta_j/\sqrt{h_i}$, for $0 < j \leq k$. Also, $u_i^* = u_i/\sqrt{h_i}$. Notice that x^* variables may not have useful interpretation but we can always interpret the coefficients based on the original equation. If the original equation satisfies the first four Gauss-Markov assumptions, the transformed equation satisfies all five Gauss-Markov assumptions. Also, if u_i has a normal distribution, then u_i^* has a normal distribution with variance σ^2. Therefore, the transformed equation satisfies the six classical linear model assumptions (MLR.1 through MLR.6) if the original model does so except for the homoskedasticity assumption.

We can now estimate x^* based on OLS. These estimators β_j^* will be different from the one in the original OLS equation. They are examples of generalized least squares (GLS) estimators. Because the transformed equation satisfies all conditions, standard errors, t and F statistics are all valid. The R-squared that is obtained, while useful for computing F

statistics, is not especially informative as a goodness-of-fit measure, because it tells us how much the variation in y^* is explained by the x_j^*, and this is seldom very useful.

The GLS estimators for correcting heteroskedasticity are caller weighted least squares (WLS) estimators because β_j^* minimizes the weighted sum of squares residual, where each squared residual is weighted by $1/h_i$. The idea is that *less weight is given to the observations with a higher error variance*; while OLS gives the same weight to each observation.

Example 8.6. We estimate an equation to explain net total financial wealth $nettfa$ in terms of income and other variables including age, gender, and eligibility for 401(k). We use data on single people $fsize = 1$. And use a quadratic term on age $(age - 25)^2$ as the minimum age in the sample is 25 for simplified interpretation. We report simple OLS with heteroskedasticity-robust errors and WLS with the assumption $Var(u|inc) = \sigma^2\ inc$.

```
df = woo.data('401ksubs')
dfs = df[df.fsize == 1]
model_ols = smf.ols(formula='nettfa~inc', data=dfs).fit(cov_type='HC0')
wts = 1/dfs.inc
model_wls = smf.wls(formula='nettfa~inc', data=dfs, weights=wts).fit()
model_ols_full = smf.ols(formula='nettfa~inc+male+e401k+I((age-25)**2)',
                    data=dfs).fit(cov_type='HC0')
model_wls_full = smf.wls(formula='nettfa~inc+male+e401k+I((age-25)**2)',
                    data=dfs, weights=wts).fit()
     ols_simple  wls_simple  ols_complex  wls_complex
r2        0.083       0.071        0.128        0.112
ar2       0.082       0.070        0.126        0.110
```

```
                          ols_simple  wls_simple  ols_complex  wls_complex
coeff  I((age - 25) ** 2)         NaN         NaN        0.025        0.018
       Intercept              -10.571      -9.581      -20.985      -16.703
       e401k                      NaN         NaN        6.886        5.188
       inc                      0.821       0.787        0.771        0.740
       male                       NaN         NaN        2.478        1.841
pval   I((age - 25) ** 2)         NaN         NaN        0.000        0.000
       Intercept                0.000       0.000        0.000        0.000
       e401k                      NaN         NaN        0.003        0.002
       inc                      0.000       0.000        0.000        0.000
       male                       NaN         NaN        0.228        0.239
stderr I((age - 25) ** 2)         NaN         NaN        0.004        0.002
       Intercept                2.529       1.653        3.491        1.958
       e401k                      NaN         NaN        2.284        1.703
       inc                      0.104       0.063        0.099        0.064
       male                       NaN         NaN        2.056        1.564
tval   I((age - 25) ** 2)         NaN         NaN        5.791        9.080
       Intercept               -4.180      -5.795       -6.011       -8.530
       e401k                      NaN         NaN        3.015        3.046
       inc                      7.926      12.398        7.749       11.514
       male                       NaN         NaN        1.205        1.177
```

In both the models, we notice that β_{inc} is very slightly smaller for WLS versus OLS, but the standard error is significantly lower for WLS versus OLS, provided we assume the model $Var(nettfa|inc) = \sigma^2\ inc$. Adding other controls reduces the inc coefficient somewhat, with the OLS estimate still larger than the WLS estimate. Again, the WLS estimate of β_{inc} is more precise. The WLS estimate for $e401k$ is substantially below the OLS estimate and suggests a misspecification of the functional form in the mean equation. An interaction term might help.

```
f = model_wls_full.f_test(['male=0', 'e401k=0', 'I((age - 25) ** 2)=0'])
print(f.statistic[0][0], f.pvalue)
>> 30.66851806330892 2.205574009502316e-19
```

The F statistic for joint significance of $(age - 25)^2, male$, and $e401k$ is about 30.8, with p-value very close to 0; this is not surprising given the very large t statistics for the age and 401(k) variable. □

The functional form assumption is essentially arbitrary. There is one case where WLS is a natural choice. This happens when, instead of using individual-level data, we only have averages of data across some group. For example, suppose we are interested in determining the relationship between the amount a worker contributes to their 401(k) as a function of the plan generosity. Let i denote a particular firm and let e denote an employee within the firm. A simple model is $contrib_{i,e} = \beta_0 + \beta_1 earns_{i,e} + \beta_2 age_{i,e} + \beta_3 mrate_i + u_{i,e}$, where $contrib_{i,e}$ is the annual contribution by employee e who works for firm i, $earns_{i,e}$ is the annual earnings for this person, and $age_{i,e}$ is the person's age. The variable $mrate_i$ is the amount the firm puts into an employee's account for every dollar the employee contributes.

If only the average across various employers is available and we know the number of employees at firm i as m_i then we the equation becomes $\overline{contrib}_i = \beta_0 + \beta_1 \overline{earns}_i + \beta_2 \overline{age}_i + \beta_3\ mrate_i + \overline{u}_i$. The estimators are unbiased if the original model satisfies the Gauss-Markov assumptions and the individual errors $u_{i,e}$ are independent of the firm's size, m_i. If the individual-level equations satisfies the homoskedasticity assumption then $Var(\bar{u}_i) = \sigma^2/m_i$. The variance of the error term $\overline{u}_i$ decreases with firm size. Hence, we can use $h_i = 1/m_i$ in our weighted least squares. This ensures that larger firms receive more weight. This gives us an efficient way of estimating the parameters in the individual-level model when we only have averages at the firm level.

Uncertainty about the form of $Var(\bar{u}_i)$ is why researchers simply use OLS and compute robust standard errors and test statistics when estimating models using per capita data. An alternative is to weight by group size but to report the heteroskedasticity-robust statistics in the WLS estimation. This ensures that, while the estimation is efficient if the individual-level model satisfies the Gauss-Markov assumptions, heteroskedasticity at the individual level or within-group correlation are accounted for through robust inference.

8.3.2 Heteroskedasticity function must be estimated: Feasible GLS

It is generally difficult to find the function $h(\boldsymbol{x}_i)$. We can instead estimate it to $\hat{h}_i$ and then use that in the GLS transformation yielding Feasible GLS (FGLS) estimator. There are many way to model heteroskedasticity. We follow the following general form

$$Var(u|\boldsymbol{x}) = \sigma^2 exp\left(\delta_0 + \delta_1 x_1 + \ldots + \delta_k x_k\right)\nu,$$

where ν has a mean equal to unity, conditional on $\boldsymbol{x}$. Hence, $h(\boldsymbol{x}) = exp(\delta_0+\delta_1 x_1+\ldots+\delta_k x_k)$. The linear alternative to model heteroskedasticity, like in Breusch-Pagan test, are fine when testing, but they can be problematic when correcting for heteroskedasticity using WLS, because they don't ensure positive values, while exponentiation does. If we now assume ν is independent of $\boldsymbol{x}$ we can write:

$$log(u^2) = \delta_0' + \delta_1 x_1 + \ldots + \delta_k x_k + \varepsilon.$$

Since this equation satisfies the Gauss-Markov assumptions, we can get unbiased estimators of δ_j by using OLS. We replace the unobserved u with OLS residuals $log(\hat{u}^2)$. We then get the fitted values $\hat{g}_i$ and then estimate h_i as $\hat{h}_i = exp(\hat{g}_i)$. We can then use WLS with weights $1/\hat{h}_i$ in place of $1/h_i$. Having to estimate h_i using the same data means the FGLS estimator is no longer unbiased, but it is consistent and asymptotically more efficient than OLS. Hence FGLS is an attractive alternative to OLS for large sample sizes, where there is evidence of heteroskedasticity. However, if we have some doubts about the original general form, we can use heteroskedasticity-robust standard errors and test statistics in the transformed equation.

Another useful alternative for estimating h_i is to obtain $\hat{g}_i$ as the fitted values from the regression of $log(\hat{u}^2)$ on $\hat{y}, \hat{y}^2$ and then obtain the $\hat{h}_i$ as before. The previous general regression formulation, called Park test can't be used for testing heteroskedasticity because it needs stronger null hypothesis requiring u must be independent of $\boldsymbol{x}$ and using $\hat{u}$ in place of u can cause F statistic to deviate from F distribution even in large samples. Regression works well here because we are only interested in consistent estimate of δ_j which it delivers.

When computing F statistics for testing multiple hypothesis after estimation by WLS, it is important that the same weights be used to estimate the restricted and unrestricted models. The OLS and WLS estimates will always differ due to sampling error, the issue is whether their differences are enough to change important conclusions. A big difference, particularly in sign, indicates that one of the other Gauss-Markov assumptions is false, particularly the zero conditional mean assumption MLR.4, indicating a functional form misspecification in $E(y|\boldsymbol{x})$. The Hausman test can be used to formally compare the OLS and WLS estimates to see if they differ by more than sampling error suggess they should.

Example 8.7. We estimate the demand function for daily cigarette consumption. Since most of the people do not smoke a linear model is not ideal, but we fit it for illustration. We first estimate the simple OLS regression with OLS standard errors to get

$$\begin{aligned}\widehat{cigs} = &-\underset{(24.08)}{3.64} + \underset{(0.73)}{0.88}\, log(income) - \underset{(5.77)}{0.75}\, log(cigpric)\\ &- \underset{(0.50)}{0.17}\, educ + \underset{(0.16)}{0.77}\, age - \underset{(0.0017)}{0.0090}\, age^2 - \underset{(1.11)}{2.83}\, resaturn,\end{aligned}$$

with 800 degrees of freedom for residual and $R^2 = 0.053$. Here *resaturn* is a binary indicator of if the state restaurant restrict smoking. We note that neither income nor cigarette price is statistically significant and their effects are not particularly large. However education, age and its quadratic effect and *resaturn* seem to be statistically significant.

To check if the errors in the above regression contain heteroskedasticity we look at the Breusch-Pagan test.

```
y, X = pt.dmatrices('cigs~lincome+lcigpric+educ+age+I(age ** 2)+restaurn',
                    data=df, return_type='dataframe')
bp = het_breuschpagan(result_ols.resid, X)
print(np.round(bp,4))
>> [32.2584  0.      5.5517  0.    ]
```

We get a LM statistic of 32.2584 for variable χ^2_6 with a p-value of almost 0, suggesting strong evidence of heteroskedasticity. Therefore, we estimate the equation using feasible GLS procedure.

```
df['lu2'] = np.log(result_ols.resid**2)
model_lu2 = smf.ols(formula='lu2~lincome+lcigpric+educ+age+I(age ** 2)+restaurn',
                    data=df).fit()
wts = 1/np.exp(model_lu2.fittedvalues)
model_fgls = smf.wls(formula='cigs~lincome+lcigpric+educ+age+I(age ** 2)+restaurn',
                     weights=wts, data=df).fit()
model_fgls.summary2()
```

This gives a fitted model of

$$\widehat{cigs} = \underset{(17.803)}{-5.635} + \underset{(0.533)}{1.295}\ log(income) - \underset{(4.46)}{2.94}\ log(cigpric) - \underset{(0.12)}{0.463}\ educ + \underset{(0.097)}{0.482}\ age - \underset{(0.001)}{0.006}\ age^2 - \underset{(0.796)}{3.461}\ resaturn,$$

with 800 degrees of freedom for residual and $R^2 = 0.113$. We compare this with heteroscedasticity-robust standard errors in the following table.

	coeff		pval		stderr		tval	
	fgls	robust	fgls	robust	fgls	robust	fgls	robust
I(age ** 2)	-0.006	-0.006	0.000	0.000	0.001	0.001	-5.990	-4.802
Intercept	5.635	5.635	0.752	0.879	17.803	37.161	0.317	0.152
age	0.482	0.482	0.000	0.000	0.097	0.114	4.978	4.209
educ	-0.463	-0.463	0.000	0.002	0.120	0.148	-3.857	-3.123
lcigpric	-2.940	-2.940	0.510	0.742	4.460	8.931	-0.659	-0.329
lincome	1.295	1.295	0.003	0.015	0.437	0.533	2.964	2.431
restaurn	-3.461	-3.461	0.000	0.000	0.796	0.713	-4.351	-4.856

The income effect is now significant and larger in magnitude. □

Under the situation when $Var(y|\boldsymbol{x}) \neq \sigma^2 h(\boldsymbol{x})$, the estimates are still unbiased and consistent. Large differences between OLS and WLS estimators are in indicator of functional form misspecification. It can become biased only if we estimate parameters in a function say $h(\boldsymbol{x}, \hat{\delta})$ but will still be consistent. If we use WLS with misspecified variance function the estimated standard errors are no longer valid. The easy fix is to obtain heteroskedasticity-robust standard errors for WLS that allows the variance function to be arbitrarily misspecified. In fact, it is always a good idea to compute fully robust standard errors and test statistics after WLS estimation. As a final note, though if the variance function is misspecified WLS is not guaranteed to be more efficient that OLS; in case of strong heteroskedasticity, it is often better to use a wrong form of heteroskedasticity and apply WLS than to ignore it altogether in the estimation and use OLS. Even approximate variance models will produce smaller asymptotic variances.

8.3.3 Prediction and prediction intervals with Heteroskedasticity

Point prediction of y is affected by heteroskedasticity only insofar as it affects estimation of the β_j. our prediction of an unobserved outcome y^0 is estimated as $\hat{y}^0 = \hat{\beta}_0 + \boldsymbol{x}^0\hat{\boldsymbol{\beta}}$, where $\boldsymbol{x}^0$ is the known values of the explanatory variables. Once we know $E(y|\boldsymbol{x})$ our predictions are based on it and $Var(y|\boldsymbol{x})$ does not effect it. However, prediction intervals do depend directly on the nature of $Var(y|\boldsymbol{x})$. Under all the CLM assumptions with heteroskedasticity WLS estimators are BLUE and normally distributed. Hence, we can use the previous method to find $se(\hat{y}^0)$. We estimate $y_i = \theta_0 + \beta_1(x_{i1} - x_1^0) + \ldots + \beta_k(x_{ik} - x_k^0) + u_i$. We then use the WLS estimates to obtain $\hat{y}^0 = \hat{\theta}_0$ and $se(\hat{y}^0) = se(\hat{\theta}_0)$. Further, $Var(u|\boldsymbol{x} = \boldsymbol{x}^0) = \sigma^2 h(\boldsymbol{x}^0)$, hence $se(u^0) = \hat{\sigma}\sqrt{h(\boldsymbol{x}^0)}$, where $\hat{\sigma}$ is the standard error of the regression from the WLS estimation. Therefore, $se(\hat{e}^0) = \sqrt{se(\hat{y}^0)^2 + \hat{\sigma}^2 h(\boldsymbol{x}^0)}$. This is exact if we have access to exact $h(\boldsymbol{x}^0)$; so under estimation we do not get exact intervals. Generally, though, the error in u^0 swamps the error due to parameter estimation hence we can simply use $h(\boldsymbol{x}^0)$ swapped by $\hat{h}(\boldsymbol{x}^0)$. For the same reasons we can drop $se(\hat{y}^0)$ from the expression too; $se(\hat{y}^0)$ converges to zero at the rate $1/\sqrt{n}$, while $se(\hat{u}^0)$ is roughly constant.

We can also make predictions for the model $log(y) = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n + u$, where u is heteroskedastic. We add the normality assumption to the form of variance model

$$u|\boldsymbol{x} \sim \mathcal{N}(0, exp(\delta_0 + \delta_1 x_1 + \ldots + \delta_k x_k)).$$

This implies

$$E(y|\boldsymbol{x}) = exp\left(\beta_0 + \boldsymbol{x}\boldsymbol{\beta} + \frac{1}{2}\sigma^2 \exp(\delta_0 + \boldsymbol{x}\boldsymbol{\delta})\right)$$

To estimate β_j and δ_j using WLS, we first do the OLS to obtain the residual, then model $log(\hat{u}^2)$ linearly on the exogenous variables to obtain the fitted value $\hat{g}_i = \hat{\alpha}_0 + \hat{\delta}_1 x_{i1} + \ldots + \hat{\delta}_k x_{ik}$, then calculate $\hat{h}_i = exp(\hat{g}_i)$. These weights are used in WLS to obtain $\hat{\beta}_j$ and $\hat{\sigma}^2$. The R-squared can be calculated as the correlation between y_i and the fitted values $\hat{y}_i = exp(\widehat{\log(y_i)} + \frac{1}{2}\hat{\sigma}^2\hat{h}_i)$. For any values of the explanatory variables $\boldsymbol{x}^0$, we can estimate $E(y|\boldsymbol{x}^0)$ as

$$\hat{E}(y|\boldsymbol{x}^0) = exp\left(\hat{\beta}_0 + \boldsymbol{x}^0\hat{\boldsymbol{\beta}} + \frac{1}{2}\hat{\sigma}^2 \exp(\hat{\alpha}_0 + \boldsymbol{x}^0\hat{\boldsymbol{\delta}})\right)$$

Obtaining a proper standard error for prediction $\hat{y}^0$ in this case is quite complicated but can be fairly easy to obtain using resampling methods such as bootstrap. Obtaining a prediction interval is again complicated. However, generally the error variance swamps the estimation error and in this case the 95% prediction interval, for large samples, is $\exp(-2\hat{\sigma}\sqrt{\hat{h}(\boldsymbol{x}^0)}\exp(\hat{\beta}_0\boldsymbol{x}^0\hat{\boldsymbol{\beta}}))$ to $\exp(2\hat{\sigma}\sqrt{\hat{h}(\boldsymbol{x}^0)}\exp(\hat{\beta}_0\boldsymbol{x}^0\hat{\boldsymbol{\beta}}))$; again noticing that the interval may not be symmetric.

8.4 The Linear Probability model revisited

When the dependent variable y is a binary variable, the model must contain heteroskedasticity, unless all the slope parameters are zero. We can now deal with that issue. One way is to simply do OLS and report robust standard errors. This ignores the fact that we know the form of heteroskedasticity for the LPM. We know that $Var(y|\boldsymbol{x}) = p(\boldsymbol{x})(1-p(\boldsymbol{x}))$, where $p(\boldsymbol{x}) = \beta_0 + \boldsymbol{x}\boldsymbol{\beta}$ is the response probability, i.e. probability of success $y = 1$. We can use the estimated values based on unbiased estimators of β_j to get $\hat{h}_i = \hat{y}_i(1-\hat{y}_i)$, where $\hat{y}_i$ is the OLS fitted values for observation i. After which we can apply feasible GLS. Unfortunately, if the predicitons are negative we need to make adjustments. For example we can set $\hat{y}_i = 0.01$ if $\hat{y}_i < 0$ and $\hat{y}_i = 0.99$ if $\hat{y}_i > 1$. If many fitted values are outside the unit interval, it is probably best to just us OLS with robust standard errors.

Example 8.8. We estimate the probability of owning a computer by a college student based on high school GPA ($hsGPA$), ACT score and a binary indicator if at least one parent attended college ($parcoll$). We apply OLS and WLS with and without robust standard errors.

```
df = woo.data('gpa1')
df['parcoll'] = ((df.fathcoll+df.mothcoll)>0).astype(int)
ols = smf.ols(formula='PC~hsGPA+ACT+parcoll', data=df).fit()
olsR = smf.ols(formula='PC~hsGPA+ACT+parcoll', data=df).fit(cov_type='HC0')
wt = 1/ols.fittedvalues/(1-ols.fittedvalues)
wls = smf.wls(formula='PC~hsGPA+ACT+parcoll', data=df, weights=wt).fit()
wlsR = smf.wls(formula='PC~hsGPA+ACT+parcoll', data=df, weights=wt).fit(cov_type='HC0')

print_compare({'ols': ols, 'olsR': olsR, 'wls': wls, 'wlsR': wlsR})
         ols     olsR      wls     wlsR
r2     0.042    0.042    0.046    0.046
ar2    0.021    0.021    0.026    0.026
dfr  137.000  137.000  137.000  137.000
dfm    3.000    3.000    3.000    3.000
            coeff                          pval
              ols   olsR    wls   wlsR    ols   olsR    wls   wlsR
ACT         0.001  0.001  0.004  0.004  0.971  0.972  0.783  0.789
Intercept -0.000 -0.000  0.026  0.026  0.999  0.999  0.956  0.957
hsGPA       0.065  0.065  0.033  0.033  0.635  0.639  0.802  0.817
parcoll     0.221  0.221  0.215  0.215  0.019  0.011  0.014  0.013
            stderr                         tval
            ols    olsR    wls   wlsR    ols   olsR    wls   wlsR
ACT         0.015  0.016  0.015  0.016  0.036  0.036  0.276  0.268
Intercept   0.491  0.489  0.477  0.486 -0.001 -0.001  0.055  0.054
```

```
hsGPA      0.137  0.139  0.130  0.141  0.476  0.469  0.252  0.232
parcoll    0.093  0.087  0.086  0.087  2.378  2.547  2.494  2.472
```

There are no important differences in the OLS and WLS estimates. The only significant explanatory variable is *parcoll*. In all the cases the estimate that the probability of PC ownership is about 0.22 higher if at least one parent attended college. □

9 More on specification and Data Issues

9.1 Functional form misspecification

Functional form misspecification happens when the model does not properly represent the relationship between dependent and the observed explanatory variables. It leads to biased estimators for all the coefficients. The amount of bias depends on the strength of the omitted variable and the correlation among included explanatory variables. Also using a misspecified form of a variable, e.g. level vs log, can also cause misspecification.

Misspecification is a minor issue if we have data on all the necessary variables for obtaining a functional relationship that fits the data well. This can be contrasted with the problem addressed in the next section, where a key variable is omitted on which we cannot collect data.

It can be difficult to pinpoint the precise reason that a functional form is misspecified. We can add quadratic terms of any significant variables to a model and perform a test of join significance. However, significant quadratic terms can be symptomatic of other functional form problems, such as using the level of a variable when the logarithm is more appropriate, or vice versa. Fortunately, in many cases, using logarithms of certain variables and adding quadratics are sufficient for detecting many important nonlinear relationships in economics. We can use F test for joint exclusion restrictions.

Ramsey's **regression specification error test** (RESET) is useful to detect general functional form misspecification. For the original model $y = \beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k + u$ if it satisfies MLR.4, then on nonlinear functions of the independent variables should be significant when added to this equation. If $\hat{y}$ denotes the OLS fitted values, we consider the following expanded equation $y = \beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k + \delta_1 \hat{y}^2 + \delta_2 \hat{y}^3 + \varepsilon$ and test the null hypothesis $H_0 : \delta_1 = 0, \delta_2 = 0$ using a F statistic. A significant F statistic, distributed as $F_{2,n-k-3}$, suggest some kind of functional problem.

Example 9.1. We estimate two models for housing prices.

```
df = woo.data('hprice1')
levmod = smf.ols(formula='price~lotsize+sqrft+bdrms', data=df).fit()
logmod = smf.ols(formula='lprice~llotsize+lsqrft+bdrms', data=df).fit()
print_results(levmod)
             coeff   pval  stderr   tval
Intercept -21.770  0.462  29.475 -0.739
```

```
bdrms      13.853  0.128   9.010  1.537
lotsize     0.002  0.002   0.001  3.220
sqrft       0.123  0.000   0.013  9.275
print_results(logmod)
           coeff   pval  stderr   tval
Intercept -1.297  0.050   0.651 -1.992
bdrms      0.037  0.183   0.028  1.342
llotsize   0.168  0.000   0.038  4.388
lsqrft     0.700  0.000   0.093  7.540
```

We now calculate the RESET statistics as follows:

```
from statsmodels.stats.diagnostic import linear_reset
linear_reset(levmod, power=3, use_f=True)
<F test: F=array([[4.66820553]]), p=0.01202171144289935, df_denom=82, df_num=2>
linear_reset(logmod, power=3, use_f=True)
<F test: F=array([[2.5650462]]), p=0.08307546624339071, df_denom=82, df_num=2>
```

A p-value of 0.012 for the level model is evidence of functional form misspecification; while a p-value of 0.084 for the log model shows we can reject the null at 5% significance level. On the basis of RESET log-log model is preferred. □

A drawback of RESET is that it really provided on direction on how to proceed if the model is rejected. RESET is simply a functional form test and should not be used, in general, for testing omitted variables or heteroskedasticity. RESET has no power for detecting omitted variables whenever they have expectations that are linear in the included independent variables in the model. Further, if the functional form is properly specified, RESET has no power for detecting heteroskedasticity.

Trying to decide whether an independent variable should appear in level or logarithmic form comes under non-nested models, e.g. comparing $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + u$ versus $y = \beta_0 + \beta_1 \log(x_1) + \beta_2 \log(x_2) + u$. We can construct a comprehensive model that contains each of the models as a special case and then test for restrictions using F statistics. For example, the comprehensive model $y = \gamma_0 + \gamma_1 x_1 + \gamma_2 x_2 + \gamma_3 \log(x_1) + \gamma_4 \log(x_2) + u$ can be tested first for $H_); \gamma_3 = 0, \gamma_4 = 0$ for the first equation and then $H_0 : \gamma_1 = 0, \gamma_2 = 0$ as a test for the second equation.

Another approach is the **Davidson-MacKinnon test**. If the first equation is true then the fitted values from the other model should be insignificant. Thus, to test the first equation, we first estimate the second model by OLS to obtain the fitted values $\hat{y}$ and then look at the t-statistic on $\hat{\hat{y}}$ in $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \theta_1 \hat{\hat{y}} + u$. A significant t statistic against a two-sided alternative is a rejection of the first equation. Similarly the second equation can be tested against the first.

```
from statsmodels.stats.diagnostic import compare_encompassing
levmod = smf.ols(formula='price~lotsize+sqrft+bdrms', data=df).fit()
```

```
logmod = smf.ols(formula='price~llotsize+lsqrft+bdrms', data=df).fit()
compare_encompassing(levmod, logmod)
       stat    pvalue  df_num  df_denom
x  7.861289  0.000753       2        82
z  7.050765  0.001494       2        82
```

As we see in this example, there is no clear winner, in fact both the tests reject each other. In the case none of the models are rejected we can use adjusted R-squared to choose between them. Also a rejection at DM test could be for a variety of functional form misspecifications. An even more difficult problem is obtaining non-nested tests when the competing models have different dependent variables, e.g., y versus $\log(y)$. Using goodness-of-fit test has to be done carefully as described in the previous chapters.

When a key variable is excluded because of data unavailability, we can mitigate the issue by obtaining a **proxy variable**, something that is correlated to the omitted variable, and include it in our regression. Say, the real functional form is $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3^* + u$, with data on x_3^* unavailable. Say data is available on a proxy variable x_3. At a minimum x_3 should have some relationship to x_3^*. This is captured by $x_3^* = \delta_0 + \delta_3 x_3 + \nu_3$, with $\delta_3 \neq 0$. The **plug-in solution to the omitted variables problem** simply regresses y on x_1, x_2, x_3. We need the following assumptions for the plug-in solution to provide consistent estimators of β_1 and β_2:

1. The error u is uncorrelated with x_1, x_2 and x_3^*. In addition u is uncorrelated with x_3. It means that it is x_3^* that directly affects y, not x_3.

2. The error ν_3 is uncorrelated with x_1, x_2, and x_3, i.e. x_3 is a good proxy for x_3^*. In other words $E(x_3^*|x_1, x_2, x_3) = E(x_3^*|x_3) = \delta_0 + \delta_3 x_3$.

Under these assumptions we get unbiased coefficient estimates for β_1 and β_2 from the original equation, and $\alpha_3 = \beta_3 \delta_3$. It is easy to see that if the proxy variable is dependent on other exogenous variables as well, we get biased estimates of the coefficients.

Example 9.2. As a method to account for omitted ability bias, we use IQ as a proxy variable to explain log wages.

```
df = woo.data('wage2')
m1 = smf.ols(formula='lwage~educ+exper+tenure+married+south+urban+black',
       data=df).fit()
m2 = smf.ols(formula='lwage~educ+exper+tenure+married+south+urban+black'+
       '+IQ', data=df).fit()
m3 = smf.ols(formula='lwage~educ+exper+tenure+married+south+urban+black'+
       '+IQ+I(educ * IQ)', data=df).fit()
          1        2        3
r2    0.253    0.263    0.263
ar2   0.247    0.256    0.256
dfr 927.000  926.000  925.000
dfm   7.000    8.000    9.000
             coeff                pval              stderr               \
```

```
                    1      2      3      1      2      3      1      2      3
I(educ * IQ)      NaN    NaN  0.000    NaN    NaN  0.375    NaN    NaN  0.000
IQ                NaN  0.004 -0.001    NaN  0.000  0.855    NaN  0.001  0.005
Intercept       5.395  5.176  5.648  0.000  0.000  0.000  0.113  0.128  0.546
black          -0.188 -0.143 -0.147  0.000  0.000  0.000  0.038  0.039  0.040
educ            0.065  0.054  0.018  0.000  0.000  0.653  0.006  0.007  0.041
exper           0.014  0.014  0.014  0.000  0.000  0.000  0.003  0.003  0.003
married         0.199  0.200  0.201  0.000  0.000  0.000  0.039  0.039  0.039
south          -0.091 -0.080 -0.080  0.001  0.002  0.002  0.026  0.026  0.026
tenure          0.012  0.011  0.011  0.000  0.000  0.000  0.002  0.002  0.002
urban           0.184  0.182  0.184  0.000  0.000  0.000  0.027  0.027  0.027
                  tval
                     1       2       3
I(educ * IQ)       NaN     NaN   0.888
IQ                 NaN   3.589  -0.182
Intercept       47.653  40.441  10.339
black           -5.000  -3.624  -3.695
educ            10.468   7.853   0.449
exper            4.409   4.469   4.378
married          5.107   5.148   5.173
south           -3.463  -3.054  -3.056
tenure           4.789   4.671   4.670
urban            6.822   6.791   6.835
m3.f_test(['IQ=0', 'I(educ * IQ)=0'])
>> <F test: F=array([[6.83181683]]), p=0.0011341745040462897, df_denom=925, df_num=2>
m3.f_test(['educ=0', 'I(educ * IQ)=0'])
>> <F test: F=array([[31.223758]]), p=7.547730433836841e-14, df_denom=925, df_num=2>
```

Our primary interest is in what happens to the estimated return to education. When we don't include IQ, the proxy for ability we estimate returns to education is 6.5%. If ability is positively correlated to *educ*, then this estimate is too high. When IQ is added to the equation, the return to education falls to 5.4%, which corresponds with our prior beliefs about omitted ability bias.

Model 2, shows that IQ does have a statistically significant, positive effect on earnings, after controlling for several other factors. It is clear that *educ* still have an important role in increasing earnings, even though the effect is not as large as originally estimated in model 1. The R-squared increases from 0.253 to 0.263, and adding IQ does not eliminate the race effect.

In model 3 we include the interaction term $educ * IQ$. We might think that the return to education is higher for people with more ability, but this turns about not to be the case: the interaction term is not significant and its addition makes *educ* and IQ individually insignificant while complicating the model. We do joint F test on IQ and the interaction term and then on *educ* and the interaction term; both suggesting the importance of the two factors. Therefore, model 2 is preferred. □

When we suspect that one or more of the independent variables is correlated with an omitted variable, but we have no idea how to obtain a proxy for that omitted variable, we can include as a control, the value of the dependent variables from an earlier time period. Using a **lagged**

dependent variable in a cross-sectional equation increases the data requirement.

Example 9.3. We estimate the crime from 46 cities in 1987. Crime rate from 1982 can be used as a lagged dependent variable. Out main aim is to estimate the ceteris paribus effect of $\log(lawexpc)$ on $\log(crmrte)$. Without the lagged crime rate the effect of expenditures on law enforcement is counterintuitive. The t-statistic on it is 1.178 and is not significant.

```
df = woo.data('crime2').dropna()
m1 = smf.ols(formula='lcrmrte~unem+llawexpc', data=df).fit()
m2 = smf.ols(formula='lcrmrte~unem+llawexpc+lcrmrt_1', data=df).fit()
print_compare({'1': m1, '2': m2})
          1       2
r2    0.057   0.680
ar2   0.013   0.657
dfr  43.000  42.000
dfm   2.000   3.000
           coeff           pval         stderr           tval
               1      2      1      2      1      2      1      2
Intercept  3.343  0.076  0.011  0.926  1.251  0.821  2.673  0.093
lcrmrt_1     NaN  1.194    NaN  0.000    NaN  0.132    NaN  9.038
llawexpc   0.203 -0.140  0.245  0.206  0.173  0.109  1.178 -1.285
unem      -0.029  0.009  0.375  0.661  0.032  0.020 -0.897  0.442
```

Adding the log of crime rate from five years earlier ($\log(crmrt\_1)$) has a large effect on the expenditure coefficient. The elasticity of the crime rate with respect to expenditures becomes -0.14, with $t = -1.285$. This is not strongly significant, but it suggests that a more sophisticated model with more cities in the sample could produce significant results.

Adding the lagged variable increases the R-squared significantly, as the current crime rate is strongly related to the past crime rate. The estimates indicates that if the crime rate in 1982 was 1% higher, then the crime rate in 1987 is predicted to be about 1.19% higher. We can test the hypothesis if this is different from 1 with a $t = (1.194 - 1)/0.1321.47$, which can't be rejected. □

A less structured, more general approach to multiple regression is to forego specifying models with unobservables. Rather, we fit models on the variables we have access to. It is better to control for a variable we have access to than to do nothing because we could not find suitable proxies. When are are primarily interested in predicting the outcome y, given a set of explanatory variables, it makes little sense to think in terms of 'bias' in estimated coefficients due to omitted variables. Instead, we should focus on obtaining a model that predicts as well as possible, and make sure we do not include as regressors variables that cannot be observed at the time of prediction.

9.2 Models with random slopes and measurement error

9.2.1 Random slopes

If the slopes differ we can think of a model like

$$y_i = a_i + \boldsymbol{x}_i \boldsymbol{b}_i,$$

which in an OLS correspond to $\boldsymbol{b}_i = \boldsymbol{\beta}$ and $a_i = u_i$. This is called the **random coefficient model** or **random slope model**. We can hope to estimate the average slope and intercept, where the average is across the population. Therefore, $\alpha = E(a_i)$ and $\boldsymbol{\beta} = E(\boldsymbol{b}_i)$. $\boldsymbol{\beta}$ is the average of the partial effect of $\boldsymbol{x}$ on y, called the **average partial effect**. If we write $a_i = \alpha + c_i$ and $\boldsymbol{b}_i = \boldsymbol{\beta} + \boldsymbol{d}_i$, and $E(c_i) = 0$ and $E(\boldsymbol{d}_i) = 0$, then $\boldsymbol{d}_i$ is the individual-specific deviation from the APE. This gives $y_i = \alpha + \boldsymbol{\beta}\boldsymbol{x}_i + u_i$, where $u_i = c_i + \boldsymbol{x}_i \boldsymbol{d}_i$. Hence we can write the random coefficient model as a constant coefficient model but with the error term containing an interaction between and unobservable $\boldsymbol{d}_i$ and the observed explanatory variables $\boldsymbol{x}_i$. If $E(u_i|\boldsymbol{x}_i) = 0$ then OLS will give an unbiased estimate of α and $\boldsymbol{\beta}$. This corresponds to $E(c_i|\boldsymbol{x}_i) = E(c_i) = 0$ and $E(\boldsymbol{d}_i|\boldsymbol{x}_i) = E(\boldsymbol{d}_i) = 0$ which is by construction assumed to be true. Thus OLS consistently estimates the population average of those slopes when they are mean independent of the explanatory variables.

However, the error term almost certainly contains heteroscedasticity. For illustration, take the univariate case. We can write $Var(u_i|x_i) = \sigma_c^2 + \sigma_d^2 x_i^2$, where $Var(c_i|x_i) = \sigma_c^2$ and $Var(d_i|x_i) = \sigma_d^2$ and $Cov(c_i, d_i|x_i) = 0$. We can use OLS and compute heteroskedasticity-robust standard errors and test statistics, or we can estimate the variance function and apply weighted least squares. Multivariate case would be similarly heteroskedastic. Finally, if we allow $\boldsymbol{b}_i$ to depend on cross observable explanatory variables as well as unobservables it is equivalent of introducing interaction terms.

9.2.2 Measurement error in y

We begin the investigation of measurement error with the case where only the dependent variable is measured with error. Let y^* denote the variable that we would like to explain but what is available after measurement is $y = y^* + e_0$, where e_0 is the **measurement error** in the population. The intended regression model is $y^* = \beta_0 + \boldsymbol{x}\boldsymbol{\beta} + u$ and we assume it satisfies the Gauss-Markov assumptions. For a random draw i from the population we can write $e_{i0} = y_i - y_i^*$. To obtain an estimable model we plug $y^* = y - e_0$ into the equation to get $y = \beta_0 + \boldsymbol{x}\boldsymbol{\beta} + u + e_0$. This model can be estimated using usual OLS. This produces unbiased and consistent estimates of β_j when $E(e_0) = 0$ and the measurement error e_0 in y is statistically independent of each explanatory variable x_j. Further, the usual OLS inference procedures (t, F, and LM statistics) are valid. The general assumption is that e_0 and u are uncorrelated, which means $Var(u + e_0) = \sigma_u^2 + \sigma_0^2 > \sigma_u^2$. This means the measurement error in the dependent variable results in larger error variance.

When the dependent variable is in logarithmic form, so that $log(y^*)$ is the dependent variable, it is natural for the measurement error equation to be of the form $log(y) = log(y^*) + e_0$.

This follows from a multiplicative measurement error for y : $y = y^* a_0$, where $a_0 > 0$ and $e_0 = \log(a_0)$.

9.2.3 Measurement error in x

We begin with a simple regression model $y = \beta_0 + \beta_1 x_1^* + u$, which satisfies the first four Gauss-Markov assumptions. Instead of x_1^* we actually have $x_1 = x_1^* + e_1$ available, where e_1 is the measurement error; with the natural assumption that $E(e_1) = 0$. We maintain the assumption that u is uncorrelated with x_1^* and x_1, implying $E(y|x_1^*, x_1) = E(y|x_1^*)$, which says that x_1 does not affect y after x_1* has been controlled for. This also implies $Cov(u|e_1) = 0$.

We want to know the properties of the OLS when we substitute x_1 for x_1^* giving equation $y = \beta_0 + \beta_1 x_1 + (u - \beta_1 e_1)$. If we assume $Cov(x_1, e_1) = 0$ then $Cov(x_1^*, e_1) \neq 0$ but $E(u - \beta_1 e_1) = 0$ and $Cov(u - \beta_1 e_1, x_1) = 0$ with $Var(u - \beta_1 x_1) = \sigma_u^2 + \beta_1^2 \sigma_{e_1}^2$. Hence, OLS estimates are unbiased and consistent, though with higher variance, which is a natural result of additional measurement error. The case is same for multiple regression as well.

The **classical errors-in-variables** assumption is a more appropriate model of measurement error which assumes $Cov(x_1^*, e_1) = 0$ for $x_1 = x_1^* + e_1$. Under this assumption we have $Cov(x_1, e_1) = \sigma_{e_1}^2$. This will cause the OLS estimates to be biased and inconsistent since $Cov(u - \beta_1 e_1, x_1) = -\beta_1 \sigma_{e_1}^2 \neq 0$. For large samples $plim(\hat{\beta}_1) = \beta_1 \frac{Var(x_1^*)}{Var(x_1)} \leq \beta_1$. This is called the **attenuation bias** in the OLS due to classical error-in-variables. For multiple regression $y = \beta_0 + \beta_1 x_1^* + \beta_2 x_2 + \beta_3 x_3 + u$, OLS will be biased and inconsistent, in general, for all OLS estimators, not just $\hat{\beta}_1$. The attenuation bias for estimating β_1 for large sample case is $plim(\hat{\beta}_1) = \beta_1 \left(\frac{\sigma_{r_1^*}^2}{\sigma_{r_1^*}^2 + \sigma_{e_1}^2} \right)$, where r_1^* is the population error in the equation $x_1^* = \alpha_0 + \alpha_1 x_2 + \alpha_2 x_3 + r_1^*$. In the special case that x_1^* is uncorrelated with x_2 and x_3, $\hat{\beta}_2$ and $\hat{\beta}_3$ are consistent. But that is rare in practice. One consequence of the downward bias is that a test of $H_0 : \beta_1 = 0$ will have less chance of detecting $\beta_1 > 0$.

CEV assumption while more believable, is a strong assumption and may not always be true. The truth is probably somewhere in between, and if e_1 is correlated with both x_1^* and x_1, OLS is inconsistent. We will show in a later section that under certain assumptions, the parameters can be consistently estimated in the presence of general measurement error.

9.3 Missing Data, nonrandom samples and outliers

- *Missing Data*: If data are missing for an observation on either the dependent or one of the independent variables, then it can't be use in standard multiple regression and is generally dropped. If the data are missing at random, then the size of the random sample available from the population is simply reduced, making the estimator less precise but still unbiased. There are ways to use the information on observations where only some variables are missing, but in practice the improvements are usually slight and hence not usually done.

- *Nonrandom Samples*: If the data is missing due to nonrandom sample we need to worry about any possible bias it might induce. Fortunately, in the case of **exogenous sample selection**, i.e. sample selection based on the independent variables there is no bias of inconsistency caused in the OLS. The reason OLS on the nonrandom sample is unbiased is that the regression function is same for any subset of the population described by independent variable, provided there is enough variation in the independent variables in the sub population. It simply reduces the sample size. The situation is much different for **endogenous sample selection**, i.e. sample selection based on the dependent variable. This cause bias and inconsistency because the population regression function is not the same for different subset of dependent variables value. In **stratified sampling** population is divided into non overlapping, exhaustive groups/strata and then sampled with different frequencies than dictated by their population representation. Again, if the stratification is based on exogenous variables it causes no bias or inconsistency, which is not the case if stratification is based on endogenous variable.

- *Outliers and influential observations*: An observation is an influential observation if dropping it from the analysis changes the key OLS estimates by a large amount. OLS is susceptible to outlying observations too because it minimizes the sum of squared residuals. From practical perspective, outlying observations can occur for two reasons. The easiest case to deal with is when a mistake has been made in entering data, like adding extra zeros or misplacing a decimal. Computing statistical summary on the data helps in catching mistakes in data entry. Outliers can also arise when sampling from a small population if one or several members of the population are very different from rest of the population. OLS results reported with and without outlying observations can serve as a good investigation tool. Defining *leverage* of an observation, formalizes the notion that an observation has a large or small influence on the OLS estimates.

9.4 Least Absolute deviations estimation

Least absolute deviations (LAD) is a method that is less sensitive to the outliers. We minimize the sum of the absolute values of the residuals,

$$\underset{b_1,b_1,\ldots,b_k}{min} \sum_{i=1}^{n} |y_i - b_0 - b_1 x_{i1} - \ldots - b_k x_{ik}|.$$

Because LAD does not give increasing weight to larger residuals, it is must less sensitive to changes in the extreme values of the data than OLS. LAD is designed to estimate the parameters of the conditional median of y given $x_1, x_2, \ldots, x_k$ rather than conditional mean. All estimate of LAD are justified only as the sample size grows.

There is no closed form solution for LAD. But a more important drawback to LAD is that it does not always consistently estimate the parameters appearing in the conditional mean function $E(y|x_1, \ldots, x_k)$. OLS produces unbiased and consistent estimators of the parameters whether or not the error distribution is symmetric. When the error distribution

is asymmetric LAD estimates can be very different form OLS, reflecting the different between the mean and median and not necessarily an outlier. If we assume u is independent of $(x_1, \ldots, x_k)$ then the OLS and LAD slope estimates should differ only by sampling error whether or not the distribution of u is symmetric. The intercept can, however, differ to reflect the difference in mean and median. Independent rules out heteroskedasticity. But in practice it is a big problem with asymmetric distributions.

Because LAD estimates median, it is easy to obtain partial effects - and predictions - using monotonic transformations. If $\log(y) = \beta_0 + \boldsymbol{x}\boldsymbol{\beta} + u$ with $Med(u|\boldsymbol{x}) = 0$, then $Med(\log(y)|\boldsymbol{x}) = \beta_0 + \boldsymbol{x}\boldsymbol{\beta}$. Conditional median passes through increasing functions, i.e. $Med(y|\boldsymbol{x}) = \exp(\beta_0 + \boldsymbol{x}\boldsymbol{\beta})$. In other words, the partial effect of x_j in the log linear equation can be used to uncover the partial effect in the nonlinear model. This holds for any distribution of u till $Med(u|\boldsymbol{x}) = 0$, and we need not assume u and $\boldsymbol{x}$ are independent. By contrast, with a linear model for $E(\log(y)|\boldsymbol{x})$, in general, there is no way to uncover $E(y|\boldsymbol{x})$, unless we make a full distributional assumption for u given $\boldsymbol{x}$.

Least absolute deviations is a special case of what is often called *robust regression*, which is insensitive to extreme observations. Effectively, observations with large residuals are given less weight than in least squares. LAD is not a robust estimator of the conditional mean because it requires extra assumptions in order to consistently estimate the conditional mean parameters. Either the distribution of u given $(x_1, \ldots, x_k)$ has to be symmetric about zero, or u must be independent of $(x_1, \ldots, x_k)$. Neither of these is required for OLS. Finally, LAD is also a special case of *quantile regression*, which is used to estimate the effect of the x_j on different parts of the distribution - not just the median or mean.

Example 9.4. We look at infant mortality data from 50 stats and DC to study the effect of outliers and how robust regression can help. The variable $infmort$ is the number of deaths within the first year per 1000 live births, *pcinc* is per capital income, *physic* is the physicians per 100,000 members of the civilian population, and *popul* is the population.

```
df = woo.data('infmrt')
m1 = smf.ols(formula='infmort~lpcinc+lphysic+lpopul', data=df).fit()
m2 = smf.quantreg(formula='infmort~lpcinc+lphysic+lpopul', data=df).fit(q=0.5)
print_compare({'ols': m1, 'lad': m2})
            coeff           pval         stderr          tval
              ols     lad    ols    lad     ols    lad    ols    lad
Intercept  36.226  34.393  0.001  0.000  10.135  8.826  3.574  3.897
lpcinc     -4.884  -3.033  0.000  0.008   1.293  1.126 -3.777 -2.693
lphysic     4.028  -0.030  0.000  0.969   0.891  0.776  4.521 -0.039
lpopul     -0.054   0.565  0.776  0.001   0.187  0.163 -0.286  3.458
```

For the OLS model, higher per capital income is, as expected, estimated to lower infant mortality. But more physicians per capital is associated with higher infant mortality rates, which is counterintuitive. Infant mortality rate does not appear to be related to population size.

The DC data is an outlier here. It is unusual in that it has pockets of extreme poverty and great wealth in a small area. We do a LAD regression to find that more physicians per capita lowers infant mortality, but the estimates are statistically not different from 0. The effect of per capital income has fallen. We also find that with robust regression, infant mortality rates are higher in more populous states and the relationship is very significant. □

10 Basic Regression Analysis with Time Series Data

Temporal ordering is the main difference versus the cross-sectional data. Past can effect the future, but not vice versa. Like in OLS time series is clearly an outcomes of random variables. Formally, a sequence of random variables indexed by time is called a **stochastic process** or a **time series process**. When we collect a time series data set, we obtain one possible outcome, or realization, of that stochastic process. We can only see a single realization. The set of all possible realizations of a time series process plays the role of the population in cross-sectional analysis.

Suppose that we have time series data available on variable y and set of variables $\boldsymbol{z}$, where y_t and $\boldsymbol{z}_t$ are dated contemporaneously. A **static model** relating y to $\boldsymbol{z}$ is $y_t = \beta_0 + \boldsymbol{z}_t\boldsymbol{\beta} + u_t$, for $t = 1, 2, \ldots, n$. We are modelling a contemporaneous relationship between y and $\boldsymbol{z}$. Usually, a static model is postulated when a change in vector $\boldsymbol{z}$ at time t is believed to have an immediate effect on y, $\Delta y_t = \Delta \boldsymbol{z}_t\boldsymbol{\beta}$, when $\Delta u_t = 0$. In a **finite distributed lag (FDL) model**, we allow one or more variables to affect y with a lag. The model $y_t = \alpha_0 + \delta_0 z_t + \delta_1 z_{t-1} + \delta_2 z_{t-2} + u_t$, is an FDL *of order two*. δ_0 is called the **impact propensity**; it is the immediate change in y due to one-unit increase in z at time t. When we graph the δ_j as a function of j, we obtain the **lag distribution**, which summarizes the dynamic effect that a temporary increase in z has on y. The sum of coefficients on current and lagged z, $\delta_0 + \delta_1 + \delta_2$, is the long-run change in y given a permanent unit increase in z and is called **long-run propensity**.

A finite distributed lag model of order q is written as

$$y_t = \alpha_0 + \delta_0 z_t + \delta_1 z_{t-1} + \ldots + \delta_q z_{t-q} + u_t.$$

Static model is a special case of this by setting $\delta_1, \delta_2, , \delta_q$ equal to zero. In general case the lag distribution can be plotted by graphing the estimated δ_j as a function of j. For an horizon h, we can define the **cumulative effect** as $\sum_{i=0}^{h} \delta_i$, which is interpreted as the change in the expected outcome h periods after a permanent, one-unit increase in x. The long run propensity, LRP is simply $\sum_{i=0}^{q} \delta_i$. Because of the often substantial correlation in z at different lags - that is, due to multicollinearity - it can be difficult to obtain precise estimates of the individual δ_j. Interestingly, even when the δ_j cannot be precisely estimated, we can often get good estimates of the LRP. Finally, we can have more than on explanatory variables appearing with lags, or we can add contemporaneous variables to an FDL model.

10.1 Finite Sample properties of OLS under classical assumptions

We let $\boldsymbol{x}_t = (x_{t1}, x_{t2}, \ldots, x_{tk})$ denote the set of all independent variables in the equation at time t. Further let $\boldsymbol{X}$ denote the collection of all independent variables for all time periods. We state the **Gauss-Markov assumptions for time series regression** analysis.

- TS.1 Linear in parameters - The stochastic process $\{(\boldsymbol{x}_t, y_t) : t = 1, 2, \ldots, n\}$ follows the linear model $y_t = \beta_0 + \beta_1 x_{t1} + \ldots + \beta_k x_{tk} + u_t$, where $\{u_t : t = 1, 2, \ldots, n\}$ is the sequence of errors and n is the number of observations.

- TS.2 No perfect collinearity - In the sample and underlying time series process, no independent variable is constant nor a perfect linear combination of the others.
- TS.3 Zero conditional mean - For each t, the expected value of the error u_t, given the explanatory variables for all time periods, is zero. That is, $E(u_t|\boldsymbol{X}) = 0$, for $t = 1, 2, \ldots, n$. This means error at time t, u_t is uncorrelated with each explanatory variable in *every* time period.

If u_t is independent of $\boldsymbol{X}$ and $E(u_t) = 0$, then assumption TS.3 automatically holds. Assumption TS.3 implies x_{tj} to be **contemporaneously exogenous** given by $E(u_t|\boldsymbol{x}_t) = 0$, i.e. $Cov(x_{tj}, u_t) = 0$, for all j. Assumption TS.3 is stronger than it requiring explanatory variables to be **strictly exogenous**, i.e. u_t must be uncorrelated with x_{sj}, even when $s \neq t$. Contemporaneous exogeneity is sufficient for consistency but for unbiasedness we need strict exogeneity. Under random sampling, u_i was automatically independent of the explanatory variables for observations other than i in the case of OLS. In the time series context random sampling is almost never appropriate. It is important to note that assumption TS.3 puts no restriction on correlation in the independent variables or in the u_t across time, but only requires the average value of u_t to be unrelated to the independent variables in all time periods.

Two main reasons why assumption TS.3 can fail is omitted variables and measurement error in some of the regressors. The other subtle point is that strict exogeneity excludes the possibility that changes in the error term today can cause future changes in z, effectively ruling out feedback from y to future values of z. Feedback from u to future z is generally a practical concern in many problems and the strict exogeneity assumption may be violated.

Theorem 10.1. *(Unbiasedness of OLS) Under the assumptions TS.1, TS.2, and TS.3, the OLS estimators are unbiased conditional on $\boldsymbol{X}$, and therefore unconditionally as well $E(\hat{\beta}_j) = \beta_j$, for $j = 0, 1, \ldots, k$.*

We have been able to drop the random sampling assumption by assuming that, for each t, u_t has zero mean given explanatory variables at all the time periods. If this assumption does not hold, OLS cannot be shown to be unbiased.

- TS.4 Homoskedasticity - Conditional on $\boldsymbol{X}$, the variance of u_t is the same for all t, i.e. $Var(u_t|\boldsymbol{X}) = Var(u_t) = \sigma^2$, for $t = 1, 2, \ldots, n$.
- TS.5 No serial correlation - Conditional on $\boldsymbol{X}$, the errors in two different time periods are uncorrelated, i.e $Cor(u_t, u_s|\boldsymbol{X}) = 0$, for all $t \neq s$.

When $Var(u_t|\boldsymbol{X})$ does depend on $\boldsymbol{X}$, it often depends on the explanatory variables at time t, $\boldsymbol{x}_t$. Assumption TS.5 is new for time series analysis. When this is false, we say the errors suffer from **serial correlation** or **autocorrealtion**. Importantly, assumption TS.5 assumes nothing about temporal correlation in the independent variables, as it only talks about the error term. This was taken care of by random sampling in the cross-sectional regression case.

Theorem 10.2. *(OLS sampling variances) Under the time series Gauss-Markov assumptions TS.1 through TS.5, $Var(\hat{\beta}_j|\boldsymbol{X}) = \frac{\sigma^2}{SST_j(1-R_j^2)}$, for $j = 1, \ldots, k$, where SST_j is the*

total sum of squares of x_{tj} and R_j^2 is the R-squared from the regression of x_j on the other independent variables.

Theorem 10.3. *(Unbiased estimation of σ^2) Under assumptions TS.1 through TS.5, the estimator $\hat{\sigma}^2 = \frac{SSR}{n-k-1}$ is an unbiased estimator of σ^2.*

Theorem 10.4. *(**Gauss-Markov theorem**) Under the assumptions TS.1 through TS.5, the OLS estimators are the best linear unbiased estimators conditional on $\boldsymbol{X}$.*

In order to use the usual OLS standard errors, t statistics, and F statistics, in finite sample, we need to add a final assumption.

- TS.6 Normality - The errors u_t are independent of $\boldsymbol{X}$ and are independently and identically distributed as $\mathcal{N}(0, \sigma^2)$.

Assumption TS.6 implies TS.3, TS.4 and TS.5, but is stronger because of the independence and normality assumptions.

Theorem 10.5. *(Normal sampling distributions) Under the assumptions TS.1 through TS.6, the CLM assumptions for time series, the OLS estimators are normally distributed, conditional on $\boldsymbol{X}$. Further, under the null hypothesis, each t statistic has a t distribution, and each F statistic has an F distribution. The usual construction of confidence intervals is also valid.*

Thus if these six assumptions are true, everything for before can be used for time series as well. Note that the CLM assumptions for time series data are more restrictive than those for cross-sectional data - in particular, the strict exogeneity and no serial correlation assumptions can be unrealistic. Nevertheless, the CLM framework is a good starting point for many application.

Example 10.1. We determine if there is a tradeoff between unemployment and inflation, we can test $h_0 : \beta_1 = 0$ against $H_1 : \beta_1 < 0$. If CLM holds we can use the usual OLS t statistic.

```
df = woo.data('phillips')
df = df[df.year <= 1996]
m1 = smf.ols(formula='Q("inf")~unem', data=df).fit()
m1.summary2()
```

```
                    Results: Ordinary least squares
=================================================================
Model:              OLS              Adj. R-squared:     0.033
Dependent Variable: Q("inf")         AIC:                252.8529
Date:               2020-10-29 22:24 BIC:                256.6365
No. Observations:   49               Log-Likelihood:     -124.43
Df Model:           1                F-statistic:        2.616
Df Residuals:       47               Prob (F-statistic): 0.112
R-squared:          0.053            Scale:              9.8004
-----------------------------------------------------------------
             Coef.    Std.Err.     t      P>|t|     [0.025   0.975]
-----------------------------------------------------------------
Intercept    1.4236     1.7190   0.8282   0.4118   -2.0346   4.8818
unem         0.4676     0.2891   1.6174   0.1125   -0.1140   1.0493
-----------------------------------------------------------------
```

We get the following model

$$\widehat{inf}_t = \underset{(1.72)}{1.42} + \underset{(0.289)}{0.468}\ unem_t,$$

with 47 degrees of freedom of the residual and $R^2 = 0.053$. we see that the t statistic for $\hat{\beta}_1$ is 1.62 and hence not statistically significant with a p-value against a two-sided alternative of 0.11. We will later see that CLM assumptions are not valid here. □

10.2 Functional Form, Dummy variables, and Index numbers

The natural logarithm form: time series regression with constant percentage effects appear often in applied work. We can use logarithmic functional form in distributed lag models too. In the equation relating money demand M_t and gross domestic product GDP_t $log(M_t) = \alpha_0 + \sum_{i=0}^{4} \delta_i log(GDP_i) + u_t$, the impact propensity δ_0 is also called the **short-run elasticity**: it measures the immediate percentage change in money demand given a 1% increase in GDP. The long run propensity, $\sum_{i=0}^{4} \delta_i$, is sometimes called **long-run elasticity**: it measures the percentage increase in money demand after four quarters given a permanent 1% increase in GDP.

Since unit of observation is time, a dummy variable represents whether, in each time period, a certain event has occurred. Often, dummy variables are used to isolate certain periods that may be systematically different from other periods covered by a data set.

Example 10.2. We look at the general fertility rate, number of children born to every 1000 women of childbearing age(gfr) from 1913 to 1984. We account for the factors - average dollar value of personal tax exemption (pe) and the binary variables $ww2$, world war 2 years 1941 to 1945, and $pill$, availability of contraception since 1963. Each variable is significant at 1% level against a two-sided alternative.

```
df = woo.data('fertil3')
m1 = smf.ols(formula='gfr~pe+ww2+pill', data=df).fit()
m1.summary2()
m2 = smf.ols(formula='gfr~pe+pe_1+pe_2+ww2+pill', data=df).fit()
m2.summary2()
     static  lagdist
r2    0.473    0.499
ar2   0.450    0.459
dfr  68.000   64.000
dfm   3.000    5.000
            coeff           pval         stderr            tval
           static lagdist static lagdist static lagdist  static lagdist
Intercept  98.682  95.870  0.000   0.000  3.208   3.282  30.760  29.211
pe          0.083   0.073  0.007   0.565  0.030   0.126   2.784   0.579
pe_1          NaN  -0.006    NaN   0.970    NaN   0.156     NaN  -0.037
pe_2          NaN   0.034    NaN   0.790    NaN   0.126     NaN   0.268
pill      -31.594 -31.305  0.000   0.000  4.081   3.982  -7.742  -7.862
ww2       -24.238 -22.126  0.002   0.043  7.458  10.732  -3.250  -2.062
```

We see that the fertility rate was lower during world war 2: given pe, there were about 24 fewer births for every 1000 women of childbearing age, which is a large reduction. Similarly, the fertility rate has been substantially lower since the introduction of the birth control pill. The coefficient of pe implies that a \$12 increase in pe increases gfr by about one birth per 1000 women of childbearing age. This effect is hardly trivial.

We then look at how the fretility rate may react to changes in pe with a lag by estimating a distributed lag model. This regression has two less data points with two more variables and hence the degree of freedom for residuals reduces from 68 to 64. The coefficients on the pe variables are estimated very imprecisely, each of them being insignificant. It turns out that there is substantial correlation between pe_t, pe_{t-1} and pe_{t-2}, and this multicollinearity makes it difficult to estimate the effect at each lag.

```
m2.f_test(['pe=0','pe_1=0','pe_2=0'])
<F test: F=array([[3.97296405]]), p=0.011652005303126168, df_denom=64, df_num=3>
m2.f_test(['pe_1=0','pe_2=0'])
<F test: F=array([[0.05343035]]), p=0.9480142085486433, df_denom=64, df_num=2>
```

However, pe_t, pe_{t-1} and pe_{t-2} are jointly significant: the F statistic has a p-value of 0.012. Thus, pe does have an effect on gfr, but we do not have good enough estimates to determine whether it is contemporaneously or with a one- or two-year lag. Actually, pe_{t-1} and pe_{t-2} are jointly insignificant in this equation with p-value of 0.95, so we would be justified in using the static model.

To estimate LRP we add the coefficients on pe to get ≈ 0.101. However we do not have enough information to obtain the standard error of this estimate. We use the trick of substituting $\theta_0 = \delta_0 + \delta_1 + \delta_2$ and substituting $\delta_0 = \theta_0 - \delta_1 - \delta_2$ in the regression equation. We can then obtain $\hat{\theta}_0$ and its standard error by regressing gfr_t on $pe_t, (pe_{t-1} - pe_t), (pe_{t-2} - pe_t), ww2_t$, and $pill_t$.

```
m3 = smf.ols(formula='gfr~pe+I(pe_1-pe)+I(pe_2-pe)+ww2+pill', data=df).fit()
m3.summary2()
-------------------------------------------------------------------
                Coef.   Std.Err.    t    P>|t|   [0.025   0.975]
-------------------------------------------------------------------
Intercept       95.8705   3.2820 29.2114 0.0000  89.3140 102.4270
pe               0.1007   0.0298  3.3795 0.0012   0.0412   0.1603
I(pe_1 - pe)    -0.0058   0.1557 -0.0371 0.9705  -0.3168   0.3052
I(pe_2 - pe)     0.0338   0.1263  0.2679 0.7896  -0.2184   0.2861
ww2            -22.1265  10.7320 -2.0617 0.0433 -43.5661  -0.6869
pill           -31.3050   3.9816 -7.8625 0.0000 -39.2591 -23.3509
-------------------------------------------------------------------
```

The coefficient is $\hat{\theta}_0 = 0.101$ for pe_t and $se(\hat{\theta}_0) = 0.30$. Therefore the t statistic for $\hat{\theta}_0$ is about 3.38, so $\hat{\theta}_0$ is statistically different from zero at small significance levels. Even though

none of the $\hat{\delta}_j$ is individually significant, the LRP is very significant. The 95% confidence interval for the LRP is about 0.0412 to 0.1603. □

Binary variables are the key component in what is called an **event study**. The goal is to see whether a particular event influences some outcome. A simple version of an equation used for such event studies is $R_t^f = \beta_0 + \beta_1 R_t^m + \beta_2 d_t + u_t$, where R_t^f is the stock return for firm f during period t, R_t^m is the market return, and d_t might is a dummy variable indicating when the event occurred. Sometimes, multiple dummy variables are used. For example dummy variable few days/weeks before and after an event might detect the presence of inside information and after effects respectively. **Index numbers** are also widely used a aggregated series with some **base value** in the **base period**. They are used to differentiate between nominal (current) and real (constant) economic variables. It is easy to change the base period for any index number, and sometimes we must do this to a given index number reported with different base years to a common base year for direct comparison.

An important example of an index number is a *price index*, such as the consumer price index (CPI). CPI is only meaningful when compared across different time periods. In addition to being used to compute inflation rates, price index are necessary for turning a time series measured in nominal dollars into real dollars. Most economic behavior is assumed to be influenced by real, not nominal, variables. If w denotes the hourly wage in nominal dollars and $p = CPI/100$, the *real wage* is simply w/p. This wage is measured in dollars for the base period of the CPI. Further, standard measures of economic output are in real terms. The most important of these is *gross domestic product* (GDP), which when reported in popular press, is always *real* GDP growth.

If we use real dollar variable in the regression in combination with natural logarithms in the equation $log(hours) = \beta_0 + \beta_1 log(w/p) + u$, we can write it as $log(hours) = \beta_0 + \beta_1 log(w) + \beta_2 log(p) + u$, but with the restriction that $\beta_2 = -\beta_1$. Therefore, the assumption that only the real wage influences labor supply imposes a restriction on the parameters of the model consisting of both nominal wage and price index. If $\beta_2 \neq \beta_1$, then the price level has an effect on labor supply. Since, the magnitudes of index numbers are not especially informative, they often appear in logarithmic form, so that regression coefficients have percentage change interpretations. Use of interaction terms is also very common in time series analysis.

Example 10.3. *(It's the economy stupid!)* Economist are interested in explaining presidential election outcomes in terms of economic performance. Proportion of the two-party vote going to the Democratic candidate using data from 1916 through 1992 for a total of 20 observations, can be modeled using variables:- $partyWH$: +1 if a Democrat is in the White House or -1 if a Republican is in the White House; $incum$: +1 if a Democratic incumbent is running, -1 if a Republican incumbent is running, and zero otherwise; $gnews$: number of quarters, during the administration's first 15 quarters, when the quarterly growth in real per capita output was above 2.9% annually; and inf: the average annual inflation rater over the first 15 quarters of the administration.

We are interested in the interaction terms $partyWH.gnews$ and $partyWH.inf$. $\beta_{partyWH.gnews}$

measures the effect of good economic news on the party in power; we expect this coefficient to be positive. Similary $\beta_{partyWH.inf}$ measures the effect that inflation has on the party in power; we expect this coefficient to be negative.

```
df = woo.data('fair')
m1 = smf.ols(formula='demvote~partyWH+incum+partyWH:gnews+partyWH:Q("inf")',
     data=df).fit()
m1.summary2()
==================================================================
Model:              OLS              Adj. R-squared:     0.574
Dependent Variable: demvote          AIC:                -62.7861
Date:               2020-10-30 13:33 BIC:                -57.5635
No. Observations:   21               Log-Likelihood:     36.393
Df Model:           4                F-statistic:        7.725
Df Residuals:       16               Prob (F-statistic): 0.00115
R-squared:          0.659            Scale:              0.0024008
------------------------------------------------------------------
                Coef.   Std.Err.     t     P>|t|    [0.025   0.975]
------------------------------------------------------------------
Intercept       0.4842    0.0115  42.2101  0.0000   0.4599   0.5085
partyWH        -0.0324    0.0375  -0.8623  0.4013  -0.1120   0.0472
incum           0.0564    0.0230   2.4490  0.0262   0.0076   0.1052
pWHgnews        0.0097    0.0038   2.5389  0.0219   0.0016   0.0177
pWHinf         -0.0083    0.0031  -2.6717  0.0167  -0.0150  -0.0017
------------------------------------------------------------------
```

All coefficients, except $\beta_{partyWH}$, are statistically significant at the 5% level. incumbency is worth about 5.4 percentage points in the share of the vote. Further, as expected the economic news variable has a positive effect: one more quarter of good news is worth about 1.1 percentage points. Inflation, again as expected, has a negative effect: if average inflation of 1% higher makes the party in power lose about 0.83% percentage points of the two-party vote.

We could use this equation to predict the outcome of 1996 presidential election between Bill Clinton, the Democrat, and Bob Dole, the Republican. Because Clinton ran as an incumbent, $partyWH = 1$ and $incum = 1$. To predict the election outcome, we need the variables $gnews$ and inf. During Clinton's first 15 quarters in office, the annual growth rate of per capita real GDP exceeded 2.9% three times, so $gnews = 3$. Further, the average inflation rate from the fourth quarter in 1991 to third quarter in 1996 was 3.019. This gives $\widehat{demvote} = 0.481 - 0.0435 + 0.0544 + 0.0108 \times 3 = 0.0077 \times 3.019 \approx 0.5011$. Therefore, Clinton was predicted to receive 50.1% of the two party vote and hence a majority. He, in fact, got 54.65% of the two-party vote share. □

10.3 Trend and Seasonality

Many economic time series have a common tendency of growing over time, called **time trend**. Ignoring the fact that two sequences are trending in the same or opposite directions can lead us to falsely conclude that changes in one variable are actually caused by changes

in another variable. This is called **spurious regression problem**. Very often, two time series processes appear to be correlated only because they are both trending over time for reasons related to other unobserved factors. These can be modelled using a **linear time trend** $y_t = \alpha_0 + \alpha_1 t + e_t$, with $= 1, 2, \ldots$, where $\{e_t\}$ is an independent, i.i.d. sequence with $E(e_t) = 0$ and $Var(e_t) = \sigma_e^2$. This can be interpreted using $\Delta y_t = \alpha_1$, if $\Delta e_t = 0$, i.e. α_1 measures the change in y_t from one period to the next due to passage of time. If $\{e_t\}$ is an i.i.d. sequence, then $\{y_t\}$ is an independent, though not identically, distributed sequence. A more realistic characterization of trending time series allows $\{e_t\}$ to be correlated over time, which we will discuss in later sections.

Many economic time series are better approximated by an **exponential trend**, when the series have the same average growth rate from period to period. They can be modeled by $log(y_t) = \beta_0 + \beta_1 t + e_t$, where $t = 1, 2, \ldots$. This can be interpreted as $\Delta log(y_t) = \beta_1$ for all t, if $\Delta e_t = 0$. Time trends can be more complicated as well. For example, we might have a quadratic term like $y_t = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + e_t$. It is advisable to be as parsimonious as possible.

Including the trending variable in a regression does not violate any of the CLM assumptions TS.1 to TS.6. Consider a model where two factors x_{t1} and x_{t2} affect y_t. In addition, there are unobserved factors that are systematically growing or shrinking over time. A model to capture this is $y_t = \beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} + \beta_3 t + u_t$. The trend term recognizes that y_t may grow or shrink over time for reasons essentially unrelated to x_{t1} and x_{t2}. If the equation satisfies TS.1, TS.2 and TS.3, then omitting t from the regression and regressing y_t on x_{t1} and X_{t2} will generally yield biased β_1 and β_2: we have effectively omitted an important variable from the regression. This is especially true if x_{t1} and x_{t2} are themselves trending, because they can then be highly correlated with t.

Example 10.4. We want to investigate how per capita housing investment *invpc* is affected by housing price index *price*. Doing a simple regression on their logarithmic form gives

$$\widehat{log(invpc)} = \underset{(0.043)}{-0.550} + \underset{(0.382)}{1.241}\ log(price)$$

with very significant $\hat{\beta}_{log(price)}$. Both *invpc* and *price* have upward trends.

```
df = woo.data('hseinv').set_index('year')
m1 = smf.ols(formula='linvpc~lprice', data=df).fit()
m2 = smf.ols(formula='linvpc~lprice+t', data=df).fit()
print_compare({'nt':m1, 'lt':m2})
         nt      lt
r2    0.208   0.341
ar2   0.189   0.307
dfr  40.000  39.000
dfm   1.000   2.000
           coeff          pval        stderr           tval
              nt     lt     nt     lt     nt     lt      nt     lt
Intercept -0.550 -0.913  0.000  0.000  0.043  0.136 -12.788 -6.733
lprice     1.241 -0.381  0.002  0.578  0.382  0.679   3.245 -0.561
t            NaN  0.010    NaN  0.008    NaN  0.004     NaN  2.798
```

We must be careful here. To account for the trending behavior of the variables, we add a time trend to get

$$\widehat{log(invpc)} = \underset{(0.136)}{-0.913} - \underset{(0.679)}{0.381}\ log(price) + \underset{(0.004)}{0.01}\ t.$$

The story is much different now. The estimated price elasticity is negative and not statistically different from zero. The time trend is statistically significant, and its coefficient implies an approximate 1% increase in *invpc* per year, on average. There are other factors, captured in time trend, that affect *invpc*, but we have not modelled these. The result in the first regression show a spurious relationship between *invpc* and *price* due to the fact that price is also trending upward over time. □

If the dependent and independent variable have different kind of trends, but the movement in the independent variable about its trend line causes movement in the dependent variable away from its trend line, adding a time trend can make the key explanatory variable more significant.

Example 10.5. For the general fertility rate model we considered we now add a time trend and then an additional quadratic trend to see the effect on the variable of interest pe_t.

```
df = woo.data('fertil3').set_index('year')
m1 = smf.ols(formula='gfr~pe+ww2+pill', data=df).fit()
m2 = smf.ols(formula='gfr~pe+ww2+pill+t', data=df).fit()
m3 = smf.ols(formula='gfr~pe+ww2+pill+t+tsq', data=df).fit()
print_compare({'nt': m1, 'lt': m2, 'qt':m3})
         nt      lt      qt
r2    0.473   0.662   0.727
ar2   0.450   0.642   0.706
dfr  68.000  67.000  66.000
dfm   3.000   4.000   5.000
            coeff                      pval                stderr                \
               nt       lt       qt     nt     lt     qt     nt     lt     qt
Intercept  98.682  111.769  124.092  0.000  0.000  0.000  3.208  3.358  4.361
pe          0.083    0.279    0.348  0.007  0.000  0.000  0.030  0.040  0.040
pill      -31.594    0.997  -10.120  0.000  0.874  0.115  4.081  6.262  6.336
t             NaN   -1.150   -2.531    NaN  0.000  0.000    NaN  0.188  0.389
tsq           NaN      NaN    0.020    NaN    NaN  0.000    NaN    NaN  0.005
ww2       -24.238  -35.592  -35.880  0.002  0.000  0.000  7.458  6.297  5.708
             tval
               nt      lt      qt
Intercept  30.760  33.287  28.457
pe          2.784   6.968   8.639
pill       -7.742   0.159  -1.597
t             NaN  -6.119  -6.501
tsq           NaN     NaN   3.945
ww2        -3.250  -5.652  -6.286
```

The coefficient on *pe* increase both in value and significance. Interestingly, *pill* is not significant once an allowance is made from a linear trend. As can be seen by the estimate, gfr was

falling, on average, over this period, other factors being equal. Since the general fertility rate exhibited both upward and downward trends during the period from 1913 through 1984, we also look at the quadratic trend. The coefficient of pe is even larger and more statistically significant. Nothing prevents us from adding t^3, but we should keep the model relatively simple unless warranted otherwise. □

Including a time trend in a regression model is equivalent to **detrending** the original data series before using them in regression analysis. Concretely, to do the regression $\hat{y}_t = \hat{\beta}_0 + \hat{\beta}_1 x_{t1} + \hat{\beta}_2 x_{t2} + \hat{\beta}_3 t$, we can first detrend the variables y_t, x_{t1}, and x_{t2} by regressing them against t with an intercept to get the residuals $\tilde{y}_t, \tilde{x}_{t1}$ and $\tilde{x}_{t2}$. We then regress $\tilde{y}_t$ on $\tilde{x}_{t1}$ and $\tilde{x}_{t2}$ without intercept to get back $\hat{\beta}_1$ and $\hat{\beta}_2$.

If the trend term is statistically significant, and the results change in important ways when a time trend is added to a regression, then the initial results without a trend should be treated with suspicion. This analysis suggests that it is a good idea to include a trend in the regression if any independent variable is trending, even if y_t is not. If y_t has no noticeable trend, but say x_{t1} is growing over time, then excluding a trend from the regression may make it look as if x_{t1} has no effect on y_t, even though movements of x_{t1} about its trend may affect y_t. This will be captured if t is included in the regression.

Example 10.6. We take up the Puerto Rican employment data and add a trend variable to it.

```
df = woo.data('prminwge')
m1 = smf.ols(formula='lprepop~lmincov+lusgnp', data=df).fit()
m2 = smf.ols(formula='lprepop~lmincov+lusgnp+t', data=df).fit()
print_compare({'nt': m1, 'lt':m2})
         nt      lt
r2    0.660   0.847
ar2   0.641   0.834
dfr  35.000  34.000
dfm   2.000   3.000
            coeff           pval         stderr           tval
               nt     lt      nt     lt      nt     lt      nt     lt
Intercept -1.054 -8.696   0.177  0.000   0.765  1.296 -1.378 -6.711
lmincov   -0.154 -0.169   0.023  0.001   0.065  0.044 -2.380 -3.813
lusgnp    -0.012  1.057   0.891  0.000   0.089  0.177 -0.138  5.986
t            NaN -0.032     NaN  0.000     NaN  0.005     NaN -6.442
```

The coefficient on $log(usgnp)$ has changed dramatically from -0.012 and insignificant to 1.06 and very significant. The coefficient of minimum wage has also increased in significance. The variable $prepop_t$ has no clear trend, but $log(usgnp)$ has an upward linear trend. We can think of the estimate 1.06 as follows: when $usgnp$ increases by 1% above its long-run trend, $prepop$ increases by about 1.06%. □

Typical R-squared are higher in time series than in cross-sectional regressions. This is understandable when we are looking at aggregated data, which is typically less noisy. But it can be artificially high when the dependent variable is trending. Adjusted R-squared

$\bar{R}^2 = 1 - \hat{\sigma}_u^2/\hat{\sigma}_y^2$, shows that if y_t is trending and we include the time trend in the regression, $\hat{\sigma}_u^2$ will be calculated correctly. On the other hand $\hat{\sigma}_y^2 = SST/(n-1)$, where $SST = \sum(y_t - \bar{y})^2$ is no longer biased or consistent; in fact, it could substantially overestimate the variance in y_t, because it does not account for the trend in y_t. Hence, the simplest method is to compute the usual R-squared in a regression where the dependent variable has already been detrended. For example, in the two variable multi-regression model $\hat{y}_t = \hat{beta}_0 + \hat{\beta}_1 x_{t1} + \hat{\beta}_{t2} + \hat{\beta}_3 t$; we can first get $\tilde{y}_t$ as the residual of y_t regressed on t and then regressing $\tilde{y}_t$ on x_{t1}, x_{t2}, and t. The R-squared form this regression is usable. For the adjusted R-squared we use $(n-2)$ as the degree of freedom in the denominator, as there are two parameters in estimating $\tilde{y}_t$. Example 10.4 shows a misleading R-squared of 34.1%. If we first detrend $log(invpc)$ and regress the detrended variable on $log(price)$ and t, the R-squared becomes 0.8%. This is consistent with the small t statistic on $log(price)$.

```
df = woo.data('hseinv').set_index('year')
m1 = smf.ols(formula='linvpc~t', data=df).fit()
df['tildey'] = m1.resid
m2 = smf.ols(formula='tildey~lprice+t', data=df).fit()
m2.summary2()
=================================================================
Model:              OLS              Adj. R-squared:     -0.043
Dependent Variable: tildey           AIC:                -40.9186
Date:               2020-10-31 19:26 BIC:                -35.7056
No. Observations:   42               Log-Likelihood:     23.459
Df Model:           2                F-statistic:        0.1575
Df Residuals:       39               Prob (F-statistic): 0.855
R-squared:          0.008            Scale:              0.020633
-----------------------------------------------------------------
                Coef.   Std.Err.     t     P>|t|    [0.025  0.975]
-----------------------------------------------------------------
Intercept      -0.0718    0.1356  -0.5292  0.5997  -0.3461  0.2025
lprice         -0.3810    0.6788  -0.5612  0.5779  -1.7540  0.9921
t               0.0017    0.0035   0.4791  0.6345  -0.0054  0.0088
-----------------------------------------------------------------
```

In computing R-squared form of an F statistic for testing multiple hypotheses, we just use the usual R-squareds without any detrending. The usual formula is appropriate for F statistic.

Many time series exhibit **seasonality**, e.g. quarterly sales are typically higher in the 4th quarter due to Christmas holiday. Quarterly US GDP is an example where the series is reported after being **seasonally adjusted**. Generally, we can include a set of **seasonal dummy variables** to account for seasonality in the dependent variable, the independent variables, or both. If the data are quarterly, then we would include dummy variables for three of the four quarters, with the omitted category being the base quarter. If there is no seasonality, then the coefficients to the dummy variables would be all zero. This is easily tested via an F test. Sometimes, it is useful to consider interaction effects to with these seasonal dummies.

Including seasonal dummies in a regression can be interpreted as **deseasonalizing** the data. One can regress each of the variables on the seasonal dummies and then do the regression on the residuals, as before. In cases with pronounced seasonality, a better goodness-of-fit measure is an R-squared based on the deseasonalized y_t. Time series exhibiting seasonal patterns can be trending as well, in which case we should estimate a regression model with a time trend and seasonal dummy variables. Goodness-of-fit statistics can be determined based on detrended and deseasonalized y_t.

11 Further Issues in using OLS with Time Series Data

11.1 Stationary and weakly dependent time series

The stochastic process $\{x_t : t = 1, 2, \dots\}$ is **stationary** if for every collection of time indices $1 \leq t_1 < t_2 < \dots < t_m$, the joint distribution of $(x_{t_1}, x_{t_2}, \dots, x_{t_m})$ is the same as the joint distribution of $(x_{t_1+h}, x_{t_2+h}, \dots, x_{t_m+h})$ for all integers $h \geq 1$. For $m = 1$ and $t_1 = 1$ it means that the sequence $\{x_t : 1, 2, \dots\}$ is identically distributed. Further, stationarity also requires the nature of any correlation between adjacent terms is the same across all time periods. For example, the joint distribution (x_1, x_2) must be the same as the joint distribution of (x_t, x_{t+1}) for any $t \geq 1$ (though there is no restriction on what that correlation could be). Stationarity is a property of the underlying stochastic process and not of the available single realization; though it is tested on the data collected for that single realization. It is easy to spot certain sequences that are not stationary, e.g. a process with a time trend.

Sometimes, a weaker form of stationarity suffices. A stochastic process $\{x_t : t = 1, 2, \dots\}$ with a finite second moment $E(x_t^2) < \infty$ is **covariance stationary** if $E(x_t)$ is constant; $Var(x_t)$ is constant; and for any t, $h \geq 1$, $Cov(x_t, x_{t+h})$ depends on h and not on t. Here we concern ourselves only with the first two moments and the covariance between x_t and x_{t+h} depends only on the distance between the two terms, h, and not on the location of the initial time t. If a stationary process has a finite second moment, then it must be covariance stationary, but the converse is certainly not true.

Stationarity assumption allows us to assume stability of relationships in variables over time, and, hence, allowing us to model the relationships if we only have access to a single time series realization. In multiple regression model we are assuming that β_j does not change over time. Further, TS.4 and TS.5 imply the variance is of error process is constant over time and that the correlation between errors in two adjacent periods is equal to zero.

A very different concept is that of weak dependence, which places restrictions on how strongly related the random variables x_t and x_{t+h} can be as the time distance between them, h, gets large. this notion of weak dependence is most easily discussed for a stationary time series. A stationary time series process $\{x_t, t = 1, 2, \dots\}$ is said to be **weakly dependent** if x_t and x_{t+h} are almost independent as h increases without bound. For Covariance stationary time series is weakly dependent if the correlation between X_t and x_{t+h} goes to zero sufficiently

quickly as $h \to \infty$. They are also said to be **asymptotically uncorrelated**.

For regression, the assumption of weak dependence essentially replaces the assumption of random sampling in implying the law of large numbers and the central limit theorem hold. Time series that are not weakly dependent do not generally satisfy the CLT, and their use in multiple regression can be tricky. A sequence that is independent and identically distributed (called white noise) is trivially weakly dependent. There are two main category of weakly dependent time series.

1. **MA process**: $x_t = e_t + \alpha_1 e_{t-1}$, for $t = 1, 2, \ldots$, where $\{e_t : t = 0, 1, \ldots\}$ is an iid sequence with zero mean and variance σ_e^2. The process $\{x_t\}$ is called a moving average process of order one MA(1). We can easily see that $Corr(x_t, x_{t+1}) = \alpha_1/(1+\alpha_1^2)$. Due to the identical distribution assumption on the e_t, $\{x_t\}$ is actually stationary.

2. **AR process**: $y_t = \rho_1 y_{t-1} + e_t$, $t = 1, 2, \ldots$, where $\{e_t : t = 1, 2, \ldots\}$ is an iid sequence with zero mean and variance σ_e^2; and e_t are independent of y_0 and $E(y_0) = 0$. This is called an autoregressive process of order one AR(1). For weak dependence we require $|\rho_1| < 1$ and call it a stable AR(1) process. If we assume covariance stationarity, we can show that $E(y_t) = 0$ and $Corr(y_t, y_{t+h}) = \rho_1^h$. This shows that a stable AR(1) process is weakly dependent.

An important point to remember is that a trending series, though certainly non-stationary, can be weakly dependent. A series that is stationary about its time trend, as well as weakly dependent, is often called a **trend stationary process**.

11.2 Asymptotic properties of OLS

We state the assumptions and main results that justify OLS for large samples.

- TS.1' Linearity and weak dependence: The stochastic process $\{(\boldsymbol{x}_t, y_t) : t = 1, 2, \ldots\}$ follows the linear model $y_t = \beta_0 + \beta_1 x_{t1} + \ldots + \beta_k x_{tk} + u_t$, where $\{u_t : t = 1, 2, \ldots, n\}$ is the sequence of errors and n is the number of observations. We assume the stochastic process, additionally, to be stationary and weakly dependent.

- TS.2' No perfect collinearity: In the sample and underlying time series process, no independent variable is constant nor a perfect linear combination of the others.

- TS.3' Zero conditional mean: The explanatory variables $\boldsymbol{x}_t = (x_{t1}, x_{t2}, \ldots, x_{tk})$ are **contemporaneously exogenous**, i.e. $E(u_t|\boldsymbol{x}_t) = 0$.

TS.1' implies law of large numbers and the central limit theorem can be applied to sample averages. Unlike TS.1 x_{tj} can include lags of dependent variable (and the usual lags of explanatory variables are allowed as well). The assumption of stationarity in TS.1' is not at all critical for OLS to have its standard asymptotic properties, since we already assume β_j to be constant. The important extra restriction is the weak dependence assumption. Further, TS.3' is much weaker than TS.3 because it puts no restrictions on how u_t is related to the explanatory variables in other time periods. By stationarity, if contemporaneous exogeneity

holds for one time period, it holds for them all. Relaxing stationarity would simply require us to assume the condition holds for all $t = 1, 2, \ldots$.

Theorem 11.1. *(Consistency of OLS) Under TS.1', TS.2' and TS.3', the OLS estimators are consistent: plim$\hat{\beta}_j = \beta_j$, $j = 0, 1, \ldots, k$. In fact, the only required conditions are $E(u_t) = 0$, $Cov(x_{tj}, u_t) = 0$, $j = 1, \ldots, k$.*

Notice, the OLS estimators are not necessarily unbiased under the above theorem. Also, we have weakened the sense in which the explanatory variables must be exogenous and imposed the weak dependence requirement.

Consider a static model with two explanatory variables $y_t = \beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} + u_t$. Under weak dependence, the condition sufficient for consistency of OLS is $E(u_t|z_{t1}, z_{t2}) = 0$. This rules out omitted variables and missepecified functional form issues. Measurement error in dependent variables can cause OLS to fail. Importantly, assumption TS.3' does not rule out correlation between, say, u_{t-1} and z_{t1}, which could be caused if z_{t1} depends on past lag of y_t. Similarly, in the finite distributed lag model, $y_t = \alpha_0 + \delta_0 z_t + \delta_1 z_{t-1} + \delta_2 z_{t-2} + u_t$, the natural assumption of $E(u_t|z_t, z_{t-1}, \ldots) = 0$ is more than enough to satisfy assumption TS.3' making the OLS consistent. Again, it does not rule out feedback from y to future values of z. These two examples do not require asymptotic theory because the explanatory variables could be strictly exogenous.

For the AR(1) model $y_t = \beta_0 + \beta_1 y_{t-1} + u_t$, where $E(u_t|y_{t-1}, y_{t-2}, \ldots) = 0$, we have $E(y_t|y_{t-1}, y_{t-2}, \ldots) = E(y_t|y_{t-1}) = \beta_0 + \beta_1 y_{t-1}$. Assumption TS.3' holds, but the strict exogeneity assumption needed for unbiasedness, TS.3 does not hold. This is because $Cov(y_t, u_t) = Var(u_t) > 0$. Therefore, a model with lagged dependent variable cannot satisfy the strict exogeneity assumption TS.3. For the weak dependence condition to hold, we must assume $|\beta_1| < 1$, to produce consistent estimators of β_0 and β_1. Unfortunately, $\hat{\beta}_1$ is biased, especially for small sample size and when β_1 is near 1 (where it can have severe downward bias).

- TS.4' Homoskedasticity: The errors are contemporaneously homoskedastic, that is $Var(u_t|\boldsymbol{x}_t) = \sigma^2$.
- TS.5' No serial correlation: For all $t \neq s$, $E(u_t u_s|\boldsymbol{x}_t, \boldsymbol{x}_s) = 0$.

In both these conditions we assume only contemporaneous relationships of the time periods involved. Serial correlation is often a problem in static and finite distributed lag regression models. Importantly, TS.5' *does* hold for AR(1) model. We can show that $E(u_t u_s|\boldsymbol{y}_{t-1}, \boldsymbol{y}_{s-1}) = 0$ for $t \neq s$. Hence, as long as only one lag is involved here, the error must be serially uncorrelated.

Theorem 11.2. *(Asymptotic normality of OLS) Under TS.1' through TS.5', the OLS estimators are asymptotically normally distributed and the usual OLS standard errors, t statistics, F statistics, and LM statistics are asymptotically valid.*

Example 11.1. *(Efficient Markets Hypothesis)* Let y_t be the weekly percentage return on the New York Stock Exchange composite index. A strict form of the efficient markets

hypothesis states that information observable to the market prior to week t should not help to predict the return during week t. If we use only past information on y, the EMH is stated as $E(y_t|y_{t-1}, y_{t-2}, \ldots) = E(t_t)$. We can test this by using an AR(1) model on the returns with the null hypothesis $H_0 : \beta_1 = 0$. Under null hypothesis we have $y_t = \beta_0 + u_t$ and hence TS.1' to TS.5' hold and OLS estimates are asymptotically normal and we can use the usual OLS t statistic for $\hat{\beta}_1$ to test $H_0 : \beta_1 = 0$ against $H_1 : \beta_1 \neq 0$.

```
df = woo.data('nyse').set_index('t')
dff = df[['return', 'return_1']].dropna().astype('f8')
m1 = smf.ols(formula='Q("return")~return_1', data=dff).fit()
m1.summary2()
==================================================================
Model:              OLS              Adj. R-squared:     0.002
Dependent Variable: Q("return")      AIC:                2986.4884
Date:               2020-11-08 17:54 BIC:                2995.5589
No. Observations:   689              Log-Likelihood:     -1491.2
Df Model:           1                F-statistic:        2.399
Df Residuals:       687              Prob (F-statistic): 0.122
R-squared:          0.003            Scale:              4.4538
------------------------------------------------------------------
              Coef.    Std.Err.     t      P>|t|     [0.025   0.975]
------------------------------------------------------------------
Intercept     0.1796     0.0807   2.2248   0.0264    0.0211   0.3382
return_1      0.0589     0.0380   1.5490   0.1218   -0.0158   0.1336
------------------------------------------------------------------
```

The average weekly return over this period was 0.196 in percentage form, with the largest weekly return being 8.45% and the smallest being -15.32%. The AR(1) model is estimated as

$$\widehat{return}_t = \underset{(0.0807)}{0.1796} + \underset{(0.0380)}{0.0589}\ return_{t-1}$$

with residual degree of freedom 687 and model degree of freedom 1, and $R^2 = 0.0035$ and $\overline{R}^2 = 0.002$. The t statistic for the coefficient on $returns_{t-1}$ is about 1.55 and so $H_0 : \beta_1 = 0$ cannot be rejected against the two-sided alternative, even at the 10% significance level.

We now fit an AR(2) model to see if an extra week of lag and test EMH. We fit $y_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + u_t$. There are stability conditions on β_1 and β_2 that are needed to ensure that the AR(2) process is weakly dependent, but this is not an issue here because the null hypothesis states that the EMH holds: $H_0 : \beta_1 = \beta_2 = 0$. We can use the F statistic to test the joint hypothesis.

```
dff['return_2'] = dff.return_1.shift(1)
dff = dff.dropna()
m2 = smf.ols(formula='Q("return")~return_1+return_2', data=dff).fit()
m2.summary2()
==================================================================
Model:              OLS              Adj. R-squared:     0.002
Dependent Variable: Q("return")      AIC:                2984.0120
```

```
Date:               2020-11-08 18:12 BIC:                2997.6134
No. Observations:   688              Log-Likelihood:     -1489.0
Df Model:           2                F-statistic:        1.659
Df Residuals:       685              Prob (F-statistic): 0.191
R-squared:          0.005            Scale:              4.4593
-------------------------------------------------------------------
             Coef.    Std.Err.      t      P>|t|     [0.025   0.975]
-------------------------------------------------------------------
Intercept    0.1857     0.0812    2.2889   0.0224    0.0264   0.3451
return_1     0.0603     0.0382    1.5799   0.1146   -0.0146   0.1353
return_2    -0.0381     0.0381   -0.9982   0.3185   -0.1130   0.0368
-------------------------------------------------------------------
m2.f_test(['return_1=0', 'return_2=0'])
<F test: F=array([[1.65857123]]), p=0.19117456962590035, df_denom=685, df_num=2>
```

The fitted AR(2) model is

$$\widehat{return}_t = \underset{(0.0812)}{0.1857} + \underset{(0.0382)}{0.0603}\, return_{t-1} - \underset{(0.0381)}{0.0381}\, return_{t-2}$$

with residual degree of freedom 685 and model degree of freedom 2, and $R^2 = 0.005$ and $\overline{R}^2 = 0.002$. The two lags are individually insignificant at 10% level. They are also jointly insignificant since the F statistic of 1.66 gives a p-value of 0.191. Thus we do not reject the null hypothesis even at 15% significance level. □

11.3 Highly persistent time series

The previous section shows that, provided the time series we use are weakly dependent, usual OLS inference procedures are valid under assumptions weaker than the classical linear model assumptions. We now look at highly persistent or strongly dependent time series and show how they can be transformed for use in regression analysis.

The process $y_t = y_{t-1} + u_t$, $t = 1, 2, \ldots$, is called a **random walk**. It can be easily seen that $E(y_t) = E(y_0)$ for all $t \geq 1$. However, the variance does depend on t, $Var(y_t) = \sigma_e^2 t$. Since the variance is not constant, it is not a stationary process. Random walk is very persistent in the sense that $E(y_{t+h}|y_t) = y_t$, for all $h \geq 1$. Comparing this to AR(1) process we have $E(y_{t+h}|y_t) = \rho_1^h y_t$, for all $h \geq 1$ and this approaches zero as $h \to \infty$. We can also see the correlation between y_t and y_{t+h} is close to 1 for large t when $\{y_t\}$ follows a random walk. If $Var(y_0) = 0$ then $Corr(y_t, y_{t+h}) = \sqrt{t/(t+h)}$, and hence it is not covariance stationary. Further, although for fixed t the correlation tends to zero as $h \to \infty$, it does not do so very quickly. Therefore, a random walk does not satisfy the requirement of an asymptotically uncorrelated sequence.

A random walk is a special case of what is known as a **unit root process**. The name comes from the fact that $\rho_1 = 1$ in the AR(1) model. A more general class of unit root processes is generated when $\{e_t\}$ is allowed to be a general, weakly dependent series. For example, $\{e_t\}$ could be itself a MA(1) or AR(1) process. When $\{e_t\}$ is not an i.i.d. sequence,

the properties of the random walk we derived earlier no longer hold. But the key feature of $\{y_t\}$ is preserved: the value of y today is highly correlated with y even in the distant future.

It is extremely important not to confuse trending and highly persistent behaviors. A series can be trending but not highly persistent. On the other hand they can be persistent, but have no obvious trend. However, it is often the case that a highly persistent series also contains a clear trend. One model that leads to this behavior is the **random walk with drift**: $y_t = \alpha_0 + y_{t-1} + e_t$, for $t = 1, 2, \ldots$, where $\{e_t\}$ and y_0 satisfy the same properties as in the random walk model. α_0 is called the drift term. If $y_0 = 0$ then $E(y_t) = \alpha_0 t$, the expected value of y_t growing over time if $\alpha_0 > 0$. Further $E(y_{t+h}|y_t) =_0 h + y_t$. This is another example of unit root process.

Using time series with strong persistence as displayed by unit root process in a regression can lead to spurious regression problem if the CLM assumptions are violated. Simple transformations are available that render a unit root process weakly dependent. Weakly dependent processes are said to be **integrated of order zero** or **I(0)**, i.e. averages of such sequences already satisfy the standard limit theorems. Unit root processes, such as random walk (with or without drift), are said to be **integrated of order one**, or **I(1)**. This means that the **first difference** of the process is weakly dependent (and often stationary). A time series that is I(1) is staid to be **difference-stationary process**, although the name is somewhat misleading with its emphasis on stationarity after differencing rather than weak dependence in the differences. Many time series y_t that are strictly positive are such that $log(y_t)$ is integrated of order one. We can use the first differences in logs, $log(y_t) = log(y_t) - log(y_{t-1}) \approx (y_t - y_{t-1})/y_{t-1}$, or the proportionate changes directly.

Differencing time series before using them in regression analysis has another benefit: it removes any linear time trend. Therefore, rather than including time trend in a regression, we can instead difference those variables that show obvious trends. An informal method to check for I(1) process is to look at **first order autocorrelation**, the sample correlation between y_t and y_{t-1}, denoted by $\hat{\rho}_1$. By apply the law of large numbers, $\hat{\rho}_1$ can be shown to be consistent for ρ_1 provided $|\rho_1| < 1$. However, $\hat{\rho}_1$ is not an unbiased estimator of ρ_1. Calculating sampling distributions of the estimator $\hat{\rho}_1$ is hard particularly when they are close to one or much less than one. As a rough guide, differencing is warranted if $\hat{\rho} > 0.8$ to 0.9.

Example 11.2. To explain general fertility rate grf, in terms of the value of the personal exemption pe we get via OLS:

$$\widehat{gfr} = \underset{(4.3047)}{96.3443} - \underset{(0.0359)}{0.0071}\ pe$$

with degree of freedom for residuals of 70 and degree of freedom for the model of 1, and $R^2 = 0.001$.

```
df = woo.data('fertil3')
df.corrwith(df.shift(1))[['gfr', 'pe']]
gfr    0.976452     pe     0.963580
```

```
m1 = smf.ols(formula='gfr~pe', data=df).fit()
df['Dgfr'] = df.gfr.diff()
df['Dpe'] = df.pe.diff()
m2 = smf.ols(formula='Dgfr~Dpe', data=df).fit()
```

We look at the autocorrelation of the series which turn out to be very high and suggestive of unit root behavior. This raises questions about our use of OLS t statistics. We now estimate the equation using first differences and contrast against the original one.

$$\widehat{\Delta gfr} = \underset{(0.5020)}{-0.7848} - \underset{(0.0284)}{0.0427}\ \Delta pe$$

with degree of freedom for residuals of 69 and degree of freedom for the model of 1, and $R^2 = 0.032$. Though the new estimates are not statistically different from zero at 5% significance, they give a very different results than when we estimated the model in levels, and casts doubt on our earlier analysis. If we add two lags of Δpe, we see

$$\widehat{\Delta gfr_t} = \underset{(0.4678)}{-0.9637} - \underset{(0.0268)}{0.0362}\ \Delta pe_t - \underset{(0.0276)}{0.0140}\ \Delta pe_{t-1} + \underset{(0.0269)}{0.1100}\ \Delta pe_{t-2}$$

with degree of freedom for residuals of 65 and degree of freedom for the model of 3, and $R^2 = 0.232$.

```
df['Dpe_1'] = df.Dpe.shift(1)
df['Dpe_2'] = df.Dpe.shift(2)
m3 = smf.ols(formula='Dgfr~Dpe+Dpe_1+Dpe_2', data=df).fit()
m3.f_test(['Dpe=0', 'Dpe_1=0'])
<F test: F=array([[1.28941437]]), p=0.2823824025023764, df_denom=65, df_num=2>
m3.f_test(['Dpe=0', 'Dpe_1=0', 'Dpe_2=0'])
<F test: F=array([[6.56265636]]), p=0.000605410952860825, df_denom=65, df_num=3>
```

The joint F test for Δpe_t and Δpe_{t-1} shows they are jointly insignificant. But the joint F test of Δpe_t, Δpe_{t-1} and Δpe_{t-2} shows they are jointly significant and also the coefficient of Δpd_{t-2} is highly significant. This indicates a positive relationship between changes in pe and subsequent changes in gfr two years hence. □

When the series in has an obvious upward or downward trend, it makes more sense to obtain the first order autocorrelation after detrending. If the data are not detrended, the autoregressive correlation tends to be overestimated, which biases toward finding a unit root in a trending process.

Example 11.3. One way to estimate the elasticity of hourly wage with respect to output per hour is to estimate the following

```
df = woo.data('earns')
m1 = smf.ols(formula='lhrwage~loutphr+t', data=df).fit()
m1.summary2()
```

```
-------------------------------------------------------------------
                Coef.   Std.Err.     t      P>|t|    [0.025   0.975]
-------------------------------------------------------------------
Intercept      -5.3285    0.3744  -14.2301  0.0000  -6.0865  -4.5704
loutphr         1.6396    0.0933   17.5650  0.0000   1.4507   1.8286
t              -0.0182    0.0017  -10.4278  0.0000  -0.0218  -0.0147
-------------------------------------------------------------------
```

with degree of freedom for residuals of 38 and degree of freedom for the model of 2, and $R^2 = 0.971$. The time trend is included because both $log(hrwage_t)$ and $log(outphr_t)$ display clear, upwards, linear trends. The estimated elasticity seems to be to too large: a 1% increase in productivity increases the real wages by about 1.64%. We need to be cautious here.

```
df['DTlhrwage']=smf.ols(formula='lhrwage~t',data=df).fit().resid
df['DTloutphr']=smf.ols(formula='loutphr~t',data=df).fit().resid
df.corrwith(df.shift(1))[['DTlhrwage', 'DTloutphr']]
>>> DTlhrwage    0.967159    DTloutphr    0.945293
```

The high auto-correlations show presence of unit root in detrended series. We, thus, take the first difference and estimate the equation again to get

```
df['DDTlhrwage'] = df['DTlhrwage'].diff()
df['DDTloutphr'] = df['DTloutphr'].diff()
m2 = smf.ols(formula='DDTlhrwage~DDTloutphr', data=df).fit()
m2.summary2()
-------------------------------------------------------------------
               Coef.   Std.Err.     t      P>|t|    [0.025   0.975]
-------------------------------------------------------------------
Intercept     -0.0006    0.0027  -0.2302  0.8192  -0.0060   0.0048
DDTloutphr     0.8093    0.1735   4.6659  0.0000   0.4582   1.1605
-------------------------------------------------------------------
```

with degree of freedom for residuals of 38 and degree of freedom for the model of 1, and $R^2 = 0.364$. Now, a 1% increase in productivity is estimated to increase real wages by about 0.81%, and the estimate is not statistically different from one. The R^2 shows that growth in output explains about 36% of the growth in real wages. □

11.4 Dynamically complete models

In the general model $y_t = \beta_0 + \beta_1 x_{t1} + \ldots + \beta_k x_{tk} + u_t$, where the explanatory variables $\boldsymbol{x}_t = (x_{t1}, \ldots, x_{tk})$ may or may not contain lags of y or z we assume $E(u_t|\boldsymbol{x}_t, y_{t-1}, \boldsymbol{x}_{t-1}, \ldots) = 0$. Written in terms of y_t, $E(y_t|\boldsymbol{x}_t, y_{t-1}, \boldsymbol{x}_{t-1}, \ldots) = E(y_t|\boldsymbol{x}_t)$. When $\boldsymbol{x}_t$ has enough lags included so that further lags of y and the explanatory variables do not matter for explaining y_t we have a **dynamically complete model**. This can be a very strong assumption for static and finite distributed lag models. Since the assumption is equivalent to

$E(u_t|\boldsymbol{x}_t, u_{t-1}, \boldsymbol{x}_{t-1}, u_{t-2}, \ldots) = 0$, we can show that a dynamically complete model must satisfy TS.5', i.e. $E(u_t u_s|\boldsymbol{x}_t, \boldsymbol{x}_s) = 0$. This means that for dynamically complete models the errors $\{u_t\}$ must be serially uncorrelated.

For forecasting purposes it is best to have dynamically complete models. Serial correlation in the errors is a sign of misspecification but could be allowed if we are really interested in a static model or finite distributed lag models specifically. In example 11.2 we estimated the distributed lag model for Δgfr on Δpe, allowing for two lags of Δpe. For this model to be dynamically complete no further lags of Δgfr or Δpe should appear in the equation.

```
df = woo.data('fertil3')
df['Dgfr'] = df.gfr.diff()
df['Dpe'] = df.pe.diff()
df['Dpe_1'] = df.Dpe.shift(1)
df['Dpe_2'] = df.Dpe.shift(2)
df['Dgfr_1'] = df.Dgfr.shift(1)
m1 = smf.ols(formula='Dgfr~Dpe+Dpe_1+Dpe_2+Dgfr_1', data=df).fit()
m1.summary2()
-----------------------------------------------------------------
               Coef.   Std.Err.     t     P>|t|    [0.025  0.975]
-----------------------------------------------------------------
Intercept     -0.7022    0.4538  -1.5473  0.1267  -1.6087  0.2044
Dpe           -0.0455    0.0256  -1.7734  0.0809  -0.0967  0.0058
Dpe_1          0.0021    0.0268   0.0771  0.9388  -0.0514  0.0556
Dpe_2          0.1051    0.0256   4.1084  0.0001   0.0540  0.1563
Dgfr_1         0.3002    0.1059   2.8351  0.0061   0.0887  0.5118
-----------------------------------------------------------------
```

We can easily see that this is false by adding Δgfr_{t-1}: the coefficient estimate is 0.3002 with a t statistics of 2.8351. Thus the model is not dynamically complete with just two lags of Δpe. This suggest that there may be serial correlation in the errors of the original model.

```
m2 = smf.ols(formula='Dgfr~Dpe+Dpe_1+Dpe_2', data=df).fit()
m2.summary2()
m2.resid.corr(m2.resid.shift(1))
>>> 0.2912470311914683
m1.resid.corr(m1.resid.shift(1))
>>> 0.02647671283248059
```

We see that, indeed, the errors in original model has serial correlation which reduces considerably when we add the term Δgfr_{t-1}.

In a regression model the explanatory variables $\boldsymbol{x}_t$ are said to be **sequentially exogenous** if $E(u_t|\boldsymbol{x}_t, \boldsymbol{x}_{t-1},) = E(u_t) = 0$, for $t = 1, 2, \ldots$. Sequential exogeneity is implied by strict exogeneity and sequential exogeneity implies contemporaneous exogeneity. Further dynamic completeness implies sequential exogeneity because $(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}, \ldots)$ is a subset of $(\boldsymbol{x}_t, y_{t-1}, \boldsymbol{x}_{t-1}, \ldots)$. If $\boldsymbol{x}_t$ contains y_{t-1}, the dynamic completeness and sequential exogeneity

are the same conditions. A sequentially exogenous model like distributed lag model may not be dynamically complete and we may not care. In addition, the explanatory variables in an FDL model may or may not be strictly exogenous.

12 Serial Correlation and Heteroskedasticity in Time Series Regression

12.1 Properties of OLS with serially correlated errors

Under first three Gauss-Markov assumptions for time series regressions OLS estimators are unbiased. There is no assumption about serial correlation in the errors. Hence as long as explanatory variables are strictly exogenous, the $\hat{\beta}_j$ are unbiased, regardless of the degree of serial correlation in the errors. This is analogous to the observation that heteroskedasticity in the errors does not cause bias in the $\hat{\beta}_j$. When we relax the strict exogeneity assumption to contemporaneous exogeneity, and when the data is weakly dependent, the $\hat{\beta}_j$ are still consistent, although not necessarily unbiased. This result did not hinge on any assumption about serial correlation in the errors.

Because the Gauss-Markov Theorem requires both homoskedasticity and serially uncorrelated errors, OLS is no longer BLUE in the presence of serial correlation. Even more importantly, the usual OLS standard errors and test statistics are not valid, even asymptotically. For an AR(1) serial correlation model for the error term $u_t = \rho u_{t-1} + e_t$, $t = 1, 2, \ldots, n$ and $|\rho| < 1$, where e_t are uncorrelated random variables with mean zero and variance σ_e^2 we consider the variance of the OLS slope estimator in the regression model $y_t = \beta_0 + \beta_1 x_t + u_t$, with $\bar{x} = 0$. Then, the OLS estimator $\hat{\beta}_1$ of β_1 can be written as

$$\hat{\beta}_1 = \beta_1 + \sum_{t=1}^{n} x_t u_t / \sum_{t=1}^{n} x_t^2.$$

To compute the variance of $\hat{\beta}_1$ conditional on $\boldsymbol{X}$ we must account for the serial correlation in the u_t,

$$Var(\hat{\beta}_1) = \frac{1}{\left(\sum x_t^2\right)^2} Var\left(\sum x_t u_t\right) = \frac{Var(u_t)}{\sum x_t^2} + 2\frac{Var(u_t)^2}{\left(\sum x_t^2\right)^2} \sum_{t=1}^{n-1} \sum_{j=1}^{n-t} \rho^j x_t x_{t+j}.$$

When $\rho = 0$ we get the usual OLS result. Under serial correlation be get biased estimates of the standard error. In most practical applications, under presence of serial correlation OLS estimates underestimates the true variance of the estimator. Beyond t statics no longer being valid, F and LM statistics are also invalid.

Provided the data are stationary and weakly dependent, serial correlation does not invalidate R-squared and adjusted R-squared. R-squared is defined as $1 - \sigma_u^2/\sigma_y^2$ and serial correlation does not change the variance. By law of large numbers, R^2 and $\overline{R}^2$ both consistently estimate the population R-squared. The argument is essentially the same as in the case of

heteroskedasticity. This argument does not go through if $\{y_t\}$ is an I(1) process because $Var(y_t)$ is no longer constant; goodness of fit does not make much sense in this case. Trends and seasonality in the mean of y_t should be accounted for in computing R^2 and $\overline{R}^2$. Other deviation from stationarity do not cause difficulty in interpreting the goodness of fit in the usual ways.

OLS can be inconsistent in the presence of lagged dependent variables when the errors have serial correlation based on lagged dependence model, e.g. AR(1). Often, serial correlation in the errors of a dynamic model simply indicates that the dynamic regression function has not been completely specified.

12.2 Testing for serial correlation

A t test for AR(1) serial correlation with strictly exogenous regressors: The simplest model of serial correlation in errors is AR(1) model. The null hypothesis is that there is no serial correlation, $H_0 : \rho = 0$. For $u_t = \rho u_{t-1} + e_t$, we assume $E(e_t|u_{t-1}, u_{t-2}, \ldots) = 0$ and $Var(e_t|u_{t-1}) = Var(e_t) = \sigma_e^2$. Since u_t is unknown, we use $\hat{u}_t$ from the OLS residuals. Fortunately, it turns out that, because of strict exogeneity assumption, the large-sample distribution of the t statistic is not affected by using the OLS residuals in place of the errors. We run the regression of $\hat{u}_t$ on $\hat{u}_{t-1}$ for all $t = 1, \ldots, n$, obtaining the coefficient $\hat{\rho}$ on $\hat{u}_{t-1}$ and it t statistic $t_{\hat{\rho}}$. We use $t_{\hat{\rho}}$ to test $H_0 : \rho = 0$ against $H_1 : \rho \neq 0$ in usual way. Hence, serial correlation is a problem only if H_0 is rejected. We should remember the difference between practical and statistical significance. With a large sample size, it is possible to find statistically significant serial correlation even though $\hat{\rho}$ is practically small and close to zero. In such a case, OLS inference procedures will not be far off.

Example 12.1. We look a static Phillips curve model of inflation versus unemployment tradeoff in the United States. We compare it against adaptive expectations augmented Phillips curve. We look at the AR(1) test of serial correlation in the errors of these models.

```
df = woo.data('phillips')
df = df[df.year<=1996]
m1 = smf.ols(formula='Q("inf")~unem', data=df).fit()
------------------------------------------------------------------------
             Coef.    Std.Err.      t      P>|t|     [0.025   0.975]
------------------------------------------------------------------------
Intercept    1.4236     1.7190   0.8282   0.4118   -2.0346   4.8818
unem         0.4676     0.2891   1.6174   0.1125   -0.1140   1.0493
------------------------------------------------------------------------
m2 = smf.ols(formula='cinf~unem', data=df).fit()
------------------------------------------------------------------------
               Coef.   Std.Err.      t      P>|t|    [0.025   0.975]
------------------------------------------------------------------------
Intercept      3.0306    1.3768   2.2012   0.0328   0.2592   5.8020
unem          -0.5426    0.2302  -2.3575   0.0227  -1.0059  -0.0793
------------------------------------------------------------------------
```

```
rf1=pd.DataFrame({'hu': m1.resid, 'hu_1':m1.resid.shift(1)})
r1 = smf.ols(formula='hu~hu_1', data=rf1).fit()
------------------------------------------------------------------------
               Coef.    Std.Err.      t      P>|t|     [0.025   0.975]
------------------------------------------------------------------------
Intercept      -0.1134     0.3594   -0.3155   0.7538   -0.8368   0.6100
hu_1            0.5730     0.1161    4.9337   0.0000    0.3392   0.8067
------------------------------------------------------------------------

rf2 = pd.DataFrame({'hu': m2.resid, 'hu_1':m2.resid.shift(1)})
r2 = smf.ols(formula='hu~hu_1', data=rf2).fit()
------------------------------------------------------------------------
               Coef.    Std.Err.      t      P>|t|     [0.025   0.975]
------------------------------------------------------------------------
Intercept       0.1942     0.3004    0.6464   0.5213   -0.4108   0.7992
hu_1           -0.0356     0.1239   -0.2873   0.7752   -0.2851   0.2139
------------------------------------------------------------------------
```

We find that the model $\widehat{inf}_t = 1.4236 + 0.4676\ unem_t$ has a $\hat{\rho} = 0.573$ with $t_{\hat{\rho}} = 4.93$, giving strong evidence of positive, first order serial correlation; and making the estimated regression model doubtful. The second model $\widehat{\Delta inf}_t = 3.0306 - 0.5426\ unem_t$ has a $\hat{\rho} = -0.0346$ with $t_{\hat{\rho}} = 0.1239$ with p-value of 0.7752: giving no evidence of AR(1) serial correlation in the expectations augmented Phillips curve. □

We must assume that when estimating $u_t = u_{t-1} + e_t$, the assumption of homoskedasticity holds. If not, we simply use the usual, heteroskedasticity-robust t statistic.

The Durbin-Watson test under classical assumptions: Another test for AR(1) serial correlation is the Drubin-Watson test. The **Durbin-Watson statistic** is also based on the the OLS residuals:

$$DW = \frac{\sum_{t=2}^{n}(\hat{u}_t - \hat{u}_{t-1})^2}{\sum_{t=1}^{n}\hat{u}_t^2} \approx 2(1-\hat{\rho}).$$

Therefore, for large sample size, tests based on DW and t test based on $\hat{\rho}$ are conceptually the same. The distribution of DW, conditional on $\boldsymbol{X}$ depends on the full set of CLM assumptions. Usually, the DW test is computed for the alternative $H_1 : \rho > 0$. $\hat{\rho} \approx 0$ implies that $DW \approx 2$, and $\hat{\rho} > 0$ implies that $DW < 2$. Thus, to reject the null hypothesis we are looking for a value of DW that is significantly less than 2. Unfortunately, because of the problems in obtaining the null distribution of DW, we must compare DW with two sets of critical values. These are usually labeled d_U for upper and d_L for lower. If $DE < d_L$, then we reject H_0; if $DW > d_U$, we fail to reject H_0. If $d_L \leq d_U$, the test is inconclusive.

For the static Phillips curve regression, we see that the DW value is 0.803. For $k = 1$ and $n = 48$ at 1% significance level we have $d_L = 1.32$ and hence we reject the null of no serial correlation against and alternative of positive serial correlation. Notice that finding the critical value depends on the matrix X and hence a thumb rule of $d_L = 1.5$ and $d_U = 2.5$ is generally used.

```
------------------------------------------------------------------------
             Coef.    Std.Err.      t      P>|t|     [0.025    0.975]
------------------------------------------------------------------------
Intercept    1.4236     1.7190   0.8282   0.4118    -2.0346    4.8818
unem         0.4676     0.2891   1.6174   0.1125    -0.1140    1.0493
------------------------------------------------------------------------
Omnibus:              8.905        Durbin-Watson:           0.803
Prob(Omnibus):        0.012        Jarque-Bera (JB):        8.336
Skew:                 0.979        Prob(JB):                0.015
Kurtosis:             3.502        Condition No.:           23
========================================================================
```

For the expectations augmented Phillips curve, $DW = 1.77$, which is well within the fail-to-reject region at even the 5% level of $d_U = 1.59$. This test is not valid in presence of heteroskedasticity as the CLM assumptions are violated.

```
------------------------------------------------------------------------
               Coef.    Std.Err.      t      P>|t|     [0.025    0.975]
------------------------------------------------------------------------
Intercept      3.0306     1.3768   2.2012   0.0328     0.2592    5.8020
unem          -0.5426     0.2302  -2.3575   0.0227    -1.0059   -0.0793
------------------------------------------------------------------------
Omnibus:             22.805        Durbin-Watson:           1.770
Prob(Omnibus):        0.000        Jarque-Bera (JB):       52.071
Skew:                -1.239        Prob(JB):                0.000
Kurtosis:             7.460        Condition No.:           24
========================================================================
```

Testing for AR(1) serial correlation without strictly exogenous regressors: In this case neither the t-test or the DW statistic are valid, even in large samples. This generally happens when y_{t-1} is a dependent variable. Durbin suggested an alternative statistic for this case. We first run OLS regression of y_t on $x_{t1}, , x_{tk}$ and obtain the residual $\hat{u}_t$. Thereafter, we run the regression of $\hat{u}_t$ on $x_{t1}, \ldots, x_{tk}, \hat{u}_{t-1}$ including an intercept and obtain the coefficient $\hat{\rho}$ on $\hat{u}_{t-1}$ and its t statistic $t_{\hat{\rho}}$. we use this statistic to test $H_0 : \rho = 0$ against $H_1 : \rho \neq 0$ in the usual way. The inclusion of $x_{t1}, \ldots, x_{tk}$ explicitly allows for each x_{tj} to be correlated with u_{t-1}, and this ensures that $t_{\hat{\rho}}$ has an approximate t distribution in large samples. This t statistic is easily made robust to heteroskedasticity of unknown form by using heteroskedasticity-robust t statistic on $\hat{u}_{t-1}$.

Example 12.2. We estimate the effect of minium wage on the Puerto Rican employment rate. We then check the errors for first order serial correlation using Durbin's alternative statistic.

```
df = woo.data('prminwge')
m1 = smf.ols(formula='lprepop~lmincov+lprgnp+lusgnp+t', data=df).fit()
```

```
------------------------------------------------------------------------------
                 Coef.    Std.Err.      t      P>|t|     [0.025    0.975]
------------------------------------------------------------------------------
Intercept      -6.6634     1.2578   -5.2976   0.0000   -9.2225   -4.1044
lmincov        -0.2123     0.0402   -5.2864   0.0000   -0.2940   -0.1306
lprgnp          0.2852     0.0805    3.5437   0.0012    0.1215    0.4490
lusgnp          0.4860     0.2220    2.1896   0.0357    0.0344    0.9377
t              -0.0267     0.0046   -5.7629   0.0000   -0.0361   -0.0173
------------------------------------------------------------------------------
Omnibus:                2.155        Durbin-Watson:           1.014
Prob(Omnibus):          0.340        Jarque-Bera (JB):        1.744
Skew:                   0.521        Prob(JB):                0.418
Kurtosis:               2.872        Condition No.:           5892
==============================================================================
df['hatu'] = m1.resid
df['hatu_1'] = m1.resid.shift(1)
m2 = smf.ols(formula='hatu~lmincov+lprgnp+lusgnp+t+hatu_1', data=df).fit()
------------------------------------------------------------------------------
                 Coef.    Std.Err.      t      P>|t|     [0.025    0.975]
------------------------------------------------------------------------------
Intercept      -0.8508     1.0927   -0.7786   0.4421   -3.0793    1.3778
lmincov         0.0375     0.0352    1.0650   0.2951   -0.0343    0.1093
lprgnp         -0.0785     0.0705   -1.1126   0.2744   -0.2223    0.0654
lusgnp          0.2039     0.1952    1.0450   0.3041   -0.1941    0.6020
t              -0.0035     0.0041   -0.8509   0.4013   -0.0118    0.0048
hatu_1          0.4805     0.1664    2.8869   0.0070    0.1410    0.8200
------------------------------------------------------------------------------
sm.stats.stattools.durbin_watson(m1.resid)
>>> 1.0137087802831086
```

In the first regression we see Durbin-Watson statistic of 1.014 suggesting presence of first order serial correlation in the errors. We conduct the regression on the residual in presence of all independent variables and $\hat{u}_{t-1}$ and get $\hat{\rho} = 0.4805$ with a t stat of 2.887, suggesting that the null hypothesis is rejected establishing the presence of serial correlation in the errors. This means that the estimates of $\hat{\beta}_j$ in the first regression are not valid for inference. $\hat{\beta}_j$ are still consistent if u_t is contemporaneously uncorrelated with each explanatory variable. □

Testing for higher order serial correlation: More generally, we can easily extend the test to higher orders of serial correlation. We can test for serial correlation in the autoregressive model of order q: $u_t = \rho_1 u_{t-1} + \rho + 2u_{t-2} + \ldots + \rho_q u_{t-q} + e_t$. The null hypothesis is $H_0 : \rho_1 = \rho_2 = \ldots = \rho_q = 0$. We first run the regression of y_t on $x_{t1}, \ldots, x_{tk}$ and obtain the OLS residuals, $\hat{u}_t$, for all $t = 1, 2, \ldots, n$. Thereafter, we run the regression of $\hat{u}_t$ on $x_{t1}, \ldots, x_{tk}, \hat{u}_{t-1}, \ldots, \hat{u}_{t-q}$, for all $t = (q+1), \ldots, n$. We compute the F test for joint significance of $\hat{u}_{t-1}, \ldots, \hat{u}_{t-q}$. Including the x_{tj} in the regression makes the test valid with or without the strict exogeneity assumption. The test requires the homoskedasticity assumption $Var(u_t|\boldsymbol{x}_t, u_{t-1}, \ldots, u_{t-q}) = \sigma^2$. A heteroskedasticity-robust version can be computed as well.

An alternative to computing the F test is to use the Lagrange multiplier (LM) form of

the statistic. The LM statistic to test the null hypothesis is $LM = (n-q)R^2_{\hat{u}}$, where $R^2_{\hat{u}}$ is the usual R-squared from the regression. Under the null hypothesis, $LM \overset{a}{\sim} \chi^2_q$. This is usually called the **Breusch-Godfrey test** for AR(q) serial correlation. The LM statistic requires the homoskedasticity assumption but can be made robust to heteroskedasticity.

Example 12.3. In the event study of the barium chloride industry, we used monthly data. We wish to test for higher order of serial correlation. We use AR(3) here.

```
df = woo.data('barium')
m1 = smf.ols(formula='lchnimp~lchempi+lgas+lrtwex+befile6+affile6+afdec6', data=df)
res = m1.fit()
import statsmodels.api as sm
sm.stats.diagnostic.acorr_breusch_godfrey(res, nlags=3)
(14.76817281786161,
 0.0020258705854483334,
 5.124668960539397,
 0.0022637006749126826)
```

We see that the F statistic is 5.12 with a p-value of 0.002 giving strong evidence of AR(3) serial correlation. The LM statistic is 14.77 with p-value of 0.002 confirming the same.

To test for seasonal form of serial correlation, here for monthly data, we can test for AR(1) seasonal serial correlation with $u_t = u_{t-12} + e_t$ when the regressors are strictly exogenous, otherwise we include the regressors along with $\hat{u}_{t-12}$. A regression of $\hat{u}_t$ on $\hat{u}_{t-12}$ gives

```
df['uhat'] = m1.resid
df['uhat_12'] = m1.resid.shift(12)
m2 = smf.ols(formula='uhat~uhat_12', data=df).fit()
m2.summary()
```

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.0379	0.050	0.751	0.454	-0.062	0.138
uhat_12	-0.1874	0.084	-2.225	0.028	-0.354	-0.021

$\hat{\rho} = -0.187$ with p-value of 0.028, so there is evidence of negative seasonal autocorrelation. □

12.3 Correcting for serial correlation with exogenous regressors

If we detect serial correlation and our goal is to estimate a model with complete dynamics, we need to re-specify the model. In other cases, we need to find a way to carry out statistical inference robust to serial correlation. We begin with the important case of AR(1) serial correlation. We need strictly exogenous regressors and hence can't use these corrections when the explanatory variables include lagges dependent variables.

Obtaining the BLUE in the AR(1) error model: We assume the Gauss-Markov assumptions TS.1 through TS.4, but relax assumption TS.5 and model the errors as $u_t = \rho u_t + e_t$, for all $t = 1, 2, \ldots$. We have $Var(u_t|\boldsymbol{X}) = \sigma_e^2/(1-\rho^2)$. For a single explanatory case $y_t = \beta_0 + \beta_1 x_1 + u_t$ we can manipulate the equation to $\tilde{y}_t = (1-\rho)\beta_0 + \beta_1 \tilde{x}_1 + e_t$, for $t \geq 2$, where $\tilde{y}_t = y_t - \rho y_{t-1}$ and $\tilde{x}_t = x_t - \rho x_{t-1}$ and is called the **quasi-differenced data**. If $\rho = 1$ we get the usual differenced data, but here we assume $|\rho| < 1$. This equation satisfies all Gauss-Markov assumptions but not quite BLUE because they do not use the first period. We can use the first equation with equalized variance $\tilde{y}_1 = \sqrt{1-\rho^2}\beta_0 + \beta_1 \tilde{x}_1 + \tilde{u}_t$, where $\tilde{u}_1 = \sqrt{1-\rho^2}u_1$, $\tilde{y}_1 = \sqrt{1-\rho^2}y_1$, $\tilde{x}_1 = \sqrt{1-\rho^2}x_1$, with $Var(\tilde{u}_1) = \sigma_e^2$. We can now use these to get BLUE estimators of β_0, β_1 under assumptions TS.1 through TS.4 with AR(1) model for u_t. This is another example of a *generalized least squares (GLS)* estimator. Adding more regressors changes very little. The t and F statistics from the transformed equations are valid asymptotically, and exactly if the errors e_t are normally distributed.

Feasible GLS estimation with AR(1) errors: Since ρ is rarely known in practice, we can get a consistent estimate of ρ by regressing the OLS residuals on their lagged counterparts. Next we use this estimate $\hat{\rho}$ in place of ρ to obtain the quasi-differenced variables and finally use OLS on the transformed equation. This results in the **feasible GLS (FGLS)** estimator of the β_j. The error terms now consist of e_t and also the terms involving the estimation error in $\hat{\rho}$. Fortunately, the estimation error in $\hat{\rho}$ does not affect the asymptotic distribution of the FGLS estimators. The usual standard errors, t statistics, and F statistics are asymptotically valid. The cost of using ρ in place of $\hat{\rho}$ is that the feasible GLS estimator has no tractable finite sample properties. In particular, it is not unbiased, although it is consistent when the data is weakly dependent. Since FGLS is not unbiased, it is not BLUE, but is asymptotically more efficient than the OLS estimator with the AR(1) model for serial correlation holds. We assume that the time series are weakly dependent.

There are several names for the FGLS estimation of the AR(1) model. **Cochrane-Orcutt (CO) estimation** omits the first observation and uses $\hat{\rho}$ from the regression of $\hat{u}_t$ on $\hat{u}_{t-1}$ whereas **Prais-Winsten (PW) estimation** uses the first observation as suggested above. Asymptotically, it makes no difference. In practice, these methods can be used in an iterative scheme, but the iterations don't help much because theoretically, the large-sample properties of the iterated estimator are the same as the estimator that uses only the first iteration.

Example 12.4. Using the Barium data set we comare the estimations using simple OLS, Cochrane-Orcutt and Prais-Winsten estimation.

```
df = woo.data('barium')
m1 = smf.ols(formula='lchnimp~lchempi+lgas+lrtwex+befile6+affile6+afdec6',
             data=df).fit()

def ols_ar1(model,rho,drop1=True):
    x = model.model.exog
    y = model.model.endog
    ystar = y[1:]-rho*y[:-1]
    xstar = x[1:, ]-rho*x[:-1, ]
```

```
    if not drop1:
        ystar = np.append(np.sqrt(1-rho**2)*y[0], ystar)
        xstar = np.append([np.sqrt(1-rho**2)*x[0, ]], xstar, axis=0)
    return sm.OLS(ystar, xstar).fit()

df['hu'] = m1.resid
df['hu_1'] = m1.resid.shift(1)
r1 = smf.ols(formula='hu~hu_1', data=df).fit()
print(r1.params.hu_1)
>>> 0.2707530212152067
r2 = ols_ar1(m1, r1.params.hu_1, drop1=True)
r3 = ols_ar1(m1, r1.params.hu_1, drop1=False)
pd.DataFrame({'ols': m1.params, 'CO': r2.params, 'PW': r3.params},
             index=m1.params.index)
                ols         CO         PW
Intercept -17.803001 -35.598952 -35.395059
lchempi     3.117193   2.964914   2.959406
lgas        0.196350   0.978515   0.971471
lrtwex      0.983018   1.123438   1.119983
befile6     0.059574  -0.008092  -0.008227
affile6    -0.032406  -0.032864  -0.032929
afdec6     -0.565245  -0.575676  -0.575388
pd.DataFrame({'ols': m1.tvalues, 'CO': r2.tvalues, 'PW': r3.tvalues},
             index=m1.params.index)
               ols        CO        PW
Intercept -0.845934 -1.543588 -1.564096
lchempi    6.504965  4.728984  4.809537
lgas       0.216575  0.993475  0.999970
lrtwex     2.456602  2.241054  2.263493
befile6    0.228279 -0.025705 -0.026240
affile6   -0.122613 -0.103560 -0.104185
afdec6    -1.977521 -1.703557 -1.709827
# automatic CO estimation
y, X = pt.dmatrices('lchnimp~lchempi+lgas+lrtwex+befile6+affile6+afdec6',
                    data=df, return_type='dataframe')
m2 = sm.GLSAR(y, X).fit()
```

The coefficients that are statistically significant in OLS remain so in the Cochrane-Orcutt and Prais-Winsten estimation. The low t statistics indicate that the correct standard errors are uniformly higher than OLS estimation, which understates the standard error. Therefore, the effect on Chinese imports after the International Trade Commission's decision is now less statistically significant than we thought. The R^2 should not be compared against each other as they are transformed models and it is not clear what is R^2 measuring in the corrected models. □

When OLS and FGLS estimates differ in practically important ways, we need to be a bit careful. The required condition for consistency of OLS estimates is $Cov(x_t, u_t) = 0$. While the required condition for the consistency of FGLS estimates is $Cov((x_{t-1} + x_{t+1}), u_t) = 0$. Hence OLS and FLGS might give significantly different estimates if the required condition fails for FGLS. In this case OLS - which is still consistent - is preferred to FGLS, which is

inconsistent. If x has a lagged effect on y, or x_{t+1} reacts to changes in u_t, FGLS can produce misleading results.

Example 12.5. We compare the OLS and Prais-Winsten estimates of the static Phillips curve.

```
df = woo.data('phillips')
df = df[df.year <= 1996]
m1 = smf.ols(formula='Q("inf")~unem', data=df).fit()
m1.summary2()
df['hu'] = m1.resid
df['hu_1'] = m1.resid.shift(1)
r1 = smf.ols(formula='hu~hu_1', data=df).fit()
print(r1.params.hu_1)
>>> 0.5729694848063343
r2 = ols_ar1(m1, r1.params.hu_1, drop1=False)
pd.DataFrame({'ols': m1.params, 'PW': r2.params}, index=m1.params.index)
                ols        PW
Intercept  1.423610  6.239994
unem       0.467626 -0.362041
pd.DataFrame({'ols': m1.tvalues,  'PW': r2.tvalues}, index=m1.params.index)
                ols        PW
Intercept  0.828155  3.194348
unem       1.617376 -1.145993
```

The coefficient of interest is on *unem*, and it differs markedly between PW and OLS. PW estimates are consistent with the inflation-unemployment tradeoff and are fairly close to what is obtained by first differencing both *inf* and *unem*. FGLS, here has the advantage of eliminating unit roots. □

Correction for higher orders of serial correlation is similar but involved. Fortunately, econometrics packages geared towards time series analysis easily estimate models with general $AR(q)$ errors. Differencing is recommended when the error term in $y_t = \beta_0 + \beta_1 x_1 + u_t$ has unit root, due to y_t and x_t being I(1) processes. Even if u_t does not follow a random walk, but ρ is positive and large, first differencing is often a good idea to eliminate most of the serial correlation.

12.4 Serial correlation-robust inference

If the explanatory variables are no strictly exogenous FGLS is no longer consistent. Also it may be better to compute standard errors for the OLS estimates that are robust to more general forms of serial correlation and not just AR(1). For the model $y_t = \beta_0 + \beta_1 x_{t1} + \ldots + \beta_k x_{tk} + u_t$, for $t = 1, 2, \ldots, n$, which we have estimated by OLS, we are interested in obtaining a serial correlation-robust standard error for $\hat{\beta}_1$. We write $x_{t1} = \delta_0 + \delta_2 x_{t2} + \ldots + \delta_k x_{tk} + r_t$, where r_t has zero mean and is uncorrelated with $x_{t2}, x_{t3}, \ldots, x_{tk}$. The asymptotic variance of the OLS estimator $\hat{\beta}_1$ is $Avar(\hat{\beta}_1) = \left(\sum_{t=1}^{n} E(r_t^2)\right)^{-1} Var\left(\sum_{t=1}^{n} r_t u_t\right)$, with $a_t = r_t u_t$. To estimate $Avar(\hat{\beta}_1)$ we begin with $se(\hat{\beta}_1)$ the usual OLS standard error

and $\hat{\sigma}$ the usual standard error. Let $\hat{r}_t$ denote the residuals form the auxiliary regression of x_{t1} on $x_{t2}, x_{t3}, \ldots, x_{tk}$ including a constant. For a chosen $g > 0$ we define

$$\hat{\nu} = \sum_{t=1}^{n} \hat{a}_t^2 + 2 \sum_{h=1}^{g} \left(1 - \frac{h}{g+1}\right) \left(\sum_{t=h+1}^{n} \hat{a}_t \hat{a}_{t-h}\right),$$

where $\hat{a}_t = \hat{r}_t \hat{u}_t$. The integer g controls how much serial correlation we are allowing in computing the standard error. The serial correlation-robust standard error of $\hat{\beta}_1$ is simply $se(\hat{\beta}_1)_{HAC} = \left(\frac{se(\hat{\beta}_1)}{\hat{\sigma}}\right)^2 \sqrt{\hat{\nu}}$. This can be used to construct confidence intervals and t statistics for $\hat{\beta}_1$. This standard error is also robust to arbitrary heteroscedasticity, in fact the first term is the usual heteroskedasticity-robust standard error. This error is called **heteroscedasticity and autocorrelation consistent (HAC) standard error**. The choice of g should grow with n. Newey and West recommend $g = 4(n/100)^{2/9}$, others have suggested $n^{1/4}$.

Empirically, the serial correlation-robust standard errors are typically larger than the usual OLS standard errors when there is serial correlation. Computing an SC-robust standard error after quasi-differencing would ensure that any extra serial correlation is accounted for in statistical inference. The SC-robust standard errors are specially useful when we have doubts about some of the explanatory variables being strictly exogenous, when Prais-Winsten and Cochrane-Orcutt methods are not not even consistent. It is also valid to use the SC-robust standard errors in models with lagged dependent variables, assuming there is good reason for allowing serial correlation in such model.

Example 12.6. We obtain an SC-robust standard error for the minimum wage effect in the Puerto Rican employment equation.

```
df = woo.data('prminwge')
m1 = smf.ols(formula='lprepop~lmincov+lprgnp+lusgnp+t',
             data=df).fit()
m2 = smf.ols(formula='lprepop~lmincov+lprgnp+lusgnp+t',
             data=df).fit(cov_type='HAC', cov_kwds={'maxlags': 2})
print_compare({'OLS': m1, 'HAC': m2})
           coeff          pval        stderr          tval
             OLS    HAC    OLS    HAC    OLS    HAC    OLS    HAC
Intercept -6.663 -6.663  0.000  0.000  1.258  1.432 -5.298 -4.654
lmincov   -0.212 -0.212  0.000  0.000  0.040  0.043 -5.286 -4.982
lprgnp     0.285  0.285  0.001  0.002  0.080  0.093  3.544  3.072
lusgnp     0.486  0.486  0.036  0.062  0.222  0.260  2.190  1.869
t         -0.027 -0.027  0.000  0.000  0.005  0.005 -5.763 -4.971
np.sqrt(m1.mse_resid)
>>> 0.03276561893350027
```

The OLS estimate of the elasticity of the employment rate with respect to the minimum wage is $\hat{\beta}_1 = -0.2123$, and the usual OLS standard error is $se(\hat{\beta}_1) = 0.0402$. The standard error of the regression is $\hat{\sigma} = 0.0328$. This gives the SC/heteroskedasticity-robust standard error of 0.0427 and a t stats of -4.98. □

12.5 Heteroskedasticity in time series regression

Just as in cross-sectional case the presence of heteroskedasticity, while not causing bias or inconsistency in the $\hat{\beta}_j$, does invalidate the usual standard errors, t statistics, and F statistics. It is important to remember that serially correlated errors cause problems that adjustment for heteroskedasticity are not able to address. Hence we want u_t to have no serial correlation, by testing for serial correlation first, before using a heteroskedasticity test. Secondly, for the Breusch-Pagan test for heteroskedasticity $u_t^2 = \delta_0 + \delta_1 x_{t1} + \ldots + \delta_k x_{tk} + v_t$, with null hypothesis $H_0 : \delta_0 = \delta_2 = \ldots = \delta_k = 0$ the F statistics is valid only if $\{v_t\}$ is themselves homoskedastic and serially uncorrelated. If heteroskedasticity is found, one can use heteroskedasticity-roubust test statistic or weighted least squares.

Example 12.7. We tested EMH by regressing returns on lagged returns. The EMH states that $\beta_1 = 0$. When we tested this hypothesis using the data in NYSE we botained $t_{\beta_1} = 1.55$ with $n = 689$. With such large sample, this is not much evidence against the EMH.

```
df = woo.data('nyse').set_index('t')
dff = df[['return', 'return_1']].dropna().astype('f8')
m1 = smf.ols(formula='Q("return")~return_1', data=dff).fit()
m1.summary2()
------------------------------------------------------------------
              Coef.    Std.Err.     t      P>|t|     [0.025   0.975]
------------------------------------------------------------------
Intercept     0.1796     0.0807   2.2248   0.0264    0.0211   0.3382
return_1      0.0589     0.0380   1.5490   0.1218   -0.0158   0.1336
------------------------------------------------------------------
df['hu2'] = m1.resid**2
m2 = smf.ols(formula='hu2~return_1', data=df).fit()
m2.summary2()
------------------------------------------------------------------
               Coef.   Std.Err.     t     P>|t|    [0.025   0.975]
------------------------------------------------------------------
Intercept       4.6565    0.4277  10.8878  0.0000   3.8168   5.4962
return_1       -1.1041    0.2014  -5.4822  0.0000  -1.4996  -0.7087
------------------------------------------------------------------
m3 = smf.ols(formula='Q("return")~return_1', data=dff).fit(cov_type='HC0')
m3.summary2()
print_compare({'OLS': m1, 'HC0': m3})
           coeff           pval          stderr           tval
             OLS    HC0     OLS    HC0     OLS    HC0     OLS    HC0
Intercept  0.180  0.180  0.026  0.035  0.081  0.085  2.225  2.109
return_1   0.059  0.059  0.122  0.394  0.038  0.069  1.549  0.852
```

The Breusch-Pagan test for heteroskedasticity entails regressing the squared OLS residuals $\hat{u}_t^2$ on $returns_{t-1}$. The t statistics on $returns_{t-1}$ is about -5.5, indicating strong evidence of heteroskedasticity. Because the coefficient on $returns_{t-1}$ is negative, we have the interesting finding that volatility in stock returns is lower than when the previous return was high, and vice versa. We find that the expected value of stock returns does not depend on past returns, but the variance of returns does. □

Dynamic model of heteroskedasticity have gain grounds recently. Even if variance of u_t given $\boldsymbol{X}$ is constant, there are other ways that heteroskedasticity can arise. Engle suggested what is known as the **autoregressive conditional heteroskedasticity (ARCH)** model, with $E(u_t^2|u_{t-1}) = \alpha_0 + \alpha_1 u_{t-1}^2$, or $u_t^2 = \alpha_0 + \alpha_1 u_{t-1}^2 + \nu_t$, where the expected value of ν_t given $u_{t-1}, u_{t-2} \ldots$ is zero. This is an autoregressive model in u_t^2. OLS is still consistent under ARCH model of heteroskedasticity. The usual heteroskedasticity-robust standard errors and test statistics are valid, as they don't depend on the form for heteroskedasticity. In the previous example, we can better characterize heteroskedasticity by the ARCH model. The t statistic on $\hat{u}_{t-1}^2$ is over 9, indicating strong ARCH.

```
df['hu2_1'] = m1.resid.shift(1)**2
m4 = smf.ols(formula='hu2~hu2_1', data=df).fit()
m4.summary()
```

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	2.9474	0.440	6.695	0.000	2.083	3.812
hu2_1	0.3371	0.036	9.377	0.000	0.266	0.408

```
df['hu'] = m1.resid
df['hu_1'] = m1.resid.shift(1)
m5 = smf.ols(formula='hu~hu_1', data=df).fit()
m5.summary()
```

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-0.0012	0.081	-0.015	0.988	-0.159	0.157
hu_1	0.0014	0.038	0.037	0.971	-0.074	0.076

It is important to see that, thought the squared OLS residuals are autocorrelated, the OLS residuals themselves are not, as is consistent with the EMH.

Both heteroskedasticity and serial correlation can be present in the regression model. If we are unsure, we can always use OLS and compute fully robust standard errors. Alternatively, if we detect serial correlation we can employ the Cochrane-Orcutt or Prais-Winsten transformation and, in the transformed equation, use heteroskedasticity-robust standard errors. Or, we can even test for heteroskedasticity using the Breusch-Pagan or White tests. Moreover, we can model and correct for both through a combined weighted least squares AR(1) procedure.

Specifically, consider the model

$$
\begin{aligned}
y_t &= \beta_0 + \beta_1 x_{t1} + \ldots + \beta_k x_{tk} + u_t \\
u_t &= \sqrt{h_t}\nu_t \\
\nu_t &= \rho\nu_{t-1} + e_t, \ |\rho| < 1
\end{aligned}
$$

where the explanatory variables $\boldsymbol{X}$ are independent of e_t for all t, and h_t is a function of the x_{tj}. The process $\{e_t\}$ has zero mean and constant variance ^2_e and is serially uncorrelated. Therefore, $\{\nu_t\}$ satisfies a stable $AR(1)$ process. The error u_t is heteroskedastic, in addition to containing serial correlation $Var(u_t|\boldsymbol{x}_t) = \sigma^2_\nu h_t$, where $\sigma^2_\nu = \sigma^2_e/(1-\rho^2)$. But $\nu_t = u_t/\sqrt{h_t}$ is homoskedastic and follows a stable AR(1) model. Therefore, the transformed equation

$$\frac{y_t}{\sqrt{h_t}} = \beta_0 \frac{1}{\sqrt{h_t}} + \beta_1 \frac{x_{t1}}{\sqrt{h_t}} + \ldots + \beta_k \frac{x_{tk}}{\sqrt{h_t}} + \nu_t$$

has AR(1) errors. Now we can estimate this equation using CO or PW methods. For a feasible GLS with heteroskedasticity and AR(1) serial correlation we can estimate the OLS residual $\hat{u}_t$. We then regression $log(\hat{u}^2_t)$ on $x_{t1}, \ldots x_{tk}$ (or on $\hat{y}_t, \hat{y}^2_t$) and obtain the fitted values $\hat{g}_t$. Obtain the estimates $\hat{h}_t = exp(g_t)$. Finally, we can estimate the transformed equation by standard Cochrane-Orcutt or Prais-Winsten methods.

The feasible GLS estimator is asymptotically efficient. All standard errors and test statistics from the CO or PW estimation are asymptotically valid. For the possibility of serial correlation not beign AR(1) or variance function misspecified, we can apply quasi-differencing to the transformed equation, estimating the resulting equation by OLS, and then obtain the Newey-West standard errors. This would be asymptotically efficient while ensuring that inference is asymptotically valid, even for misspecified model of either heteroskedasticity or serial correlation.

13 Pooling Cross Sections across Time: Simple Panel Data Methods

Data sets that have both cross-sectional and time-series dimensions are being used more often in empirical research. There are two kinds of data sets in consideration here

- Independently pooled cross-section: sampling randomly from a large population at different points in time. They consist of *independently* sampled observations ruling out correlation in the error terms across different observations. The sampling of the population at different point of time may not be *identically* distributed. This time varying nature can be captured by allowing intercepts in a multiple regression model, and in some cases the slopes, to change over time.
- Panel data or longitudinal data: While having both a cross-sectional and a time-series dimension, the data correspond to the same entity over time. We cannot assume that the observations are independently distributed over time. Moreover, the distribution may not be identically distributed.

13.1 Pooling Independent Cross Sections across Time

If a random sample is drawn at each time period, pooling the resulting random samples give us an independently pooled cross section, increasing the sample size giving more precise estimators. Pooling in helpful in this regard only insofar as the relationship between variable and independent variables remain constant over time. Typically, to reflect the fact that the population may have different distributions in different time periods, we allow the intercept to differ across time, accomplished by including dummy variables for all but one period, with the earliest sample chosen as the base case. It is also possible that the error variance changes over time.

Example 13.1 (Women's fertility over time)**.** We are interested in the question: After controlling for education, has the pattern of fertility among women over age 35 changed between 1972 and 1984? The model estimated explains the total number of kids born to woman from 1972 to 1984. To understand the fertility rates over time, we control for the factors like years of education, age, race, region, environment to get

```
import statsmodels.api as sm

df = woo.dataWoo('fertil1')
df['age2'] = df.age**2
y = df.kids
x = df[['educ', 'age', 'age2', 'black', 'east', 'northcen',
        'west', 'farm', 'othrural', 'town', 'smcity'] +
       ['y74', 'y76', 'y78', 'y80', 'y82', 'y84']]
X = sm.add_constant(x)
model = sm.OLS(y, X)
results = model.fit()
print(results.summary())
```

```
==============================================================================
Dep. Variable:                   kids   R-squared:                       0.130
Model:                            OLS   Adj. R-squared:                  0.116
Method:                 Least Squares   F-statistic:                     9.981
Date:                Sat, 12 Sep 2020   Prob (F-statistic):           4.27e-25
Time:                        09:05:37   Log-Likelihood:                -2091.2
No. Observations:                1129   AIC:                             4218.
Df Residuals:                    1111   BIC:                             4309.
Df Model:                          17
Covariance Type:                  HC3
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -7.7425      3.104     -2.494      0.013     -13.827      -1.658
educ          -0.1284      0.021     -5.994      0.000      -0.170      -0.086
age            0.5321      0.140      3.789      0.000       0.257       0.807
age2          -0.0058      0.002     -3.636      0.000      -0.009      -0.003
black          1.0757      0.204      5.265      0.000       0.675       1.476
east           0.2173      0.129      1.691      0.091      -0.035       0.469
northcen       0.3631      0.118      3.087      0.002       0.133       0.594
west           0.1976      0.165      1.200      0.230      -0.125       0.520
farm          -0.0526      0.147     -0.357      0.721      -0.341       0.236
othrural      -0.1629      0.183     -0.890      0.373      -0.521       0.196
town           0.0844      0.129      0.652      0.515      -0.169       0.338
smcity         0.2119      0.156      1.362      0.173      -0.093       0.517
y74            0.2682      0.189      1.419      0.156      -0.102       0.639
y76           -0.0974      0.202     -0.483      0.629      -0.493       0.298
y78           -0.0687      0.199     -0.344      0.731      -0.460       0.322
y80           -0.0713      0.195     -0.365      0.715      -0.454       0.312
y82           -0.5225      0.190     -2.757      0.006      -0.894      -0.151
y84           -0.5452      0.188     -2.907      0.004      -0.913      -0.178
==============================================================================
Omnibus:                        9.775   Durbin-Watson:                   2.011
Prob(Omnibus):                  0.008   Jarque-Bera (JB):                9.966
Skew:                           0.227   Prob(JB):                      0.00685
Kurtosis:                       2.920   Cond. No.                     1.32e+05
==============================================================================
results.f_test(['y74=0','y76=0','y78=0','y80=0','y82=0','y84=0'])
<F test: F=array([[5.86950867]]), p=4.85518986757229e-06, df_denom=1.11e+03, df_num=6>
```

The base year is 1972. The coefficients of the year dummy variables show a sharp drop in fertility in the early 1980s. Holding control factors fixed, a woman had on average 0.52 less children, or about one-half a child, in 1982 than in 1972. This drop is separate from the decline in fertility that is due to the increase in average education levels. Given the coefficients on y82 and y84 are significant the joint significance F statistic is 5.87 and p-value ≈ 0.

Women with more education have fewer children, and the estimate is very statistically significant. Other things being equal, 100 women with a college education will have about 51 fewer children on average than 100 women with only a high school education: $0.128 \times 4 = 0.512$. Age has a diminishing effect on fertility. The turning point in the quadratic is about

$age = 0.5321/(2 \times 0.0058) \approx 46$, by which time most women have finished having children.

The model estimated assumes that the effect of each explanatory variables, particularly education, has remained constant over years, i.e. the betas are constant. Finally, there my be heteroskedasticity in the error term, the error variance may change over time even if it does not change with the values of the explanatory variables. The heteroskedasticity-robust standard errors and test statistics are nevertheless valid. The Breusch-Pagan test would be obtained by regressing the squared OLS residuals on all of the independent variables including the year dummies. For the case of White statistic, the fitted values $\widehat{kids}$ and the squared fitted values are used as the independent variables. A weighted least squares procedure should account for variances that possibly change over time, with year dummies included. □

We can also interact a year dummy variable with key explanatory variables to see if the effect of that variable has changed over a certain time period.

Example 13.2. We estimate the effect of education and gender on wage growth from 1978 to 1985, with 1978 serving as the base year. The dummy variable $y85$ is a dummy variable equal to one if the observation comes from 1985 and zero if it comes from 1978. The intercept for 1978 is β_0 and the intercept for 1985 is $\beta_0 + \beta_{y85}$. The return to education in 1978 is β_{educ}, and the return to education in 1985 is $\beta_1 + \beta_{y85.educ}$. The $\log(wage)$ differential between women and men is β_{female} in 1978 and $\beta_{y85.female}$ in 1985.

```
df = woo.data('cps78_85')
m1 = smf.ols(formula='lwage~y85+educ+y85educ+exper+expersq+union+female+y85fem',
             data=df).fit()
------------------------------------------------------------------
               Coef.   Std.Err.      t     P>|t|    [0.025   0.975]
------------------------------------------------------------------
Intercept      0.4589    0.0934   4.9111  0.0000   0.2756   0.6423
y85            0.1178    0.1238   0.9517  0.3415  -0.1251   0.3607
educ           0.0747    0.0067  11.1917  0.0000   0.0616   0.0878
y85educ        0.0185    0.0094   1.9735  0.0487   0.0001   0.0368
exper          0.0296    0.0036   8.2932  0.0000   0.0226   0.0366
expersq       -0.0004    0.0001  -5.1513  0.0000  -0.0006  -0.0002
union          0.2021    0.0303   6.6722  0.0000   0.1427   0.2616
female        -0.3167    0.0366  -8.6482  0.0000  -0.3886  -0.2449
y85fem         0.0851    0.0513   1.6576  0.0977  -0.0156   0.1857
------------------------------------------------------------------
```

The hourly wage here is in nominal dollars. We are interested in real wage growth. We can use CPI index to convert 1985 wage in terms of 1978 dollars, it turns our that this is not necessary, provided a 1985 year dummy is included in the regression and $\log(wage)$ (as opposed to $wage$) is used as the dependent variable. Using real or nominal wage in a logarithmic functional form only affects the coefficient on the year dummy $y85$. We see that $log(wage_i/P_{85}) = log(wage_i) - log(P_{85}$, where P_{85} is the deflation factor for 1985 wages.

Now, while $wage_i$ differs across people, P_{85} does not and hence $log(P_{85})$ will be absorbed into the intercept for 1985. Hence, to study how the return to education or the gender gap has changed, we can work with nominal wages till we work with log and include a dummy for $y85$.

The return to education in 1978 is estimated to be about 7.5%; the return to education in 1985 is about 1.85 % higher, or about 9.35%. The t stats of 1.97 shows that the difference is significant at the 5% level against a two-sided alternative, $H_1 : \beta_{y85.educ} \neq 0$. In 1978, other things being equal, a woman earned about 32% less than a man. In 1985, the gap in $log(wage)$ is about -23%, and hence the gender gap appears to have fallen by about 8.5%. The t statistic on the interaction term is about 1.67 which means it is significant at the 5% level against the positive one-sided alternative, $H_1 : \beta_{y85.female} > 0$. □

If we interact all independent variables with the dummy variable, it is identical to estimating two separate equations.

Chow test for structural changes across time: It is usually more interesting to allow the intercepts to change over time and then test whether the slope coefficients have changed over time. We can test the constancy of slope coefficients generally by interacting all of the time-period dummies with one, several or all explanatory variables and test the joint significance of the interaction terms. First, we estimate the restricted model by doing a pooled regression allowing for different time intercepts; this gives SSR_r. Then we run a regression for each of the, say, T time periods and obtain the sum of the squared residuals for each time period. The unrestricted sum of squared residuals is obtained as $SSR_{ur} = SSR_1 + SSR_2 + \ldots + SSR_T$. If there are k explanatory variables with T time periods, then we are testing $(T-1)k$ restrictions, and there are $T(1+k)$ parameters estimated in the unrestricted model. So, if $n = n_1 + n_2 + \ldots + n_T$ is the total number of observations, then the *df* of the F test are $(T-1)k$ and $n - T - Tk$. We compute the F statistic as usual $[(SSR_r - SSR_{ur})/(SSR_{ur})][(n - T - Tk)/(T-1)k]$.

13.2 Policy analysis with pooled cross sections

We investigate the effect of a garbage incinerator's location on housing prices in this example. This is an example of event study showing how two cross-sectional data sets, collected before and after the occurrence of an event, can be used to determine the effect on economic outcomes. The rumor that a new incinerator would be built began after 1978, and construction began in 1981, and began operating in 1985 but was expected to be in operation soon after the start of the construction. We use prices of house that sold in 1978 and that sold in 1981. The hypothesis is that the price of houses located near the incinerator would fall relative to the price of more distant houses.

We define a house to be near the incinerator if it is within 3 miles and look at the constant 1978 dollar effect on housing prices. Let *rprice* denote the house price in real terms. A naive analysis would be to use only 1981 data and estimate the following

$$\widehat{rprice} = \underset{(3093.0)}{101307.5} - \underset{(5827.71)}{30688.27}\ nearinc$$

with $R^2 = 0.165$ and *nearinc* is the dummy variable. This shows that the average selling price for the houses near the incinerator was \$30,688.27 less than the other houses, with t statistic of 5.266, and hence we strongly reject the hypothesis that that average value for homes near and far from the incinerator are the same. This analysis is wrong, and the equation does not imply that the siting of the incinerator is causing the lower housing values. In fact if we run the regression for 1978 we get

$$\widehat{rprice} = \underset{(2653.79)}{82517.23} - \underset{(47744.59)}{18824.37}\ nearinc$$

with $R^2 = 0.082$. Therefore, even before there was any talk of an incinerator, the average value of a home near the site was \$ 18,824.37 less than the average value of a home near the site, with t statistics 3.968. This is consistent with the view that the incinerator was built in an area with lower housing values.

```
df = woo.data('kielmc')
m1 = smf.ols(formula='rprice~nearinc', data=df[df.year==1981]).fit()
m2 = smf.ols(formula='rprice~nearinc', data=df[df.year==1978]).fit()
print(m1.params.nearinc - m2.params.nearinc)
>>> -11863.903252112555
```

To answer the question, whether building a new incinerator depresses housing values we look at how $_{nearinc}$ changes between 1978 and 1981. Our estimate of the effect of the incinerator on values of homes near the incinerator side is $\hat{\delta}_1 = -30688.27 - (-18824.37) = -11863.9$, called **difference-in-differences estimator**. We need the standard error for this to test if it is significant.

```
m3 = smf.ols(formula='rprice~y81+nearinc+y81nrinc', data=df).fit()
m4 = smf.ols(formula='rprice~y81+nearinc+y81nrinc+age+agesq', data=df).fit()
m5 = smf.ols(formula='rprice~y81+nearinc+y81nrinc+age+agesq+intst+land+area'+
                    'rooms+baths', data=df).fit()
         noc      age     fset
r2     0.174    0.414    0.660
ar2    0.166    0.405    0.649
dfr  317.000  315.000  310.000
dfm    3.000    5.000   10.000
                coeff                           tval
                  noc        age       fset      noc     age   fset
Intercept   82517.228  89116.535  13807.665   30.260  37.039  1.237
age               NaN  -1494.424   -739.451      NaN -11.333 -5.639
agesq             NaN      8.691      3.453      NaN  10.248  4.248
area              NaN        NaN     18.086      NaN     NaN  7.843
baths             NaN        NaN   6977.317      NaN     NaN  2.703
intst             NaN        NaN     -0.539      NaN     NaN -2.743
land              NaN        NaN      0.141      NaN     NaN  4.551
nearinc   -18824.370   9397.936   3780.337  -3.861   1.953  0.849
rooms             NaN        NaN   3304.227      NaN     NaN  1.989
y81         18790.286  21321.042  13928.476   4.640   6.191  4.977
y81nrinc  -11863.903 -21920.270 -14177.934  -1.591  -3.447 -2.843
```

To obtain the standard error of $\hat{\delta}_1$ we can estimate $rprice = \beta_0 + \delta_0 y_{81} + \beta_1 + \delta_1 y_{81}.nearinc + u$, which is shown as the first columne (noc) in the above table. $hata_0 = 82517.23$ is the estimate of the average price of a home not hear the incinerator in 1978. The parameter $\hat{\delta}_0 = 18790.29$ captures changes in all housing values from 1978 to 1981. The coefficient on *nearinc*, $\hat{\beta}_1 = -18824.37$ measures the location effect that is not due to the presence of the incinerator. They three coefficients are all highly statistically significant. The parameter of interest on the interaction term $y_{81}.nearinc$, $\hat{\delta}_1 = -11863.90$ measures the decline in housing values due to the new incinerator. The t statistic on it is -1.59, which is marginally significant against a one-sided alternative with p-value 0.057.

Introducing various housing characteristics in the analysis can help us further for two reasons. First, other features can control for systematic difference in the kinds of homes selling near the incinerator in 1981 than those selling near the incinerator in 1978. Second, Even if the relevant house charecteristic did not change, including them can greatly reduce the error variance, which can then shrink the standard error of $\hat{\delta}_1$. In column two above, we control for age of the houses, using a quadratic. This increases the R^2 substantially, and increases the t statistics of δ_1 to 3.447. Controlling for further characteristics in column 3, further increase the R^2 and bring the t statistic for δ_1 to 2.843. This is preferred because they control for the most factors and have the smallest standard error. *nearinc* is almost insignificant now with must lower value, indicating that the characteristics included in column 3 largely capture the housing characteristics that are most important for determining housing prices.

```
m6 = smf.ols(formula='lrprice~y81+nearinc+y81nrinc', data=df).fit()
m7 = smf.ols(formula='lrprice~y81+nearinc+y81nrinc+age+agesq', data=df).fit()
m8 = smf.ols(formula='lrprice~y81+nearinc+y81nrinc+age+agesq+lintst+lland+larea'+
                    'rooms+baths', data=df).fit()
```

	lnoc	lage	lfset
r2	0.246	0.509	0.733
ar2	0.239	0.501	0.724
dfr	317.000	315.000	310.000
dfm	3.000	5.000	10.000

	coeff			tval		
	lnoc	lage	lfset	lnoc	lage	lfset
Intercept	11.285	11.371	7.652	369.839	440.693	18.399
age	NaN	-0.018	-0.008	NaN	-12.793	-5.924
agesq	NaN	0.000	0.000	NaN	11.145	4.342
baths	NaN	NaN	0.094	NaN	NaN	3.400
larea	NaN	NaN	0.351	NaN	NaN	6.813
lintst	NaN	NaN	-0.061	NaN	NaN	-1.950
lland	NaN	NaN	0.100	NaN	NaN	4.077
nearinc	-0.340	0.007	0.032	-6.231	0.138	0.679
rooms	NaN	NaN	0.047	NaN	NaN	2.732
y81	0.193	0.220	0.162	4.261	5.952	5.687
y81nrinc	-0.063	-0.185	-0.132	-0.751	-2.712	-2.531

In fact, it makes more sense to use the logarithm than the levels to get the percentage effect. The above table shows the same estimation with log used for *price* and other features.

We see in column 3 that the approximate percentage reduction in housing value due to the incinerator is about 13.2% when all characteristics are included.

The above method can be especially used for data coming from a **natural experiment** or **quasi-experiment** - when we always have a control group which is thought to be affected by the policy change. This is in contrast to **true experiment**, in which treatment and control groups are randomly and explicitly chosen. The control and treatment groups in natural experiments arise from the particular policy change. Hence we have four categories of the sample data - control group before the change, control group after the change, treatment group before the change, and treatment group after the change. The difference-in-differences parameter is called **average treatment effect**. When explanatory variables are added, the OLS estimate of δ_1 no longer has simple form, but its interpretation is similar.

13.3 Two period Panel Data Analysis

For cross-sectional data with two periods we develop the **first difference model**. We are given the crime and unemployment data for year 1982 and 1987. If we run the regression for data in 1987 we get $\widehat{crmrte} = \underset{(20.76)}{128.38} + \underset{(3.42)}{4.16}\, unem$ with $R^2 = 0.033$ and $n = 46$.

```
df = woo.data('crime2')
m1 = smf.ols(formula='crmrte~unem', data=df[df.year == 87]).fit()
```

A causal interpretation of this estimation implies that an increase in the unemployment rate lowers the crime rate, which is certainly not what we expect. The coefficient is not statistically significant so at best we have found no link between crime and unemployment rate. This regression equation likely suffers from omitted variable problems. One way to deal with the issue is to add more exogenous factors like, age, gender, education, law enforcement efforts, and so on, in a multiple regression analysis. But many factors might be hard to control for. We can also include *crmrte* from a previous year(in this case 1982) to help control for the fact that different cities have historically different crime rates. This is one way to use two years of data for estimating a causal effect.

An alternative way is the fixed effect model- to view the unobserved factors affecting the dependent variable as consisting of two types: those that are constant and those that vary over time.

$$crmrte_{it} = \beta_0 + \delta_0 d87_t + \beta_1 unem_{it} + a_i + u_{it}, \text{ for } t = 1, 2$$

Let i denote the cross-sectional unit and t denote the time period, then $crmrte_{it}$ denotes crime rate in city i at time t, variable $d87_t$ is a dummy variable for year and does not change across i, this allows the intercept to change over time capturing the secular trend. The variable a_i captures all unobserved, fixed, time-constant factors that affect $crmrte_{it}$, called **unobserved effect** or **fixed effect**, or **unobserved heterogeneity**. The error u_{it} is often called the **idiosyncratic error** or **time-varying error**.

To estimate this equation pooling together $v_{it} = a_i + u_{it}$, called composite error, is a bad idea, because a_i is generally not uncorrelated with other exogenous variables, causing the estimates to be biased (called heterogeneity bias) and inconsistent. This also causes serial correlation in the composite error further distorting the estimation. This 'wrong' pooled ols $\widehat{crmrte} = \underset{(12.74)}{93.42} + \underset{(7.98)}{7.94}\ d87 + \underset{(1.19)}{0.427}\ unem$ with $R^2 = 0.012$ and $n = 92$ does not solve the omitted variables problem.

```
m2 = smf.ols(formula='crmrte~d87+unem', data=df).fit()
```

In most pooled data application, we want a_i to be correlated with the explanatory variables. We want the unmeasured city factors in a_i that affect the crime rate also to be correlated with the unemployment rate. This can be allowed via the operation of differencing the data across the two years. For $t = 1$ we can write $y_{i1} = \beta_0 + \beta_1 x_{i1} + a_i + u_{i1}$ and for $t = 2$ we can write $y_{i2} = \beta_0 + \delta_0 + \beta_1 x_{i2} + a_i + u_{i2}$. Taking the difference we get $\Delta y_i = \delta_0 + \beta_1 \Delta x_i + \Delta u_i$. a_i has been differenced away using this **first-differenced equation** satisfying all the assumptions of the OLS in most cases.

Δu_i is uncorrelated with Δx_i if u_{it} is uncorrelated with the explanatory variable in both time periods, i.e. strict exogeneity assumption holds. This rules out x_{it} being lagged dependent variable $y_{i,t-1}$. However, we allow x_{it} to be correlated with unobservables that are constant over time. β_1 is called **first-differenced estimator**. In the crime example Δu_i and $\Delta unem_i$ are uncorrelated but if law enforcement, which is in the idiosyncratic error, increases more in cities where the unemployment rate decreases, it can cause negative correlation between Δu_i and $\Delta unem_i$. That could lead to bias. This problem can be overcome to some extent by including more factors.

Another crucial condition is that Δx_i must have some variation across i, i.e. the explanatory variable should change over time for any cross-sectional observation by possibly different amounts. For this reason gender cannot be an explanatory variable in a first differenced equation. Finally, we also need the homoskedasticity assumption and if not we can correct for homoskedasticity. We estimate the equation to $\widehat{\Delta crmrte} = \underset{(4.70)}{15.40} + \underset{(0.88)}{2.22}\ \Delta unemp$ with $R^2 = 0.127$ and $n = 46$.

```
m3 = smf.ols(formula='ccrmrte~cunem', data=df).fit()
```

This now gives a positive, statistically significant relationship between the crime and unemployment rates. The intercept means that even if $\Delta unem = 0$ the crime rate increases by 15.40 per 1000 people, reflecting the secular increase. The differencing, however, can greatly reduce the variation in the explanatory variables, leading to large standard error for $\hat{\beta}_1$. This can be combat by using a large cross section and using longer differences over time.

```
import linearmodels as plm
df['id'] = df.index//2
```

```
dff = df.set_index(['id', 'year'])
res_plm = plm.FirstDifferenceOLS.from_formula('crmrte~d87+unem', data=dff).fit()
res_plm.summary()
                    FirstDifferenceOLS Estimation Summary
================================================================================
Dep. Variable:                 crmrte   R-squared:                        0.1961
Estimator:         FirstDifferenceOLS   R-squared (Between):              0.4064
No. Observations:                  46   R-squared (Within):               0.1961
Date:                Sat, Nov 28 2020   R-squared (Overall):              0.4041
Time:                        20:49:58   Log-likelihood                  -202.17
Cov. Estimator:            Unadjusted
                                        F-statistic:                      5.3653
Entities:                          46   P-value                           0.0082
Avg Obs:                       2.0000   Distribution:                    F(2,44)
Min Obs:                       2.0000
Max Obs:                       2.0000   F-statistic (robust):             5.3653
                                        P-value                           0.0082
Time periods:                       2   Distribution:                    F(2,44)
Avg Obs:                       46.000
Min Obs:                       46.000
Max Obs:                       46.000

                             Parameter Estimates
==============================================================================
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
d87            15.402     4.7021     3.2756     0.0021      5.9257      24.879
unem           2.2180     0.8779     2.5266     0.0152      0.4488      3.9872
==============================================================================
```

Example 13.3. To estimate the tradeoff between sleeping and working, with control variables of years of education, marriage, presence of small kid and good health variable. We do not include gender or race as it does not change over time, they are part of a_i. Our primary interest is in β_{educ}. The estimated differenced equation is

```
df = woo.data('slp75_81')
m1 = smf.ols(formula='cslpnap~ctotwrk+ceduc+cmarr+cyngkid+cgdhlth', data=df).fit()
------------------------------------------------------------------------------
             Coef.    Std.Err.      t      P>|t|     [0.025     0.975]
------------------------------------------------------------------------------
Intercept   -92.6340   45.8659   -2.0197   0.0446  -182.9989   -2.2692
ctotwrk      -0.2267    0.0361   -6.2869   0.0000    -0.2977   -0.1556
ceduc        -0.0245   48.7594   -0.0005   0.9996   -96.0901   96.0411
cmarr       104.2139   92.8554    1.1223   0.2629   -78.7295  287.1574
cyngkid      94.6654   87.6525    1.0800   0.2813   -78.0274  267.3582
cgdhlth      87.5778   76.5991    1.1433   0.2541   -63.3376  238.4933
------------------------------------------------------------------------------
m1.f_test(['ceduc=0', 'cmarr=0', 'cyngkid=0', 'cgdhlth=0'])
>>> <F test: F=array([[0.86483857]]), p=0.48573468205905057, df_denom=233, df_num=4>
df.ceduc.value_counts().to_frame().T
>>>      0   1   2  5  4  3
```

```
ceduc  183  32  20  2  1  1
```

with $R^2 = 0.15$ and $n = 239$.We assume that Δu_i is uncorrelated with the changes in all explanatory variables to get consistent estimators using OLS. One more hour of work is associated with $60 \times 0.227 = 13.62$ fewer minutes of sleeping. The t statistic of -6.29 is very significant. No other estimate, except the intercept, is statistically different from zero as shown by the join F-test p value of 0.49. The standard error of $\Delta educ$ is very high, because $\sim 77\%$ values are 0, i.e. there is no change in education over the two periods.

Panel data can also be used to estimate finite distributed lag models. To estimate the effect of lagged conviction rate on current crime we use the fixed effect model for two years and estimate the first difference equation

```
df = woo.data('crime3')
m1 = smf.ols(formula='clcrime~cclrprc1+cclrprc2', data=df).fit()
------------------------------------------------------------------
              Coef.   Std.Err.      t      P>|t|    [0.025   0.975]
------------------------------------------------------------------
Intercept     0.0857    0.0638   1.3429  0.1854  -0.0425   0.2138
cclrprc1     -0.0040    0.0047  -0.8576  0.3952  -0.0135   0.0054
cclrprc2     -0.0132    0.0052  -2.5404  0.0142  -0.0236  -0.0028
------------------------------------------------------------------
```

with $R^2 = 0.193$ and $n = 53$. The second lag is negative and statistically significant, implying that a higher clear-up percentage two years ago would deter crime this year. □

program evaluation is a natural candidate for fixed effect panel data models. We obtain some data in the first time period from participants. Then a subset of these participants, called the treatment group, take part in the program in a later time period, while the one who don't take part are the control group. This is similar to the natural experiment we discussed, with one important difference: the same cross-sectional units appear in each time period.

To evaluate the effect of job training on worker productivity we regression $log(scrap)_{it}$, log of scrap rate of firm i during year t against $grant_{it}$ the binary indicator of if firm i got grant in year t. For year 1987 and 1988, the model is $log(scrap)_{it} = \beta_0 + \delta_0 y88_t + \beta_1 grant_{it} + a_i + u_{it}$, for $t = 1, 2$, with $y88_t$ begin the year dummy. The unobserved effect, a_i contains such factors as average employee ability, capital, and managerial skill; which are constant over the two year period. a_i might be systematically related to whether the firm received a grant or not. In that case, an analysis using a single cross section or just a pooling of the cross sections will produce biased and inconsistent estimators.

Differencing removes a_i to give $\Delta log(scrap)_i = \delta_0 + \beta_1 \Delta grant_i + \Delta u_i$. Since no firm received grants in 1987, $grant_{i1} = 0$ for all i, and so $\Delta grant_i = grant_{i2}$, which simply indicates whether the firm received a grant in 1988. This can be estimated to

```
df = woo.data('jtrain')
m1 = smf.ols(formula='clscrap~cgrant', data=df[(df.year==1987)|(df.year==1988)]).fit()
------------------------------------------------------------------------------
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0574      0.097     -0.591      0.557      -0.252       0.138
cgrant        -0.3171      0.164     -1.935      0.058      -0.646       0.012
------------------------------------------------------------------------------
```

with $R^2 = 0.067$ and $n = 54$. Having a job training grant is estimated to lower the scrap rate by about $e^{-0.317} - 1 = 27.2\%$. The t statistic is about -1.94, which is marginally significant. By contrast, using pooled OLS of $log(scrap)$ on $y88$ and $grant$ gives insignificant $\hat{\beta}_{grant}$. Since this differs so much from the first-difference estimates, it suggests that firms that have lower-ability workers are more likely to receive a grant.

13.4 Differencing with more than two time periods

For more than two time periods the development is similar. If we have the same T time periods for each of N cross sectional units, we say that the data set is a **balanced panel**. A general fixed effects model is

$$y_{it} = \delta_1 + \delta_2 d2_t + \ldots \delta_T dT_t + \beta_1 x_{it1} + \ldots + \beta_k x_{itk} + a_i + u_{it}$$

for $t = 1, 2, \ldots, T$. It is a good idea to allow a separate intercept for each time period to account for secular changes that are not being modeled, especially when we have a small number of them, with base period as t=1. The key assumption is that $Cov(x_{itj}, u_{is}) = 0$ for all t, s and j, i.e. the explanatory variables are strictly exogenous after we take out the unobserved effect a_i. Both omitted variable and measurement error in one or more explanatory variables can cause this assumption to fail. Having more time periods generally does not reduce the inconsistency. Differencing a poorly measured regressor reduces its variation relative to its correlation with the differenced error caused by classical measurement error, resulting in a potentially sizable bias.

If the assumptions are satisfied, differencing the equation gives us the required form. We remove one more dummy variable and introduce the intercept (which was lost due to differencing) to make the OLS R-squared interpretation easier.

$$\Delta y_{it} = \alpha_0 + \alpha_3 \Delta d3_t + \ldots + \alpha_T \Delta dT_t + \beta_1 \Delta x_{it1} + \ldots + \beta_k \Delta x_{itk} + \Delta u_{it}, \text{ for } t = 2, 3, \ldots, T.$$

The important requirement is that Δu_{it} is uncorrelated with Δx_{itj} for all j and $t = 2, 3, \ldots, T$. We have $T - 1$ time periods on each unit i for the first-differenced equation, with total number of observations being $N(T - 1)$.

For more than two time periods, we must assume that Δu_{it} is uncorrelated over time for the usual standard errors and test statistic to be valid. If we assume u_{it} are uncorrelated over

time with constant variance, then the correlation between Δu_{it} and $\Delta u_{i,t+1}$ can be shown to be -0.5, and if u_{it} follows a stable $AR(1)$ model, then Δu_{it} will be serially correlated. Only when u_{it} follows a random walk will Δu_{it} be serially uncorrelated. To test for serial correlation we first run the pooled OLS first-differenced regression to obtain the residual $\hat{r}_{it}$. When the regress $\hat{r}_{it}$ on $\hat{r}_{i,t-1}$, for $t = 3, \ldots, T$, $i = 1, \ldots, N$ and compute a t test for the coefficient on $\hat{r}_{i,t-1}$, denoted $\hat{\rho}$, which is a consistent estimator of ρ.

We can correct for the presence of AR(1) serial correlation in r_{it} by using feasible GLS, using Prais-Winsten transformation based on $\hat{\rho}$ within each cross-sectional observation. It is important to note that AR(1) process is valid across the t index and not i since the observations are independent across i. It is also possible to compute standard errors robust to serial correlation and heteroskedasticity of unknown form. If there is no serial correlation in the errors, the usual methods for dealing with heteroskedasticity works along with Breusch-Pagan and White tests for heteroskedasticity. We can also compute robust standard errors.

We now provide the assumptions for the first-differencing estimator.

- FD.1 For each i, the model is $y_{it} = \beta_1 x_{it1} + \ldots + \beta_k x_{itk} + a_i + u_{it}$, for $t = 1, \ldots, T$, where the β_j are the parameters to estimate and a_i is the unobserved effect.
- FD.2 We have a random sample from the cross section.
- FD.3 Each explanatory variable changes over time, for at least some i, and no perfect linear relationship exist among the explanatory variables.
- FD.4 For each t, the expected value of the idiosyncratic error given the explanatory variables in all time periods and the unobserved effect is zero: $E(u_{it}|\boldsymbol{X}_i, a_i) = 0$, where $\boldsymbol{X}_i$ denote the explanatory variables for all time periods for cross-sectional observation i; thus $\boldsymbol{X}_i$ contains $x_{itj}, t = 1, \ldots, T, j = 1, \ldots, k$.

When FD.4 holds, we say x_{itj} are strictly exogenous conditional on the unobserved effect. This is stronger assumption than necessary. This form emphasizes that $E(y_{it}|\boldsymbol{X}_i, a_i) = \beta_1 x_{it1} + \ldots + \beta_k x_{itk} + a_i$ so that the β_j measures partial effects of the observed explanatory variables holding fixed, the unobserved effect a_i. An important implication of FD.4 is that $E(\Delta u_{it}|\boldsymbol{X}_t) = 0$, for $t = 2, \ldots, T$. In fact, for consistency we simply need Δx_{itj} uncorrelated with Δu_{it} for all $t = 2, \ldots, T$ and $j = 1, \ldots, k$. Under the first four assumptions, the first-difference estimators are unbiased. Further the estimator is consistent with a fixed T and $N \to \infty$.

- FD.5 The variance of the differenced errors, conditional on all explanatory variables, is constant: $Var(\Delta u_{it}|\boldsymbol{X}_i) = \sigma^2$, $t = 2, \ldots, T$.
- FD.6 For all $t \neq s$ the differences in the idiosyncratic errors are uncorrelated, conditional on all explanatory variables: $Cov(\Delta u_{it}, \Delta u_{is}|\boldsymbol{X}_i) = 0$, $t \neq s$.

FW.5 ensures the differenced errors, are homoskedastic. FW.6 ensures that they are serially uncorrelated, i.e. u_{it} follow a random walk across time. Under assumptions FD.1 through FD.6, the FD estimator of β_j is the best linear unbiased estimator, BLUE, conditional on the explanatory variables.

- FD.7 Conditional on $\boldsymbol{X}_i$, the Δu_{it} are independent and identically distributed normal random variables.

With this last assumption, the t and F statistics from pooled OLS on the differences have exact t and F distributions. Without FD.7, we can rely on the usual asymptotic approximations.

Example 13.4. To study the effect of enterprise zone program on unemployment claims, we look at the data from 22 cities from 1980 to 1988, with 12 of the cities never receiving an enterprise zone over this period, serving as control group. A simple policy evaluation model is $log(uclms_{it}) = \theta_t + \beta_1 ez_{it} + a_i + u_{it}$, where $uclms_{it}$ is the number of unemployment claims filed during year t in city i. The binary variable ez_{it} is equal to one if city i at time t was an enterprise zone; we are interested in β_1. The unobserved effect a_i represents fixed factors that affect the economic climate in city i. Usually enterprise zones are usually economically depressed areas - it is likely that ez_{it} and a_i are positively correlated. We difference the equation to eliminate a_i:

$$\Delta log(uclms_{it}) = \alpha_0 + \alpha_1 d82_t + \ldots + \alpha_7 d88_t + \beta \Delta ez_{it} + \Delta u_{it}.$$

```
df = woo.data('ezunem')
# direct
m1 = smf.ols(formula='guclms~d82+d83+d84+d85+d86+d87+d88+cez', data=df).fit()
# using linearmodels
dff = df.set_index(['city', 'year'])
p1 = plm.FirstDifferenceOLS.from_formula('luclms'+
    '~d81+d82+d83+d84+d85+d86+d87+d88+ez', data=dff).fit()
------------------------------------------------------------------------
               Coef.   Std.Err.      t     P>|t|    [0.025    0.975]
------------------------------------------------------------------------
Intercept     -0.3216    0.0461  -6.9823  0.0000  -0.4126  -0.2307
d82            0.7788    0.0651  11.9544  0.0000   0.6501   0.9074
d83           -0.0331    0.0651  -0.5084  0.6118  -0.1617   0.0955
d84           -0.0171    0.0685  -0.2500  0.8029  -0.1525   0.1182
d85            0.3231    0.0667   4.8454  0.0000   0.1914   0.4547
d86            0.2922    0.0651   4.4847  0.0000   0.1635   0.4208
d87            0.0539    0.0651   0.8281  0.4088  -0.0747   0.1826
d88           -0.0171    0.0651  -0.2618  0.7938  -0.1457   0.1116
cez           -0.1819    0.0782  -2.3262  0.0212  -0.3362  -0.0275
------------------------------------------------------------------------
# heteroskedasticity test
from statsmodels.stats.diagnostic import het_breuschpagan
import patsy as pt
y, X = pt.dmatrices('guclms~d82+d83+d84+d85+d86+d87+d88+cez',
                    data=df, return_type='dataframe')
bp = het_breuschpagan(m1.resid, X)
>>> (6.913966430356902, 0.5459429362157462, 0.85358350531213, 0.5570473394318125)
# serial correlation test
yr = pd.DataFrame({'year': df.year[df.year!=1980],  'resid': m1.resid,
                   'city': df.city[df.year!=1980]})
resid = pd.pivot_table(yr, 'resid', 'year', 'city')
dff = df.set_index(['city', 'year'])
```

```
lagr = resid.shift(1).unstack()
data = pd.concat([dff, lagr], axis=1).rename(columns={0:'lagr'})
m2 = smf.ols(formula='guclms~d82+d83+d84+d85+d86+d87+d88+cez+lagr', data=data).fit()
------------------------------------------------------------------------------
               Coef.    Std.Err.      t      P>|t|    [0.025    0.975]
------------------------------------------------------------------------------
Intercept     -0.1111     0.0156  -7.1049   0.0000   -0.1419   -0.0802
d82            0.5682     0.0422  13.4551   0.0000    0.4847    0.6516
d83           -0.2437     0.0422  -5.7710   0.0000   -0.3272   -0.1602
d84           -0.2383     0.0453  -5.2624   0.0000   -0.3278   -0.1488
d85            0.1054     0.0431   2.4451   0.0157    0.0202    0.1907
d86            0.0816     0.0422   1.9318   0.0553   -0.0019    0.1650
d87           -0.1566     0.0422  -3.7091   0.0003   -0.2401   -0.0732
d88           -0.2276     0.0422  -5.3905   0.0000   -0.3111   -0.1442
cez           -0.1430     0.0785  -1.8217   0.0706   -0.2982    0.0122
lagr          -0.1965     0.0807  -2.4348   0.0161   -0.3561   -0.0370
------------------------------------------------------------------------------
# serial correlation of residual time series
data = pd.DataFrame({'t': resid.unstack(), 't1': resid.shift(1).unstack()})
m2 = smf.ols(formula='t~t1', data=data).fit()
------------------------------------------------------------------------------
               Coef.    Std.Err.      t      P>|t|    [0.025    0.975]
------------------------------------------------------------------------------
Intercept     -0.0000     0.0167  -0.0000   1.0000   -0.0331    0.0331
t1            -0.1884     0.0773  -2.4389   0.0159   -0.3411   -0.0358
------------------------------------------------------------------------------
```

This give $\hat{\beta}_1 = -0.182$ with t stats of -2.33. Therefore, it appears that the presence of an EZ causes about $e^{-0.182} - 1 = 16.6\%$ fall in unemployment claims. There is no evidence of heteroskedasticity: the Breusch-Pagan F test yields F=0.85, p-value=0.557. The lagged OLS residuals show $\hat{\rho} = -0.197$ with t stats of -2.44, so there is evidence of minimal negative serial correlation in the first-differenced errors. Because it is negative, the OLS standard errors may not be greatly understated, thus the significance of the enterprise zone dummy variable will probably not be affected. □

Example 13.5. We use data on 90 counties in North Carolina, for years 1981 to 1987, to estimate an unboserved effects model of crime. Various factors including geographical location, attitudes toward crime, historical records, and reporting conventions might be contained in a_i. The crime rate *crmrte* is number of crimes per person, *prbarr* is the estimated probability of arrest, *prbconv* is the estimated probability of conviction, *prbpris* is the probability of serving time in perison, *avgsen* is the average sentence length served, and *polpc* is the number of police officers per capita. We use the logs of all variables to estimate elasticities and include full set of year dummies. We run the model using simple OLS, heteroscedasticity robust, and then robust to to both serial correlation and heteroskedasticity.

```
m1 = smf.ols(formula='clcrmrte~d83+d84+d85+d86+d87'+
                     '+clprbarr+clprbcon+clprbpri+clavgsen+clpolpc', data=df)
df = woo.data('crime4')
m1 = smf.ols(formula='clcrmrte~d83+d84+d85+d86+d87'+
```

```
                    '+clprbarr+clprbcon+clprbpri+clavgsen+clpolpc', data=df)
r1 = m1.fit()
r2 = m1.fit(cov_type='HC0')
r3 = m1.fit(cov_type='HAC', cov_kwds={'maxlags': 2})
           coeff               stderr                   tval
             ols    hrr    hsr    ols    hrr    hsr     ols    hrr    hsr
Intercept  0.008  0.008  0.008  0.017  0.014  0.014   0.452  0.533  0.535
clavgsen  -0.022 -0.022 -0.022  0.022  0.025  0.026  -0.985 -0.880 -0.843
clpolpc    0.398  0.398  0.398  0.027  0.075  0.075  14.821  5.301  5.330
clprbarr  -0.327 -0.327 -0.327  0.030  0.051  0.053 -10.924 -6.429 -6.235
clprbcon  -0.238 -0.238 -0.238  0.018  0.031  0.033 -13.058 -7.717 -7.219
clprbpri  -0.165 -0.165 -0.165  0.026  0.035  0.040  -6.356 -4.747 -4.085
d83       -0.100 -0.100 -0.100  0.024  0.021  0.022  -4.179 -4.671 -4.547
d84       -0.048 -0.048 -0.048  0.024  0.020  0.020  -2.040 -2.391 -2.369
d85       -0.005 -0.005 -0.005  0.023  0.024  0.023  -0.196 -0.193 -0.198
d86        0.028  0.028  0.028  0.024  0.021  0.021   1.139  1.295  1.300
d87        0.041  0.041  0.041  0.024  0.023  0.024   1.672  1.754  1.735
from statsmodels.stats.diagnostic import het_white
import patsy as pt
y, X = pt.dmatrices('clcrmrte~d83+d84+d85+d86+d87+clprbarr'+
            '+clprbcon+clprbpri+clavgsen+clpolpc', data=df, return_type='dataframe')
wt = het_white(r1.resid, X)
>>> (257.5728753581, 1.003636505357e-29, 8.9193370650, 3.559273636015e-43)
```

The three probability variables are all statistically significant. The average sentence variable shows a modest deterrent effect, but it is not statistically significant. The coefficient on police per capita is somewhat surprising with a t stats of 15 in OLS. Interpreted causally, it says that a 1% increase in police per capita increases crime rate by about 0.4%. It is possible that when there are more police, more crimes are reported. But also that the police variable might be endogenous in the equation.

Assuming there is no serial correlation we can test for heteroskedasticity using White test, giving F value of 8.91 with p-value of almost 0, so there is strong evidence of heteroskedasticity. Technically, this test is not valid if there is also serial correlation, but it is strongly suggestive. The test for AR(1) yields $\hat{\rho} = -0.233$ with $t = -4.77$, so negative serial correlation exists. Since the robust standard errors are bigger, the t statistics decrease and the corresponding confidence intervals increase.

```
yr = pd.DataFrame({'year': df.year[df.year!=81], 'resid': r1.resid,
                   'county': df.county[df.year!=81]})
resid = pd.pivot_table(yr, 'resid', 'year', 'county')
dff = df.set_index(['county', 'year'])
lagr = resid.shift(1).unstack()
data = pd.concat([dff, lagr], axis=1).rename(columns={0:'lagr'})
m2 = smf.ols(formula='clcrmrte~d83+d84+d85+d86+d87+clprbarr+clprbcon'+
            '+clprbpri+clavgsen+clpolpc+lagr', data=data).fit()
------------------------------------------------------------------------
               Coef.   Std.Err.      t      P>|t|    [0.025    0.975]
------------------------------------------------------------------------
Intercept     -0.0076    0.0062   -1.2305  0.2192  -0.0197    0.0045
```

```
d83           -0.0853    0.0148   -5.7729  0.0000  -0.1143  -0.0562
d84           -0.0338    0.0150   -2.2537  0.0247  -0.0632  -0.0043
d85            0.0106    0.0148    0.7172  0.4736  -0.0185   0.0397
d86            0.0442    0.0149    2.9592  0.0033   0.0149   0.0736
d87            0.0566    0.0150    3.7886  0.0002   0.0273   0.0860
clprbarr      -0.3192    0.0331   -9.6525  0.0000  -0.3842  -0.2542
clprbcon      -0.2417    0.0200  -12.0768  0.0000  -0.2811  -0.2024
clprbpri      -0.1633    0.0279   -5.8637  0.0000  -0.2181  -0.1086
clavgsen      -0.0343    0.0245   -1.4018  0.1617  -0.0825   0.0138
clpolpc        0.4199    0.0283   14.8544  0.0000   0.3644   0.4755
lagr          -0.2332    0.0489   -4.7711  0.0000  -0.3293  -0.1371
------------------------------------------------------------------
```

□

Chow test can be applied to panel data model estimated by first differencing. We rarely want to test whether the intercepts are constant over time, instead it is more interesting to test whether slope coefficients have changed over time. This can be easily carried out by interacting the explanatory variables of interest with time-period dummy variables. Interestingly, while we cannot estimate the slopes on variables that do not change over time, we can test whether the partial effects of time-constant variables have changed over time. When we first difference we eliminate the intercepts for the first year and coefficient of variables that don't change over time, but the interaction terms survive.

As a general statement, it is important to return to the original model and remember that the differencing is used to eliminate a_i.

14 Advanced Panel Data Methods

14.1 Fixed effects estimation

An alternative method to first differencing, which works better under certain circumstances, is called the **fixed effects transformation**. For a single explanatory variable for each i we have $y_{it} = \beta_1 x_{it} + a_i + u_{it}$, for $t = 1, 2, , T$. Now for each i we average the equation over time to get $\bar{y}_i = \beta_1 \bar{x}_i + a_i + \bar{u}_i$, where $\bar{y}_i = \frac{1}{T}\sum y_{it}$ etc. Because a_i is fixed over time, we can take the difference of the two equations to get $\ddot{y}_{it} = \beta_1 \ddot{x}_{it} + \ddot{u}_{it}$, $t = 1, 2, \ldots, T$, where $\ddot{z}_{it} = z_{it} - \bar{z}_i$ is the time-demeaned data on variable z. The fixed effect transformation is also called the **within transformation**. This pooled OLS estimator is called the fixed effects or within estimator. The *between* estimator is the mean equation, and is not estimated separately as it is biased when a_i is correlated with $\bar{x}_i$.If a_i is uncorrelated with x_{it}, it is instead better to use random effects estimator, covered in the next section.

For fixed effects estimator to work we need u_{it} uncorrelated with each explanatory variable across all time periods. Further we need u_{it} to be homoskedastic and serially uncorrelated across t. In determining the degrees of freedom for the fixed effects estimator with NT observations and k independent variables (intercepts are eliminated by fixed effect estimators)

we need to use $df = NT - N - k$, where the extra loss is due to the constraint of $\ddot{u}_{it}$ summing to 0 for each i. The assumptions for fixed effect model are as follows.

- FE.1 For each i, the model is $y_{it} = \beta_1 x_{it1} + \ldots + \beta_k x_{itk} + a_i + u_{it}$, $t = 1, 2, \ldots, T$, where β_j are the parameters to estimate and a_i is the unobserved effect.
- FE.2 We have a random sample from the cross section.
- FE.3 Each explanatory variable changes over time, for at least some i, and no perfect linear relationship exist among the explanatory variables.
- FE.4 For each t, the expected value of the idiosyncratic error given the explanatory variables in all time periods and the unobserved effect is zero: $E(u_{it}|\boldsymbol{X}_i, a_i) = 0$.

These 4 assumptions are identical to the assumptions for the first-differencing estimator, and make the fixed effect estimator unbiased. Under FE.4 the estimator is consistent with fixed T as $N \to \infty$.

- FE.5 $Var(u_{it}|\boldsymbol{X}_i, a_i) = Var(u_{it}) = \sigma_u^2$ for all $t = 1, \ldots, T$.
- FE.6 For all $t \neq s$, the idiosyncratic errors are uncorrelated, conditional on all explanatory variables and a_i: $Cov(u_{it}, u_{is}|\boldsymbol{X}_i, a_i) = 0$.

Under FE.1 through FE.6, the FE estimator is BLUE. Because of FE.6 FE is is a better estimator than FD.

- FE.7 Conditional on $\boldsymbol{X}_i$ and a_i, the u_{it} are independent and identically distributed as $\mathcal{N}(0, \sigma_u^2)$.

FE.7 implies FE.4, FE.5, FE.6, but is stronger because it assumes a normal distribution for the idiosyncratic errors. With FE.7, the FE estimator is normally distributed, and t and F statistics have exact t and F distributions. Without FE.7, we can rely on asymptotic approximations requiring large N and small T.

Example 14.1. We use the scrap data on 54 firms from three years 1987, 1988 and 1989 and study the effect of grants - no firms received grants before 1988, 19 firms received grants in 1988 and 10 different firms received grants in 1989. We allow for the possibility that additional job training in 1988 made workers more productive in 1989 by including a lagged value of the grant indicator. We also include year dummies for 1988 and 1989.

```
df = woo.data('jtrain')
df = df.set_index(['fcode', 'year'])
df = df[['lscrap', 'd88', 'd89', 'grant', 'grant_1']].dropna()
p1 = plm.PanelOLS.from_formula(formula='lscrap~d88+d89+grant+grant_1+EntityEffects',
                               data=df).fit()
==============================================================================
Dep. Variable:                 lscrap   R-squared:                        0.2010
Estimator:                   PanelOLS   R-squared (Between):             -0.1103
No. Observations:                 162   R-squared (Within):               0.2010
Date:                Sat, Nov 28 2020   R-squared (Overall):             -0.0839
```

```
Time:                          21:19:51   Log-likelihood                -80.946
Cov. Estimator:            Unadjusted
                                          F-statistic:                   6.5426
Entities:                         156     P-value                        0.0001
Avg Obs:                       1.0385     Distribution:                F(4,104)
Min Obs:                       0.0000
Max Obs:                       3.0000     F-statistic (robust):          6.5426
                                          P-value                        0.0001
Time periods:                       3     Distribution:                F(4,104)
Avg Obs:                       54.000
Min Obs:                       54.000
Max Obs:                       54.000

                            Parameter Estimates
==============================================================================
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
d88           -0.0802     0.1095    -0.7327     0.4654     -0.2973      0.1369
d89           -0.2472     0.1332    -1.8556     0.0663     -0.5114      0.0170
grant         -0.2523     0.1506    -1.6751     0.0969     -0.5510      0.0464
grant_1       -0.4216     0.2102    -2.0057     0.0475     -0.8384     -0.0048
==============================================================================
F-test for Poolability: 24.661
P-value: 0.0000
Distribution: F(53,104)
Included effects: Entity
```

Time demeaning allows us to estimate β_j but we interpret the equation as follows

$$\widehat{log(scrap)}_{it} = \underset{(0.11)}{-0.08}\, d88_t - \underset{(0.13)}{0.25}\, d89_t - \underset{(0.15)}{0.25}\, grant_{it} - \underset{(0.21)}{0.42}\, grant_{i,t-1} + a_i + u_{it}$$

The estimate of lagged grant is substantially larger than the contemporaneous effect: job training has an effect at least one year later. Obtaining a grant in 1988 is predicted to lower the firm scrap rate in 1989 by $e^{-0.422} - 1 \approx -0.344$, i.e. 34.4%, with the t-stats of -2.01 significant at 5% level against a two-sided alternative. The df is obtained at $df = N(T-1) - k = 54 * (3-1) - 4 = 104$.

The coefficient on $d89$ shows that the scrap rate in 1989 was substantially lower than in base year of 1987, even in the absence of job training grants. Thus, it is important to allow for these aggregate effects. Omitting the year dummies would attribute the secular increase in productivity to job training grant. The regression shows that even after controlling for the time trends the grants have had a large effect. Finally, it is crucial to allow for the lagged effect in the model. If we omit $grant_{t-1}$, then we are assuming that the effect of job training does not last into the next year. Usually we use the within transformation based R-squared, but there are other ways to do it too. □

Although time-constant variables cannot be included by themselves in a fixed effects model, they can be interacted with variables that change over time and, in particular, with year dummy variables to see how the effect in each year differs from that in the base period.

Further, when we include a full set of year dummies (except the first year) we cannot estimate the effect of any variable whose change across time is constant, e.g. years of experience which increases at the same rate as time. The presence of a_i accounts for the base year differences, but the effect of constant change over years cannot be distinguished from the aggregate time effects.

Example 14.2. We include interaction of *educ* with year dummies for 1981 through 1987 to test whether the return to education was constant over this time period. We use $log(wage)$ as the dependent variable, dummy variables for marital and union status, a full set of year dummies, and the interaction term with *educ*.

```
df = woo.data('wagepan')
df.index = df.set_index(['nr', 'year']).index
df = df[['lwage', 'married', 'union', 'year', 'educ']]
results0 = plm.PanelOLS.from_formula(formula='lwage~married+union + C(year)'+
                        ' + EntityEffects', data=df, drop_absorbed=True).fit()
results1 = plm.PanelOLS.from_formula(formula='lwage~married+union+C(year) * educ'+
                        ' + EntityEffects', data=df, drop_absorbed=True).fit()
```

Dep. Variable:	lwage	R-squared:	0.1708
Estimator:	PanelOLS	R-squared (Between):	0.0905
No. Observations:	4360	R-squared (Within):	0.1708
Date:	Sun, Nov 29 2020	R-squared (Overall):	0.1277
Time:	11:53:43	Log-likelihood	-1350.7
Cov. Estimator:	Unadjusted		
		F-statistic:	48.907
Entities:	545	P-value	0.0000
Avg Obs:	8.0000	Distribution:	F(16,3799)
Min Obs:	8.0000		
Max Obs:	8.0000	F-statistic (robust):	5454.7
		P-value	0.0000
Time periods:	8	Distribution:	F(16,3799)
Avg Obs:	545.00		
Min Obs:	545.00		
Max Obs:	545.00		

Parameter Estimates

	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
C(year)[1980]	1.3625	0.0162	83.903	0.0000	1.3306	1.3943
C(year)[1981]	1.3400	0.1452	9.2307	0.0000	1.0554	1.6247
C(year)[1982]	1.3567	0.1451	9.3481	0.0000	1.0722	1.6412
C(year)[1983]	1.3729	0.1452	9.4561	0.0000	1.0882	1.6575
C(year)[1984]	1.4468	0.1452	9.9617	0.0000	1.1621	1.7316
C(year)[1985]	1.4122	0.1451	9.7315	0.0000	1.1277	1.6967
C(year)[1986]	1.4281	0.1451	9.8404	0.0000	1.1435	1.7126
C(year)[1987]	1.4529	0.1452	10.006	0.0000	1.1682	1.7376
married	0.0548	0.0184	2.9773	0.0029	0.0187	0.0909
union	0.0830	0.0194	4.2671	0.0000	0.0449	0.1211
C(year)[T.1981]:educ	0.0116	0.0123	0.9448	0.3448	-0.0125	0.0356
C(year)[T.1982]:educ	0.0148	0.0123	1.2061	0.2279	-0.0093	0.0388

```
C(year)[T.1983]:educ     0.0171     0.0123     1.3959     0.1628     -0.0069      0.0412
C(year)[T.1984]:educ     0.0166     0.0123     1.3521     0.1764     -0.0075      0.0406
C(year)[T.1985]:educ     0.0237     0.0123     1.9316     0.0535     -0.0004      0.0478
C(year)[T.1986]:educ     0.0274     0.0123     2.2334     0.0256      0.0033      0.0515
C(year)[T.1987]:educ     0.0304     0.0123     2.4798     0.0132      0.0064      0.0545
==============================================================================================
F-test for Poolability: 8.0932
P-value: 0.0000
Distribution: F(544,3799)

n = results0.df_resid-results1.df_resid
N = results1.df_resid
F = ((results1.rsquared-results0.rsquared)/(results0.df_resid-results1.df_resid))/
                            ((1-results1.rsquared)/results1.df_resid)
c = stats.f.ppf(1-0.05, n, N)
pval = 1-stats.f.cdf(F, n, N)
print(F, c, pval)
>>> 1.2364845415577246 2.0119908920045506 0.2786750113417985
```

The estimates on these interaction terms are all positive, and they generally get larger for more recent years. The largest coefficient of 0.030 is on $d87.educ$, with $t = 2.48$. The return to education is estimated to be about 3% larger in 1987 than in the base year 1980. If we do a joint F test for the significance of all 7 interaction terms, we get p-value of 0.28: so the set of variables is jointly insignificant though some variables are individually significant. □

14.1.1 The dummy variable regression

If we have more than 1 time periods, we can introduce dummy variables for each cross-sectional observation i and each time period t. This method is called the dummy variable regression. This gives us exactly the same estimates of β_j that we would obtain from the regression on time-demeaned data, and the standard errors and other major statistics are identical. Therefore, *the fixed effects estimator can be obtained by the dummy variable regression*, with an additional advantage that it properly computes the degrees of freedom directly. With too many explanatory variables R-squared is usually very high. They can still be used to compute F tests in usual way.

Occasionally, the estimated intercepts, say $\hat{a}_i$ are of interest. These are directly available from the dummy variable regression. After fixed effect estimation with N of any size, the $\hat{a}_i$ are pretty easy to compute by $\hat{a}_i = \bar{y}_i - \hat{\beta}_1\bar{x}_{i1} - \ldots - \hat{\beta}_k\bar{x}_{ik}$, $i = 1, \ldots, N$. Notice that time demeaning eliminates all time-constant variables, including overall intercepts. Sometimes, the average of $\hat{a}_i$ is called the intercept in these situations, where it is a consistent estimator of $\alpha = E(a_i)$. It is best to consider a_i as omitted variables that we control for through the within transformation. Even though $\hat{a}_i$ is unbiased, it it not consistent for fixed T with $N \to \infty$. If T increases a_i estimates are more accurate.

14.1.2 Fixed effects or first differencing?

For a two period case FE and FD are identical if FE (with no intercepts) includes a dummy variable for the second time period. It is same as the FD case that includes an intercept.

When $T \geq 3$, the FE and FD estimators are not the same, though both are unbiased and consistent with T fixed and $N \to \infty$. When u_{it} are serially uncorrelated, fixed effects is more efficient than first differencing, and the standard errors reported from fixed effects are valid. When unobserved factors change over time to be serially correlated, then differencing Δu_{it} can help. It is easy to test for serial correlation of Δu_{it}. In FE we can test for the serial correlation of time-demeaned errors $\ddot{u}_{it}$ but not u_{it}. If we find very little correlated in Δu_{it} FD is better, under substantial negative serial correlation FE is probably better. It is a good idea to test both to check for sensitivities.

When T is large and N is not very large, fixed effect estimator should be used with caution. Under unit root process it can give spurious regression results. First differencing has the advantage of turning an integrated process into a weakly dependent process. Inference with fixed effects estimator is potentially more sensitive to non-normality, heteroskedasticity, and serial correlation in the idiosyncratic errors. Both the estimators are very sensitive to measurement error in one ore more explanatory variables. If each x_{it} are uncorrelated with u_{it} but the strict exogeneity assumption is violated using a lagged dependent variable then FE estimator has substantially less bias than FD estimator. Theoretically, bias in FD estimator does not depend on T, while for FE estimator tends to zero at the rate $1/T$.

14.1.3 Fixed effects with unbalanced panels

If some cross-sectional values are missing for some periods, we call the sample unbalanced panel. We simply demean based on the available data and the degree of freedom is adjusted accordingly. Units for which we only have 1 observation plays no role. Provided the reason we have missing data for some i is not correlated with the idiosyncratic errors, u_{it}, the unbalanced panel causes no problems. However If the reason a unit leaves a sample (called attrition) is correlated with the idiosyncratic error - then the resulting sample section problem can cause biased estimators. However, fixed effect analysis does allow attrition to be correlated with a_i, the unobserved effect. Solving the attrition problems in panel data is complicated in general.

Example 14.3. We add two variables to our analysis $log(sales_{it})$, annual sales, and $log(employ_{it})$, number of employees. Three of the 54 firms drop out of the analysis as there is no data on them. 5 observations are lost do to missing data on one or both for some years, leaving us with $n = 148$. The unbalanced panel does not change the result much, increasing the effect of lagged grants.

```
df = woo.data('jtrain')
df = df.set_index(['fcode', 'year'])
df = df[['lscrap', 'd88', 'd89', 'grant', 'grant_1', 'lsales', 'lemploy']].dropna()
p1 = plm.PanelOLS.from_formula(formula='lscrap~d88+d89+grant+grant_1'+
```

```
                      '+lsales+lemploy+EntityEffects', data=df).fit()
==============================================================================
Dep. Variable:                  lscrap   R-squared:                      0.2131
Estimator:                    PanelOLS   R-squared (Between):           -2.2478
No. Observations:                  148   R-squared (Within):             0.2131
Date:                Sun, Nov 29 2020   R-squared (Overall):           -2.0639
Time:                        18:56:12   Log-likelihood                -68.887
Cov. Estimator:            Unadjusted
                                        F-statistic:                    4.1063
Entities:                         156   P-value                         0.0011
Avg Obs:                       0.9487   Distribution:                  F(6,91)
Min Obs:                       0.0000
Max Obs:                       3.0000   F-statistic (robust):           4.1063
                                        P-value                         0.0011
Time periods:                       3   Distribution:                  F(6,91)
Avg Obs:                       49.333
Min Obs:                       47.000
Max Obs:                       51.000

                             Parameter Estimates
==============================================================================
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
d88           -0.0040     0.1195    -0.0331     0.9736     -0.2414      0.2335
d89           -0.1322     0.1537    -0.8601     0.3920     -0.4375      0.1731
grant         -0.2968     0.1571    -1.8891     0.0621     -0.6088      0.0153
grant_1       -0.5356     0.2242    -2.3888     0.0190     -0.9809     -0.0902
lsales        -0.0869     0.2597    -0.3345     0.7388     -0.6027      0.4290
lemploy       -0.0764     0.3503    -0.2180     0.8279     -0.7722      0.6194
==============================================================================
F-test for Poolability: 20.748
P-value: 0.0000
Distribution: F(50,91)
```

□

14.2 Random effects models

We begin with the same unobserved effect model as before, $y_{it} = \beta_0 + \beta_1 x_{it1} + \ldots + \beta_k x_{itk} + a_i + u_{it}$, were we include an intercept so that we can make the assumption that the unobserved effect, a_i has zero mean, without loss of generality, with the time dummies allowed as explanatory variables. Suppose a_i is uncorrelated with each x_{itj}, then using a transformation to eliminate a_i, like FD and FE, results in inefficient estimators. Under the assumption $Cov(x_{itj}, a_i) = 0$ for $t = 1, \ldots, T$, $j = 1, \ldots, k$ we have the **random effects model**.

The ideal random effects assumptions include FE.1, FE.2. We replace FE.3 by

- RE.3 There are no perfect linear relationships among the explanatory variables.

We now allow time-constant regressors in random effect models, requiring following assumptions.

- RE.4 In addition to FE.4 we expect value of a_i given all explanatory variables is constant $E(a_i|\boldsymbol{X}_i) = \beta_0$.
- RE.5 In addition to FE.5, the variance of a_i given all explanatory variables is constant $Var(a_i|\boldsymbol{X}_i) = \sigma_a^2$.

We further assume FE.6 to hold. RE.3 rules out correlation between the unobserved effect and the explanatory variables, and is the key distinction between fixed effects and random effects. Under the first four random effects assumptions the RE estimator is consistent and asymptotically normally distributed as N gets large for fixed T. The last two assumptions are needed for the RE standard errors and test statistics to be valid. Under the six assumptions the RE estimator is asymptotically efficient.

If we define the **composite error term** as $\nu_{it} = a_i + u_{it}$ then v_{it} is serially correlated and $Cov(_{it}, \nu_{is}) = \sigma_a^2/(\sigma_a^2 + \sigma_u^2)$ for $t \neq s$. For this reason we can't estimate RE with pooled OLS, instead we use GLS, assuming large N and small T. Define $\theta = 1 - \sqrt{\sigma_u^2/(\sigma_u^2 + T\sigma_a^2)}$, which is between 0 and 1. Then the transformed equation turns out to be

$$y_{it} - \theta\bar{y}_i = \beta_0(1-\theta) + \beta_1(x_{it1} - \theta\bar{x}_{i1}) + \ldots + \beta_k(x_{itk} - \beta\bar{x}_{ik}) + (_{it} - \beta\bar{\nu}_i),$$

where the overbar denotes time average. The GLS estimator is simply the pooled OLS estimator of this **quasi-demeaned** equation, which has errors which are serially uncorrelated.

This transformation allows for explanatory variables that are constant over time, unlike FE and FD, under the assumption that a_i is uncorrelated with them. To estimate θ we generally use the form $\hat{\theta} = 1 - \sqrt{1/(1 + T\frac{\hat{\sigma}_a^2}{\hat{\sigma}_u^2})}$, where $\hat{\sigma}_a^2$ is a consistent estimator of σ_a^2 and $\hat{\sigma}^u$ is a consistent estimator of σ_u^2, based on either pooled OLS or fixed effects residuals. One possibility is to do pooled OLS on the equation with composite error and estimate $\hat{\sigma}_\nu^2$ as the square of the usual standard error of the regression. Then we can estimate $\hat{\sigma}_u^2 = \hat{\sigma}_\nu^2 - \hat{\sigma}_a^2$, where $\hat{\sigma}_a^2 = [\frac{NT(T-1)}{2} - (k+1)]^{-1} \sum_{i=1}^{N} \sum_{t=1}^{T-1} \sum_{s=t+1}^{T} \hat{\nu}_{it}\hat{\nu}_{is}$. The feasible GLS estimator that uses $\hat{\theta}$ in place of θ is called the **random effects estimator**. The estimator is consistent, but not necessarily unbiased and is asymptotically normal as N gets large with fixed T.

For $\theta = 0$ we obtain the pooled OLS estimator (when a_i is relatively unimportant) and for $\theta = 1$ be obtain the FE estimator, which is the more common case. As T gets large, $\hat{\theta}$ tends to one, and this makes RE and FE estimates very similar. Noting that $\nu_{it} - \theta\bar{\nu}_i = (1-\theta)a_i + u_{it} - \bar{u}_i$, it is clear that in the transformed equation the unobserved effect is weighted by $(1-\theta)$. If a_i is correlated with any of the x_{itj} causing inconsistency, it is attenuated by the factor $(1-\theta)$. As $\theta \to 1$, the bias term goes to zero, and RE becomes same as FE estimator. If θ is close to zero, the asymptotic bias of the RE estimator will be larger. Generally we compare pooled OLS, RE and FE to get a complete picture, enough though pooled OLS under serial correlation are invalid.

Example 14.4. We estimate the wage equation for men.

```
df = woo.data('wagepan')
df.index = df.set_index(['nr', 'year']).index
results_pols = plm.PooledOLS.from_formula(formula='lwage~C(year)+educ+black+hisp'+
                                '+exper+expersq+married+union', data=df).fit()
results_re = plm.RandomEffects.from_formula(formula='lwage~C(year)+educ+black+hisp'+
                                '+exper+expersq+married+union', data=df).fit()
results_fe = plm.PanelOLS.from_formula(formula='lwage~C(year)+educ+black+hisp'+
    '+exper+expersq+married+union+EntityEffects', data=df, drop_absorbed=True).fit()

print(results_re.theta.iloc[0,0])
>>> 0.6450593029243452
```

We us three methods: pooled OLS, random effects and fixed effects. In the first two methods, we can include *educ* and race dummies (*black* and *hisp*) but these drop out of the fixed effect analysis. The time-varying variables are *exper*, $exper^2$, *union*, *married*. *exper* also drops out of FE because it changes at a constant rate and is absorbed in the year dummies inclusion. Each regression also contains full set of year dummies.

```
from linearmodels.panel import compare
compare({'pooledOLS': results_pols, 'RE': results_re, 'FE': results_fe})
========================================================================
                            pooledOLS                RE             FE
------------------------------------------------------------------------
Dep. Variable                   lwage             lwage          lwage
Estimator                  PooledOLS     RandomEffects       PanelOLS
No. Observations                4360             4360           4360
Cov. Est.                 Unadjusted       Unadjusted     Unadjusted
R-squared                     0.1893           0.1806         0.1806
R-Squared (Within)            0.1692           0.1799         0.1806
R-Squared (Between)           0.2066           0.1853        -0.0052
R-Squared (Overall)           0.1893           0.1828         0.0807
F-statistic                   72.459           68.409         83.851
P-value (F-stat)              0.0000           0.0000         0.0000
====================     ============   ===============   ============
C(year)[1980]                 0.0921           0.0234         1.4260
                             (1.1761)         (0.1546)       (77.748)
C(year)[1981]                 0.1504           0.0638         1.5772
                             (1.7935)         (0.3988)       (72.966)
C(year)[1982]                 0.1548           0.0543         1.6790
                             (1.7335)         (0.3211)       (63.258)
C(year)[1983]                 0.1541           0.0436         1.7805
                             (1.6323)         (0.2450)       (53.439)
C(year)[1984]                 0.1825           0.0664         1.9161
                             (1.8437)         (0.3551)       (45.982)
C(year)[1985]                 0.2013           0.0811         2.0435
                             (1.9523)         (0.4136)       (39.646)
C(year)[1986]                 0.2340           0.1152         2.1915
                             (2.1920)         (0.5617)       (34.771)
C(year)[1987]                 0.2659           0.1583         2.3510
                             (2.4166)         (0.7386)       (30.867)
```

```
educ                          0.0913           0.0919
                             (17.442)         (8.5744)
black                         -0.1392          -0.1394
                             (-5.9049)        (-2.9054)
hisp                          0.0160           0.0217
                             (0.7703)         (0.5078)
exper                         0.0672           0.1058
                             (4.9095)         (6.8706)
expersq                       -0.0024          -0.0047        -0.0052
                             (-2.9413)        (-6.8623)      (-7.3612)
married                       0.1083           0.0638         0.0467
                             (6.8997)         (3.8035)       (2.5494)
union                         0.1825           0.1059         0.0800
                             (10.635)         (5.9289)       (4.1430)
===================== ============== ================ ==============
Effects                                                      Entity
-------------------------------------------------------------------------
```

The coefficients on *educ*, *black*, and *hisp* are similar for the pooled OLS and random effects estimates, which lower standard error for pooled OLS because it ignores the positive serial correlation. The *exper* profile is somewhat different. But $exper^2$, *married*, and *union* all fall as we go from Pooled OLS to Random effects to Fixed effects where we remove all the unobserved effect entirely. The drop in marriage premium is consistent with the idea that men who are more able - as captured by higher unobserved effect a_i - are more likely to be married. Therefore, in the pooled OLS estimation, a large part of the marriage premium reflects the fact that men who are married would earn more even if they were not married. The remaining 4.7% has two possible explanations: marriage really makes men more productive or employers pay married men a premium as a sign of stability. We can not distinguish between these two hypotheses.

The estimate of θ for the random effects estimation is $\hat{\theta} = 0.643$, which helps explain why on the time-varying variables, the RE estimates like closer to the FE estimates than to the pooled OLS estimates. □

14.2.1 Random effects or fixed effects?

Because FE allows arbitrary correlation between a_i and the x_{itj} it is preferred to RE. RE is used when the key explanatory variable is constant over time and we can assume the unobserved effect is uncorrelated with all explanatory variables. RE is preferred to pooled OLS because RE is generally more efficient.

It is fairly common to see researchers apply both random effects and fixed effects, and then formally test for statistically significant differences in the coefficients on the time-varying explanatory variables. **Hausman test** is used under full set of random effects assumptions. One uses the random effects estimates unless the Hausman test is rejected. Failure to reject the test means either RE and FE estimates are sufficiently close, or the sampling variation is so large in FE estimates that one cannot distinguish the difference statistically. The re-

jection of the test means the key RE assumption, is false, and then the FE estimates are used.

Using FE is mechanically the same as allowing a different intercept for each cross-sectional unit. Fortunately, whether or not we engage in the philosophical debate about the nature of a_i, FE is almost always much more convincing than RE for policy analysis using aggregated data.

Example 14.5. We compare the FE and RE models from the previous example using Hausman test here. The null hypothesis is random effect model and is strongly rejected here. Hence, it is more appropriate to use FE model in this case.

```
col = results_fe.params.index.intersection(results_re.params.index)
psi = results_fe.cov.loc[col, col] - results_re.cov.loc[col, col]
diff = results_fe.params.loc[col] - results_re.params.loc[col]
W = np.abs(diff.dot(np.linalg.inv(psi)).dot(diff))
dof = len(col)
pvalue = stats.chi2(dof).sf(W)
print("Hausman Test: chisq = {0}, df = {1}, p-value = {2}".format(W, dof, pvalue))
>>> Hausman Test: chisq = 43.427071177106974, df = 11, p-value = 9.150613846026183e-06
```

□

14.3 The correlated random effects approach

Because a_i is, by definition, constant over time, allowing it to be correlated with the average level of x_{it}, i.e $\bar{x}_i = \frac{1}{T}\sum_{t=1}^{T} x_{it}$, models the correlation between a_i and x_{it} directly as $a_i = \alpha + \gamma\bar{x}_i + r_i$. Here, r_i is uncorrelated with each x_{it}. This directly implies that $Cov(\bar{x}_i, r_i) = 0$. The **correlated random effects** (CRE) approach gives the combined model of $y_{it} = \alpha + \beta x_{it} + \gamma\bar{x}_i + r_i + u_{it}$. u_{it} is also uncorrelated with $\bar{x}_i$ due to strict exogeneity assumption. It is the addition of $\bar{x}_i$ that controls for the correlation between a_i and the sequence x_{it}. What is left over, r_i, is uncorrelated with the x_{it}. We can now apply the RE to estimate this equation and it turns out that $\hat{\beta}_{CRE} = \hat{\beta}_{FE}$. In other words, adding the time average $\bar{x}_i$ and using random effects is the same as subtracting the time averages and using pooled OLS. In CLE we control for the average level $\bar{x}_i$ when measuring the partial effect of x_{it} on y_{it}.

CRE synthesises the FE and RE approaches. But more importantly, provides a simple way to choose between them by constructing a t test of $H_0 : \gamma = 0$ against $H_1 : \gamma \neq 0$. If we reject H_0 we reject RE in favor of FE. Cluster based heteroskedasticity and serial correlation adjustment can be used to do this estimation and testing. CRE also provides a way to include time-constant explanatory variables in what is effectively a fixed effects analysis. Once we include $\bar{x}_i$, we can include any other time-constant variables in the equation, estimate it by RE, and obtain $\hat{\beta}_{FE}$ as the coefficient on x_{it}. In addition, we obtain an estimate of coefficient on time-constant variables.

When the panel is balanced there is no need to include the time averages of variables that change over time as it is covered by the intercept. For unbalance panel data set, time averages of any variable, which depends on how many periods we have for cross-sectional unit i, that changes over time must be included. In particular, for y or any x_j a time period contributes to the time average, $\bar{y}_i$ or $\bar{x}_{ij}$, only if data on all of $(y_{it}, x_{it1}, \ldots, x_{itk})$ are observed. This is denoted by dummy selection indicator s_{it} which is zero when at least one element of the cross sectional data is missing. Hence, $\bar{y}_i = \frac{1}{T_i} \sum_{t=1}^{T} s_{it} y_{it}$, where T_i is the total number of complete time periods for cross-sectional observation i. When time period dummies are included in the model, or any other variable that changes only by t and not i, we must include their time averages.

Once the time averages have been properly obtained, using a RE estimation is same as in the balanced case. In the pure random effects case, the selection indicator s_{it} cannot be correlated with the composite error $a_i + u_{it}$, in any time period, otherwise the RE estimator is inconsistent. The FE estimator allows for arbitrary correlation between the selection indicator s_{it} and fixed effect a_i. Therefore, FE estimator is more robust in the context of unbalanced panels.

Example 14.6. We apply the CRE method on the wage data and compare it against FE and RE numbers. We find that the coefficients are same as the FE model. Hence CRE seem to be a valid way to get FE estimates. We can now compare between the FE and RE model by testing the coefficients on the averaged variables included in CRE.

```
df = woo.data('wagepan')
df['t'] = df.year
df['entity'] = df.nr
df = df.set_index(['nr'])
gmean = df.groupby('nr').mean()
df['married_b'] = gmean['married']
df['union_b'] = gmean['union']
df = df.set_index(['year'], append=True)

results_fe = plm.PanelOLS.from_formula(formula='lwage~C(t)*educ+married+union'+
                        '+EntityEffects', data=df, drop_absorbed=True).fit()
results_cre = plm.RandomEffects.from_formula(formula='lwage~C(t)*educ+married'+
                        '+union+married_b+union_b', data=df).fit()
results_re = plm.RandomEffects.from_formula(formula='lwage~C(t)*educ+married'+
                        '+union', data=df).fit()
from linearmodels.panel import compare
a=compare({ 'FE': results_fe, 'CRE': results_cre, 'RE': results_re})

a.params.iloc[-6:]
                          FE       CRE        RE
C(t)[T.1987]:educ  0.030433  0.030433  0.029699
married            0.054820  0.054820  0.077258
union              0.082978  0.082978  0.107505
educ                    NaN  0.058594  0.059473
married_b               NaN  0.127336       NaN
union_b                 NaN  0.160484       NaN
```

```
a.tstats.iloc[-6:]
                          FE       CRE        RE
C(t)[T.1987]:educ  2.479819  2.484283  2.419458
married            2.977340  2.982700  4.605448
union              4.267104  4.274786  5.991787
educ                    NaN  4.917946  4.987051
married_b               NaN  2.856625       NaN
union_b                 NaN  3.190079       NaN

wtest = results_cre.wald_test(formula='married_b = union_b = 0')
H0: Linear equality constraint is valid
Statistic: 19.4058
P-value: 0.0001
Distributed: chi2(2)
```

We do a F-test or a very similar Wald-test which clearly rejects the null hypothesis that the RE model is appropriate with a time p value. As an advantage of the CRE approach we can add time-constant regressors to the model. We add back *educ*, *black*, and *hisp*.

```
results_cre = plm.RandomEffects.from_formula(formula='lwage~educ+married+union'+
                        '+married_b+union_b+educ+black+hisp', data=df).fit()
================================================================================
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
--------------------------------------------------------------------------------
educ           0.1257     0.0023     55.484     0.0000      0.1213      0.1302
married        0.2417     0.0177     13.677     0.0000      0.2070      0.2763
union          0.0700     0.0207     3.3804     0.0007      0.0294      0.1107
married_b     -0.0436     0.0450    -0.9685     0.3329     -0.1318      0.0446
union_b        0.2105     0.0519     4.0576     0.0001      0.1088      0.3122
black         -0.0892     0.0499    -1.7864     0.0741     -0.1871      0.0087
hisp           0.0784     0.0426     1.8428     0.0654     -0.0050      0.1619
================================================================================
```

Finally, we can use cross-sectional cluster based heteroskedasticity and serial-correlation robust estimation. We see that robust version has the same coefficients but the standard errors differ, but are pretty close to the non-robust version.

```
results_cre_robust = plm.RandomEffects.from_formula(formula='lwage~educ+married+union'+
                     '+married_b+union_b+educ+black+hisp', data=df)
                     .fit(cov_type='clustered', cluster_entity=True)
a=compare({ 'CRE': results_cre, 'CRE_robust': results_cre_robust})
==================================================
                                CRE      CRE_robust
--------------------------------------------------
educ                         0.1257          0.1257
                            (55.484)        (52.073)
married                      0.2417          0.2417
                            (13.677)        (10.994)
union                        0.0700          0.0700
```

```
                            (3.3804)        (2.7858)
married_b                    -0.0436         -0.0436
                            (-0.9685)       (-0.9463)
union_b                       0.2105          0.2105
                             (4.0576)        (4.3000)
black                        -0.0892         -0.0892
                            (-1.7864)       (-1.7360)
hisp                          0.0784          0.0784
                             (1.8428)        (1.9461)
-------------------------------------------------------
```

□

The various panel methods can be applied to certain data structures that do not involve time. For example differencing is used in between siblings to remove family effect. These are examples of **matched pair samples**. More generally, fixed and random effects methods can be applied to a **cluster sample**. Cluster sample has the same appearance as a cross-sectional data set, but clusters of units are sample from a population of clusters rather than sampling individuals from the population of individuals. Fixed effects estimation is preferred when we think the unobserved cluster effect, e.g. a_i, is correlated with one or more of the explanatory variables. The correlated random effects approach can be applied as well, since a cluster sample acts like an unbalanced panel, with averages calculated within the clusters. The notion of serial correlation is no longer relevant here, but cluster-robust standard errors should be used. However, if the set of data is obtained from a random sample from the population, then there is no reason to account for cluster effects in computing standard errors after OLS estimation. The fact that the units can be put into groups ex post - after the random sample has been obtained - is not a reason to make inference robust to cluster correlation. In a true cluster sample, the clusters are first drawn from a population of clusters, and then individuals are drawn from the clusters. With large cluster sizes the resulting cluster correlation is generally unimportant, but with small cluster sizes one should use the cluster-robust standard errors.

15 Instrumental Variables Estimation and Two Stage Least Squares

We further study the problem of endogenous explanatory variables in multiple regression models. OLS is generally inconsistent under omitted variables. This bias can be mitigated when a suitable proxy variable is given for an unobserved explanatory variable, unfortunately, which is not not always possible. The Panel data methods only account for time-constant omitted variables, and not for time-varying omitted variables that are correlated with the explanatory variables. In this section we introduce the method of instrumental variables (IV) to solve the problem of endogeneity of one or more explanatory variable. The method of two stage least sqaures (2SLS) is second in popularity to OLS. IV can also be used to solve the errors-in-variables problem under certain conditions.

15.1 Omitted variables

An IV approach may not be necessary at all if a good proxy exists. In the absence of a good proxy variable, with IV approach we leave the unobserved variable in the error term, but rather than estimating the model by OLS, we use an estimation method that recognizes the presence of the omitted variable. For $y = \beta_0 + \beta_1 x + u$ with $Cov(x, u) \neq 0$ we need some additional information through a new variable z, that satisfies the following two properties:

1. **instrument exogeneity**: $Cov(z, u) = 0$, i.e. Z is exogenous in the equation. It means that z should have no partial effect on y, after x and omitted variables have been controlled for, and z should be uncorrelated with the omitted variables. This assumption can't be tested in general. Hence, this assumption should be maintained by appealing to economic behavior or introspection.

2. **instrument relevance**: $Cov(z, x) \neq 0$, i.e. z is relevant for explaining variation in x. It means that z must be related, either positively or negatively, to the endogenous explanatory variable x. This assumption is easy to test by the regression $x = \pi_0 + \pi_1 z + \nu$ with null hypothesis $H_0 : \pi_1 = 0$ against a two sided alternative $H_1 : \pi_1 \neq 0$.

Example 15.1. Consider the problem of unobserved ability in a wage equation for working adults $log(wage) = \beta_0 + \beta_1\ educ + \beta_2\ abil + e$. A proxy variable for *abil* such as IQ can be substituted for ability. But if a suitable proxy is not available then we put *abil* in the error term $log(wage) = \beta_0 + \beta_1\ educ + u$, where u contains *abil*. An OLS estimation of this will be biased and inconsistent estimator of β_1 if *educ* and *abil* are correlated. An instrumental variable z for *educ* must be (1) uncorrelated with ability and (2) correlated with education. Here are some bad choices for the instrumental variable

- The last digit of an individual's Social security number - it satisfies the first requirement but not second.

- Proxy variable for *abil* like IQ - since it is highly correlated with *abil*, while and IV must be uncorrelated with *abil*.

Here are some good choices for instrumental variables:

- Economist have uses family background variables, e.g. mother's education, as IVs for education, since it is positively correlated with child's education. Mother's education could also be correlated with child's ability but that is, generally, less likely.

- Another IV choice for *educ* is number of siblings while growing up, since it is typically associated with lower average levels of education and is uncorrelated with ability.

□

It is important to take note of the sign of $\hat{\pi}_1$ and not just the statistical significance. If we expect it to be positive and get a negative value, it would suggest that there are important omitted variables driving negative correlation - variables that might themselves have to be included in the model.

Given the two assumption we can now identify $\beta_1 = \frac{Cov(z,y)}{Cov(z,x)}$ and we can get **instrumental variables (IV) estimator** of β_1 as $\hat{\beta}_1 = \frac{\sum(z_i-\bar{z})(y_i-\bar{y})}{\sum(z_i-\bar{z})(x_i-\bar{x})}$ and $\hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x}$. When x is exogenous, then x can be its own IV, i.e. $z = x$, and IV estimator is identical to the OLS estimator. IV estimators are consistent but essentially never unbiased and hence large samples are preferred.

IV estimators have approximate normal distribution in large sample sizes. We impose the heteroskedasticity assumption as $E(u^2|z) = \sigma^2 = Var(u)$. Under these conditions the asymptotic variance of $\hat{\beta}_1$ is $\sigma_u^2/(n\sigma_x^2\rho_{xz}^2)$ and decreases to 0 at the rate of $1/n$, where n is the sample size. This can be consistently estimated using sample quantities to be $\hat{\sigma}^2_{\hat{\beta}_1} = \hat{\sigma}_u^2/(SST_x R^2_{x,z})$, where $\hat{\sigma}_u^2 = \frac{1}{n-2}\sum \hat{u}_i^2$ and SST_x is the total sum of squares of the x_i. This can be used to construct the t statistics for hypotheses involving β_1. Under Gauss-Markov assumptions the variance of the OLS estimator is σ^2/SST_x. The IV estimator only differs by $R^2_{x,z}$ in the denominator, showing that the IV variance is always larger than OLS variance.

Example 15.2. We estimate the return to education for married women as

```
df = woo.data('mroz')
res_ols = smf.ols(formula='lwage~educ', data=df).fit()
                coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.1852      0.185     -1.000      0.318      -0.549       0.179
educ           0.1086      0.014      7.545      0.000       0.080       0.137
```

The estimate for β_1 implies an almost 11% return for another year of education with a t statistic of 7.55. Next, we use father's education as an instrumental variable for *educ*. We maintain that $fatheduc$ is uncorrelated with u. To check the second assumption we note that the correlation between $fatheduc$ and *educ* is 44%. Using $fatheduc$ as an IV for *educ* gives

```
print(df.fatheduc.corr(df.educ))
>>> 0.4424582341056476
import linearmodels.iv as iv
res_iv = iv.IV2SLS.from_formula(formula='lwage~1+[educ ~ fatheduc]', data=df).fit()
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
Intercept      0.4411     0.4643     0.9501     0.3421     -0.4689      1.3511
educ           0.0592     0.0369     1.6017     0.1092     -0.0132      0.1316
```

The IV estimate os the return to education is 5.9%, which i about half of the OLS estimate with a t statistic of 1.6, primarily because the standard deviation is much higher now for β_{educ}. This suggests that the OLS estimate is too high and is consistent with omitted ability bias. We can never know whether 0.109 is above the true return to education, or whether

0.059 is closer to the true return to education. The differences between the two are practically large, but the confidence interval of IV estimate incorporates the OLS estimate, so we can't say whether the difference is statistically significant.

Similarly, we can find the return to education for men as

```
df = woo.data('wage2')
res_ols = smf.ols(formula='lwage~educ', data=df).fit()
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      5.9731      0.081     73.403      0.000       5.813       6.133
educ           0.0598      0.006     10.035      0.000       0.048       0.072
```

The OLS estimate of β_1 is 0.059 with t statistic of 10. We use number of siblings as an instrument for *educ*. These are negative correlated as seen from the -24% correlation. Using *sibs* as an IV for *educ* gives

```
print(df.sibs.corr(df.educ))
>>> -0.23928810445331136
res_iv = iv.IV2SLS.from_formula(formula='lwage~1+[educ ~ sibs]', data=df).fit()
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
--------------------------------------------------------------------------------
Intercept      5.1300     0.3304     15.528     0.0000      4.4825      5.7776
educ           0.1224     0.0246     4.9850     0.0000      0.0743      0.1706
```

This has a negative R^2! The IV estimate of β_1 is 0.1224 with t-statistic of 5 with much higher standard deviation. The difference is statistically significant as the confidence intervals do not intersect. This is not in accordance with the omitted ability bias from OLS. It could either be that *sibs* is also correlated with ability or that the OLS estimator is biased towards 0 because of measurement error in *educ*, which is very unlikely as *educ* does not seem to have errors-in-variables issue. □

Nothing prevents the explanatory variable or IV from being binary variables. For example, binary variable with is 1 if the man was born in the first quarter of the year can serve as IV for education, since it is unrelated to ability. It is, however, related to education - students born early in the year typically begin school at an older age and hence reach the compulsory schooling age of 16 with somewhat less education than students who begin school at a younger age. The value $R^2_{x,z}$ is very small and hence needs large sample size to get reasonably precise IV estimates.

If the IV estimate is equal to OLS estimate it shows that there is no omitted variable bias. Small correlation between z and x cause large standard errors. However, weak correlation between z and x can cause large asymptotic bias in case of even a small amount of correlation between z and u. This is highlighted by the probability limit of the IV estimator $plim\ \hat{\beta}_{1,IV} = \beta_1 + \frac{Corr(z,u)}{Corr(z,x)}\frac{\sigma_u}{\sigma_x}$. For the OLS case we have $plim\ \hat{\beta}_{1,OLS} = \beta_1 + Corr(x,u)\frac{\sigma_u}{\sigma_x}$.

When $Corr(z, x)$ is small then a seemingly small correlation between z and u can be magnified and make IV worse than OLS, even if restrict attention to bias. This is especially bad when z is uncorrelated with x, and hence we should always check this assumption.

Of practically greater interest is the so-called problem of **weak instruments**, which is defined as low correlation between z and x. Modelling the correlation between z and x as a function of sample size; in particular, the correlation is assumed to shrink to zero at the rate $1/\sqrt{n}$. Not surprisingly, the asymptotic distribution of the instrumental variables estimator is different compared with the usual asymptotics, where the correlation is assumed to be fixed and nonzero. Finally, when x and u are correlated, we can not decompose the variance of y into $\beta_1^2 Var(x) + Var(u)$, and so the R-squared has no natural interpretation, and can be negative. These R-squared cannot be used in the usual way to compute F tests of joint restrictions. A high R-squared resulting from OLS is of little comfort if we cannot consistently estimate β_1. IV methods are intended to provide better estimates of the ceteris paribus effect of x on y when x and u are correlated; goodness-of-fit is not a factor.

15.2 IV estimation

We now consider a multivariate regression case where only one of the explanatory variables is correlated with the error $y_1 = \beta_0 + \beta_1 y_2 + \beta_2 z_1 + u_1$. We call it the **structural equation**.The dependent variable y_1 is clearly endogenous, as it is correlated with u_1. The variables y_2 and z_1 are the explanatory variables, and u_1 is the error with the assumption $E(u_1) = 0$. We use z_1 to indicate that this variable is exogenous ($Cov(z_1, u_1) = 0$) and use y_2 to indicate that this variable is suspected of being correlated with u_1. We can think of u_1 containing an omitted variable correlated with y_2. An OLS estimation gives biased and inconsistent estimators.

We seek an instrumental variable for y_2. Since, z_1 appear as an explanatory variable, it cannot serve as an instrumental variable for y_2 even it if it correlated with y_2. We need another exogenous variable - call it z_2. Thus the assumptions are $E(u_1) = 0, Cov(z_1, u_1) = 0, Cov(z_2, u_1) = 0$. We can take the sample counterpart of these 3 equations to solve for $\hat{\beta}_0$, $\hat{\beta}_1$, and $\hat{\beta}_2$.

$$\sum_{i=1}^{n}(y_{i1} - \hat{\beta}_0 - \hat{\beta}_1 y_{i2} - \hat{\beta}_s z_{i1}) = 0$$
$$\sum_{i=1}^{n} z_{i1}(y_{i1} - \hat{\beta}_0 - \hat{\beta}_1 y_{i2} - \hat{\beta}_2 z_{i1}) = 0$$
$$\sum_{i=1}^{n} z_{i2}(y_{i1} - \hat{\beta}_0 - \hat{\beta}_1 y_{i2} - \hat{\beta}_2 z_{i1}) = 0$$

The estimators are called instrumental variables estimators. If we think y_2 is exogenous and we choose $z_2 = y_2$, then it is exactly same as the first order conditions for the OLS estimators.

We also need z_2 to be correlated with y_2. The easiest way to state this condition is to write the endogenous explanatory variable as a linear function of exogenous variables

$y_2 = \pi_0 + \pi_1 z_1 + \pi_2 z_2 + \nu_2$ (called the **reduced form equation**), where $E(\nu_2) = 0$, $Cov(z_1, \nu_2) = 0$, and $Cov(z_2, \nu_2) = 0$. The key identification condition is that $\pi_2 \neq 0$. This should always be tested using a t test after possibly making it robust to heteroskedasticity. Unfortunately, we can not test that z_1 and z_2 are uncorrelated with u_1, we hope that is implied by economic reasoning or introspection.

Extending it to more exogenous explanatory variables is straightforward. The structural model as $y_1 = \beta_0 + \beta_1 y_2 + \beta_2 z_1 + \ldots + \beta_k z_{k-1} + u_1$, where y_2 is thought to be correlated with u_1. Let z_k bet he IV not in this equation already, so we assume, $E(u_1) = 0$, $Cov(z_j, u_1) = 0$ for $j = 1, \ldots, k$. $z_1, \ldots, z_{k-1}$ are the exogenous variables acting as their own instrumental variables in estimating the β_j The reduced form for y_2 is $y_2 = \pi_0 + \pi_1 z_1 + \ldots + \pi_k z_k + \nu_2$, where we want $\pi_k \neq 0$ for z_k to be a valid IV for y_2.

Example 15.3. We use wage and education data for a sample of men to estimate the return to education. We use the dummy variable for whether someone grew up near a four-year college (*nearc4*) as an instrumental variable for education, in a log(*wage*) equation. In order for it to be a valid IV it must be partially correlated with *educ* after controlling for other exogenous variables, and uncorrelated with the error term. We estimate the reduced form equation for *educ* and see a coefficient of 0.32 with a t statistic of 3.77 giving credence to our assumption.

```
df = woo.data('card')
reduced = smf.ols(formula='educ~nearc4+exper+expersq+black+smsa+south+smsa66+'
    'reg662+reg663+reg664+reg665+reg666+reg667+reg668+reg669', data=df).\
    fit(cov_type='HC0')
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     16.6383      0.215     77.456      0.000      16.217      17.059
nearc4         0.3199      0.085      3.770      0.000       0.154       0.486
exper         -0.4125      0.032    -12.896      0.000      -0.475      -0.350
expersq        0.0009      0.002      0.510      0.610      -0.002       0.004
black         -0.9355      0.092    -10.138      0.000      -1.116      -0.755
smsa           0.4022      0.111      3.625      0.000       0.185       0.620
south         -0.0516      0.142     -0.365      0.715      -0.329       0.226
```

The OLS and IV estimates are next calculated. The IV estimates of the return to education is almost twice as large as the OLS estimate, but the standard error of IV estimate is over 18 times larger than the OLS standard error, and hence the IV t statistics is much lower than that of OLS. As states, we should not make anything out oof the smaller R^2 in the IV estimation, as OLS R-squared will always be later because OLS minimizes the sum of squared residuals.

```
res_ols = smf.ols(formula='lwage~educ+exper+expersq+black+smsa+south+smsa66+'
    'reg662+reg663+reg664+reg665+reg666+reg667+reg668+reg669',
    data=df).fit()
==============================================================================
```

```
                coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      4.2332      0.061     69.316      0.000       4.113       4.353
educ           0.0747      0.003     21.351      0.000       0.068       0.082
exper          0.0848      0.007     12.806      0.000       0.072       0.098
expersq       -0.0023      0.000     -7.223      0.000      -0.003      -0.002
black         -0.1990      0.018    -10.906      0.000      -0.235      -0.163
smsa           0.1364      0.020      6.785      0.000       0.097       0.176
south         -0.1480      0.026     -5.695      0.000      -0.199      -0.097

res_iv = iv.IV2SLS.from_formula(formula='lwage~1+exper+expersq+black+smsa+south+'
    +'smsa66+reg662+reg663+reg664+reg665+reg666+reg667+reg668+reg669+[educ ~ nearc4]',
     data=df).fit()
==============================================================================
             Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
Intercept      3.6662     0.9085     4.0352     0.0001      1.8855      5.4468
exper          0.1083     0.0233     4.6376     0.0000      0.0625      0.1540
expersq       -0.0023     0.0003    -6.7128     0.0000     -0.0030     -0.0017
black         -0.1468     0.0524    -2.8031     0.0051     -0.2494     -0.0441
smsa           0.1118     0.0311     3.5995     0.0003      0.0509      0.1727
south         -0.1447     0.0291    -4.9775     0.0000     -0.2016     -0.0877
smsa66         0.0185     0.0205     0.9035     0.3663     -0.0217      0.0587
educ           0.1315     0.0540     2.4353     0.0149      0.0257      0.2373
```

□

We can also write the reduced form equation for y_1 which is used to study the effects of policy interventions. For z_k being the IV for y_2, the reduced form for y_1 has the form $y_1 = \gamma_0 + \gamma_1 z_1 + \ldots + \gamma_k z_k + e_1$, where $\gamma_j = \beta_j + \beta_1 \pi_j$ for $j < k$ and $\gamma_k = \beta_1 \pi_k$, and $e_1 = u_1 + \beta_1 \nu_2$. Since z_j are exogenous variables, γ_j can be consistently estimated by OLS. We need to apply the IV only if we need the β_1.

When y_2 is a binary variable denoting participation and z_k is a binary variable denoting eligibility for program participation γ_k has an interesting interpretation. Rather then an estimate of the program itself, it is an estimate of the effect of offering the program. β_1 measures the effect of the program itself. γ_k accounts for the possibility that some units made eligible will choose not to participate. γ_k is called the intention-to-treat parameter. The intention-to-treat coefficient $\gamma_k = \beta_1 \pi_k$, depends on the effect of participating β_1, and the change in probability of participating due to being eligible π_k.

15.3 Two stage least squares

We now discuss how to use multiple instrumental variables for a single endogenous explanatory variable. Consider again the model $y_1 = \beta_0 + \beta_1 y_2 + \beta_2 z_1 + u_1$ with one endogenous and one exogenous explanatory variables. Say we have two exogenous variables excluded and z_2 and z_3 are uncorrelated with u_1 (exclusion restrictions) and are both correlated with y_2, we could use both of them as IV individually. In fact, any linear combination of the exogenous variables is a valid IV and we need to find the best, the most correlated with y_2.

The reduced form of y_2 is $y_2 = \pi_0 + \pi_1 z_1 + \pi_2 z_2 + \pi_3 z_3 + \nu_2$, where $E(\nu_2) = 0$, $Cov(z_j, \nu_2) = 0$ for $j = 1, 2, 3$. Then, the best IV for y_2 is $y_2^* = \pi_0 + \pi_1 z_1 + \pi_2 z_2 + \pi_3 z_3$. For this IV not to be perfectly correlated with z_1 we need $\pi_2 \neq 0$ or $\pi_3 \neq 0$. We can test $H_0 : \pi_2 = \pi_3 = 0$ against the desired hypothesis using an F statistic.

Essentially, we are breaking y_2 into two pieces, y_2^* - uncorrelated with u_1 and ν_2 - possibly correlated with u_1 - which is why y_2 is possibly endogenous. We can estimate y_2^* via OLS to get $\hat{y}_{i2}$. At this point we should verify z_2 and z_3 are jointly significant. If not IV estimation is useless. Now we can use $\hat{y}_2$ as an IV for y_2 and estimate the three variables β_0, β_1, and β_2 via a set of three linear equations. With multiple instruments, the IV estimator using $\hat{y}_{i2}$ as the instrumental variable is called **two stage least squares** (2SLS) estimator. This is because, simple algebra shows that the above estimates are identical to the OLS estimates from the regression of y_1 on $\hat{y}_2$ and z_1. Hence, we first estimate $\hat{y}_2$ using the reduced form and then regress y_1 on $\hat{y}_2$ and z_1. The first step can be seen as purging y_2 of its correlation with u_1 before doing the OLS regression. Standard errors and test statistics from second equation are not valid directly and need to be decomposed before use. Adding more exogenous variables changes very little.

Example 15.4. We estimate return to education on wages with mother's and father's education as instrumental variables. We first estimate the reduced equation and test $H_0 : \pi_3 = 0, \pi_4 = 0$ using an F test, which clearly rejects the null, suggesting that *educ* is partially correlated with parent's eduction.

```
reduced = smf.ols(formula='educ~exper+expersq+motheduc+fatheduc', data=df).
            fit(cov_type='HC0')
reduced.f_test(['motheduc=0', 'fatheduc=0'])
<F test: F=array([[107.44292177]]), p=9.598906539192378e-42, df_denom=748, df_num=2>
res_ols = smf.ols(formula='lwage~educ+exper+expersq', data=df).fit()
res_iv = iv.IV2SLS.from_formula(formula='lwage~1+exper+expersq+'+
        '[educ~motheduc+fatheduc]', data=df).fit()

            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
--------------------------------------------------------------------------------
Intercept      0.0481     0.4278     0.1124     0.9105     -0.7903      0.8865
exper          0.0442     0.0155     2.8546     0.0043      0.0138      0.0745
expersq       -0.0009     0.0004    -2.1001     0.0357     -0.0017  -5.997e-05
educ           0.0614     0.0332     1.8503     0.0643     -0.0036      0.1264
```

Next we use 2SLS to obtain the IV estimates. The estimated return to education is 6.1% compares with and OLS estimate of about 10.8%. Because of its relatively large standard error, the 2SLS estimate is barely statistically significant at the 5% level against a two-sided alternative. □

We now layout the large sample assumptions required for 2SLS estimation for cross-sectional random sampling.

- 2SLS.1 The model can be written linear in parameters as $Y = \beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k + u$. The instrument variables are denoted by z_j, not a part of the linear combination directly.

- 2SLS.2 We have a random sample on y, x_j and z_j.
- 2SLS.3 There are no perfect linear relationships among the instrumental variables and the rank condition (at least one of the coefficients on instrumental variable in the reduced equation is non-zero) for identification holds.
- 2SLS.4 The error term u has zero mean, and each IV is uncorrelated with u.

Any x_j uncorrelated with u acts as its own IV. Under assumptions 2SLS.1 through 2SLS.4, the 2SLS estimator is consistent.

- 2SLS.5 Homoskedasticity holds, i.e., $E(u^2|\boldsymbol{z}) = \sigma^2$.

Under assumptions 2SLS.1 through 2SLS.5, the 2SLS estimator are asymptotically normally distributed. It is also efficient in the class of IV estimators that uses linear combinations of the exogenous variables as instruments. If the homoskedasticity assumption does not hold, 2SLS estimators are still asymptotically normal, but the standard errors need to be adjusted, and the IV estimator is no longer asymptotically efficient IV estimator.

Multicollinearity can be a more serious issue with 2SLS. The asymptotic variation of the 2SLS estimator of β_1 can be approximated as $\frac{\sigma^2}{\widehat{SST}_2(1-\hat{R}_2^2)}$, where $\sigma^2 = Var(u_1)$, $\widehat{SST}_2$ is the total variation in $\hat{y}_2$ and $\hat{R}_2^2$ is the R-squared from a regression of $\hat{y}_2$ on all other exogenous variables appearing in the structural equation. First, by construction $haty_2$ has less variation than y_2 and second, the correlation between $\hat{y}_2$ and the exogenous variables is often much higher than the correlation between y_2 and these variables. But, as with OLS, a large sample size can help offset a large $\hat{R}_2^2$. As an example, when $educ$ is regressed on the exogenous variables we get an R-squared of 0.477 with moderate degree of multicollinearity. The OLS standard error for $\hat{beta}_e duc$ is only 0.075. When we first obtain $\widehat{educ}$ and regress these on the exogenous variables we get an R-squared of 0.995, which indicates a very high degree of multicollinearlity.

```
reduced.rsquared
>>> 0.4771162094852581
res_ols.params.educ
>>> 0.07469325559311754
s2 = res_iv.resids.var()
sst2 = reduced.mse_total * reduced.nobs
df['hat_y2'] = df.educ-reduced.resid
r2 = smf.ols(formula='hat_y2~exper+expersq+black+smsa+south+smsa66+'
                     'reg662+reg663+reg664+reg665+reg666+reg667+reg668+reg669',
                     data=df).fit().rsquared
stderr = np.sqrt(s2 / sst2 /(1-r2))
print(r2, stderr)
>>> 0.9951478449115145, 0.03786434773912782
```

A small correlation between the instrument and error can lead to very large inconsistencies if the instrument z also has little correlation with the explanatory variable x. Even with

large sample sizes the 2SLS estimator can be biased and a distribution very different from normal if we have weak instruments, even if they are exogenous. As a thumb rule, one need to adjust a statistical rejection of the null hypothesis in the first stage regression at the usual significance levels - with first stage t statistic with absolute value more than $\sqrt{10} = 3.2$ for one IV, and first-stage F statistic for exclusion of the instrumental variables for y_2 if $F > 10$ for multiple IV. As an illustration the F value f 107.44 in the previous example is well past the required value of 10.

2SLS can also be used in models with more than one endogenous explanatory variables like $y_1 = \beta_0 + \beta_1 y_2 + \beta_2 y_3 + \beta_3 z_1 + \beta_4 z_2 + \beta_5 z_3 + u_1$, where $E(u_1) = 0$ and u_1 is uncorrelated with x_1, z_2, z_3 but may be correlated to y_2, y_3. We need at least two exogenous variables as IV. We can use F test on the reduced forms of y_2 and y_3, but to be sufficient we need **rank condition** to be satisfied - we need at least as many excluded exogenous variables are there are included endogenous explanatory variables in the structural equation. The R-squared from 2SLS estimation can be negative and hence can't be used for usual F testing. There are other methods to do it generally available in econometric packages.

15.4 IV solutions to errors-in-variables problems

IV can also be used to deal with the measurement error problem. For $y = \beta_0 + \beta_1 x_1^* + \beta_2 x_2 + u$, where y and x_2 are observed but x_1^* is not and instead $x_1 = x_1^* + e_1$ is available. If the classical errors-in-variables (CEV) assumptions hold, OLS, where we use x_1 in place of x_1^* will be biased and inconsistent. This is because $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + (u - \beta_1 e_1)$ have x_1 correlated with the error term breaking the exogeneity assumption. The bias in the OLS estimator of β_1 is toward zero.

We assume that u is uncorrelated with x_1^*, x_1, x_2; CEV case assumption is e_1 is uncorrelated with x_1^* and x_2. Thus, x_2 is exogenous, but x_1 is correlated with e_1. We need an IV for x_1, which is correlated with x_1, uncorrelated with u and uncorrelated with the measurement error e_1. One possibility is to obtain a second measurement on x_1^*, say z_1. Because x_1^* affects y it is only natural to assume that z_1 is uncorrelated with u. And the measurement error in $z_1 = x_1^* + a_1$ can be assumed to be independent of e_1 measurement error. Certainly, x_1 and z_1 are correlated though their dependence on x_1^*, so we can use z_1 as an IV for x_1. An alternative is to use other exogenous variables as IVs.

IV methods can also be adopted when using things like test scores to control for unobserved characteristics. For the wage equation $\log(wage) = \beta_0 + \beta_1\ educ + \beta_2\ exper + abil + u$, we can get two different test scores as an indicator of ability: $test_1 = \gamma_1\ abil + e_1$ and $test_2 = \delta_1\ abil + e_2$. We can assume that $test_1$ and $test_2$ are uncorrelated with u. We can substitute $test_1$ in the original model to get $\log(wage) = \beta_0 + \beta_1\ educ + \beta_2 + \alpha_1 test_1 + (u - \alpha_1 e_1)$. If we assume e_1 is uncorrelated with all the explanatory variables including $abil$, then e_1 and $test_1$ must be correlated. OLS estimation will produce inconsistent estimators of the β_j and α_1. $test_1$ does not satisfy the proxy variable assumption, under the assumptions we have made. If we assume e_2 is also uncorrelated with all the explanatory variables and that e_1 and e_2 are uncorrelated, then e_1 is uncorrelated with the second test score $test_2$. Therefore,

$test_2$ can be used as an IV for $test_1$.

Example 15.5. We use the wage data and implement the procedure where IQ plays the role of the first test score and KWW is the second test score. The explanatory variables are $educ, exper, tenure, married, south, urban, black$. We add IQ and use KWW as its instrument.

```
df = woo.data('wage2')
reduced = smf.ols(formula='IQ~educ+exper+tenure+married+south+urban+black+KWW',
                  data=df).fit(cov_type='HC0')
#                 coef    std err          z      P>|z|      [0.025      0.975]
# ------------------------------------------------------------------------------
# KWW            0.3853      0.060      6.388      0.000       0.267       0.504
res_ols = smf.ols(formula='lwage~educ+exper+tenure+married+south+urban+black++IQ',
                  data=df).fit()
res_iv = iv.IV2SLS.from_formula(formula='lwage~1+educ+exper+tenure+married'+
                  '+south+urban+black+[IQ~KWW]', data=df).fit()
#                 coef    std err          t      P>|t|      [0.025      0.975]
# ------------------------------------------------------------------------------
# IQ OLS         0.0036      0.001      3.589      0.000       0.002       0.006
# IQ IV          0.0130     0.0055     2.3835     0.0171      0.0023      0.0238

# educ OLS       0.0544      0.007      7.853      0.000       0.041       0.068
# educ IV        0.0250     0.0187     1.3410     0.1799     -0.0116      0.0616
```

We look at the reduced equation and see that KWW could be a decent instrument for IQ. We then run the OLS and IV estimations for the models and see the standard error of the $\hat{\beta}_{IQ}$ increase. More puzzling is the drop in the effect of $educ$ both in value and statistical significance. It suggests that one our assumptions fails; perhaps e_1 and e_2 are correlated. □

15.5 Testing for endogeneity and overidentifying restrictions

We need a test for endogeneity of an explanatory variable that shows whether 2SLS is even necessary. Suppose we have a single suspected endogenous variable, $y_1 = \beta_0 + \beta_1 y_2 + \beta_2 z_1 + \beta_3 z_2 + u_1$, where z_1, z_2 are exogenous. We have two additional exogenous variables, z_3, z_4 which do not appear here. If y_2 is uncorrelated with u_1 we should prefer direct OLS estimation. Hausman test suggests to compare the OLS and 2SLS estimators and significant difference implies that y_2 must be endogenous. The reduced equation is given by $y_2 = \pi_0+\pi_1 z_1+\pi_2 z_2+\pi_3 z_3+\pi_4 z_4+v_2$. Now, since each z_j is uncorrelated with u_1, y_2 is uncorrelated with u_1 iff v_2 is uncorrelated with u_1. Since v_2 is unavailable, we use the residuals $\hat{v}_2$ from the reduced form. Therefore, we estimate $y_1 = \beta_0+\beta_1 y_2+\beta_2 z_1+\beta_3 z_2+\delta_1 \hat{v}_2+\varepsilon$ by OLS and test $H_0 : \delta_1 = 0$ using a t-statistic, preferably heteroskedasticity robust. If we reject H_0 at a small significance level we conclude that y_2 is endogenous because v_2 and u_1 are correlated.

Example 15.6. We can test for the endogeneity of $educ$ by obtaining the residuals $\hat{v}_2$ from estimating the reduced form.

```
df = woo.data('mroz')
reduced = smf.ols(formula='educ~exper+expersq+motheduc+fatheduc', data=df).
                    fit(cov_type='HC0')
df['resid'] = reduced.resid
res_ols = smf.ols(formula='lwage~educ+exper+expersq+resid', data=df).
                    fit(cov_type='HC0')
res_ols.f_test(['resid=0'])
<F test: F=array([[2.67337396]]), p=0.10278339444663086, df_denom=423, df_num=1>
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0114      0.382     -0.030      0.976      -0.760       0.737
educ           0.0639      0.031      2.092      0.036       0.004       0.124
exper          0.0463      0.015      3.059      0.002       0.017       0.076
expersq       -0.0009      0.000     -2.276      0.023      -0.002      -0.000
resid          0.0559      0.034      1.635      0.102      -0.011       0.123
==============================================================================
```

When we do this, the coefficient on $\hat{v}_2$ is $\hat{\delta}_1 = 0.056$ and $t = 1.64$. This is moderate evidence of positive correlation between u_1 and v_2. It is a good idea to report both estimates because the 2SLS estimate of the return to education (6.1%) is well below the OLS estimate (10.8%). □

We can also test for endogeneity of multiple explanatory variables. For each suspected endogenous variable, we obtain the reduced from residuals. The, we test for join significance of these residuals in the structural equation, using and F test. Joint significance indicates that at least one suspected explanatory variable is endogenous.

In the context of the simple IV estimator, we noted that the exogeneity requirement cannot be tested. However, if we have more instruments than we need, we can effectively test whether some of them are uncorrelated with the structural error. The procedure of comparing different IV estimates of the same parameter is an example of testing **overidentifying restrictions**. We first estimate the structural equation by 2SLS an obtain $\hat{u}_1$. We regression $\hat{u}_1$ on all exogenous variables and obtain R_1^2. Under the null hypothesis all IVs are uncorrelated with u_1 and $nR_1^2 \overset{a}{\sim} \chi_q^2$, where q is the number of instrumental variables from outside the model minus the total number of endogenous explanatory variables. If nR_1^2 exceeds say 5% critical value in the χ_q^2 distribution, we reject H_0 and conclude that at least some of the IVs are not exogenous. If we have just enough instruments, the model is said to be just identified, and the R-squared will be identically zero.

When q=1 we can use one IV at at time and identify the two IV estimator coefficients $\check{\beta}_1$ and $\tilde{\beta}_1$ and test the null hypothesis $H_0 : \check{\beta}_1 - \tilde{\beta}_1 = 0$, failing which we conclude that either z_3, z_4, or both fail the exogeneity requirement. Unfortunately, we know which is the case. This test is asymptotically same as the previous test we described for overidentifying restrictions.

Example 15.7. When we use *montheduc* and $fatheduc$ as IVs for *edu*, we have a single overidentifying restriction. Regression 2SLS residuals $\hat{u}_1$ on $exper, exper^2, motheduc, fatheduc$,

produces $R_1^2 = 0.0009$. Therefore we have a very small value in χ_1^2 distribution with a p-value of 0.539 and hence parent's education variables pass the overidentification test.

```
df = woo.data('mroz')
res_iv = iv.IV2SLS.from_formula(formula='lwage~1+exper+expersq+[educ~motheduc+fatheduc]',
                        data=df).fit()
df['resid'] = res_iv.resids
res_ht = smf.ols(formula='resid~exper+expersq+motheduc+fatheduc',
                        data=df).fit()
r2 = res_ht.rsquared
n = res_ht.nobs
teststat = n * r2
q = 1
pval = 1 - stats.chi2.cdf(teststat, q)
print(r2, n, teststat, 1, pval)
>>> 0.0008833442569248229 428.0 0.3780713419638242 1 0.5386372330714875
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
Intercept      0.0481     0.4278     0.1124     0.9105     -0.7903      0.8865
exper          0.0442     0.0155     2.8546     0.0043      0.0138      0.0745
expersq       -0.0009     0.0004    -2.1001     0.0357     -0.0017  -5.997e-05
educ           0.0614     0.0332     1.8503     0.0643     -0.0036      0.1264
```

When we add husband's education to the the IV list, we get two overidentifying restrictions and the three variables pass the overidentification test as well.It seems reasonable to add *huseduc* to the IV list, as it resused the standard error of the 2SLS estimate: the 2SLS estimate on *educ* using all three instruments is 0.08 (0.022) so this makes *educ* much more significant than when *huseduc* is not used as an IV with coefficient 0.061(0.031).

```
df = woo.data('mroz')
res_iv = iv.IV2SLS.from_formula(formula='lwage~1+exper+expersq+[educ~motheduc+
                        fatheduc+huseduc]', data=df).fit()
df['resid'] = res_iv.resids
res_ht = smf.ols(formula='resid~exper+expersq+motheduc+fatheduc+huseduc',
                        data=df).fit()
r2 = res_ht.rsquared
n = res_ht.nobs
teststat = n * r2
q = 1
pval = 1 - stats.chi2.cdf(teststat, q)
print(r2, n, teststat, 1, pval)
0.0026052406571422937 428.0 1.1150430012569017 1 0.29098833327344065
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
Intercept     -0.1869     0.2999    -0.6232     0.5332     -0.7746      0.4008
exper          0.0431     0.0152     2.8289     0.0047      0.0132      0.0730
expersq       -0.0009     0.0004    -2.0558     0.0398     -0.0017  -4.023e-05
educ           0.0804     0.0216     3.7216     0.0002      0.0381      0.1227
```

□

Adding new instruments requires that they in fact are exogenous - otherwise, 2SLS will not be consistent. Adding too many instruments can cause severe biases in 2SLS as well.

15.6 2SLS with heteroskedasticity

Heteroskedasticity in the context of 2SLS raises essentially the same issues as with OLS and we can obtain standard errors and test statistics that are asymptotically robust to heteroskedasticity of arbitrary and unknown form. If $\hat{r}_{ij}$ are obtained as the residuals form regression $\hat{x}_{ij}$ on the other $\hat{x}_{ih}$ where these are fitted values from the first stage regression (for endogenous explanatory variables), we can apply the usual LM statistic test.

We can also test for heteroskedasticity using Breusch-Pagan test, White test of the conserved df White test. For BP, let $\hat{u}$ denote the 2SLS residuals and let $x_1, \ldots, z_m$ denote all the exogenous variables including those used as IVs for the endogenous explanatory variables. Then under reasonable assumptions, an asymptotically valid statistic is the usual F statistic for join significance in a regression of $\hat{u}^2$ on $z_1, \ldots, z_m$. The null hypothesis of homoskedasticity is rejected if the z_j are jointly significant. Applying this on the previous example gives us a p-value of 0.029 for the F statistic. This is evidence of heteroskedasticity at the 5% level. We might want to compute heteroskedasticity-robust standard errors to account for this.

```
from statsmodels.stats.diagnostic import het_breuschpagan
import patsy as pt
y, X = pt.dmatrices('resid~exper+expersq+motheduc+fatheduc+huseduc',
                    data=df, return_type='dataframe')
bp = het_breuschpagan(y, X)
>>> (12.435255667725004, 0.029286768099771654, 2.5255645303654766, 0.02869518569783332)
```

If we know how the error variance depends on the exogenous variables, we can use a weighted 2SLS procedure. After estimating a model for $Var(u|z_1, z_2, \ldots, z_m)$, we divide the dependent variable, the explanatory variables, and all the instrumental variables for observations i, along with the constant, by $\sqrt{\hat{h}_i}$, the estimated standard deviation. Then we apply 2SLS on the transformed equation using the transformed instruments.

15.7 2SLS for time series

For time series applications we have the structural form as $y_t = \beta_0 + \beta_1 x_{t1} + \ldots + \beta_k x_{tk} + u_k$ where one more more explanatory variables x_{tj} might be correlated with u_t. We denote the set of exogenous variables by $x_{t1}, \ldots, z_{tm}$ and assume $E(u_t) = 0$ and $Cov(z_{tj}, u_t) = 0$ for $j = 1, \ldots, m$. Any exogenous explanatory variable is also a z_{tj} and for identification we need $m \geq k$. We must be careful to include trends if we have trending dependent or explanatory variables. The same is true of season dummy variables, if monthly or quarterly data are used. Series that have strong persistence, or unit roots, can be differenced, and it applies to instruments as well. Under standard assumptions the asymptotic properties

of OLS, 2SLS using time series data is consistent and asymptotically normally distributed. The homoskedasticity assumption is stated as $E(u_t^2|z_{t1},\ldots,z_{tm}) = \sigma^2$. As is with OLS, we must assume all series, including IVs, are weakly dependent for law of large numbers and central limit theorem to work. Additionally we require

- 2SLS.6 No serial correlation i.e. $E(u_t u_s|\boldsymbol{z}_s, \boldsymbol{z}_s) = 0$ for all $t \neq s$.

As in the case of OLS, the no serial correlation assumption can often be violated with time series data. It is easy to test for AR(1) serial correlation. We estimate the structural equation by 2SLS and obtain $\hat{u}_t$. We then estimate $y_t = \beta_0 + \beta_1 x_{t1} + \ldots + \beta_k x_{tk} + \rho\hat{u}_{t-1} + \varepsilon_t$, for $t = 2,\ldots,n$ by 2SLS, using the same instruments in addition to $\hat{u}_{t-1}$. We test $H_0 : \rho = 0$ using the t statistic on $\hat{\rho}$. This only has asymptotic justification. A heteroskedasticity-robust version should be used. Further, lagged residuals can be added to the equation to test for higher forms of serial correlation using a joint F test.

If serial correlation is detected, we can use standard errors robust to fairly general forms of serial correlation and heteroskedasticity (HAC). Alternatively we can use the AR(1) model and correct for serial correlation using FGLS estimator (either OC or PW). The procedure is similar and places additional restrictions on the instrumental variables. The quasi-differenced equation is $\tilde{y}_t = \beta_0(1-\rho) + \beta_1\tilde{x}_{t1} + \ldots + \beta_k\tilde{x}_{tk} + e_t$, $t \geq 2$, where $\tilde{x}_{tj} = x_{tj} - \rho x_{t-1,j}$. We can use the quasi-differenced instrument variables $\tilde{z}_{tj} = z_{tj} - \rho z_{t-1,j}$ if the original error u_t is uncorrelated with the instruments at time $t, t-1$, and $t+1$, i.e. the instrument variable should be strictly exogenous. This rules out lagged dependent variables as IVs, for example. It also eliminates cases where future movements in the IVs react to current and past changes in the error u_t.

For feasible GLS, we estimate $\hat{u}_t$ using 2SLS. Then obtain $\hat{\rho}$ from regression of $\hat{u}_t$ on $\hat{u}_{t-1}$, $t = 2,\ldots,n$ and construct the quasi-differenced variables $\tilde{y}_t = y_t - \hat{\rho}y_{t-1}$, $\tilde{x}_{tj} = x_{tj} - \hat{\rho}x_{t-1,j}$, and $\tilde{z}_t = y_{tj} - \hat{\rho}z_{t-1,j}$. We use 2SLS to estimate the quasi-differenced equation using $\tilde{z}_{tj}$ as instruments. If the 2SLS assumptions are satisfied, the usual 2SLS test statistics are asymptotically valid. We can also use the first time period as in Prais-Winsten estimation by simply multiplying all the first-period values by $\sqrt{1-\hat{\rho}^2}$.

15.8 2SLS for pooled cross sections and panel data

Applying IV methods to independently pooled cross sections raises no new difficulties. As with OLS, we should often include time period dummy variables to allow for aggregate time effect, and these are exogenous because of passage of time is exogenous - so it acts as its own instrument.

Example 15.8. We used pooled cross section to estimate the effect of education on women's fertility, controlling for various factors. If *educ* is endogenous in the equation, we can use mother's and father's education level as instruments and do a 2SLS estimation. The OLS estimate for *educ* is -0.128 (se=0.018) and the 2SLS estimate is -0.153 (se=0.04). The 2SLS estimate shows a somewhat larger effect of education on fertility, but the 2SLS standard errors are over twice as large as the OLS standard errrors.

```
df = woo.data('fertil1')
res_ols = smf.ols('kids~educ+age+I(age**2)+black+east+northcen+west+farm'+
                  '+othrural+town+smcity+y74+y76+y78+y80+y82+y84', data=df).fit()
# using IV
reduced_ols = smf.ols('educ~meduc+feduc+age+I(age**2)+black+east+northcen+west+farm'+
                  '+othrural+town+smcity+y74+y76+y78+y80+y82+y84', data=df).fit()
#                    coef    std err          t      P>|t|      [0.025      0.975]
# ------------------------------------------------------------------------------
# meduc           0.1723      0.022      7.763      0.000       0.129       0.216
# feduc           0.2074      0.025      8.147      0.000       0.157       0.257

# 2SLS
res_iv = iv.IV2SLS.from_formula(formula='kids~1+age+I(age**2)+black+east+northcen'+
                '+west+farm+othrural+town+smcity+y74+y76+y78+y80+y82+y84'+
                '+[educ~meduc+feduc]', data=df).fit()

#                    coef    std err          t      P>|t|      [0.025      0.975]
# ------------------------------------------------------------------------------
# educ OLS       -0.1284      0.018     -6.999      0.000      -0.164      -0.092
# educ IV        -0.1527     0.0402     -3.7949     0.0001     -0.2316     -0.0739
```

We can perform a test for endogeneity of *educ* to see if they are statistically different. When the reduced form residual $\hat{v}_2$ is included with the other regressors including *educ* the t stats of 0.702 shows that they are not different at any reasonable level.

```
df['resid'] = reduced_ols.resid
res_test = smf.ols(formula='kids~educ+age+I(age**2)+black+east+northcen+west+farm'+
                  '+othrural+town+smcity+y74+y76+y78+y80+y82+y84+resid', data=df).fit()
res_test.f_test(['resid=0'])
# <F test: F=array([[0.49262419]]), p=0.4829061767889229, df_denom=1.11e+03, df_num=1>
#                    coef    std err          t      P>|t|      [0.025      0.975]
# ------------------------------------------------------------------------------
# resid           0.0311      0.044      0.702      0.483      -0.056       0.118
```

□

Instrument variables estimation can be combined with the panel data methods, particularly first differencing, to estimate parameters consistently in the presence of unobserved effects and endogeneity in one more more time-varying explanatory variables.

Example 15.9. We model the productivity $\log(scrap_{it})$ based on $hrsemp_{it}$, hours of job training per employee using the model $\log(scrap)_{it} = \beta_0 + \delta_0 d88_t + \beta_1 hrsemp_{it} + a_i + u_{it}$, $t = 1, 2$, where we allow different year intercepts and a constant, unobserved firm effect a_i. We might be concerned that $hrsemp_{it}$ might be correlated to with a_i (which might contain unmeasured worker ability). We difference to remove a_i to get $\Delta \log(scrap)_i = \delta_0 + \beta_1 \Delta hrsemp_i + \Delta u_i$. The OLS estimation of FD model gives a coefficient of -0.0076 (sd = 0.005).

```
df = woo.data('jtrain')
df = df[df.year==1988][['clscrap', 'chrsemp', 'cgrant']].dropna()
reduced_ols = smf.ols(formula='chrsemp~cgrant', data=df).fit()
#                 coef    std err          t      P>|t|      [0.025      0.975]
# ------------------------------------------------------------------------------
# Intercept      1.5806      3.185      0.496      0.622      -4.844       8.005
# cgrant        24.4369      5.183      4.715      0.000      13.985      34.889

res_fd = smf.ols(formula='clscrap~chrsemp', data=df).fit()
res_iv = iv.IV2SLS.from_formula(formula='clscrap~1+[chrsemp~cgrant]', data=df).fit()
#                 coef    std err          t      P>|t|      [0.025      0.975]
# ------------------------------------------------------------------------------
# chrsemp FD    -0.0076      0.005     -1.685      0.099      -0.017       0.001
# chrsemp IV    -0.0142     0.0082    -1.7156     0.0862     -0.0303      0.0020
```

What if $\Delta hrsemp_i$ is correlated with u_i, e.g. the firm might hire more skilled workers while reduce the level of job training at same time. In this case, we need a instrumental variable for $\Delta hrsemp_i$. We can exploit the fact that some firms received job training grants in 1988. If we assume that grant designation is uncorrelated with Δu_i - which is reasonable because the grants were given at the beginning of 1988 - then $\Delta grant_i$ is a valid IV, provided it is correlated to $\Delta hrsemp$. The reduced equation confirms this and the IV 2SLS estimate gives a coefficient of -0.0142 (se = 0.0082). □

When $T \geq 3$, the differenced equation may contain serial correlation. The same test and correction for AR(1) serial correlation can be used, where all regressions are pooled across i as well as t. Prais-Winsten transformation should allow us to use the initial time period. Unobserved effects models containing lagged dependent variables also require IV methods for consistent estimation. The reason is that, after differencing, $\delta y_{i,t-1}$ is correlated with Δu_{it} because $y_{i,t-1}$ and $u_{i,t-1}$ are correlated. We can use two or more lags of y as IVs for $\delta y_{i,t-1}$.

Instrumental variables after differencing can be used on matched pairs samples as well. We difference the wage equation across twins to eliminate unobserved ability to get $\log(wage_2) - \log(wage_1) = \delta_0 + \beta_1(educ_{2,2} - educ_{1,1}) + \Delta u$, where $educ_{i,j}$ is the year of schooling for the ith twin as reported by the jth twin. To account for possible measurement error in the self-reported schooling measures, one can use $(educ_{2,1} - educ_{1,2})$ as in IV for $(educ_{2,2} - educ_{1,1})$.

16 Simultaneous Equations Models

Instrumental variables can solve two kinds of endogeneity problems: omitted variables and measurement error. Another important form of endogeneity of explanatory variables is **simultaneity**. This arises when one ore more of the explanatory variables is jointly determined with the dependent variable, typically through an equilibrium mechanism. The leading method for estimating Simultaneous equation models (SEM) is the method of instrumental variables.

16.1 Nature of simultaneous equations model

Since SEM should have ceteris paribus, causal interpretation, even though the outcome is in equilibrium, we are required to use counterfactual reasoning in constructing equations. A classical example of SEM is supply and demand. A simple labor supply function, called the **structural equation** in context of SEM, is $h_s = \alpha_1 w + \beta_1 z_1 + u_1$, where h_s is the annual labor hours supplied by workers in agriculture, w is the average hourly wage offered to such workers, z_1 is some observed variable affecting labor supply like average manufacturing wage in the county and u_1 is the error term containing other factors that affect labor supply. This equation differs from those we studied previously in a very subtle way. Although the equation is supposed to hold for all possible values of wage, we only have access to the equilibrium value of wages and hours worked at which the market cleared!

To describe how equilibrium wages and hours are determined, we need to bring in the demand for labor $h_d = \alpha_2 w + \beta_2 z_2 + u_2$, where h_d is the hours demanded, z_2 is say the observable variable agricultural land area and u_2 contains all other unobservable factors. This too is a structural equation. The two equations describe entirely different relationships. Labor supply is a behavioral equation of workers, and labor demand is a behavioral relationship for farmers. In equilibrium, for each county i, observed hours h_i and observed wage w_i are determined by the condition $h_{is} = h_{id}$. This equilibrium hours for each county i, is denoted by h_i. We, thus, get the **simultaneous equations model (SEM)** as $h_i = \alpha_1 w_i + \beta_1 z_{i1} + u_{i1}$ and $h_i = \alpha_2 w_i + \beta_2 z_{i2} + u_{i2}$.

We need to assume $\alpha_1 \neq \alpha_2$ and given z_{i1}, z_{i2} (exogenous variables), u_{i1}, u_{i2} (structural error) determine h_i and w_i (endogenous variables). The key assumption is that z_{i1} and z_{i2} are uncorrelated to u_{i1} and u_{i2} respectively. If z_1 and z_2 are the same, there is no way to tell which equation is supply or demand function - called the identification problem - and estimation is not possible. Finally, just because two variables are determined simultaneously does not mean that a simultaneous equation model is suitable. For an SEM to make sense, each equation in the SEM should have a ceteris paribus interpretation in isolation from the other equation. Generally the two structural equations will have two different drivers for SEM to make sense.

16.2 Simultaneity bias in OLS

When an explanatory variable is determined simultaneously with the dependent variable it is generally correlated with the error term, leading to the usual bias and inconsistency in OLS. Consider the two equation structural model

$$y_1 = \alpha_1 y_2 + \beta_1 z_1 + u_1$$
$$y_2 = \alpha_2 y_1 + \beta_2 z_2 + u_2$$

which focus on estimating the first equation. z_1, z_2 are exogenous, so that each is uncorrelated with u_1 and u_2. We can substitute one equation into another to get the **reduced form equation** $y_2 = \pi_{21} z_1 + \pi_{22} z_2 + v_2$, where $\pi_{21} = \alpha_2 \beta_1 / (1 - \alpha_1 \alpha_2)$, $\pi_{22} = \beta_2 / (1 - \alpha_1 \alpha_2)$, and $v_2 = (\alpha_2 u_1 + u_2)/(1 - \alpha_1 \alpha_2)$ when $\alpha_1 \alpha_2 \neq 1$. π_{21}, π_{22} are called the **reduced form**

parameters and are a nonlinear function of the structural parameters. The **reduced from error** v_2 is a linear function of u_1 and u_2 and is also uncorrelated with z_1, z_2. Therefore, we can consistently estimate π_{21}, π_{22} by OLS. A reduced form also exists for y_1 with similar properties.

If we use OLS to estimate α_1, β_1 in the first equation, we need to consider if y_2 and u_1 are uncorrelated or not. From the reduced form we see that y_2 and u_1 are correlated only if v_2 and u_1 are correlated. Since v_2 is a linear function of u_1 and u_2, it is in general correlated with u_1. Only when $\alpha_2 = 0$ and u_1 and u_2 are uncorrelated, i.e. the two equations are unlinked, we have y_2 and u_1 uncorrelated. When y_2 is correlated with u_1 because of simultaneity, we say that OLS suffers from **simultaneity bias**. If we drop z_1 from the equation, and assume u_1 and u_2 are uncorrelated, then the covariance between y_2 and u_1 is $\frac{\alpha_2}{1-\alpha_1\alpha_2}\sigma_1^2$, hence the bias is the same sign as $\frac{\alpha_2}{1-\alpha_1\alpha_2}$. In general cases, it is complicated to determine the bias direction.

16.3 Identifying and estimating a structural equation

If we have some instrumental variables we can still identify the parameters in an SEM equation, just as with the omitted variables or measurement error. A general two-equation model is written as

$$y_1 = \beta_{10} + \alpha_1 y_2 + \boldsymbol{z}_1^T \boldsymbol{\beta}_1 + u_1$$
$$y_2 = \beta_{20} + \alpha_2 y_1 + \boldsymbol{z}_2^T \boldsymbol{\beta}_2 + u_2$$

where y_1 and y_2 are the endogenous variables and u_1 and u_2 are the structural error terms. The variable $\boldsymbol{z}_1 = (z_{11}, z_{12}, \ldots, z_{1k_1})$ denote a set of k_1 exogenous variables appearing the first equation. Similarly $\boldsymbol{z}_2 = (z_{21}, z_{22}, \ldots, z_{2k_2})$ denote a set of k_2 exogenous variables in the second equation. $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ will overlap in many cases. Imposing **exclusion restriction** means $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ generally contain different exogenous variables. This allows us to distinguish the two structural equations. We can solve these two equations for y_1 and y_2 if $\alpha_1\alpha_2 \neq 1$, under which the reduced form exists for y_1 and y_2. To address the **identification issue** of a single equation we need the **rank condition**: the first equation is identifiable iff the second equation contains at least one exogenous variable (with a nonzero coefficient) that is excluded from the first equation. This is the necessary and sufficient condition for the first equation to be identified. We can test these using t or an F test. Identification of the second equation is, naturally, just the mirror image of the statement for the first equation. Once identified, the equations can be estimated using 2SLS with the instrumental variables consisting of the exogenous variables appearing in either equation.

Example 16.1. Consider the labor supply equation for married women already in the workforce. The two structural equations are

$$hours = \alpha_1 \log(wage) + \beta_{10} + \beta_{11}\ educ + \beta_{12}\ age + \beta_{13}\ kidslt6 + \beta_{14}\ nwifeinc + u_1$$
$$log(wage) = \alpha_2\ hours + \beta_{20} + \beta_{21}\ educ + \beta_{22} exper + \beta_{23}\ exper^2 + u_2$$

The variable *age* is the women's age, in years, *kidslt*6 is the number of children less than six years old, $nwifeinc$ is the woman's non wage income which includes husband's earnings,

and *educ* and *exper* are years of education and prior experience, respectively. All variables except *hours* and log(*wage*) are assumed to be exogenous. The first equation is the supply function and satisfies the order condition because two exogenous variables, *exper* and $exper^2$ are omitted from the labor supply equation. The rank condition for identifying the first equation is that at least one of *exper* and $exper^2$ has a nonzero coefficient in the demand equation. We can state the rank condition for identification equivalently in terms of the reduced form for log(*wage*) and estimate it as follows:

```
df = woo.data('mroz')
reduced_wage = smf.ols(formula='lwage~educ+age+kidslt6+nwifeinc+exper+expersq',
                       data=df).fit()
reduced_wage.f_test(['exper=0', 'expersq=0'])
>>> <F test: F=array([[9.32933313]]), p=0.0001085, df_denom=421, df_num=2>
```

The null hypothesis is resoundingly rejected given very low p-value. The wage offer demand function is identified if at least one of $age, kidslt6$, and $nwifeinc$ has a nonzero coefficient in the supply equation. This is identical to assuming that the reduced form for *hours* depends on at least on of $age, kidslt6$ or $nwifeinc$, which is the case as we see below.

```
df = woo.data('mroz')
reduced_hours = smf.ols(formula='hours~educ+age+kidslt6+nwifeinc+exper+expersq',
                        data=df).fit()
reduced_hours.f_test(['age=0', 'kidslt6=0', 'nwifeinc=0'])
>>> <F test: F=array([[26.66873626]]), p=2.157443e-16, df_denom=746, df_num=3>
```

We can now use 2SLS to estimate the supply equation as

```
supply_2sls = iv.IV2SLS.from_formula(formula='hours~1+educ+age+kidslt6+'+
                    'nwifeinc+[lwage~exper+expersq]', data=df).fit()
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
--------------------------------------------------------------------------------
Intercept      2225.7     603.10     3.6904     0.0002      1043.6      3407.7
educ          -183.75     67.787    -2.7107     0.0067     -316.61     -50.890
age           -7.8061     10.487    -0.7443     0.4567     -28.361      12.749
kidslt6       -198.15     208.42    -0.9507     0.3417     -606.66      210.35
nwifeinc      -10.170     5.2875    -1.9233     0.0544     -20.533      0.1937
lwage          1639.6     593.31     2.7634     0.0057      476.69      2802.4
```

This shows an upward sloping supply curve. Holding other factors fixed $\widehat{\Delta hours} \approx 16.4(\%\Delta wage)$. For comparison when the supply equation is 'wrongly' estimated via OLS we get the coefficient on $log(wage)$ as -2.05 with t stats of -0.037. To confirm that $log(wage)$ is in fact endogenous we can test it out by adding the reduced form residual $\hat{v}_2$ to the equation and estimate by OLS, giving a t statistic on $\hat{v}_2$ of -6.608, which is very significant, and so $log(wage)$ appears to be endogenous.

```
df['resid'] = reduced_wage.resid
supply_endo_test = smf.ols(formula='hours~educ+age+kidslt6+nwifeinc+lwage+resid',
                           data=df).fit()
                coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
resid      -1714.3580    259.439     -6.608      0.000   -2224.316   -1204.400
```

The wage offer demand equation can also be estimate by 2SLS. This differs from the previous wage equations in that *hours* is included as an explanatory variable and 2SLS is used to account for endogeneity of *hours*. The coefficient on *hours* is statistically insignificant, which means there is no evidence that the wage offer increases with hours worked. The other coefficients are similar to what we get by dropping *hours* and estimating the equation by *OLS*.

```
demand_2sls = iv.IV2SLS.from_formula(formula='lwage~1+educ+exper+'+
                  'expersq+[hours~age+kidslt6+nwifeinc]', data=df).fit()
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
Intercept     -0.6557     0.4098    -1.6002     0.1095     -1.4589      0.1474
educ           0.1103     0.0148     7.4458     0.0000      0.0813      0.1394
exper          0.0346     0.0185     1.8688     0.0617     -0.0017      0.0709
expersq       -0.0007     0.0004    -1.6547     0.0980     -0.0015      0.0001
hours          0.0001     0.0003     0.4305     0.6668     -0.0004      0.0007

demand_ols = smf.ols(formula='lwage~educ+exper+expersq',data=df).fit()
                coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.5220      0.199     -2.628      0.009      -0.912      -0.132
educ           0.1075      0.014      7.598      0.000       0.080       0.135
exper          0.0416      0.013      3.155      0.002       0.016       0.067
expersq       -0.0008      0.000     -2.063      0.040      -0.002   -3.82e-05
```

□

Example 16.2. To test the hypothesis that more 'open' countries should have lower inflation rates, we explain the average annual inflation rates in terms of the average share of imports in GDP, which is a measure of openness. The two-equation system is given by

$$\begin{aligned} inf &= \beta_{10} + \alpha_1\ open + \beta_{11}\ \log(pcinc) + u_1 \\ open &= \beta_{20} + \alpha_2\ inf + \beta_{21}\ \log(pcinc) + \beta_{22}\ \log(land) + u_2, \end{aligned}$$

where *pinc* is per capita income, and *land* is the land area of the country. The first equation is of interest with the hypothesis that $\alpha_1 < 0$. The second equation reflects the fact that the degree of openness might depend on the average inflation rate, as well as other factors. The first equation is identified provided $_{22} \neq 0$. The second equation, however, is not identified because it contains both exogenous variables. The reduced form equation for *open* is

```
df = woo.data('openness')
reduced_inf = smf.ols(formula='open~lpcinc+lland', data=df).fit()
                coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    117.0845     15.848      7.388      0.000      85.680     148.489
lpcinc         0.5465      1.493      0.366      0.715      -2.412       3.505
lland         -7.5671      0.814     -9.294      0.000      -9.181      -5.954
```

A t statistic of -9.294 on $log(land)$ verifies the assertion that smaller countries are more open. We now estimate the first equation using 2SLS to get

```
lwage_2sls = iv.IV2SLS.from_formula(formula='I(inf)~1+lpcinc+[open~lland]',
                                    data=df).fit()
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
------------------------------------------------------------------------------
Intercept      26.899     10.775     2.4964     0.0125      5.7802      48.018
lpcinc         0.3758     1.3603     0.2763     0.7823     -2.2903      3.0419
open          -0.3375     0.1504    -2.2435     0.0249     -0.6323     -0.0427
```

The coefficient on *open* is statistically significant at about the 1% level against a one-sided alternative ($\alpha_1 < 0$). For every percentage point increase in the import share of GDP, annual inflation is about one-third of a percentage point lower. For comparison the OLS estimate is -0.215 (se=0.095). □

16.4 Systems with more than two equations

Simultaneous equation models with more than two equations need matrix algebra to study the identification condition. But some an equation in a general system has been shown to be identified, it can be estimated by 2SLS. It is generally much easier to see if an equation is not identified, if it has no IVs in the remaining equations. An equation in any SEM satisfies the order condition for identification if the number of excluded exogenous variables from the equation is at least as large as the number of right-hand side endogenous variables. The order condition, however, is not necessarily sufficient, for identification, because it generally depends on the value of the parameters in the other equations which are yet to be estimated. There are subtle ways in which identification can fail in complicated SEMs. To obtain sufficient conditions, we need to extend the rank condition for identification for identification using matrix algebra. The nomenclature of overidentified, just identified and unidentified equation works as usual.

Regardless of the number of equations in an SEM, each identified equation can be estimated by 2SLS with instruments for a particular equation consist of the exogenous variables appearing anywhere in the system. Test of endogeneity, heteroskedasticity, serial correlation, and overidentifying restrictions can be obtained as usual. When two ore more equations are correctly specified, system estimation methods like 3SLS are generally more efficient than 2SLS.

16.5 Simultaneous equations models with time series

We use a simple Keynesian model of aggregate demand, ignoring exports and imports, to show the application of SEMs to time series.

$$\begin{aligned} C_t &= \beta_0 + \beta_1(Y_t - T_t) + \beta_2 r_t + u_{t1} \\ I_t &= \gamma_0 + \gamma_1 r_t + u_{t2} \\ Y_t &\equiv C_t + I_t + G_t \end{aligned}$$

where, the first equation is the aggregate consumption (C_t) function which depends on disposable income ($Y_t - T_t$ = income - tax receipts), the interest rate (r_t), and the unobserved structural error u_{t1}. The second equation is the simple investment function. The third equation is an identity and holds by definition, without error. C_t, I_t, and Y_t are endogenous and T_t, r_t, and G_t are exogenous, so they are uncorrelated with u_{t1} and u_{t2}. To estimate the first equation we can use (T_t, G_t, r_t) as IV using 2SLS. The assumptions we have made above are not generally valid and also these models are completely static, which make them unsuitable.

We often expect adjustment lags in time series models. We could add lagged income to the second equation to get $I_t = \gamma_0 + \gamma_1 r_t + \gamma_2 Y_{t-1} + u_{t2}$. That is, we add a **lagged endogenous variable**, generally called a predetermined variable. We generally assume that the error term is uncorrelated with the current exogenous variables and all past endogenous and exogenous variables. The presence of dynamics in aggregate SEMs is, at least for the purposes of forecasting, a clear improvement over static SEMs.

The validity of the usual OLS or 2SLS inference procedures in time series applications depends on the notion of weak dependence. Aggregate consumption, income, investment, and even interest rate seems to have unit roots in them, apart from exponential trends, turning the estimates biased and inconsistent. Problems with trends and high persistence can be avoided by specifying systems in first differences or growth rates, though it is a different SEM than one specified in levels. If a structural model contains a time trend - which if exogenous - then trend can act as its own IV.

Example 16.3. We test the permanent income hypothesis, PIH by estimating $gc_t = \beta_0 + \beta_1 gy_t + \beta_2 r3_t + u_t$, where $gc_t = \Delta \log(c_t)$ the annual growth in real per capita consumption, gy_t is the growth in real disposable income, and $r3_t = i3_t - inf_t$ is the real interest rate as measured by the return on three month T-bill rates. We assume that none of these time series are trending so we can apply standard asymptotic theory. PIH implies that $E(u_t|I_{t-1}) = 0$, where I_{t-1} is all the information till time $t-1$. However, u_t is not necessarily uncorrelated with gy_t or $r3_t$.

Because u_t is uncorrelated with all the variables dates $t-1$ or earlier, valid instruments for estimating the equation are lagged values of gc, ty, and $r3$. The hypothesis of interest here is $H_) : \beta_1 = \beta_2 = 0$. β_1 will be positive if some fraction of the population consumes current income, rather than permanent income and $\beta_2 > 0$ would mean that there is non-constant real interest rate. We use 2SLS to estimate the equation.

```
df = woo.data('consump')
reduced_gy = smf.ols(formula='gy~gc_1+gy_1+r3_1', data=df).fit()
reduced_r3 = smf.ols(formula='r3~gc_1+gy_1+r3_1', data=df).fit()
print(reduced_gy.f_pvalue, reduced_r3.f_pvalue)
>>> 0.016170854727587457 2.9427201368159195e-07
res_iv = iv.IV2SLS.from_formula(formula='gc~1+[gy+r3~gc_1+gy_1+r3_1]', data=df).fit()
            Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
--------------------------------------------------------------------------------
Intercept      0.0081     0.0034     2.3694     0.0178      0.0014      0.0147
gy             0.5862     0.1371     4.2761     0.0000      0.3175      0.8549
r3            -0.0003     0.0009    -0.2961     0.7671     -0.0021      0.0015
Model F-statistic
H0: All parameters ex. constant are zero
Statistic: 18.3050
P-value: 0.0001
```

Therefore, the pure form of PIH is strongly rejected because the coefficient on gy is economically large and statistically significant. By contrast, the real interest rate coefficient is very small and statistically insignificant. The PIH also implies the errors are serially uncorrelated. We can check the residual $\hat{u}_t$ by regressing against $\hat{u}_{t-1}$ as an additional explanatory variable. A small negative correlation with insignificant t-stats holds the assumptions we used.

```
df['resid'] = res_iv.resids
df['resid_1'] = res_iv.resids.shift(1)
sc_ols = smf.ols(formula='resid~resid_1', data=df).fit()
                coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.0001      0.001      0.098      0.923      -0.002       0.003
resid_1       -0.1050      0.180     -0.584      0.563      -0.471       0.261
```

□

16.6 Simultaneous equations models with panel data

The basic approach to estimating SEMs with panel data involves two steps: (1) eliminate the unobserved effects from the equations of interest using the fixed effects transformation or first differencing and (2) find instrumental variables for the endogenous variables in the transformed equation. An SEM with panel data can be written as

$$y_{it1} = \alpha_1 y_{it2} + \boldsymbol{z}_{it1}\boldsymbol{\beta}_1 + a_{i1} + u_{it1}$$
$$y_{it2} = \alpha_2 y_{it1} + \boldsymbol{z}_{it2}\boldsymbol{\beta}_2 + a_{i2} + u_{it2}$$

where i denotes cross section, t denotes time period, and $\boldsymbol{z}_{it1}\boldsymbol{\beta}_1$ or $\boldsymbol{z}_{it2}\boldsymbol{\beta}_2$ denotes linear functions of a set of exogenous variables in each equation. The idiosyncratic structural errors u_{it1} and u_{it2}, are uncorrelated with the $\boldsymbol{z}$ in both equations and across all time periods;

while y_{it2} is correlated with u_{it1} and y_{it1} is correlated with u_{it2}. If we are interested in the first equation we can not estimate it by OLS, as the composite error $a_{i1} + u_{it1}$ is potentially correlated with all explanatory variables. Suppose we difference over time to remove the unobserved effect, a_{i1} to get $\Delta y_{it1} = \alpha_1 \Delta y_{it2} + \Delta \boldsymbol{z}_{it1} \boldsymbol{\beta}_1 + \Delta u_{it1}$. Now the error term in this equation is uncorrelated with $\Delta \boldsymbol{z}_{it1}$ by assumption. But Δy_{it2} and Δu_{it1} are possibly correlated. Therefore, we need an IV for Δy_{it2}.

We need time-varying elements in $\boldsymbol{z}_{it2}$ that are not also in $\boldsymbol{z}_{it1}$. This is because we need an instrument for Δy_{it2} that is correlated with it. Thus we need a time-varying element in $\Delta \boldsymbol{z}_{it2}$ that are not also in $\Delta \boldsymbol{z}_{it1}$.

Example 16.4. To estimate the causal effect of prison population increase on crime rates we estimate the model $\log(crime_{it}) = \theta_t + \alpha_t \log(prison_{it}) + \boldsymbol{z}_{it1} \boldsymbol{\beta}_1 + a_{i1} + u_{it1}$, where θ_t are the year dummies vector. The exogenous variables consist of log of police per capita, log of income per capita, the unemployment rate, proportions of black and those living in metropolitan areas, and age distribution proportions. Differencing the equation gives $\Delta log(crime_{it}) = \zeta_t + \alpha_t \Delta log(prison_{it}) + \Delta \boldsymbol{z}_{it1} \boldsymbol{\beta}_1 + \Delta u_{it1}$. Simultaneity between crime rates and prison population, or more precisely in the growth rates, makes OLS estimate generally inconsistent. The pooled OLS estimate of α_1 is -0.181 (se=0.048).

```
df = woo.data('prison')
res_pnl = smf.ols(formula='gcriv~C(year)+gpris+gpolpc+gincpc+cunem+cblack+cmetro'+
                '+cag0_14+cag15_17+cag18_24+cag25_34', data=df).fit()
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
gpris            -0.1809      0.048     -3.798      0.000      -0.274      -0.087
gpolpc            0.0514      0.056      0.926      0.355      -0.058       0.160
gincpc            0.7384      0.166      4.438      0.000       0.412       1.065
cunem             0.4113      0.394      1.045      0.297      -0.362       1.184
cblack           -0.0147      0.033     -0.445      0.657      -0.080       0.050
cmetro            0.5383      0.996      0.541      0.589      -1.417       2.493
```

We also estimate this equation using pooled 2SLS, where the instuments for $\Delta log(prison)$ are two binary variables, one each for whether a final decision was reached on overcrowding litigation in the current year or in the previous two years.

```
res_iv = iv.IV2SLS.from_formula(formula='gcriv~1+C(year)+[gpris~final1+final2]+'+
            'gpolpc+gincpc+cunem+cblack+cmetro+cag0_14+cag15_17+cag18_24+cag25_34',
            data=df).fit()
               Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
--------------------------------------------------------------------------------
gpolpc            0.0353     0.0603     0.5859     0.5579     -0.0828      0.1534
gincpc            0.9102     0.3258     2.7941     0.0052      0.2717      1.5487
cunem             0.5237     0.4809     1.0889     0.2762     -0.4189      1.4663
cblack           -0.0158     0.0403    -0.3928     0.6945     -0.0949      0.0632
cmetro           -0.5915     1.6039    -0.3688     0.7123     -3.7352      2.5522
gpris            -1.0320     0.3314    -3.1139     0.0018     -1.6815     -0.3824
```

The pooled 2SLS estiamte of α_1 is -1.032(se=0.370). Therefore, the 2SLS estimated effect is much larger, but less precise.

Testing for AR(1) serial correlation in $r_{it1} = \Delta u_{it1}$ is easy as well. A rough estimate is obtained as follows showing modest serial correlation.

```
df['resid'] = res_iv.resids
df['resid_1'] = res_iv.resids.shift(1)
res_iv = smf.ols(formula='resid~resid_1', data=df).fit()
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept   1.592e-16   6.86e-17      2.322      0.021    2.46e-17    2.94e-16
resid_1        0.0858      0.037      2.296      0.022       0.012       0.159
```

□

An alternative approach to estimating SEMs with panel data is to use the fixed effects transformation and then apply the IV techniques such as pooled 2SLS.

17 Limited Dependent Variable Models and Sample Selection Corrections

A limited dependent variable (LDV) model is defined as a dependent variable whose range of values is substantially restricted. A binary takes only two values. A percentage variable can take values only between 0 and 100. Most economic variables are limited to be positive. The Logit and Probit model takes care of these cases. Optimizing behaviors often lead to a corner solution response for some nontrivial fraction of the population. The Tobit model covers this. Poisson regression models are well suited for modelling count variables.

17.1 Logit and probit models for binary response

In a **binary response models**, interest lies primarily in the response probability $P(y = 1|\boldsymbol{x}) = P(y = 1|x_1, \ldots, x_k)$, where $\boldsymbol{x}$ is the full set of explanatory variables. To avoid linear probability model limitations, consider a class of binary response models of the form $P(y = 1|\boldsymbol{x}) = G(\beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k) = G(\boldsymbol{x\beta})$, where G is a function such that $0 < G(z) < 1$ for all real numbers z, representing response probability and $\boldsymbol{x\beta}$ subsumes the constant. In the **logit model**, we have the logistic function $G(z) = \Lambda(z) = e^z/(1 + e^z)$, which is the cdf of a standar logistic random variable. In the **probit model**, we have the standard normal cdf $G(z) = \Phi(z) = \int_{-\infty}^{z} \phi(\nu)d\nu$, where $\phi(z) = (2\pi)^{-1/2}e^{-z^2/2}$, the standard normal density. For both these functions, $G(z)0$ as $z \to -\infty$ and $G(z) \to \infty$ as $z \to \infty$.

Logit and probit models can be derived from an underlying **latent variable model**. Let y^* be an unobserved, or latent, variable and $y^* = \boldsymbol{x\beta} + e$, and $y = \boldsymbol{I}_{y^*>0}$, the indicator function.

We assume e to be independent of $\boldsymbol{x}$ and has a standard logistic or standard normal distribution, symmetric about zero ($1-G(-z)=G(z)$). We can thus show $P(y=1|\boldsymbol{x})=G(\boldsymbol{x}\boldsymbol{\beta})$.

In order to explain the effect of the x_j on the response probability $P(y=1|\boldsymbol{x})$, the magnitudes of each β_j are not, themselves, useful. If x_j is roughly continuous we can write $\frac{\partial p(\boldsymbol{x})}{\partial x_j}=g(\boldsymbol{x}\boldsymbol{\beta})\beta_j$, where $g(z)=G'(z)$, the probability density function and hence $g(z)>0$ for all z. Therefore, the partial effect of x_j on $p(\boldsymbol{x})$ depends on $\boldsymbol{x}$ through the positive quantity $g(\boldsymbol{x}\boldsymbol{\beta})$, which means that the partial effect always has the same sign as β_j. Consequently, the relative effects of any two continuous explanatory variables x_j and x_h is β_j/β_h. For a symmetric distribution with unique mode at zero for g, the largest individual effect occurs when $\boldsymbol{x}\boldsymbol{\beta}=0$. Thus, for probit case at $g(0)=\phi(0)\approx 0.4$ and for logit case at $g(0)=e^z/(1+e^z)^2|_{z=0}=0.25$. For a binary explanatory variable x_j, the partial effect from changing x_j from zero to one, holding all other variables fixed, is $G(\beta_0+\ldots+\beta_i+\ldots+\beta_k x_l)-G(\beta_0+\ldots+0+\ldots+\beta_k x_k)$. This again depends on all the values of other x_j. In both cases, knowing the sign of β_1 is sufficient for determining whether the program had a positive or negative effect. But to find the magnitude of the effect, we have to estimate the nonlinear quantity that depends on all other values of xs.

To include functional form among the explanatory variables is straightforward, but the partial effect has to be carefully evaluated. The effects, generally, depend on all other variables. Models with interactions among the explanatory variables can be a bit tricky, again and one has to be cautious in evaluating the resulting partial effects at interesting values.

Maximum likelihood estimation (MLE) is the preferred way to estimate the coefficients for limited dependent variable models. Because MLE is based on the distribution of y given $\boldsymbol{x}$, the heteroskedasticity in $Var(y|\boldsymbol{x})$ is automatically accounted for. We need the log of density of y_i given $\boldsymbol{x}_i$ and β summed over all i.i.d. observations, $\mathcal{L}(\boldsymbol{\beta})=\sum_{i=1}^{n}\ell_i(\boldsymbol{\beta})$, where $\ell_i(\boldsymbol{\beta})=y_i\log(G(\boldsymbol{x}_i\boldsymbol{\beta}))+(1-y_i)\log(1-G(\boldsymbol{x}_i\boldsymbol{\beta}))$. The MLE of $\boldsymbol{\beta}$, denoted by $\hat{\boldsymbol{\beta}}$ maximizes the log-likelihood. The general theory of MLE for random samples implies that, under very general conditions, the MLE is consistent, asymptotically normal, and asymptotically efficient. Each $\hat{\beta}_j$ comes with an asymptotic standard error. Once these are available, we can construct t tests and confidence intervals.

To test $H_0:\beta_j=0$, we form the t statistic $\hat{\beta}_j/se(\hat{\beta}_j)$ and carry out the test in the usual way, once we have decided on a one-or-two sided alternative. We can also test multiple exclusions using Lagrange multiplier or score test or Wald test or F test. If both restricted and unrestricted models are easy to estimate - then the **likelihood ratio (LR) test** becomes very attractive. The LR test is based on the difference in the log-likelihood functions for the unrestricted and restricted models. Because the MLE maximizes the log-likelihood function, dropping variables generally leads to a smaller log-likelihood. The question is whether the fall in the log-likelihood is large enough to conclude that that dropped variables are important. The likelihood ratio statistic $LR=2(\mathcal{L}_{ut}-\mathcal{L}_r)$, is twice the difference of unrestricted and restricted log-likelihood. Here $LR\overset{a}{\sim}\chi^2_q$ with q exclusion restrictions. To test H_0 we

look at the p-values based on this distribution.

Percent correctly predicted serves as a reasonable goodness of fit measure. We can predict $\tilde{y}_i = 1$ if $G(\boldsymbol{x}_i\hat{\boldsymbol{\beta}}) \geq \theta = 0.5$ and 0 otherwise. We then have four possible outcomes for each pair $(y_i, \tilde{y}_i)$. The percentage correctly predicted is the percentage of time that $y_i = \tilde{y}_i$. This measure can be misleading for the least likely outcome. Often, we hope to have some ability to predict the least likely outcome. Therefore, it makes sense to also compute the percentage correctly predicted for each of the outcomes. Another alternative is to use the fraction of successes in the sample as the threshold, i.e. θ. A third possibility is to choose the threshold such that the fraction of $\tilde{y}_i = 1$ in the same is the same as $\overline{y}$.

There are also various **pseudo R-squared** measure for binary responses. One can sure $1 - \mathcal{L}_{ur}/\mathcal{L}_0$, where $\mathcal{L}_{ur}$ is the log-likelihood for the estimated model and $\mathcal{L}_0$ is the log likelihood for the model with only the intercept. If the covariates have no explanatory power then $\mathcal{L}_{ur}/\mathcal{L}_0 = 1$, and the pseudo R-squared is zero. When $\mathcal{L}_{ur}$ were zero, the pseudo R-squared would equal unity. Alternative pseudo R-squares for probit and logit are more directly related to the usual R-squared from OLS estimation. Let $\hat{y}_i = G(\boldsymbol{x}_i\hat{\boldsymbol{\beta}})$ be the fitted probabilities. Since these probabilities are also estimates of $E(y_i|\boldsymbol{x}_i)$, we can base an R-squared on the squared correlation between y_i and $\hat{y}_i$.

To estimate the effects of the x_j on the response probability $P(y = 1|\boldsymbol{x})$, for continuous x_j we have $\Delta\hat{P}(y = 1|\boldsymbol{x}) \approx g(\boldsymbol{x}\hat{\boldsymbol{\beta}})\hat{\beta}_j\Delta x_j$. So, for $\Delta x_j = 1$, the change in the estimated success probability is roughly $g(\boldsymbol{x}\hat{\boldsymbol{\beta}})\hat{\beta}_j$, which depend on all βs and $\boldsymbol{x}$s. One can use sample average values to get a representative value for this term $g(\overline{\boldsymbol{x}}\hat{\boldsymbol{\beta}})$. When multiplied by $\hat{\beta}_j$, we obtain the partial effect of x_j for the 'average' person in the sample - **partial effect at the average (PEA)**. This is potentially problematic when we have discrete explanatory variables whose average might be nonsensical. Also if we use some nonlinear functional form of a variable the average is taken of the nonlinear function of the variable, and not the variable itself.

Another approach is to calculate **average partial effect (APE)**, also called the average marginal effect (AME), where we average the individual partial effect across the sample, $\left(\frac{1}{n}\sum_{i=1}^{n} g(\boldsymbol{x}_i\hat{\boldsymbol{\beta}})\right)\hat{\beta}_j$. The two scale factors differ - and are possibly quite different. Neither of these make sense for discrete variables. Instead it is better to use the difference in the values of $G(.)$. We can either use average value of the variable of the average of the partial effect, like we did for the continuous case.

When comparing the partial effects of probit, logit and LPM models, it makes sense to compute the scale factors described above for probit and logit. A quicker way to compare the magnitudes is to use the most influential point $g(0) \approx 0.4$ for probit and $g(0) = 0.25$ for logit. We can multiply probit coefficients by $0.4/0.15 = 1.6$ to make it comparable to logit coefficient. In LPM, $g(0)$ is effectively one, so logit slope can be divided by 4 and probit slope divided by 2.5 to make them comparable to LPM estimates. Scale factors give a more accurate comparison.

Example 17.1. We estimate the labor force participation of married women by probit and logit and compare it to LPM heteroskedasticity-robust estimates.

```
df = woo.data('mroz')
res_lpm = smf.ols(formula='inlf~nwifeinc+educ+exper+expersq+age+kidslt6+kidsge6',
                  data=df).fit(cov_type='HC3')
res_logit = smf.logit(formula='inlf~nwifeinc+educ+exper+expersq+age+kidslt6+kidsge6',
                  data=df).fit()
res_probit = smf.probit(formula='inlf~nwifeinc+educ+exper+expersq+age+kidslt6+kidsge6',
                  data=df).fit()
a, b = compare({'LPM': res_lpm, 'Logit': res_logit, 'Probit': res_probit})

         LPM    Logit   Probit
r2     0.264      NaN      NaN
ar2    0.257      NaN      NaN
pr2      NaN    0.220    0.221
llf -423.892 -401.765 -401.302
dfr  745.000  745.000  745.000
dfm    7.000    7.000    7.000

           coeff                 tval                stderr
             LPM  Logit Probit    LPM  Logit Probit    LPM  Logit Probit
Intercept  0.586  0.425  0.270  3.798  0.494  0.531  0.154  0.860  0.509
age       -0.016 -0.088 -0.053 -6.476 -6.040 -6.235  0.002  0.015  0.008
educ       0.038  0.221  0.131  5.151  5.091  5.183  0.007  0.043  0.025
exper      0.039  0.206  0.123  6.962  6.422  6.590  0.006  0.032  0.019
expersq   -0.001 -0.003 -0.002 -3.227 -3.104 -3.145  0.000  0.001  0.001
kidsge6    0.013  0.060  0.036  0.986  0.804  0.828  0.013  0.075  0.043
kidslt6   -0.262 -1.443 -0.868 -7.814 -7.090 -7.326  0.034  0.204  0.119
nwifeinc  -0.003 -0.021 -0.012 -2.351 -2.535 -2.484  0.001  0.008  0.005
```

The estimates from the three models tell a consistent story. The R-squares are also comparable. As we already emphasized, the magnitudes of the coefficients estimates across models are not directly comparable. Instead we compute the scale factors PEA and APE.

```
avg_exog = df[res_logit.params.index.intersection(df.columns)].mean()
PEA_lpm = res_lpm.params
PEA_logit = stats.logistic.pdf(res_logit.predict(avg_exog))*res_logit.params
PEA_probit = stats.norm.pdf(res_probit.predict(avg_exog))*res_probit.params
PEA = pd.DataFrame({'lpm': PEA_lpm, 'logit': PEA_logit, 'probit': PEA_probit}).iloc[1:]
# automatic
PEA = pd.DataFrame({'lpm': PEA_lpm[1:], 'logit': res_logit.get_margeff('mean').margeff,
                                        'probit': res_probit.get_margeff('mean').margeff})
               lpm     logit    probit
nwifeinc -0.003405 -0.005190 -0.004696
educ      0.037995  0.053777  0.051129
exper     0.039492  0.050057  0.048177
expersq  -0.000596 -0.000767 -0.000737
age      -0.016091 -0.021403 -0.020643
kidslt6  -0.261810 -0.350950 -0.339151
kidsge6   0.013012  0.014616  0.014063
```

```
APE_lpm = res_lpm.params
APE_logit = np.mean(stats.logistic.pdf(res_logit.fittedvalues)) * res_logit.params
APE_probit = np.mean(stats.norm.pdf(res_probit.fittedvalues)) * res_probit.params
APE = pd.DataFrame({'lpm': APE_lpm, 'logit': APE_logit, 'probit': APE_probit}).iloc[1:]
# automatic
APE = pd.DataFrame({'lpm': APE_lpm[1:], 'logit': res_logit.get_margeff('overall').margeff,
                                    'probit': res_probit.get_margeff('overall').margeff})
                                                lpm     logit    probit
nwifeinc -0.003405 -0.003812 -0.003616
educ      0.037995  0.039497  0.039370
exper     0.039492  0.036764  0.037097
expersq  -0.000596 -0.000563 -0.000568
age      -0.016091 -0.015719 -0.015896
kidslt6  -0.261810 -0.257754 -0.261154
kidsge6   0.013012  0.010735  0.010829
```

The biggest difference between the LPM model and the logit and probit models is that the LPM assumes constant marginal effects for *educ*, *kidslt*6, and so on, while the logit and probit models imply diminishing magnitudes of the partial effects.

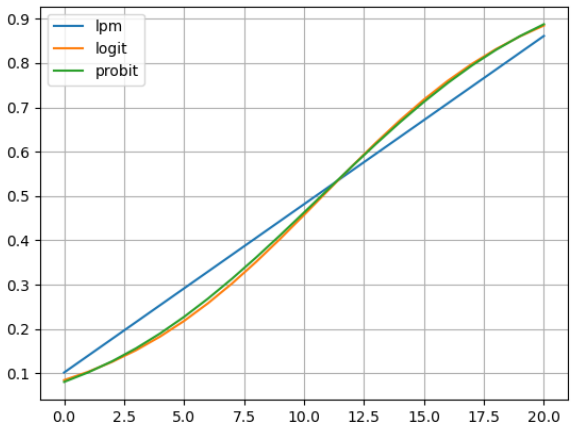

Figure 1: Response probability with respect to education for the three models.

```
inlt_lpm, inlt_logit, inlt_probit = dict(). dict(), dict()
for i in range(0, 21):
    avg_exog.update({'educ': i})
    inlt_lpm[i] = res_lpm.predict(avg_exog)[0]
    inlt_logit[i] = res_logit.predict(avg_exog)[0]
    inlt_probit[i] = res_probit.predict(avg_exog)[0]
p = pd.DataFrame({'lpm': pd.Series(inlt_lpm),
                  'logit': pd.Series(inlt_logit),
                  'probit': pd.Series(inlt_probit)})
p.plot(grid=True)
```

The figure illustrates how the estimated response probabilities from nonlinear binary response model can differ form the linear probability model. □

It is possible to test and correct for endogenous explanatory variables using methods related to two stage least squares. The other issues of interest are the distribution of e and heteroskedasticity in e. Binary response models apply with little modification to independently pooled cross sections or to other data sets where the observations are independently but not necessarily identically distributed. Similarly, linear probability model can be applied with panel data, estimated generally by fixed effects. Logit and probit models are complicated to use with panel data due to non-linearity.

17.2 Tobit model for corner solution responses

Limited dependent variable with corner response is zero for a nontrivial fraction of the population but is roughly continuously distributed over positive values. Let y be a variable that is essentially continuous over strictly positive values but that takes on a value of zero with positive probability. Because the distribution of y piles up at zero, y clearly cannot have a conditional normal distribution. So all inference would have only asymptotic justification. The **Tobit model** expresses the observed response y, in terms of an underlying latent variable $y^* = \boldsymbol{x}\boldsymbol{\beta} + u$, where $u|\boldsymbol{x} \sim \mathcal{N}(0, \sigma^2)$ with $y = max(0, y^*)$. Hence, $P(y = 0|\boldsymbol{x}) = P(y^* < 0|\boldsymbol{x}) = P(u < -\boldsymbol{x}\boldsymbol{\beta}|\boldsymbol{x}) = P(\frac{u}{\sigma} < -\frac{\boldsymbol{x}\boldsymbol{\beta}}{\sigma}|\boldsymbol{x}) = \Phi(-\frac{\boldsymbol{x}\boldsymbol{\beta}}{\sigma}) = 1-\Phi(\frac{\boldsymbol{x}\boldsymbol{\beta}}{\sigma})$, since $\frac{u}{\sigma}$ has a standard normal distribution independent of $\boldsymbol{x}$. The log-likelihood function for each observation i can, thus be written as $\ell_i(\boldsymbol{\beta}, \sigma) = \boldsymbol{I}_{y_i=0}\ log(1-\Phi(\frac{\boldsymbol{x}_i\boldsymbol{\beta}}{\sigma})) + \boldsymbol{I}_{y_i>0}\ log(\frac{1}{\sigma}\phi(\frac{y_i-\boldsymbol{x}_i\boldsymbol{\beta}}{\sigma}))$, where ϕ is the standard normal density. This is summed over all i to get the log-likelihood. The estimates of $\boldsymbol{\beta}$ and σ are obtained by maximizing the log-likelihood. Each estimate comes with a standard error, which can be used to construct t statistic for each $\hat{\beta}_j$. Testing multiple exclusion restrictions is easily done using the Wald test or the likelihood ratio test.

We are generally interested in the conditional expectation $E(y|y > 0, \boldsymbol{x})$. The unconditional expectation $E(y|\boldsymbol{x})$ is easily found via $E(y|\boldsymbol{x}) = \Phi(\frac{\boldsymbol{x}\boldsymbol{\beta}}{\sigma})E(y|y > 0, \boldsymbol{x})$. We can evaluate $E(y|y > 0, \boldsymbol{x}) = \boldsymbol{x}\boldsymbol{\beta} + \sigma\lambda(\frac{\boldsymbol{x}\boldsymbol{\beta}}{\sigma})$ where $\lambda(c) = \frac{\phi(c)}{1-\Phi(c)}$ called the **inverse Mills ratio**, the ratio between the standard normal pdf and the standard normal cdf evaluated at c. In context of an OLS, the inverse Miller ratio is an omitted variable, and it is generally correlated with the elements of $\boldsymbol{x}$. This gives $E(y|\boldsymbol{x}) = \Phi(\frac{\boldsymbol{x}\boldsymbol{\beta}}{\sigma})\boldsymbol{x}\boldsymbol{\beta} + \sigma\phi(\frac{\boldsymbol{x}\boldsymbol{\beta}}{\sigma})$.

If x_j is a continuous variable, we can find the partial effect as

$$\frac{\partial}{\partial x_j}E(y|y > 0, \boldsymbol{x}) = \beta_j \left(1 - \lambda\left(\frac{\boldsymbol{x}\boldsymbol{\beta}}{\sigma}\right)\left(\frac{\boldsymbol{x}\boldsymbol{\beta}}{\sigma} + \lambda\left(\frac{\boldsymbol{x}\boldsymbol{\beta}}{\sigma}\right)\right)\right).$$

The partial effect of x_j on $E(y|y > 0, \boldsymbol{x})$ is β_j multiplied by a factor between 0 and 1, which is a linear function of $\boldsymbol{x}$. We estimate this by plugging in the MLEs of the β_j and σ. We usually plug mean values or other interesting value of x_j to get an estimate of the effect. σ, sometimes called the ancillary parameter, is critical in estimating the partial effects magnitude but not sign. For binary and discrete variables, the effect of interest is obtained as the

difference between $E(y|y>0,\boldsymbol{x})$, with different values of the variable.

We can further establish that

$$\frac{\partial}{\partial x_j}E(y|\boldsymbol{x}) = \beta_j \Phi\left(\frac{\boldsymbol{x}\boldsymbol{\beta}}{\sigma}\right).$$

Compared to OLS, we see the tobit estimates are adjusted down. The PEA is obtained by evaluating $\Phi(\frac{\overline{\boldsymbol{x}}\hat{\boldsymbol{\beta}}}{\hat{\sigma}})$ while APE (which is preferred in most cases) is computed as $\frac{1}{n}\sum_{i=1}^{n}\Phi(\frac{\boldsymbol{x}_i\hat{\boldsymbol{\beta}}}{\hat{\sigma}})$. In fact, $\hat{P}(y_i>0|\boldsymbol{x}_i) = \Phi(\frac{\boldsymbol{x}_i\hat{\boldsymbol{\beta}}}{\hat{\sigma}})$, and so the APE scale factor and PEA scale factor tends to be closer to one when there are few observations with $y_i = 0$, and the tobit estimates look closer to OLS. Comparing OLS and Tobit for binary factors is not that easy.

Example 17.2. We look at the hours worked by women, 325 or 753 of which worked zero hours. We compare OLS model versus a Tobit model.

```
import statsmodels.base.model as smclass
df = woo.data('mroz')
y, X = pt.dmatrices('hours~nwifeinc+educ+exper+expersq+age+kidslt6+kidsge6',
                    data=df, return_type='dataframe')
res_ols = sm.OLS(y, X).fit(cov_type='HC0')
start_params = pd.concat([res_ols.params, pd.Series({'sigma': res_ols.resid.std()})])
class Tobit(smclass.GenericLikelihoodModel):
    def nloglikeobs(self, params):
        s = params[-1]
        y, X = self.endog, self.exog
        yH = X @ params[:-1]
        ll = np.empty(len(y))
        ll[y == 0] = np.log(stats.norm.cdf(-yH[y == 0]/s))
        ll[y > 0] = np.log(stats.norm.pdf((y-yH)[ y > 0]/s)/s)
        return -ll
res_tobit = Tobit(y, X).fit(start_params=start_params, maxiter=10000, disp=0)
```

We see both models give same signs to the coefficients with similar t statistics as well. Comparing the magnitudes of the coefficients of the two model is not informative, though tempting.

```
pd.DataFrame({'ols': res_ols.params.values, 'tobit': res_tobit.params[:-1]},
                index=res_ols.params.index
                   ols       tobit
Intercept  1330.482400  965.305168
nwifeinc     -3.446636   -8.814242
educ         28.761125   80.645579
exper        65.672513  131.564327
expersq      -0.700494   -1.864158
age         -30.511634  -54.405006
kidslt6    -442.089908 -894.021641
kidsge6     -32.779226  -16.217990
```

```
pd.DataFrame({'ols': res_ols.tvalues, 'tobit': res_tobit.tvalues[:-1]},
                index=res_ols.params.index)
                ols     tobit
Intercept  4.866191  2.162271
nwifeinc  -1.546459 -1.976692
educ       2.217580  3.736502
exper      6.116641  7.613950
expersq   -1.893066 -3.467159
age       -7.226509 -7.333743
kidslt6   -7.734555 -7.991047
kidsge6   -1.445233 -0.419727
print(res_ols.resid.std(), res_tobit.params[-1])
>>> 746.6789217101644 1122.0216717957192
print(res_ols.rsquared, np.corrcoef(res_tobit.exog @ res_tobit.params[:-1],
                    res_tobit.endog)[0,1]**2)
>>> 0.26562449298495205 0.26089211771972076
```

We can multiply the Tobit estimates by appropriate adjustment factors to make them roughly comparable to the OLS estimates. The APE scale factor $\frac{1}{n}\sum_{i=1}^{n}\Phi(\frac{\boldsymbol{x}_i\hat{\boldsymbol{\beta}}}{\hat{\sigma}})$ turns out to be 0.589 which can be used to obtain the average partial effects for the Tobit estimation. The Tobit APEs for *nwifeinc*, *educ* and *kidslt6* are all substantially larger in magnitude than the corresponding OLS coefficients. We can also use the effect at the average values by computing PEA scale factor $\Phi(\frac{\bar{\boldsymbol{x}}\hat{\boldsymbol{\beta}}}{\hat{\sigma}})$ which turns out to be 0.604 which gives double the effect of *educ* when compared to the OLS model.

```
avg_exog = df[res_ols.params.index.intersection(df.columns)].mean()
PEA_ols = res_ols.params
yHA = res_tobit.params[0]+np.dot(res_tobit.params[1:-1],avg_exog)
yH = res_tobit.exog @ res_tobit.params[:-1]
s  = res_tobit.params[-1]
PEA_tobit = stats.norm.cdf(yHA/s)
APE_tobit = np.mean(stats.norm.cdf(yH/s))
print(PEA_tobit, APE_tobit)
>>> 0.6042994317813641 0.5886633769773091
pd.DataFrame({'OLS': PEA_ols[1:], 'PEAtobit': PEA_tobit * res_tobit.params[1:-1],
                                  'APEtobit': APE_tobit * res_tobit.params[1:-1]})
                 OLS    PEAtobit    APEtobit
nwifeinc   -3.446636   -5.326441   -5.188621
educ       28.761125   48.734078   47.473099
exper      65.672513   79.504248   77.447101
expersq    -0.700494   -1.126510   -1.097362
age       -30.511634  -32.876914  -32.026234
kidslt6  -442.089908 -540.256770 -526.277798
kidsge6   -32.779226   -9.800522   -9.546937
```

The R-squared for the OLS model and Tobit models are similar. For Tobit, the R-squared is the square of the correlation coefficient between y_i and $\hat{y}_i$. In nonlinear models, like Tobit, the squared correlation coefficient is not identical to an R-squared based on a sum of squared

residuals. This is because the fitted values $\hat{y}_i$ and the residuals $y_i - \hat{y}_i$ are not uncorrelated in the sample. The Tobit estimates are chosen to maximize the log-likelihood, whereas the OLS estimates are the values that do produce the highest R-squared given the linear functional form.

```
avg_exog = df[res_ols.params.index.intersection(df.columns)].mean()
inlt_ols = dict()
inlt_tobit = dict()
for i in range(0, 21):
    avg_exog.update({'educ': i})
    inlt_ols[i] = np.dot(avg_exog, res_ols.params[1:])+res_ols.params[0]
    yH = np.dot(avg_exog, res_tobit.params[1:-1])+res_tobit.params[0]
    s = res_tobit.params[-1]
    inlt_tobit[i] = stats.norm.cdf(yH/s)*yH + s * stats.norm.pdf(yH/s)
p = pd.DataFrame({'ols': pd.Series(inlt_ols),
                  'tobit': pd.Series(inlt_tobit)})
ax = p.plot(grid=True)
df.plot.scatter(ax=ax, x='educ', y='hours', alpha=0.1, grid=True)
```

By construction, all of the Tobit fitted values for hours are positive, while 39 out of 753 OLS fitted values are negative. We show the estimate of $E(y|\boldsymbol{x})$ as a function of educational for the Tobit model, the other explanatory variables are set to their average values. The linear model gives notably higher estimates of the expected hours worked at even fairly high levels of education. The increasing slope of the Tobit line clearly indicates the increasing marginal effect of education on expected hours worked. □

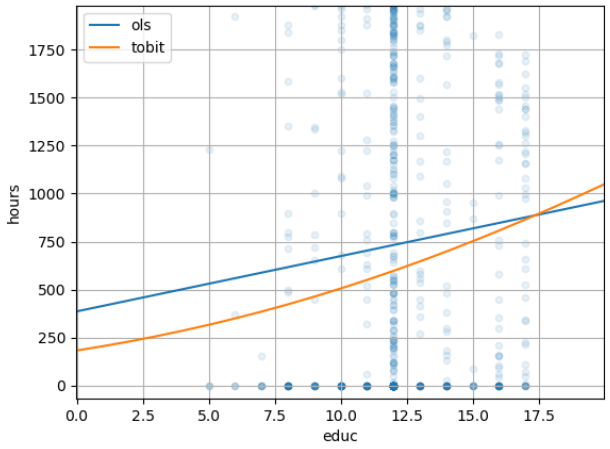

Figure 2: Estimated expected hours as a function of education with other variables at their average.

The Tobit model estimation, depends on the assumptions of normality and homoskedasticity in the underlying latent variable model. Moderate departures are acceptable but huge

deviations are problematic. The effect of x_j on $P(y > 0|\boldsymbol{x})$ and $P(y|y > 0, \boldsymbol{x})$ is proportional to β_j and depends on $\boldsymbol{x}$ only through $\boldsymbol{x\beta}/\sigma$. This rules out the corner effects on the other end.

One way to informally evaluate whether the Tobit model is appropriate is to estimate a probit model where the binary outcomes are $y > 0$ and $y = 1$. Fitting a probit model gives a coefficient of $\gamma_j = \beta_j/\sigma$ on x_j. If the Tobit model holds then $\hat{\gamma}_j$ should be close to $\hat{\beta}_j/\hat{\sigma}$ obtained from Tobit estimates. Huge deviations indicates that Tobit model might not be valid.

```
res_probit = sm.Probit(y>0, X).fit()
# nwifeinc
print(res_tobit.params[1] / res_tobit.params[-1], res_probit.params.nwifeinc)
>>> -0.00785567871258899 -0.012023739040370837
# kidslt6
print(res_tobit.params[-3] / res_tobit.params[-1], res_probit.params.kidslt6)
>>> -0.7967953415368939 -0.8683285096989993
```

If we conclude that Tobit model is inappropriate, non-linear models called hurdle or two-part models, can be used.

17.3 The Poisson regression model

Another kind of non-negative dependent variable is a **count variable**, taking non-negative integer values, e.g. number of kids born to a woman. It might be information to start with a simple linear model, but it might not provide the best fit. We can't take a log as zeros might be involved, but an exponentiation can be suitable giving the model $E(y|\boldsymbol{x}) = exp(\boldsymbol{x\beta})$. This can be interpreted as $\%\Delta E(y|\boldsymbol{x}) \approx 100\beta_j \Delta x_j$. For a more accurate measure we can evaluate the discrete changes in the expected value $\Delta E(y|\boldsymbol{x}) = e^{\beta_k \Delta x_k} - 1$, keeping all but x_k fixed. Essentially, with $log(y)$ as the dependent variable we can interpret it as a linear model.

All standard count date exhibit heteroskedasticity. We rely on maximum likelihood and the important related method of quasi-maximum likelihood estimation. Instead of normal the nominal distribution for count data is the **Poisson distribution**, which is entirely determined by its mean. Hence, $P(y = h|\boldsymbol{x}) = \frac{1}{h!} e^{-e^{\boldsymbol{x\beta}}} (e^{\boldsymbol{x\beta}})^h$, for $h = 0, 1, \ldots$. This distribution, which is the basis for the **Poisson regression model**, allows us to find conditional probabilities for any values of the explanatory variables. Given a random sample $\{(\boldsymbol{x}_i, y_i) : i = 1, 2, \ldots, n\}$, we can construct the log-likelihood function $\mathcal{L}(\boldsymbol{\beta}) = \sum_{i=1}^{n} \ell_i(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left(y_i \boldsymbol{x}_i \boldsymbol{\beta} - e^{\boldsymbol{x}_i \beta} \right)$.

As with other non-linear models, we cannot directly compare the magnitudes of the Poisson estimates of an exponential function with the OLS estimates of a linear function. Since $\frac{\partial}{\partial x_j} E(y|\boldsymbol{x}) = e^{\boldsymbol{x\beta}} \beta_j$, the APE scale factor is simply $\frac{1}{n} \sum_{i=1}^{n} e^{\boldsymbol{x}_i \hat{\boldsymbol{\beta}}} = \sum_{i=1}^{n} \hat{y}_i = \overline{y}$. Thus the OLS estimate $\hat{\gamma}_j$ should be compared to $\overline{y}\hat{\beta}_j$ in Poisson regression.

Poisson distribution are restrictive as the mean is equal to the variance and this has been shown to be violated in many application. However, Poisson distribution is robust - whether or not the Poisson distribution holds, we still get consistent, asymptotically normal estimators of the β_j (analogous to the OLS estimator). When we use Poisson MLE, but do not assume that the Poisson distribution is entirely correct, we call the analysis **quasi-maximum likelihood estimation**. The standard errors need to be adjusted in this case, though.

Under the assumption that $Var(y|\boldsymbol{x}) := \sigma^2 E(y|\boldsymbol{x})$, called overdispersion when $\sigma^2 > 1$ and underdispersion when $\sigma^2 < 1$ (less common). Let $\hat{\beta}_j$ be the Poisson QMLE and $hatu_i = y_i - \hat{y}_i$ be the residual. A consistent estimator of σ^2, in this case is $\frac{1}{n-k-1}\sum_{i=1}^{n}\frac{\hat{u}_i^2}{\hat{y}_i}$, where the division by $\hat{y}_i$ is the proper heteroskedasticity adjustment and $n-k-1$ is the degree of freedom given n observations and $k+1$ estimates. We multiply the usual Poisson standard error by $\hat{\sigma}$ to get the corrected Poisson MLE standard errors.

We can use likelihood ratio statistic to test exclusion restrictions with $LR = 2(\mathcal{L}_{ur} - \mathcal{L}_r)$. If we have q exclusion restrictions, the statistic is distributed approximately as χ_q^2 under the null. Under the less less restrictive assumption of $Var(y|\boldsymbol{x}) = \sigma^2 E(y|\boldsymbol{x})$ we divide LR by $\hat{\sigma}^2$ where $\hat{\sigma}^2$ is obtained from the unrestricted model - this is called quasi-likelihood ratio statistic.

Example 17.3. We estimate the number of times a man is arrested in 1986 ($narr86$) using Poisson regression and a linear OLS model. The standard error for the Poisson model $\hat{\sigma} = 1.23$, the standard error of the Poisson regression shoudl be inflated by this factor, reducing the significance of all variables.

```
df = woo.data('crime1')
y, X = pt.dmatrices('narr86~pcnv+avgsen+tottime+ptime86+qemp86++inc86'+
        '+black+hispan+born60', data=df, return_type='dataframe')
res_ols = sm.OLS(y, X).fit(cov_type='HC0')
res_psn = sm.Poisson(y, X).fit()
print(np.sqrt((res_psn.resid**2/res_psn.predict(X)).sum()/
            (res_psn.nobs-res_psn.df_model+1)))
>>> 1.2311261693880455
df.narr86.corr(res_psn.predict(X))**2
>>> 0.07700390941530029
```

The OLS and Poisson coefficients cannot be compared directly. For coefficient $pcnv$ implies that if $\Delta pcnv = 0.1$, the expected number of arrests falls by 0.013 using the OLS model. The Poisson coefficient implies that $\Delta pvnc = 0.1$ mean 0.402×0.1 =1% reduction in overall arrests if we can increase the probability of convictio by 0.1. The Poisson coefficient on $black$ implies that, other factors beign equal, the expected number of arrests for a black man is estimated to be about $100(e^{0.661} - 1) \approx 94\%$ higher than for a white man with the same values for the other explanatory variables.

The R-squares can be calculated for the Possion model using the correlation between y_i and $\hat{y}_i$ and comes out to 0.28 versus R-squared of 0.072 in OLS

```
pd.DataFrame({'ols': res_ols.params, 'psn': res_psn.params})
                ols       psn
Intercept  0.576566 -0.599589
pcnv      -0.131886 -0.401571
avgsen    -0.011332 -0.023772
tottime    0.012069  0.024490
ptime86   -0.040873 -0.098558
qemp86    -0.051310 -0.038019
inc86     -0.001462 -0.008081
black      0.327010  0.660838
hispan     0.193809  0.499813
born60    -0.022465 -0.051029
pd.DataFrame({'ols': res_ols.tvalues, 'psn': res_psn.tvalues})
                 ols       psn
Intercept  13.558642 -8.915805
pcnv       -3.933856 -4.725970
avgsen     -0.802811 -1.191831
tottime     0.917576  1.660318
ptime86    -6.023150 -4.762510
qemp86     -3.618744 -1.309897
inc86      -6.396118 -7.762373
black       5.606131  8.950288
hispan      4.834506  6.760930
born60     -0.701263 -0.796677
```

□

17.4 Censored and truncated regression models

There was no issue of data observability in applying logit, probit, tobit and Poisson models. In case the response variable has been censored above or below some threshold, due to survey design or institutional constraints we apply **censored regression model**. Essentially, we solve the problem of missing data on the response variable y. The sampling is still random, just that some y_i have missing values, but them being below or above a threshold, provides useful information for estimating the parameters. A **truncated regression model** arises when we exclude, on the basis of y, a subset of the population in our sampling scheme, i.e instead of random sample but a rule that determines whether y is above or below a certain threshold.

17.4.1 Censored Regression Models

In a censored normal regression model we want to explain y which follows a classical regression model $y_i = \boldsymbol{x}_i\boldsymbol{\beta} + u_i$, $u_i|\boldsymbol{x}_i, c_i \sim \mathcal{N}(0, \sigma^2)$ with $w_i = \min(y_i, c_i)$. Rather than observing y_i, we observe it only if it is less than a censoring value c_i. For example in **top coding** we

know the value only up to a certain threshold. An OLS regression using only the uncensored observations - that is, those with $y_i < c_i$ - produces inconsistent estimators of the β_j. An OLS regression of w_i on $\boldsymbol{x}_i$, using all observations, doe not consistently estimate β_j either, unless there is no censoring.

We can estimate $\boldsymbol{\beta}$ (and σ^2) by maximum likelihood, given a random sample on $(\boldsymbol{x}_i, w_i)$. The density of w_i is given by $f(w|\boldsymbol{x}_i, c_i) = \begin{cases} 1 - \Phi(\frac{c_i - \boldsymbol{x}_i\boldsymbol{\beta}}{\sigma^2}) & w = c_i \\ \frac{1}{\sigma}\phi(\frac{w - \boldsymbol{x}_i\boldsymbol{\beta}}{\sigma}) & w < c_i \end{cases}$. It is important to know that we can interpret that β_j just as in a linear regression model under random sampling, unlike in Tobit model. An important application of censored regression models is **duration analysis** where longer durations are censored. We often use natural log as the dependent variable where coefficients can be interpreted as percentage change and have distribution closer to normal than the duration distribution itself. If any of the assumptions of the censored normal regression model are violated - in particular, if there is heteroskedasticity or non-normality in u_i - the MLEs are generally inconsistent.

Example 17.4. We look at the time in months until an inmate is arrested after being released from prison, *durat*. 893 out of 1445 inmates were never arrested so we censor the values ranging from 70 to 81 months. We compare the simple OLS model against the censored normal regression model.

```
df = woo.data('recid')
y, X = pt.dmatrices('ldurat~workprg+priors+tserved+felon+alcohol+drugs'+
            '+black+married+educ+age', data=df, return_type='dataframe')
cens = df.cens == 1
res_ols = sm.OLS(y, X).fit()
start_params = res_ols.params
start_params['sigma'] = res_ols.resid.std()

class CensReg(smclass.GenericLikelihoodModel):
    def __init__(self, endog, exog, cens):
        self.cens = cens
        super(smclass.GenericLikelihoodModel, self).__init__(endog, exog, missing='none')
    def nloglikeobs(self, params):
        s = params[-1]
        y, X = self.endog, self.exog
        yH = X @ params[:-1]
        ll = np.empty(len(y))
        ll[self.cens] = np.log(1-stats.norm.cdf((y - yH)[self.cens] / s))
        ll[~self.cens] = np.log(stats.norm.pdf((y - yH)[~self.cens] / s) / s)
        return -ll
res_cens = CensReg(endog=y, exog=X, cens=cens).fit(start_params=start_params,
                                        maxiter=10000, method='BFGS', disp=0)
```

The Coefficient of censored regression gives the estimated percentage change in expected duration, given a ceteris paribus increase of one unit in the corresponding explanatory variable. The variable *priors*, number of prior convictions, and *tserved*, total month spent in

prison, have negative effects on the time until the next arrest occurs. This suggests that these variables measure proclivity for criminal activity rather than representing a deterrent effect.

```
pd.DataFrame({'ols': res_ols.params.values, 'cens': res_cens.params[:-1]},
                                            index=res_ols.params.index)
                ols      cens
Intercept  3.569168  4.099384
workprg    0.008758 -0.062565
priors    -0.059064 -0.137253
tserved   -0.009400 -0.019331
felon      0.178543  0.443993
alcohol   -0.262801 -0.634924
drugs     -0.090744 -0.298150
black     -0.179101 -0.542716
married    0.134433  0.340666
educ       0.005391  0.022919
age        0.001326  0.003910
pd.DataFrame({'ols': res_ols.tvalues, 'cens': res_cens.tvalues[:-1]},
                                      index=res_ols.params.index)
                 ols       cens
Intercept  25.870657  11.795604
workprg     0.178933  -0.521215
priors     -6.439761  -6.396155
tserved    -7.227340  -6.491316
felon       3.056836   3.060198
alcohol    -4.393985  -4.402577
drugs      -1.651780  -2.246193
black      -3.775689  -4.621118
married     2.425089   2.436066
educ        0.543184   0.902430
age         5.895805   6.450468
print(res_ols.resid.std(), res_cens.params[-1])
>>> 0.8733082368126388 1.8104689867236012
print(res_ols.rsquared,np.corrcoef(res_cens.exog @ res_cens.params[:-1],
                                                  res_cens.endog)[0,1]**2)
>>> 0.10872900679802966 0.10542409037515077
```

An inmate with one more prior conviction has a duration until next arrest that is almost 14% less. A year of time served reduces duration by about $100 \times 12 \times 0.019 = 23\%$. A man serving time for a felony has an estimated expected duration that is almost $e^{0.444} - 1 \approx 56\%$ longer than a man serving time for a non-felony. Those with a history of drug or alcohol abuse have substantially shorter expected durations until the next arrest. Older men and and men who were married at the time of incarceration, are expected to have significantly longer durations until their next arrest. Black men have substantially shorter durations, on the order $e^{-0.543} - 1 \approx 42\%$. The men who participated in the work program have estimated recidivism durations that are about 6.3% shorter than the men who did not participate, but the t statistic is small. This indicates that the work program has no effect.

The OLS estimates are quite different from censored version. In fact all are shrunk towards

zero. Although the direction of the effects are the same, the importance of these variables is greatly diminished. The censored regression estimates are much more reliable. □

17.4.2 Truncated Regression Models

In the case of data censoring, we do randomly sample units from the population, but y is observed only if not censored. With data truncation, we restrict attention to a subset of the population prior to sampling, so we have no information on explanatory variables. We aim to estimate effects for the whole population using this truncated data. The **truncated normal regression model** begins with an underlying population model that satisfies the classical linear model assumptions $y = \boldsymbol{x}\boldsymbol{\beta} + u$, $u \sim N(0, \sigma^2)$. We focus on this model because relaxing the assumptions is difficult.

Assumption MLR.2 of random sampling is violated when from the random draw $(\boldsymbol{x}_i, y_i)$ is observed only if $y_i \leq c_i$, where c_i is the truncation threshold that can depend on the exogenous variables $\boldsymbol{x}_i$. To estimate the β_j, along with σ, we need the distribution $g(y|\boldsymbol{x}_i, c_i) = \frac{f(y|\boldsymbol{x}_i\boldsymbol{\beta},\sigma^2)}{F(c_i|\boldsymbol{x}_i\boldsymbol{\beta},\sigma^2)}$, for $y \leq c_i$, where the numerator is the normal pdf and denominator is the normal cdf, using Bayes' theorem. If we take the log and sum across i, and maximize the result with respect to β_j and σ^2, we obtain the maximum likelihood estimators, leading to consistent, approximately normal estimators. The inference, including standard errors and log-likelihood statistics, is standard.

Example 17.5. We take the previous example and use it as a proxy for truncated data with censored observations removed, though you would not do this for duration data in practice, as you are throwing away information. This is for illustration only.

```
df = woo.data('recid')
y, X = pt.dmatrices('ldurat~workprg+priors+tserved+felon+alcohol+drugs+black'+
                    '+married+educ+age', data=df, return_type='dataframe')
cens = df.cens == 1
y, X= y[~cens], X[~cens]
c = np.log(df.durat[df.cens==0].max())
res_ols = sm.OLS(y, X).fit()
start_params = res_ols.params
start_params['sigma'] = res_ols.resid.std()

class TruncatedReg(smclass.GenericLikelihoodModel):
    def __init__(self, endog, exog, cens):
        self.cens = cens
        super(smclass.GenericLikelihoodModel, self).__init__(endog, exog, missing='none')
    def nloglikeobs(self, params):
        s = params[-1]
        y, X = self.endog, self.exog
        yH = X @ params[:-1]
        a, b = stats.norm.pdf((y - yH) / s), stats.norm.cdf((self.cens-yH)/s)
        return - np.log(a/b/s)
res_trct = TruncatedReg(endog=y, exog=X, cens=c).fit(start_params=start_params,
                            maxiter=10000, method='newton', disp=0)
```

We do a simple linear regression and compare it with the truncated normal regression. We find the coefficients are very similar along with t-stats.

```
pd.DataFrame({'ols': res_ols.params.values, 'trct': res_trct.params[:-1]},
                                                    index=res_ols.params.index)
                ols      trct
Intercept  2.958119  3.277945
workprg    0.111715  0.171869
priors    -0.037804 -0.052120
tserved   -0.006773 -0.009230
felon      0.113735  0.156873
alcohol   -0.203764 -0.296634
drugs      0.024398  0.029511
black     -0.012580 -0.016407
married    0.265447  0.413063
pd.DataFrame({'ols': res_ols.tvalues, 'trct': res_trct.tvalues[:-1]},
                                         index=res_ols.params.index)
                 ols       trct
Intercept  33.533090  22.807301
workprg     1.370140   1.402689
priors     -3.053842  -3.020246
tserved    -3.507343  -3.419426
felon       1.101620   1.010757
alcohol    -2.127430  -2.111980
drugs       0.273841   0.220945
black      -0.154589  -0.134439
married     2.728539   2.702621
print(res_ols.resid.std(), res_trct.params[-1])
0.8989226735595284 1.1032457190120342
print(res_ols.rsquared,np.corrcoef(res_trct.exog @ res_trct.params[:-1],
                                                    res_trct.endog)[0,1]**2)
0.06571833813241124 0.06546697062016792
```

When we plot the predictions we see that the truncated regression model line tilts upwards to account for the missing data, which simple linear regression ignores completely.

```
# effect of tserved
xavg = X.mean()
tserved = list(np.arange(0, 221))
pred_ols, pred_trct = [], []
for x in tserved:   # tserved
    xavg.update({'tserved': x})
    pred_ols.append(np.sum(xavg * res_ols.params))
    pred_trct.append(np.sum(xavg * res_trct.params[:-1]))
pred = pd.DataFrame({'ols': pred_ols, 'trct': pred_trct}, index = tserved)
ax = pred.plot(grid=True)
df[~cens].plot.scatter(ax=ax, x='tserved', y='ldurat', alpha=0.5, grid=True, color='blue')
df[cens].plot.scatter(ax=ax, x='tserved', y='ldurat', alpha=0.5, grid=True, color='red')
```

□

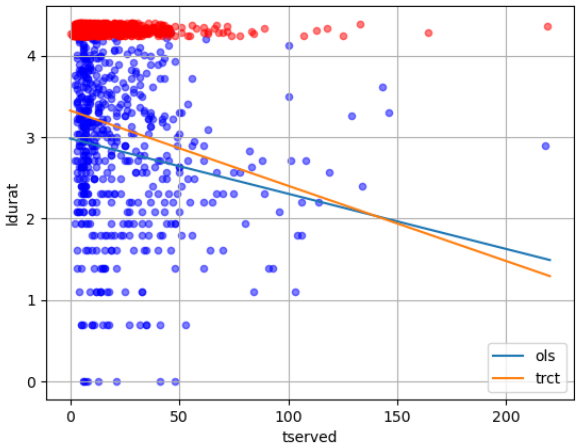

Figure 3: Estimated log duration based on time served with other variables at their mean value.

17.5 Sample selection corrections

Truncated regression is a special case of general problem known as **nonrandom sample selection**. Missing data on dependent or independent variables are generally dropped and leads to bias in our estimators. In **incidental truncation** we do not observe y because of the outcome of another variable, e.g. wage offers are not observed for people currently out of workforce. Nonrandom samples can also arise in panel data where, due to attrition, some people leave the sample.

The key distinction is between exogenous and endogenous sample selection. If our sample is determined solely by an exogenous explanatory variables, we have exogenous sample selection, we have unbiased and consistent estimation. Otherwise we have endogenous sample selection, and OLS is biased and inconsistent. The population model is $y_i = \boldsymbol{x}_i\boldsymbol{\beta} + u_i$, $E(u_i\boldsymbol{x}_i) = 0$. n is the size of a random sample from the population. We use **selection indicator** $s_i = 1$ when we observe all of $(y_i, \boldsymbol{x}_i)$, and $s_i = 0$ otherwise. We are interested in the statistical properties of the **selected sample**, where we have fewer than n observations, say n_1. We can estimate the equation $s_i y_i = s_i\boldsymbol{x}_i\boldsymbol{\beta} + s_i u_i$. OLS estimates for this model are consistent if the zero mean assumption holds $E(su) = 0$, and the zero correlation assumption $E(sx_j u) = 0$. Therefore, in the population, we need u to be uncorrelated with sx_j.

If s is a function only of the explanatory variables, then sx_j is just a function of $\boldsymbol{x}$ and hence sx_j is also uncorrelated with u. This is the case of **exogenous sample selection**, where $s_i = 1$ is entirely determined by $\boldsymbol{x}_i$ or some other random variables. In this case OLS is consistent and unbiased. If we add homoskedasticity assumption $E(u^2|\boldsymbol{x}, s) = E(u^2) = \sigma^2$, then the usual OLS standard errors and test statistics are valid. If s depends on the endogenous variable y_i or u_i, u_i and s_i will not be uncorrelated, even conditional on $\boldsymbol{x}_i$. OLS on the selected samples, here, does not consistently estimate β_j.

The results on consistency of OLS extend to instrumental variables estimation. If the selection is determined entirely by the exogenous variables $\boldsymbol{z}$, or if s depends on other factors that are independent or u and $\boldsymbol{z}$, then 2SLS on the selected sample is generally consistent. We do need to assume that the explanatory and instrumental variables are appropriately correlated in the selected part of the population. It can also be shown that, when selection is entirely a function of the exogenous variables, MLE of a nonlinear model - such as logit or probit model - produces consistent, asymptotically normal estimators, and the usual standard errors and test statistics are valid.

17.5.1 Incidental Truncation

With incidental truncation the rule determining whether we observe y does not depend directly on the outcome of y. For example, the truncation of wage offer is incidental because it depends on another variable, namely, labor force participation. We generally observe all other information about the individual. The usual approach is to explicitly model the selection equation. For $y = \boldsymbol{x}\boldsymbol{\beta} + u$, with $E(u|\boldsymbol{x}) = 0$ we have $s = \mathbf{1}_{\boldsymbol{z}\gamma+\nu\geq 0}$, where $s = 1$ if we observe y and zero otherwise. We assume that $\boldsymbol{x}$ and $\boldsymbol{z}$ are always observed. For $\boldsymbol{z}\boldsymbol{\gamma} = \gamma_0 + \gamma_1 z_1 + \ldots + \gamma_m z_m$ we make the standard assumption that $E(u|\boldsymbol{x}, \boldsymbol{z}) = 0$ and $\boldsymbol{x}$ is a strict subset of $\boldsymbol{z}$. ν is normally distributed and $E(\nu|\boldsymbol{z}, \boldsymbol{x}) = 0$ and (u, ν) is independent of $\boldsymbol{z}$.

We now see that $E(y|\boldsymbol{z}, \nu) = \boldsymbol{x}\boldsymbol{\beta} + E(u|\boldsymbol{z}, \nu) = \boldsymbol{x}\boldsymbol{\beta} + E(u|\nu)$, using the assumptions that $\boldsymbol{x}$ is a subset of $\boldsymbol{z}$ and (u, ν) is independent of $\boldsymbol{z}$. Now if (u, ν) are jointly normal we have $E(u|\nu) = \rho\nu$ for some parameter ρ. Thus, $E(y|\boldsymbol{z}, \nu) = \boldsymbol{x}\boldsymbol{\beta} + \rho\nu$. We can specialize this equation for s to get $E(y|\boldsymbol{z}, s) = \boldsymbol{x}\boldsymbol{\beta} + \rho E(\nu|\boldsymbol{z}, s)$. Using the selection equation we find $E(\nu|\boldsymbol{z}, s = 1) = \frac{\phi(\boldsymbol{z}\boldsymbol{\gamma})}{\Phi(\boldsymbol{z}\boldsymbol{\gamma})} = \lambda(\boldsymbol{z}\boldsymbol{\gamma})$, which is the inverse Mills ratio. This leads to the expression $E(y|\boldsymbol{z}, s = 1) = \boldsymbol{x}\boldsymbol{\beta} + \rho\lambda(\boldsymbol{z}\boldsymbol{\gamma})$. The equation shows that we can estimate $\boldsymbol{\beta}$ using only the selected sample, provided we include the term $\lambda(\boldsymbol{z}\boldsymbol{\gamma})$ as an additional regressor.

If $\rho = 0$ (when u and ν are uncorrelated), OLS of y on $\boldsymbol{x}$ using the selected sample consistently estimates $\boldsymbol{\beta}$, otherwise we have effectively omitted a variable $\lambda(\boldsymbol{z}\boldsymbol{\gamma})$, which is generally correlated with $\boldsymbol{x}$. From the assumptions we have made, s given z follows a probit model: $P(s = 1|\boldsymbol{z}) = \Phi(\boldsymbol{z}\boldsymbol{\gamma})$. Therefore we can estimate $\boldsymbol{\gamma}$ by probit of s_i on $\boldsymbol{z}_i$, using the entire sample and then we can estimate $\boldsymbol{\beta}$. The resulting method is called **Heckit method**.

- Using all n observations, estimate a probit model of s_i on $\boldsymbol{z}_i$ and obtain the estimates $\hat{\gamma}_h$; compute inverse Mills ratio $\hat{\lambda}_i = \lambda(\boldsymbol{z}_i\hat{\gamma})$ for each i for which $s_i = 1$.
- Using the selected sample ($s_i = 1$), run the regression of y_i on $\boldsymbol{x}_i, \hat{\lambda}_i$ to estimate $\hat{\beta}_j$ as consistent and approximately normally distributed.

To test for selection bias we look at the t statistic on $\hat{\lambda}_i$ as a test of $H_0 : \rho = 0$. Under H_0: there is no sample selection problem. If $\boldsymbol{x}$ is not a strict subset of $\boldsymbol{z}$, it can lead to inconsistency. We also need at least on element or $\boldsymbol{z}$ that is not also in $\boldsymbol{x}$ because otherwise the inverse Mills ratio can be well approximated by a linear function and cause multicollinearity and high standard errors for $\hat{\beta}_j$. In other words we need to distinguish sample selection model from a misspecified functional form.

Example 17.6. We apply the sample selection correction to the data on married women. 428 out of 753 women in the sample worked for a wage during the year.

```
df = woo.data('mroz')
y, X = pt.dmatrices('inlf~educ+exper+expersq+nwifeinc+age+kidslt6+kidsge6',
                                        data=df, return_type='dataframe')
res_probit = sm.Probit(y, X).fit()
==============================================================================
Dep. Variable:                   inlf   No. Observations:                  753
Model:                         Probit   Df Residuals:                      745
Method:                           MLE   Df Model:                            7
Date:                Mon, 28 Dec 2020   Pseudo R-squ.:                  0.2206
Time:                        23:28:41   Log-Likelihood:                -401.30
converged:                       True   LL-Null:                       -514.87
Covariance Type:            nonrobust   LLR p-value:                 2.009e-45
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.2701      0.509      0.531      0.595      -0.727       1.267
educ           0.1309      0.025      5.183      0.000       0.081       0.180
exper          0.1233      0.019      6.590      0.000       0.087       0.160
expersq       -0.0019      0.001     -3.145      0.002      -0.003      -0.001
nwifeinc      -0.0120      0.005     -2.484      0.013      -0.022      -0.003
age           -0.0529      0.008     -6.235      0.000      -0.069      -0.036
kidslt6       -0.8683      0.119     -7.326      0.000      -1.101      -0.636
kidsge6        0.0360      0.043      0.828      0.408      -0.049       0.121
==============================================================================
yH = res_probit.fittedvalues
df['imills'] = stats.norm.pdf(yH)/stats.norm.cdf(yH)
y, X = pt.dmatrices('lwage~educ+exper+expersq+imills', data=df, return_type='dataframe')
mask = df.inlf == 1
res_heckit = sm.OLS(y[mask], X[mask]).fit()

                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.5781      0.307     -1.885      0.060      -1.181       0.025
educ           0.1091      0.016      6.987      0.000       0.078       0.140
exper          0.0439      0.016      2.684      0.008       0.012       0.076
expersq       -0.0009      0.000     -1.946      0.052      -0.002    8.49e-06
imills         0.0323      0.134      0.240      0.810      -0.232       0.296

y, X = pt.dmatrices('lwage~educ+exper+expersq', data=df, return_type='dataframe')
mask = df.inlf == 1
res_ols = sm.OLS(y[mask], X[mask]).fit()
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.5220      0.199     -2.628      0.009      -0.912      -0.132
educ           0.1075      0.014      7.598      0.000       0.080       0.135
exper          0.0416      0.013      3.155      0.002       0.016       0.067
expersq       -0.0008      0.000     -2.063      0.040      -0.002   -3.82e-05
==============================================================================
```

We use *educ*, *exper* and $exper^2$ as the exogenous variables for $log(wage)$ as the dependent variable. For the sampling model we also add $nwifeinc$, *age*, *kidslt*6 and *kidsge*6 as the other exogenous variables. We compare the Heckit model with OLS model estimation and find no evidence on $\hat{\lambda}$ to be significant. The coefficients are, hence, quite similar. □

An alternative to the two step estimation method is full MLE which requires the joint distribution of y and s.

18 Advanced Time Series Topics

18.1 Infinite distributed lag models

Let $\{(y_t, z_t) : t = \ldots, -2, -1, 0, 1, 2, \ldots\}$ be a bivariate, partially observed, time series process. An **infinite distributed lag (IDL) model** relating y_t to current and past values of z is $y_t = \alpha + \delta_0 z_t + \delta_1 z_{t-1} + \ldots + u_t$, where the sum on lagged z extends back to the indefinite past. In order for this to make sense, the lag coefficients, δ_j, must tend to zero as $j \to \infty$, i.e. the impact of z_{t-j} on y_t must eventually become small as j gets large. We only observe finite history of data so we need some restriction on the δs to estimate them.

The *impact propensity* is simply δ_0. For a unit impulse at $t = 0$ (i.e. $z_s = 0$ except for $s = 0$ it is 1) we have $E(y_h) = \alpha + \delta_h$ for $h \geq 0$. It follows that δ_h is the change in $E(y_h)$ given a one-unit, temporary change in z at time zero. This means that a temporary change in z has no long-run effect: $E(y_h) = \alpha + \delta_h \to \alpha$ as $h \to \infty$. The lag distribution, which is δ_h plotted as a function of h, shows the expected path that future y follow given the one-unit, temporary increase in z.

The *long-run propensity* is the sum of all the lag coefficients: $LRP = \delta_0 + \delta_1 + \delta_2 + \ldots$, where we assume the infinite sum is well defined. This is the effect if there is a permanent unit change at $t = 0$, i.e. $z_t = 1$ for $h \geq t \geq 0$. We have $y_h = \alpha + \delta_0 + \ldots + \delta_h + u_h$, where $h \geq 0$ is any horizon. Hence, $E(y_h) = \alpha + \delta_0 + \ldots + \delta_h$. As $h \to \infty$, this becomes $\alpha + LRP$. Thus, the LRP measures the long-run change in the expected value of y given a one-unit, permanent increase in z. We are assuming strict exogeneity above by assuming that the change in z during any time period had no effect on the expected value of u_t, i.e. $E(u_t | \ldots, z_{t-2}, z_{t-1}, z_t, z_{t+1}, z_{t+2}, \ldots) = 0$. This rules out any feedback from y_t to future z, because z_{t+h} must be uncorrelated with u_t for $h > 0$, which might be an unrealistic assumption. A weaker assumption is $E(u_t | z_t, z_{t-1}, \ldots) = 0$, which allows z_t to be dependent on past y. In certain case this assumption is sufficient to estimate δ_j. Importantly, none of these assumptions disallow serial correlation.

The simplest way to make the model depend on an infinite number of lags is the **geometric (Koyck) distributed lag**. Here, $\delta_j = \gamma\rho^j$, $|\rho| < 1$, $j = 0, 1, 2, \ldots$, ensuring $\delta_j \to 0$ as $j \to \infty$. The impact propensity (IP) in the GDL is simply $\delta_0 = \gamma$. The long run propensity (LRP) is $\gamma/(1 - \rho)$. If we plug this into the original equation for y_t and y_{t-1}, multiply the second by ρ and take a difference, we get an estimable model $y_t = \alpha_0 + \gamma z_t + \rho y_{t-1} + \nu_t$, where $\alpha_0 = (1 - \rho)\alpha$ and $\nu_t = u_t - \rho u_{t-1}$. ν_t and y_{t-1} are generally correlated (because

u_{t-1} is correlated to y_{t-1}). Without further assumptions, OLS estimation gives inconsistent estimates of γ and ρ.

Under strict exogeneity z_t is uncorrelated with u_t and u_{t-1}, and therefore ν_t. Thus, if we can find a suitable instrument variable for y_{t-1}, we can estimate the equation by IV. By assumption, u_t and u_{t-1} are both uncorrelated with z_{t-1}, so ν_t is uncorrelated with z_{t-1}. If $\gamma \neq 0$, z_{t-1} and y_{t-1} are correlated even after partialling out z_t. Thus, (z_t, z_{t-1}) can serve as suitable instruments. Standard errors need to be adjusted for serial correlation in the $\{v_t\}$.

Alternatively, under the weaker exogeneity assumption $E(u_t|z_t, z_{t-1}, \ldots) = 0$, if we suppose that $\{u_t\}$ follows the AR(1) model $u_t = \rho u_{t-1} + e_t$ with $E(e_t|z_t, y_{t-1}, z_{t-1}, \ldots) = 0$, with ρ being the same as in the equation to be estimated, we can do estimation. This renders the desired equation dynamically complete in the form $y_t = \alpha_0 + \gamma z_t + \rho y_{t-1} + e_t$, which can give consistent, asymptotically normal estimators of the parameters by OLS. If e_t satisfies the homoskedasticity assumption $Var(e_t|z_t, y_{t-1}) = \sigma_e^2$, the usual inference applies. Thus we can estimate LRP as $\widehat{LRP} = \hat{\gamma}/(1-\hat{\rho})$. A simple test to check the validity of the AR(1) model is to test $u_t = \lambda u_{t-1} + e_t$ using Lagrange multipliers and test $H_0 : \lambda = \rho$. This can be naturally extended to multiple explanatory variables.

The GDL is a special case of what is generally called a **rational distributed lag (RDL) model**. A simple example is obtained by adding a lag of z to get $y_t = \alpha_0 + \gamma_0 z_t + \rho y_{t-1} + \gamma_1 z_{t-1} + \nu_t$, where $\nu_t = u_t - \rho u_{t-1}$ as before. Repeated substitution gives $y_t = \alpha + \gamma_0 z_t + (\rho\gamma_0 + \gamma_1) z_{t-1} + \rho(\rho\gamma_0 + \gamma_1) z_{t-2} + \rho^2(\rho\gamma_0 + \gamma_1) z_{t-3} + \ldots + u_t$, were we again need the assumption $|\rho| < 1$. The impact propensity is γ_0, which the coefficient on z_{t-h} is $\rho^{h-1}(\rho\gamma_0 + \gamma_1)$ for $h \geq 1$. This model allows the impact propensity to differ in sign from the other lag coefficients, even if $\rho > 0$. The LRP is given by $LRP = (\gamma_0 + \gamma_1)/(1-\rho)$.

Example 18.1. We estimate both the basic geometric and the rational distributed lag models by applying OLS for dependent variable $log(invpc)$ after a linear time trend has been removed from housing investment. For z_t, we use the growth in the price index. This allow us to estimate how residential price inflation affects the movements in housing investment around its trend.

```
df = woo.data('hseinv')
y, X = pt.dmatrices('linvpc~t+gprice+linvpc_1', data=df, return_type='dataframe')
res_geo = sm.OLS(y, X).fit()

df['gprice_1'] = df.gprice.shift(1)
y, X = pt.dmatrices('linvpc~t+gprice+linvpc_1+gprice_1', data=df, return_type='dataframe')
res_rat = sm.OLS(y, X).fit()
print_compare({'geometric': res_geo, 'rational': res_rat}, keys=['coeff', 'tval'])
     geometric  rational
r2       0.583     0.672
ar2      0.549     0.634
llf     33.404    37.355
dfr     37.000    35.000
dfm      3.000     4.000
```

```
              coeff                tval
          geometric rational geometric rational
Intercept    -0.561   -0.364    -4.653   -2.662
gprice        3.093    3.293     3.266    3.337
gprice_1        NaN   -2.932       NaN   -2.977
linvpc_1      0.340    0.542     2.548    3.512
t             0.005    0.003     2.901    1.787

g, r = res_geo.params, res_rat.params
pd.DataFrame({
    'geometric': {'IP': g.gprice, 'LRP': (g.gprice)/(1-g.linvpc_1)},
    'rational': {'IP': r.gprice, 'LRP': (r.gprice+r.gprice_1)/(1-r.linvpc_1)}
})
     geometric  rational
IP    3.092913  3.292701
LRP   4.685041  0.785987

res_rat.f_test(['gprice+gprice_1=0'])
<F test: F=array([[0.05855855]]), p=0.81020120402428, df_denom=35, df_num=1>
```

The geometric distribution lag model is clearly rejected by the data, as $gprice_1$ is very significant. The adjusted R-squareds also show that RDL model fits much better.

The two models gives very different estimates of the long-run propensity. The incorrect estimate using GLD implies 4.7% increase in housing investment on 1% increase in residential price inflation, which seems unlikely. The rational distributed lag model estimate this to be much more plausible 0.79%. We cannot reject the null hypothesis $H_0 : \gamma_0 + \gamma_1 = 0$ at any reasonable significance level (p-value = 0.81), so there is no evidence that the LRP is different from zero. □

18.2 Testing for unit roots

The simplest approach to test for a unit root begins with an AR(1) model $y_t = \alpha + \rho y_{t-1} + e_t$, $t = 1, 2, \ldots$, where y_0 is the observed initial value. We assume $E(e_t|y_{t-1}, y_{t-2}, \ldots) = 0$. $\{e_t\}$ is said to be a **martingale difference sequence** with respect to $\{y_{t-1}, y_{t-2}, \ldots\}$. This assumption is automatically satisfied if we have $\{e_t\}$ which is i.i.d. with zero mean and is independent of y_0. If $\{y_t\}$ follows the AR(1) model, it has a unit root if $\rho = 1$. If $\alpha = 0$ and $\rho = 1$ it is a random walk without drift; while $\alpha \neq 0$ with $\rho = 1$ indicated random walk with drift. The null hypothesis $H_0 : \rho = 1$ is generally compared against the one-sided alternative $H_1 : \rho < 1$. $\rho > 1$ implies y_t is explosive.

A convenient equation for carrying out the unit root test is to subtract y_{t-1} from both sides of the equation with $\theta = \rho - 1$ to get $\delta y_t = \alpha = \theta y_{t-1} + e_t$, which makes it dynamically complete. We test $H_0 : \theta = 0$ versus $H_1 : \theta < 0$. Under H_0, y_{t-1} is $I(1)$, and so the usual CLT that underlies the asymptotic standard normal distribution for the t statistic does not apply. The asymptotic distribution of t statistic under H_0 has come to be known as the **Dickey-Fuller distribution**. We can use the usual t statistic for $\hat{\theta}$ against the tabulated

appropriate critical values. The resulting test is called **Dickey-Fuller (DF) test** for unit root. We reject the null hypothesis $H_0 : \theta = 0$ against $H_1 : \theta < 0$ if $t < c$, where c is one of the negative critical values.

```
from statsmodels.tsa.stattools import adfuller
from scipy.stats import norm
adf = adfuller(np.random.randn(1000))[-2]
pd.DataFrame({'student-t': norm.ppf([0.01, 0.05, 0.1]),
              'Dickey-Fuller': adf.values()}, index=adf.keys())

     student-t  Dickey-Fuller
1%   -2.326348      -3.436919
5%   -1.644854      -2.864440
10%  -1.281552      -2.568314
```

Example 18.2. The quarterly series of 3 month T-bill rates is tested for unit root as follows:

```
df = woo.data('intqrt')
y, X = pt.dmatrices('cr3~r3_1', data=df, return_type='dataframe')
res_ols = sm.OLS(y,X).fit()
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.6253      0.261      2.398      0.018       0.109       1.142
r3_1          -0.0907      0.037     -2.473      0.015      -0.163      -0.018
res = sm.tsa.stattools.adfuller(df.r3, maxlag=0, autolag=None, regresults=True)
>> (-2.4731506198108675,
   0.1221444771592618,
  {'1%': -3.4846672514209773,
   '5%': -2.8853397507076006,
   '10%': -2.5794629869786503},
```

The standard error reported here cannot be used for DF test. The estimate of ρ is $\hat{\rho} = 1 + \theta = 0.909$. To test whether it is statistically less than 1, we compare the t statistic of -2.473 with 10% critical value of -2.57 and, therefore, fail to reject $H_0 : \rho = 1$ against $H_1 : \rho < 1$ at 10% significance level. □

When we fail to reject a unit root, we should only conclude that the data do not provide strong evidence against H_0. We should be extremely cautious to use variables that have or are close to having unit roots as explanatory variables. One solution is to use the first difference of these variables.

We also need to test for unit roots in models with more complicated dynamics. We can add p lags of Δy_t to the equation to account for the dynamics in the process $\Delta y_t = \alpha + \theta y_{t-1} + \gamma_1 \Delta y_{t-1} + \ldots + \gamma_{pt-p} + e_t$. We carry out the t test on $\hat{\theta}$. This extended version of the Dickey-Fuller test is usually called the **augmented Dickey-Fuller test**. The critical values and rejection rules are the same as before. The reliability of critical values depend on

the dynamics being completely modeled so lag length need to approximately right. The t statistics on the lagged changes have approximate t distributions so F statistics can be used for joint significance of any group of terms Δy_{t-h}.

Example 18.3. We apply Augmented Dickey-Fuller on annual US inflation rates based on CPI, allowing one lag of Δinf_t.

```
df = woo.data('phillips')
df = df[(df.year>=1948) & (df.year<=1996)]
df['cinf_1'] = df.cinf.shift(1)
df = df.dropna()
y, X = pt.dmatrices('cinf~inf_1+cinf_1', data=df, return_type='dataframe')
res_adf = sm.OLS(y, X).fit()
                coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      1.3608      0.517      2.634      0.012       0.319       2.402
inf_1         -0.3103      0.103     -3.021      0.004      -0.517      -0.103
cinf_1         0.1384      0.126      1.095      0.280      -0.116       0.393
res = sm.tsa.stattools.adfuller(df.cinf, maxlag=1, autolag=None, regresults=True)
>> -7.530314102227495,
   3.59304741980547e-11,
   {'1%': -3.584828853223594,
   '5%': -2.9282991495198907,
   '10%': -2.6023438271604937}
```

The t statistic for the unit root test is -3.01 which makes us reject H_0 at 5% statistical significance of -2.93. The estimate of ρ is about 0.69 providing strong evidence against unit root. Δinf_{t-1} has a t statistic is about 1.095 so we don not need to include it. If we drop it we get similar results. □

For series that have clear time trends, we need to modify the unit root test. A trend-stationary process - which has a linear trend in its mean but is $I(0)$ about its trend - can be mistaken for a unit root process if we do not control for a time trend in the Dickey-Fuller regression. To allow for the time trend we change the basic equation to $\Delta y_t = \alpha + \delta t + \theta y_{t-1} + e_t$, where again the null hypothesis is $H_0 : \theta = 0$ and the alternative is $H_1 : \theta < 0$. The critical values change for this test because detrending the unit root process tend to make it look more like $I(0)$ process therefore we require larger magnitude for the t statistic in order to reject H_0.

Example 18.4. We apply the unit root test with a time trend to the US GDP data. We test whether $log(GDP_t)$ has a unit root. This series has a pronounced trend that looks roughly linear. We include a single lag of $\Delta log(GDP_t)$, which is simply the growth in GDP to account for the dynamics.

```
df = woo.data('inven')
df['t'] = df.index + 1
df['lgdp_1'] = df.gdp.apply(np.log).shift(1)
df['ggdp_1'] = df.ggdp.shift(1)
```

```
y, X = pt.dmatrices('ggdp~t+lgdp_1+ggdp_1', data=df, return_type='dataframe')
ols = sm.OLS(y, X).fit()
                coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      1.6509      0.666      2.477      0.019       0.292       3.010
t              0.0059      0.003      2.177      0.037       0.000       0.011
lgdp_1        -0.2096      0.087     -2.421      0.022      -0.386      -0.033
ggdp_1         0.2638      0.165      1.601      0.120      -0.072       0.600
res = sm.tsa.stattools.adfuller(df.gdp.apply(np.log), maxlag=1,
                    autolag=None, regression='ct', regresults=True)
>> -2.4207328814761624,
 0.36865584571358057,
 {'1%': -4.243765510204081,
  '5%': -3.5443646122448977,
  '10%': -3.2046503498542274}
```

We get $\hat{\rho} = 1 - 0.21 = 0.79$. We cannot reject a unit root in the log of GDP since the t statistic of -2.42, is well above the 10% critical value of -3.2. The t statistic on $gGDP_{t-1}$ is 1.6, which is almost significant at the 10% level against a two-sided alternative. If we omit the time trend, there is much less evidence against H_0 as $\hat{\theta} = -0.023$ and $T_\theta = -1.92$. Here the estimate of ρ is much closer to one, but this is misleading due to the omitted time trend.

```
y, X = pt.dmatrices('ggdp~lgdp_1+ggdp_1', data=df, return_type='dataframe')
ols = sm.OLS(y, X).fit()
                coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.2149      0.100      2.139      0.040       0.010       0.420
lgdp_1        -0.0227      0.012     -1.908      0.065      -0.047       0.002
ggdp_1         0.1672      0.168      0.997      0.326      -0.174       0.509
res = sm.tsa.stattools.adfuller(df.gdp.apply(np.log), maxlag=1,
                                autolag=None, regresults=True)
-1.908213836453613,
 0.3282069837956807,
 {'1%': -3.6327426647230316,
  '5%': -2.9485102040816327,
  '10%': -2.6130173469387756}
```

It is wrong to compare the t statistic on the time trend with the critical value from a standard normal t distribution, to see whether the time trend is significant, unless $|\rho| < 1$. □

18.3 Spurious regression

In cross-sectional environment, the phrase 'spurious correlation' is used to describe a situation where two variables are related through their connection with a third variable; once controlled for the third variable the correlation between the first two disappears. This can happen in time series contexts with I(0) variables and we can find spurious relationship between time series that have trends. Provided the series are weakly dependent about their time trends, the problem is effectively solved by including a time trend in the regression

model.

When we are dealing with integrated processes of order one, there is an additional complication. Even if two series have means that are not trending, a simple regression involving two independent I(1) series will often result in a significant 4 statistic. For $x_t = x_{t-1} + a_t$ and $y_t = y_{t-1} + e_t$, for $t = 1, 2, \ldots$ and $x_0 = y_0 = 0$ with $\{a_t\}$ and $\{e_t\}$ being independent, iid with zero mean and unit variance, we get significant non-zero coefficients for $\hat{y}_t = \hat{\beta}_0 + \hat{\beta}_1 x_t$ using the usual t statistic for $\hat{\beta}_1$.

```
# spurious regression problem
N = 10000   # iterations
n = 10      # sample size
rho, tst = [], []
for i in range(N):
    df = pd.DataFrame(np.zeros((n, 2)), columns=['x', 'y'])
    df.iloc[1:,0] = np.random.randn(n-1)
    df.iloc[1:,1] = np.random.randn(n-1)
    df = df.cumsum()
    res = sm.OLS(df.y, df.x).fit()
    rho.append(res.params.x)
    tst.append(res.tvalues.x)
df = pd.DataFrame({'rho': rho, 'tstat': tst})
(df.tstat.apply(np.abs)>2).mean()
# 10: 0.5692, 50: 0.797, 250: 0.9043
```

Because y_t and x_t are independent, we would hope that $plim\hat{\beta}_1 = 0$ with insignificant t statistic. This however shows significance and is called **spurious regression problem**. The R-squared is nonstandard random variable and is large with high probability. Including a time trend does not really change the conclusion. For t statistic $\hat{\beta}_1$ to have an approximate standard normal distribution in large samples, $\{u_t\}$ should have mean zero, serially uncorrelated. But under $H_0 : \beta_1 = 0$ we have $y_t = \beta_0 + u_t$, and since $\{y_t\}$ is a random walk starting at $y_0 = 0$, this holds only if $\beta_0 = 0$ and $u_t = y_t = \sum_{j=1}^{t} e_t$. In other words $\{u_t\}$ should be a random walk under H_0. This clearly violates even the asymptotic version of the Gauss-Markov assumptions for the regression to be consistent.

Regression an $I(1)$ dependent variable on an $I(1)$ independent variable can be informative, if these variables are related in a precise sense.

18.4 Cointegration and error correction models

Differencing I(1) variables limits the scope of the questions that we can answer. The notion of **cointegration** due to Engle and Granger, makes regression involving I(1) variables potentially meaningful. If $\{y_t\}$ and $\{x_t\}$ are two I(1) processes, then in general $y_t - \beta x_t$ is an I(1) process. Nevertheless, it is possible that for some $\beta \neq 0$, $y_t = \beta x_t$ is an I(1) process, which means it has constant mean, constant variance, and auto-correlations that depend

only on the time distance between any two variables in the series, and it is asymptotically uncorrelated. If such β exists, we say that y and x are cointegrated, and we call β the cointegration parameter, which is unique up to a multiplicative constant.

Cointegration in many examples, has an economic interpretation. For example 6 month and 3 month T-bill rates are cointegrated, with the difference between them, the spread, a I(0) process due to the no-arbitrage argument. It also means that y_t and x_t have a long-run relationship. For a hypothesized value of β we can calculate $s_t = y_t - \beta x_t$ and run a DF or ADF test with the null hypothesis that y_t and x_t are not conitegrated. When β is unknown, it turns out that if y_t and x_t are cointegrated, the OLS estimator $\hat{\beta}$ from the regression $\hat{y}_t = \hat{\alpha} + \hat{\beta} x_t$ is consistent for β. However, the null hypothesis states that the two series are not cointegrated, thus under H_0, we are running a spurious regression. **Engle-Granger test** makes possible to tabulate critical values even when β is estimated, where we apply the Dickey-Fuller or augmented Dickey-Fuller test to the residuals $\hat{u}_t = y_t - \hat{\alpha} = \hat{\beta} x_t$, with critical values account for estimation of β and hence larger.

```
from statsmodels.tsa.stattools import coint
coint(np.random.randn(10000), np.random.randn(10000))
>>> array([-3.89753563, -3.33674114, -3.04487419])
coint(np.random.randn(10000), np.random.randn(10000), trend='ct')
>>> array([-4.32916438, -3.78152128, -3.4970183 ])
```

In the basic test we run regression of $\Delta\hat{u}_t$ on $\hat{u}_{t-1}$ and compare the t statistic on $\hat{u}_{t-1}$ to the desired critical value. We can add lags of $\Delta\hat{u}_t$ to account for serial correlation and use augmented test. If y_t and x_t contain drift terms, we can test for cointegration by adding a trend term like $y_t = \hat{\alpha} + \hat{\eta} t + \hat{\beta} x_t$ and applying the usual DF or augmented DF test to the residual of $\hat{u}_t$, though with a different set of critical values. A finding of cointegration leaves open the possibility that $y_t - \beta x_t$ has a linear trend, but not I(1).

Hence, if y_t and x_t are not cointegrated, a regression of y_t on x_t is spurious and tells are nothing meaningful. We can then, still run a regression between Δy_t and Δx_t including lags, acknowledging that it is a different relationship. If, however, y_t and x_t are cointegrated, we can use this to specify more general dynamic models.

Example 18.5. The static regression results in levels and first differences of general fertility rate (gfr) and real value of the personal tax exemption (pe) are notably different. In regression in levels, with a time trend included, we get a coefficient on pe equal to 0.187 (se=0.035) and $R^2 = 0.5$. In first difference without a trend, the coefficient on pe is -0.043 (se=0.028) and $R^2 = 0.032$. Such a discrepancy btween level and changes regression suggests that we should test for cointegration.

```
from statsmodels.tsa.stattools import adfuller, coint
df = woo.data('fertil3')
y, X = pt.dmatrices('gfr~pe+t', data=df, return_type='dataframe')
lev_ols = sm.OLS(y, X).fit()
```

```
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    109.9302      3.475     31.632      0.000     102.997     116.863
pe             0.1867      0.035      5.391      0.000       0.118       0.256
t             -0.9052      0.109     -8.305      0.000      -1.123      -0.688
y, X = pt.dmatrices('cgfr~cpe', data=df, return_type='dataframe')
dif_ols = sm.OLS(y,X).fit()
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.7848      0.502     -1.563      0.123      -1.786       0.217
cpe           -0.0427      0.028     -1.504      0.137      -0.099       0.014

         lev      dif
r2     0.500    0.032
ar2    0.486    0.018
llf -291.679 -201.972
dfr   69.000   69.000
dfm    2.000    1.000
```

To check if gfr and pe are $I(1)$ processes, we apply augmented DF tests with a single lagged change and a linear time trend, each yield t statistics of about -1.47. We then take the residuals from the regressio of gfr on t and pe and apply the augmented DF test with one lag, we obtain a t statistic on $\hat{u}_{t-1}$ of -2.43, which is much larger than the 10% critical value of -3.50. Therefore, we must conclude that there is little evidence of cointegration between gfe and pe, even allowing for seperate trends.

```
adfuller(df.gfr, maxlag=1, autolag=None, regression='ct', regresults=True)
# -1.4740696555005233, 0.8378425969614807,
# {'1%': -4.094292755102041, '5%': -3.4751863673469385, '10%': -3.164853411078717}
adfuller(df.pe, maxlag=1, autolag=None, regression='ct', regresults=True)
# -1.4712603858502238, 0.8387663909806136,
# {'1%': -4.094292755102041, '5%': -3.4751863673469385, '10%': -3.164853411078717}
adfuller(lev_ols.resid, maxlag=1, autolag=None, regresults='True')
# -2.424833604373085, 0.13484962063454553,
# {'1%': -3.527425868804664 7, '5%': -2.903810816326531, '10%': -2.589320408163265 3}
coint(df.gfr, df.pe, maxlag=1, trend='ct')
# -2.437541547330495,  0.5547559388459501,
# array([-4.55214424, -3.91691727, -3.59748361]))
```

Thus, it is very likely that the earlier regression result we obtained in levels suffer from the spurious regression problem. When we use first differences and allowed for two lags - we find an overall positive and significant long-run effect of Δpe on Δgfr.

```
y, X = pt.dmatrices('cgfr~cpe+cpe_1+cpe_2', data=df, return_type='dataframe')
dif_ols = sm.OLS(y,X).fit()
R-squared:                       0.232
Adj. R-squared:                  0.197
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
```

```
Intercept     -0.9637      0.468     -2.060      0.043     -1.898     -0.029
cpe           -0.0362      0.027     -1.352      0.181     -0.090      0.017
cpe_1         -0.0140      0.028     -0.507      0.614     -0.069      0.041
cpe_2          0.1100      0.027      4.092      0.000      0.056      0.164
```

□

When y_t and x_t are I(1) and cointegrated, we can write $y_t = \alpha + \beta x_t + u_t$, where u_t is a zero mean I(0) process. Generally $\{u_t\}$ contains serial correlation, but it does not effect the consistency of OLS but the usual inference procedures do not necessarily apply: OLS is not asymptotically normally distributed, and the t statistic for $\hat{\beta}$ does not necessarily have an approximate t distribution. The notion of cointegration implies nothing about the relationship between $\{x_t\}$ and $\{u_t\}$ - they could be arbitrarily correlated, neither does it restrict the serial dependence in $\{u_t\}$. If $\{x_t\}$ is strictly exogenous we can use the heteroskedasticity and serial correlation robust OLS estimates. If not - it can be fixed. Because x_t is I(1), the proper notion of strict exogeneity is that u_t is uncorrelated with Δx_s, for all t and s. We can accomplish that by writing $u_t = \eta + \phi_0 \Delta x_t + \phi \Delta x_{t-1} + \phi_2 \Delta x_{t-2} + \gamma_1 \Delta x_{t+1} + \gamma_2 \Delta x_{t+2} + e_t$, where by construction e_t is uncorrelated with each Δx_s. This gives, $y_t = \alpha_0 + \beta x_t + \phi_0 \Delta x_t + \phi \Delta x_{t-1} + \phi \Delta x_{t-2} + \gamma_1 \Delta x_{t+1} + \gamma_2 \Delta x_{t+2} + e_t$. x_t is now strictly exogenous and the distribution of t statistic for $\hat{\beta}$ is approximately normal. If u_t is uncorrelated with all Δx_s, $s \neq t$, we simply use $y_t = \alpha_0 + \beta x_t + \phi_0 \Delta x_t + e_t$. Endogeneity does not cause inconsistency, but we need it for asymptotically normal t statistic. This is called **leads and lags estimator** or β. Any serial correlation in e_t can be dealt with computing a serially-robust standard error for $\hat{\beta}$ by standard AR(1) correction such as Cochrane-Orcutt.

Example 18.6. We estimate the cointegration parameter between the 6 month and 3 month T-bill rates and test $H_0 : \beta = 1$.

```
df = woo.data('intqrt')
df['cr3m1'] = df.cr3.shift(1)
df['cr3m2'] = df.cr3.shift(2)
df['cr3p1'] = df.cr3.shift(-1)
df['cr3p2'] = df.cr3.shift(-2)
y, X = pt.dmatrices('r6~r3+cr3+cr3m1+cr3m2+cr3p1+cr3p2', data=df,
                                          return_type='dataframe')
llols = sm.OLS(y, X).fit(cov_type='HC3')
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.0651      0.063      1.040      0.298      -0.058       0.188
r3             1.0382      0.012     89.195      0.000       1.015       1.061
cr3           -0.0531      0.029     -1.864      0.062      -0.109       0.003
cr3m1         -0.0611      0.024     -2.516      0.012      -0.109      -0.014
cr3m2         -0.0438      0.039     -1.112      0.266      -0.121       0.033
cr3p1         -0.0036      0.019     -0.189      0.850      -0.041       0.033
cr3p2          0.0124      0.041      0.299      0.765      -0.069       0.093
```

With two leads and lags the estimate of heteroskedasticity and serially-correlated robust β is $\hat{\beta} = 1.038$ with the standard error of 0.012, giving t statistic for $H_0 : \beta = 1$ of $(1.038 -$

$1)/0.012 \approx 3.17$, which is a strong statistical rejection of H_0. Of course 1.038 may not be economically different from 1. □

If y_t and x_t are I(1) processes and are not cointegrated, we might estimate a dynamic model in the first difference, e.g. using a rational distributed lag model $\Delta y_t = \alpha_0 + \alpha_1 \Delta y_{t-1} + \gamma_0 \Delta x_t + \gamma_1 \Delta x_{t-1} + u_t$. If y_t and x_t are cointegrated with parameter β then we have additional I(0) variables that we can include $s_t = y_t - \beta x_t$, e.g. $\Delta y_t = \alpha_0 + \alpha_1 \Delta y_{t-1} + \gamma_0 \Delta x_t + \gamma_1 \Delta x_{t-1} + \delta(y_{t-1} - \beta x_{t-1}) + u_t$. The term $\delta(y_{t-1} - \beta x_{t-1})$ is called the error correction term and is an example of **error correction model**. It allows us to study the short-run dynamics in the relationship between y and x. If we know β estimation is easy by including s_{t-1} directly. Otherwise we can use **Engle-Granger two-step procedure** by using $\hat{s} = y_{t-1} - \hat{\beta} x_{t-1}$, where $\hat{\beta}$ can be an estimate of β, like standard OLS or leads and lags estimator. Fortunately, we can ignore the preliminary estimation of β, asymptotically. This implies that the asymptotic efficiency of the estimator of the parameters in the error correction model is unaffected by the estimate we use.

Example 18.7. We look at $hy6_t$, the three month holding yield from buying a six month T-bill at time $t-1$ and selling at time t as a three month T-bill, versus $hy3_{t-1}$, the three month holding yield from buying a three-month T-bill at time $t-1$. The expectation hypothesis implies that the slope coefficient should not be statistically different from one. Let us check for unit roots first.

```
df = woo.data('intqrt')
adfuller(df.hy6.dropna(), maxlag=1, autolag=None, regresults=False)
# -3.3840976105336105, 0.011513077173333101
adfuller(df.hy3.dropna(), maxlag=1, autolag=None, regresults=False)
# -2.249606717526927, 0.18873591342956964
```

We find that $hy3$ is indeed I(1) at 10% significance level, while $hy6$ is I(1) at 1% significance level. Assuming both are I(1) processes, expectation hypothesis implies that the cointegration parameter should be 1. Cointegration test shows clear dependence with cointegration parameter estimated to be 1.1 with $H_0 : \beta = 1$ with t stats of $(1.1043 - 1)/0.039 \approx 2.67$ and hence we fail to reject the null hypothesis at a 10% critical value of -3.08.

```
data = df[['hy6', 'hy3_1']].dropna()
coint(data.hy6, data.hy3_1, maxlag=1, trend='c')
# -10.90914002681641, 1.3823687595494553e-18,
# array([-3.98846223, -3.3866712 , -3.07939668]))
y, X = pt.dmatrices('hy6~hy3_1', data=data, return_type='dataframe')
reg = sm.OLS(y, X).fit()
                coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0579      0.070     -0.828      0.409      -0.196       0.081
hy3_1          1.1043      0.039     27.986      0.000       1.026       1.182
```

Thus we assume $\beta = 1$ and estimate an error correction model $\Delta hy6_t = \alpha_0 + \gamma_0 \Delta hy3_{t-1} + \delta(hy6_{t-1} - hy3_{t-2}) + u_t$ giving,

```
df['s_1'] = df['hy6_1']-df['hy3_1'].shift(1)
y, X = pt.dmatrices('chy6~chy3_1+s_1', data=df, return_type='dataframe')
reg = sm.OLS(y,X).fit()
                coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.0898      0.043      2.105      0.037       0.005       0.174
chy3_1         1.2184      0.264      4.622      0.000       0.696       1.740
s_1           -0.8400      0.244     -3.441      0.001      -1.323      -0.357
```

The error correction coefficient is negative and very significant, though not statistically different from -1. □

18.5 Forecasting

Suppose that at time t we want to forecast the outcome of y at time $t+1$ or y_{t+1}. Let I_t denote information that we can observe at time t. This **information set** includes y_t, earlier values of y, and often other variables dates at time t or earlier. We can combine this information in innumerable ways to forecast y_{t+1}. Once we specify the loss, there is always a best way to do it. Let f_t denote the forecast of y_{t+1} made at time t. We call f_t a **one-step-ahead forecast**. The forecast error is $e_{t+1} = y_{t+1} - f_t$, which we observe once the outcome on y_{t+1} is observed. The most common measure of loss is the squared error e_{t+1}^2 or the absolute value of prediction error $|e_{t+1}|$, examples of **loss function**. We do not not know e_{t+1} at time t - it is a random variable, because y_{t+1} is a random variable. It is natural to choose the forecast to minimize the expected squared forecast error, given I_t, i.e. $E(e_{t+1}^2|I_t) = E[(y_{t+1} - f_t)^2|I_t]$ which is minimized by the conditional expectation $E(y_{t+1}|I_t)$ for the squared error loss. Thus, we seek excepted value of y_{t+1} as the forecast.

If $\{y_t\}$ is a martingale difference sequence and we take I_t to be the observed past of y then $E(y_{t+1}|I_t) = 0$ for all t; the best prediction of y_{t+1} at time t is always zero! Now, an i.i.d. sequence with zero mean is a martingale difference sequence. Further, a process $\{y_t\}$ is a **martingale** if $E(Y_{t+1}|y_t, y_{t-1}, \ldots, y_0) = y_t$ for all $t \geq 0$. The predicted value of y for the next period is always the value of y for this period. For a more complicated example of **exponential smoothing** we take $E(y_{t+1}|I_t) = \alpha y_t + \alpha(1-\alpha)y_{t-1} + \ldots + \alpha(1-\alpha)^t y_0$, where $0 < \alpha < 1$ because the weights on the lagged y decline to zero exponentially. For $t \geq 1$ we can obtain the forecast as $f_t = \alpha y_t + (1-\alpha)f_{t-1}$. Exponential smoothing is suitable only for very specific time series. Regression models are more flexible.

The general issue that arise in forecasting y_{t+h} at time t, where h is any positive integer, are similar. If we use expected squared forecast error as our measure of loss, the best predictor is $E(y_{t+h}|I_t)$. When dealing with a **multiple-step-ahead forecast**, we use the notation $f_{t,h}$ to indicate the forecast of y_{t+h} made at time t.

If using the static model $y_t = \beta_0 + \beta_1 z_t + u_t$, at time $t+1$, for known values of β_0 and β_1 we have $E(y_{t+1}|I_t) = \beta_0 + \beta_1 z_{t+1}$. This forecast of y_{t+1} at time t is called a **condi-**

tional forecast because it assumes we know the value of z_{t+1} at time t! Except for dummy variables, time trends and seasonal dummies this is rare. Sometimes, we wish to generate conditional forecasts for several values of z_{t+1}. This also assumes that u_t has no serial autocorrelation which can be false in practice. If z_{t+1} is not known at t we can not include it in I_t and then we have $E(y_{t+1}|I_t) = \beta_0 + \beta_1 E(x_{t+1}|I_t)$. Thus, we must first forecast z_{t+1} before we can forecast y_{t+1}. This is called **unconditional forecast**.

Often, moving beyond static models, it makes sense to specify the model that depends only on the lagged values of y and z, such as $y_t = \delta_0 + \alpha_1 y_{t-1} + \gamma_1 z_{t-1} + u_t$, with $E(u_t|I_{t-1}) = 0$, where I_{t-1} contains y and z dated at time $t-1$ and earlier. The forecast of y_{t+1} at time t is $\delta_0 + \alpha_1 y_t + \gamma_1 z_t$, with plug in values giving us the forecast once the parameters are known.

18.5.1 One-step-ahead forecasts

Let n be the sample size. The **point forecast** of y_{n+1} is $\hat{f}_n = \hat{\delta}_0 + \hat{\alpha}_1 y_n + \hat{\gamma}_1 z_n$, where the parameters are estimated by OLS. The forecast error, as calculated at time $n+1$ is $\hat{e}_{n+1} = y_{n+1} - \hat{f}_n$. The **forecast interval** is essentially the same as prediction interval and is still approximately valid, provided u_t given I_{t-1} is normally distributed with zero means and constant variance. If $se(\hat{f}_n)$ is the standard error of the forecast and $\hat{\sigma}$ the standard error of the regression, then we can obtain $\hat{f}_n$ and $se(\hat{f}_n)$ as the intercept and its standard errors from the regression of y_t on $(y_{t-1} - y_n)$ and $(z_{t-1} - z_n)$, for $t = 1, 2, \ldots, n$. $se(\hat{e}_{n+1}) = \sqrt{se(\hat{f}_n)^2 + \hat{\sigma}^2}$ with the 95% forecast interval as $\hat{f}_n \pm se(\hat{e}_{n+1})$. Since $se(\hat{f}_n)$ is proportional to $\frac{1}{\sqrt{n}}$, $se(\hat{f}_n)$ is usually small relative to the uncertainty in the error u_{n+1}, as measured by $\hat{\sigma}$.

Example 18.8. We look to forecast the US civilian unemployment rate in 1997 based on the data till 1996. We fit an AR(1) model for *unem* and then a second model with one year lagged inflation added to it.

```
data = woo.data('phillips')
d97 = data[data.year==1997].iloc[0]
df = data[(data.year>=1948)&(data.year<=1996)]
y, X = pt.dmatrices('unem~unem_1', data=df, return_type='dataframe')
reg1 = sm.OLS(y, X).fit()
                coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      1.5717      0.577      2.723      0.009       0.410       2.733
unem_1         0.7324      0.097      7.559      0.000       0.537       0.927

y, X = pt.dmatrices('unem~unem_1+inf_1', data=df, return_type='dataframe')
reg2 = sm.OLS(y, X).fit()
sigma = reg2.mse_resid**0.5
                coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      1.3038      0.490      2.663      0.011       0.318       2.290
unem_1         0.6470      0.084      7.721      0.000       0.478       0.816
inf_1          0.1836      0.041      4.458      0.000       0.101       0.267
```

The lagged inflation rate is very significant with t statistic of 4.6 and significantly higher R-squared. Nevertheless, this does not necessarily mean that the second equation will produce a better forecast for 1997. The estimated predictions for the two models are 5.53 and 5.35 respectively, with the real value being 1.9. To find the Forecasts for 1997, we use $unem$ and inf from 1996 and regress $unem_t$ on $(unem_t - 5.4)$ and $(inf_{t-1} - 3.0)$, we obtain 5.35 as the intercept and $se(\hat{f}_n) = 0.137$. Since $\hat{\sigma} = 0.883$, we have $se(\hat{e}_{n+1}) = 0.894$ and the 95% CI are $[3.6, 7.1]$. This interval encapsulates 4.9.

```
# prediction
print(reg1.predict([1, d97.unem_1])[0],
           reg2.predict([1, d97.unem_1, d97.inf_1])[0], d97.unem)
# 5.526451978948073 5.348467851140485 4.900000095367432

# standard error
X.unem_1 -= d97.unem_1
X.inf_1 -= d97.inf_1
reg2P = sm.OLS(y, X).fit()
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      5.3485      0.137     39.172      0.000       5.073       5.623
unem_1         0.6470      0.084      7.721      0.000       0.478       0.816
inf_1          0.1836      0.041      4.458      0.000       0.101       0.267

print(reg2P.params.Intercept, reg2P.bse.Intercept, sigma)
# 5.348467851140487 0.1365394411200658 0.8829847193628465
se = np.sqrt(reg2P.bse.Intercept**2+ sigma**2)
# 0.8934791735735448

# confidence interval
print(reg2P.params.Intercept - 1.96 * se, reg2P.params.Intercept + 1.96 * se)
# [3.59724867093634 7.099687031344635]
```

□

As a professional forecaster we must produce a forecast for every time period. It is advisable to re-estimate the parameters using the new data available at each time step.

```
# static prediction
data = woo.data('phillips')
df = data[(data.year>=1948)&(data.year<=1996)]
y, X = pt.dmatrices('unem~unem_1+inf_1', data=df, return_type='dataframe')
reg2 = sm.OLS(y, X).fit()
pred_static = {i: reg2.predict([1, *data[data.year==i][['unem_1', 'inf_1']].
                        values[0]])[0] for i in range(1997, 2004)}

# running prediction
pred_running = {}
actual = {}
for t in range(1997, 2004):
    df = data[(data.year >= 1948) & (data.year < t)]
```

```
    y, X = pt.dmatrices('unem~unem_1+inf_1', data=df, return_type='dataframe')
    reg2 = sm.OLS(y, X).fit()
    pred_running[t] = reg2.predict([1, *data[data.year==t][['unem_1', 'inf_1']].
                                    values[0]])[0]
    actual[t] = data[data.year==t].unem.iloc[0]

df = pd.DataFrame({'static': pred_static, 'running': pred_running, 'actual': actual})
df.plot(grid=True, style='.-')
```

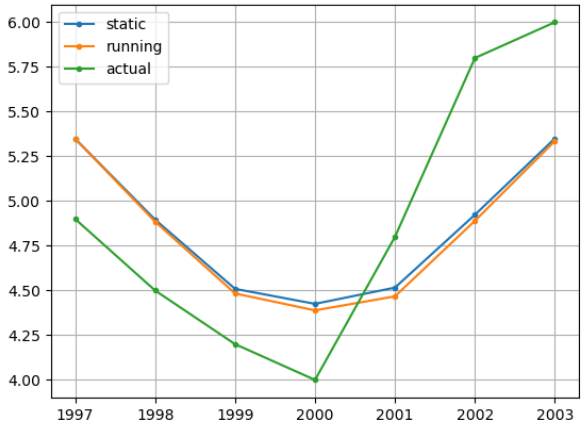

Figure 4: Static versus running predictions.

The model we estimated earlier with two variables on the right hand side is a type of **vector autoregressive (VAR) model** - we model a vector in terms of its own past. If we have two series y_t and z_t, a vector autoregression consists of equations that look like $y_t = \delta_0+\alpha_1 y_{t-1}+\gamma_1 z_{t-1}+\alpha_2 y_{t-2}+\gamma_2 z_{t-2}+\ldots$ and $z_t = \eta_0+\beta_1 y_{t-1}+\rho_1 z_{t-1}+\beta_2 y_{t-2}+\rho_2 z_{t-2}+\ldots$. VAR models can be useful for forecasting. They also allow us to test whether, after controlling for past values of y, past values of z help to forecast y_t. Generally, we say that z **Granger causes** y if $E(y_t|I_{t-1}) \neq E(y_t|J_{t-1})$, where I_{t-1} contains past information on y and z, and J_{t-1} contains only information on past y. It has nothing to say about contemporaneous causality between y and z, so it does not allow us to determine whether z_t is an exogenous variable in an equation relating y_t to z_t. This is also why the notion of Granger causality does not apply in pure cross-sectional contexts.

We can easily test the null hypothesis that z does not Granger cause y. Say $E(y_t|y_{t-1}, y_{t-2}, \ldots)$ depends on only three lags: $y_t = \delta_0 + \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \alpha_2 t_{t-3} + u_t$, $E(u_t|y_{t-1}, y_{t-2}, \ldots) = 0$. Under the null hypothesis that Z does not Granger cause y, any lags of z that we add to the equation should have zero population coefficients. If we add z_{t-1} we can simply do a t test, if we add two lags of z, we can do an F test for joint significance of z_{t-1} and z_{t-2}. We first start by estimating an autoregressive model for y and performing t and F tests to determine how many lags of y should appear. We then test for the lags of z and test of Granger causality.

In the previous example we started with AR(1) model for $unem$ and then added inf_{t-1} to find it very significant. Therefore, inflation Granger causes unemployment.

There is an extended definition of Granger causality that is often useful. Let $\{w_t\}$ be a third series, then z Granger causes y conditional on w if $E(y_t|I_{t-1}) \neq E(y_t|J_{t-1})$, where I_{t-1} contains past information on y, z, and w, while J_{t-1} contains past information on y and w. It is certainly possible that z Granger causes y, but z does not Granger cause y conditional on w. A test of the null that z does not Granger cause y conditional on w is obtained by testing for significance of lagged z in a model for y that also depends on lagged y and lagged w.

To choose among competing models - which variable to include, how many lags, logs, levels, first differences - we have **in-sample criteria**, like R-squared and especially adjusted R-squared, and **out-of-sample criteria**, which is more suitable for forecasting. We use the first part (n observations) of a sample to estimate the parameters of the model and the latter part (m observations) to gauge its forecasting capabilities. This mimics what we should have to do in practice if we did not know the future values of the variables. Let $\hat{f}_{n+h}$ be the one-step-ahead forecast of y_{n+h+1} for $h = 0, 1, \ldots, m-1$. The m forecast errors are $\hat{e}_{n+h+1} = y_{n+h+1} = \hat{f}_{n+h}$. Two measures are common to quantify the forecasting ability of the model. **root mean square error (RMSE)**, $RMSE = \sqrt{\frac{1}{m}\sum_{h=0}^{m-1} \hat{e}^2_{n+h+1}}$, is the sample standard deviation of the forecast errors, with lower values preferred. **Mean absolute error (MAE)**, which is the average of the absolute forecast errors $MAE = \frac{1}{m}\sum_{h=0}^{m-1} |\hat{e}_{n+h+1}|$, with smaller MAE preferred.

Example 18.9. We look at the previous example and find the RMSE and MAE for the two models and compare them.

```
data = woo.data('phillips')
df = data[(data.year>=1948)&(data.year<=1996)]
y, X = pt.dmatrices('unem~unem_1', data=df, return_type='dataframe')
reg1 = sm.OLS(y, X).fit()
pred_ar1 = {i: reg1.predict([1, *data[data.year==i][['unem_1']].values[0]])[0]
                                    for i in range(1997, 2004)}
y, X = pt.dmatrices('unem~unem_1+inf_1', data=df, return_type='dataframe')
reg2 = sm.OLS(y, X).fit()
pred_ar2 = {i: reg2.predict([1, *data[data.year==i][['unem_1', 'inf_1']].values[0]])[0]
                                    for i in range(1997, 2004)}

pred = pd.DataFrame({'1': pred_ar1, '2': pred_ar2})
actual = pd.Series({i: data[data.year==i].unem.values[0] for i in range(1997, 2004)})

RMSE = pred.sub(actual, axis=0).pow(2).mean().apply(np.sqrt)
MAE = pred.sub(actual, axis=0).apply(np.abs).mean()
print(RMSE, MAE)
>>> [0.57611992 0.52175432] [0.54201404 0.48419453]
```

clearly the second model is better on both criteria. □

Rather than using the first n observations to estimate the parameters of the model, we can re-estimate the models each time we add a new observation and use the new model to forecast the next itme period.

18.5.2 Multi-step-ahead forecasts

The error variance increases with the forecast horizon. If $\{y_t\}$ follows an AR(1) model $y_t = \alpha + \rho y_{t-1} + u_t$, $E(u_t|I_{t-1}) = 0$, $I_{t-1} = \{y_{t-1}, y_{t-2}, \ldots\}$, and $\{u_t\}$ has a constant variance σ^2 conditional on I_{t-1}, then at time $t+h-1$, our forecast of y_{t+h} is $\alpha + \rho y_{t+h-1}$ and the forecast error is u_{t+h}, thus a one-step-ahead forecast variance is simply σ^2. To find multiple-step-ahead forecast, we have, by repeated substitution $y_{t+h} = (1+\rho+\ldots+\rho^{h-1})\alpha + \rho^h y_t + \rho^{h-1}u_{t+1} + \rho^{t-2}u_{t+2} + \ldots + u_{t+h}$. At time t, $E(y_{t+h}|I_t) = (1+\rho+\ldots+\rho^{h-1})\alpha + \rho^h y_t$ and the forecast error is $e_{t,h} = \rho^{h-1}u_{t+1} + \rho^{h-2}u_{t+2} + \ldots + u_{t+h}$. This sum of uncorrelated random variables gives $Var(e_{t,h}) = \sigma^2(\rho^{2(h-1)} + \rho^{2(h-2)} + \ldots + \rho^2 + 1)$. This forecast variance increases with h and converges to $\sigma^2/(1-\rho^2)$, since $\rho^2 < 1$. For a random walk $\rho = 1$ and the forecast variance grows without bounds.

Once ρ is estimated the forecast of y_{n+h} at time n is $\hat{f}_{n,h} = (1+\hat{\rho}+\ldots+\hat{\rho}^{h-1})\hat{\alpha} + \hat{\rho}^h y_n$. The standard error of $\hat{f}_{n,h}$ is usually small compared with the standard deviation of the error term which can be estimated as $\hat{\sigma}\sqrt{\hat{\rho}^{2(h-1)} + \hat{\rho}^{2(h-2)} + \ldots + \hat{\rho}^2 + 1}$ where $\hat{\sigma}$ is the standard error of the regression from the AR(1) estimation. This can be used to find the required CI.

A useful, but less traditional, approach is to estimate a different model for each forecast horizon. For example, if we wish to forecast y two periods ahead, we can assume that $E(y_{t+2}|I_t) = \alpha_0 + \gamma_1 y_t$ which we can estimate by regressing y_t on y_{t-2} consistently and approximately normal, even if the errors in this equation contain serial correlation. The forecast of y_{n+2} at time n is simply $\hat{f}_{n,2} = \hat{\alpha}_0 + \hat{\gamma}_1 y_n$. The serial-correlation adjusted standard error ($\hat{\sigma}$) of this regression is the error on the forecast. This goes to zero as n gets large while the variance of the error is constant. We can, thus, approximate interval by using $\hat{f}_{n,2} \pm 1.96\hat{\sigma}$, but this ignores the error in $\hat{\alpha}_0$ and $\hat{\gamma}_1$.

For more complicated autoregressive models multiple-step-ahead forecasting We can take the example of AR(2) process at time n, we wish to forecast y_{n+2}. Now, $y_{n+2} = \alpha + \rho_1 y_{n+1} + \rho_2 y_n + u_{n+2}$, so $E(y_{n+2}|I_n) = \alpha + \rho_1 E(y_{n+1}|I_n) + \rho_2 y_n$. That is $\hat{f}_{n,2} = \alpha + \rho_1 f_{n,1} + \rho_2 y_n$, and so the two-step-ahead forecast at time n can be obtained once we get the one-step-ahead forecast. Once the coefficients are estimated via OLS we have $\hat{f}_{n,2} = \hat{\alpha} + \hat{\rho}_1 \hat{f}_{n,1} + \hat{\rho}_2 y_n$ with $\hat{f}_{n,1} = \hat{\alpha} + \hat{\rho}_1 y_n + \hat{\rho}_2 y_{n-1}$, which we can compute at time n. For any $h > 2$, obtaining any h-step-ahead forecast for an AR(2) model is easy to fin in a recursive manner: $\hat{f}_{n,h} = \hat{\alpha} + \hat{\rho}_1 \hat{f}_{n,1} + \hat{\rho}_2 \hat{f}_{n,h-2}$.

Similarly, we can obtain multiple-step ahead forecasts for VAR models. Suppose we have $y_t = \delta_0 + \alpha_1 y_{t-1} + \gamma_1 z_{t-1} + u_t$ and $z_t = \eta_0 + \beta_1 y_{t-1} + \rho_1 z_{t-1} + \nu_t$. To forecast y_{n+1} at time n we use $\hat{f}_{n,1} = \hat{\delta}_0 + \hat{\alpha}_1 y_n + \hat{\gamma}_1 z_n$. And to forecast z_{n+1} at time n we use $\hat{g}_{n,1} = \hat{\eta}_0 + \hat{\beta}_1 y_n + \hat{\rho}_1 z_n$. For two step ahead forecast of y at time n we have $E(Y_{n+2}|I_n) = \delta_0 + \alpha_1 E(y_{n+1}|I_n) + \gamma_1 E(z_{n+1}|I_n)$,

thus the forecast is $\hat{f}_{n,2} = \hat{\delta}_0 + \hat{\alpha}_1 \hat{f}_{n,1} + \hat{\gamma}_1 \hat{g}_{n,1}$. Thus the two-step ahead forecast of y depends on the one-step ahead forecast for y and z. Generally, we can build up multiple-step-ahead forecasts of $y$by using the recursive formula $hat f_{n,h} = \hat{\delta}_0 + \hat{\alpha}_1 \hat{f}_{n,h-1} + \hat{\gamma}_1 \hat{g}_{n,h-1}$, for $h \geq 2$.

Example 18.10. To forecast unemployment two years out we write $f_{t,2} = \alpha + \rho f_{t,1} + \beta E(inf_{t+1}|I_t)$, so we need a model for inflation. We estimate a simple AR(1) model to get

```
data = woo.data('phillips')
df = data[(data.year>=1948)&(data.year<=1996)]
y, X = pt.dmatrices('I(inf)~inf_1', data=df, return_type='dataframe')
mod_inf = sm.OLS(y, X).fit(cov_type='HC3')
              Coef.    Std.Err.      z      P>|z|    [0.025   0.975]
------------------------------------------------------------------
Intercept     1.2767     0.4497   2.8391   0.0045   0.3953   2.1580
inf_1         0.6653     0.1333   4.9925   0.0000   0.4041   0.9264
inf_1997 = mod_inf.predict([1, data[data.year==1996].inf.values[0]])[0]
>>> 3.27242621

y, X = pt.dmatrices('unem~unem_1+inf_1', data=df, return_type='dataframe')
mod_une = sm.OLS(y, X).fit()
              Coef.    Std.Err.      t      P>|t|    [0.025   0.975]
------------------------------------------------------------------
Intercept     1.3038     0.4897   2.6625   0.0107   0.3175   2.2901
unem_1        0.6470     0.0838   7.7206   0.0000   0.4782   0.8158
inf_1         0.1836     0.0412   4.4576   0.0001   0.1006   0.2665
unem_1007 = mod_une.predict([1, *data[data.year==1996][['unem', 'inf']].values[0]])
>>> 5.34846785
```

Now, we estimate $E(inf_{1997}|I_t) = 1.2767 + 0.6653 * inf_{1996} = 3.2724$. Also, we can estimate $E(unem_{1997}|I_t) = 1.3038 + 0.6470 unem_{1996} + 0.1836 inf_{1996} = 5.3485$. Using this we obtain the forecase for $unem_{1998}$ using 1996 data as $\widehat{unem}_{1998} = 1.3038 + 0.647\widehat{unem}_{1997} + 0.1836\widehat{inf}_{1997} \approx 5.37$. The one-step ahead forecast of $unem_{1998}$ using 1997 data was about 4.90, which is much closer to the real value of 4.5. □

18.5.3 Trending, seasonal and integrated processes

Suppose that $\{y_t\}$ has a linear trend but is unpredictable around the trend $y_t = \alpha + \beta t + u_t$, $E(u_t|I_{t-1}) = 0$, $t = 1, 2, \ldots$. To forecast y_{n+h} at time n for any $h \geq 1$ we use $E(y_{n+h}|I_n) = \alpha + \beta(n+h)$. The forecast error variance is simply $\sigma^2 = Var(u_t)$, which after OLS estimation gives a forecast of $\hat{f}_{n,h} = \hat{\alpha} + \hat{\beta}(n+h)$

Example 18.11. To forecast monthly imports of Chinese barium chloride to the united states form China we first fit the model with 131 observations. The forecast 6 months later is $249.56 + 5.15 * 137 \approx 955.11$, measured as short tons.

```
data = woo.data('barium')
y, X = pt.dmatrices('chnimp~t', data=data, return_type='dataframe')
reg = sm.OLS(y, X).fit()
```

```
              Coef.    Std.Err.     t      P>|t|     [0.025     0.975]
-------------------------------------------------------------------
Intercept   249.5563   51.1360   4.8802   0.0000   148.3824   350.7302
t             5.1467    0.6723   7.6558   0.0000     3.8166     6.4768
```

The series and its estimated trend line is shown here. □

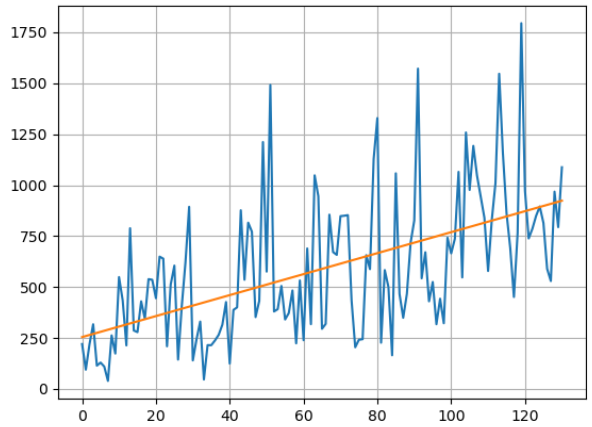

Figure 5: US imports of Chinese barium chloride and its estimated linear trand line.

Most times series are better characterized as having a constant grown rate, suggesting $log(y_t)$ follows a linear time trend. The model then is $\log(y_t) = \hat{\alpha} + \hat{\beta}t$, $t = 1, 2, , n$. Exponentiating the linear term to get the forecast is wrong! We must properly account for the error implicit in the nonlinear transformation. The simplest way to do this is to use the n observations and regress y_t against $exp(\widehat{log(y_t)})$ without an intercept. The $\hat{\gamma}$ be the slope coefficient on $exp(\widehat{log(y_t)})$, then the forecast of y in period $n + h$ is simply $\hat{f}_{n,h} = \hat{\gamma} exp\left(\hat{\alpha} + \hat{\beta}(n + h)\right)$.

Example 18.12. We use first 687 weeks of data on NYSE index to fit a trend model for $log(price_t)$ giving a secular growth of 0.2% per week. we then regress *price* on the exponentiated fitted value to get $\hat{\gamma} = 1.018$.

```
data = woo.data('nyse')
y, X = pt.dmatrices('np.log(price)~t', data=data, return_type='dataframe')
res = sm.OLS(y[:-4], X[:-4]).fit()
Coef.    Std.Err.      t       P>|t|     [0.025    0.975]
-----------------------------------------------------------------------
Intercept     3.7819     0.0085   446.2046   0.0000    3.7653    3.7986
t             0.0019     0.0000    88.7133   0.0000    0.0019    0.0019
```

To make the forecast 4 weeks in advance which is the last data point we calculate the prediction to be 165.40, with the actual value being 164.25, showing some over prediction.

```
gmm = sm.OLS(data.price.iloc[:-4], res.predict(X[:-4]).apply(np.exp)).fit()
          Coef.        Std.Err.          t          P>|t|        [0.025       0.975]
------------------------------------------------------------------------------------
x1        1.0181         0.0044       232.6045       0.0000       1.0095       1.0267

print(gmm.params.x1 * np.exp(res.predict([1, 691])))
>>> 165.40201330235257
```

□

Trend models should be used with caution, particularly when forecasting far into the future, and for integrated series that have drift. In fact, we should not use a linear trend to forecast a random walk with drift. Deterministic trends can also produce poor forecast if the trend parameters are estimated using old data and the process has a subsequent shift in the trend line. This can be mitigated by using the most recent data available to obtain the trend line parameters. Processes with deterministic seasonality can be also used for forecasting with proper dummy variables included.

with I(1) processes present, we need to be careful as well. There are two approaches to producing forecasts for I(1) processes. The first is to impost a unit root. The first is to impose a unit root. For a one-step-ahead forecast, we forecast the change in y, Δy_{t+1}, given information through time t. Then, $y_{t+1} = \Delta y_{t+1} + y_t$ implies $E(y_{t+1}|I_t) = E(\Delta y_{t+1}|I_t) + y_t$. Typically an AR model is used for Δy_t, or a vector autoregression. This can be easily extended to multiple-step-ahead forecast by noting $y_{n+h} = \Delta y_{n+h} + \Delta y_{n+h-1} + \ldots + \Delta y_{n+1} + y_n$.

The second approach to forecasting I(1) variables is to use a general AR or VAR model for $\{y_t\}$. This does not impose the unit root. For example if we use an AR(2) model, $y_t = \alpha + \rho_1 y_{t-1} + \rho_2 y_{t-2} + u_t$, with $\rho_1 + \rho_2 = 1$ we obtain the difference AR(1) equation back! But we can estimate this level equation directly from OLS. This allows us to test the t statistics of the coefficient ρ_2 to seek its significance as well.

Example 18.13. We estimate an AR(2) model for the general fertility rate. The R-squared which is very high should be ignored since the variable has a unit root as confirmed by the augmented Dickey-Fuller test. Nevertheless, we can use this model to make forecasts as we do in the following.

```
data = woo.data('fertil3')
y, X = pt.dmatrices('gfr~gfr_1+gfr_2', data=data[data.year <=1979],
                                    return_type='dataframe')
reg = sm.OLS(y, X).fit(cov_type='HC3')
Adj. R-squared:     0.947
                Coef.   Std.Err.      z      P>|z|     [0.025  0.975]
-------------------------------------------------------------------
Intercept       3.2157    2.4350   1.3206  0.1866  -1.5569  7.9882
gfr_1           1.2721    0.1866   6.8158  0.0000   0.9063  1.6379
gfr_2          -0.3114    0.1901  -1.6376  0.1015  -0.6841  0.0613
```

```
adfuller(y, maxlag=1, autolag=None, regresults=False)
# -1.2838260247532227, 0.6365390612795003

y, X = pt.dmatrices('gfr~gfr_1+gfr_2', data=data[data.year >1979],
                                        return_type='dataframe')
print(pd.DataFrame({'pred':reg.predict(X), 'actual': y['gfr']}))
         pred     actual
67  68.303383  68.400002
68  69.300525  67.400002
69  67.654783  67.300003
70  67.838964  65.800003
71  65.961987  65.400002
```

□

In context of VAR models with I(1) variables, if $\{y_t\}$ and $\{z_t\}$ both are I(1) processes and are cointegrated, we have more stationary stable variables in the information set than Δy_t and Δz_t and their lags - namely the lags of $y_t - \beta z_t$, where β is the cointegration parameter. A simple error-correction model is $\Delta y_t = \alpha_0 + \alpha_1 \Delta y_{t-1} + \gamma_1 \Delta z_{t-1} + \delta_1(y_{t-1} - \beta z_{t-1}) + e_t$, $E(e_t|I_{t-1}) = 0$. To forecast y_{n+1} we use observations up through n to estimate the cointegration parameter β, and then estimate the parameters of the error correction model by OLS. Forecasting Δy_{n+1} is then simply plugging in the values. Adding it to y_n gives us the forecast for y_{n+1}. In general, error-correction models can economize on parameters, i.e. they are parsimonious than VARs in levels.

19 Carrying out an Empirical Project

Assuming a hypothesis of interest is available and the data has been reasonably cleaned, we follow the following check list for econometric analysis:

- OLS assumptions are satisfied or not - error uncorrelated with the explanatory variables.
- Potential sources of endogeneity - namely measurement error and simultaneity - are not a serious problem.
- Functional form decision - logarithmic form, squares, dummy variables, interaction terms.
- For cross-sectional analysis heteroskedasticity need to be dealt with.
- For time series we need to decide - levels, time trends, differencing, seasonality, distributed lag dynamics.
- Misspecification analysis such as omitted variables.
- Instrument variables to solve the problem of endogeneity, including omitted variables, errors-in-variables, and simultaneity.

- Sensitivity analysis, e.g. changing continuous variable to binary variable without changing outcome much.
- outliers sensitivity.
- While using Panel data method - pooling versus random effects and fixed effects
- Apply various econometric models to fish out the assumptions that are likely to be false.
- data mining - using outcome of tests to respecify the model violates the random sampling assumption and induces serious bias. If a variable is statistically significant in only a small fraction of the models estimated, it is quite likely that the variable has no effect in the population.