Erasmus School of Economics

Statistics for Data Science
Lecture 5

Dennis Fok (Econometric Institute)

September – October, 2025

Erasmus University Rotterdam

Before next time

- Reread Chapter 4
- No new material for next week
- Reconsider/finish the in-class assignments
- Work on the take home assignment
- Final assignment (part 1 is due on Sunday)

Take home assignment

- Use the Murder rate data (Murder as dependent variable)
- Start with four independent variables: Income, Population, Illiteracy, Frost
- Do some experimentation
  - If a variable is not significant, try to remove it
    - Does the R^2 go up or go down? What about Adjusted R^2?
    - What about AIC?
- Ultimate goal: find the best model (the lowest AIC)
- Finally: check the model assumptions using the diagnostics plot
  $\rightarrow$ What do you conclude?

Plan for Lecture 5

1. Regression diagnosis
   - Normality, Independence, Linearity, Homoskedasticity
   - Multicollinearity
2. Outliers and Model Correction
3. Variable/model selection

Advanced diagnostics

The diagonostic plots are not the only way to look at the assumptions.

Let's look at/revisit the assumptions one-by-one:

1. Linearity (`sm.graphics.plot_ccpr` and `sm.stats.diagnostic.linear_reset`)
2. Normality (`qqresid` from `olsdiagnostics.py`)
3. Homoskedasticity (`sm.stats.diagnostic.het_breuschpagan`)
4. No autocorrelation (`sm.stats.stattools.durbin_watson`)
5. No multicollinearity (`sm.stats.outliers_influence.variance_inflation_factor`)

(`import statsmodels.api as sm`)

Linearity [A3]: $y_i = X_i\beta + \varepsilon_i$ holds

In the basic tool: residuals versus fitted plot

More detailed check: residual versus *each* X_i

- Component plus residual plots

$$\text{plot } e_i + \hat{\beta}_j X_{ji} \quad \text{versus} \quad X_{ji}$$

- Compare it to the observations and local fit (deviation from straight line is a bad sign)
- Use `plot_ccpr(m)` or `plot_ccpr_grid(m)` from `statsmodels.api.graphics` (with `m` a fitted model)

RESET test (from `statsmodels.stats.diagnostic`)

1. Take residuals from candidate model
2. Try to explain these using original variables and squared *fitted values* (and fitted^3, etc)
3. If model specification correct → no added value
4. Test statistic based on (joint) significance test of fitted terms

`sm.stats.diagnostic.linear_reset(m, power=2)`

Normality test [A6]

Directly testing the residuals for normality is not *really* a good idea:

- Even if $\varepsilon_i \sim N(0, \sigma^2)$, $e_i = y_i - \hat{y}_i$ is not iid normal due to
  - estimation error, and
  - all e_i are based on same b estimate
- If ε_i are iid $N(0, \sigma^2)$ → after some standardization e_i has t_{n-k-1} distribution

A fair QQ-plot

- *Studentized residuals* versus the Student-*t* distribution
- In Python implemented in `olsdiagnostics`
  `qqresid(i)`, where `i = OLSInfluence(m)` from `statsmodels.stats.outliers_influence`

Assignment

In-class Assignment 5.1 – Part I (see "starter code" on Canvas)

- Use the Murder rate data (code on Canvas adds 'labels' to observations)
- Use a QQ-plot to investigate whether "Murder" is normally distributed
  → What do you conclude?
  → Does this matter for a linear model explaining Murder?
- Create a model explaining Murder using Population, Income, Frost, and Illiteracy
- Create the basic diagnostic plot for this model
- What do you conclude?
- Continue with this model and consider the results of
  - `plot_ccpr`
  - `linear_reset`
  - `qqresid`

  → What are your conclusions?

Homoscedasticity [A4]

A4 *Homoskedasticity:* $\mathrm{E}[\varepsilon_i^2] = \mathrm{Var}(\varepsilon_i) = \sigma^2$

In the basic tool: standardized residual versus fitted value

A formal test: *Breusch–Pagan test*

- Main idea: regress e_i^2 on the X
- H_0: constant variances (homoskedasticity)
- H_a: non-constant variances (heteroskedasticity)
- Python: `sm.stats.diagnostic.het_breuschpagan(i.resid, m.model.data.exog)`
  - `m` is a fitted OLS result (also in all slides below!)
  - `i` is corresponding OLSInfluence object (also in all slides below!)
  - `m.model.data.exog` gives the variables used to explain e_i^2 (all original variables from the model)

    → Can also test with other variables!

Is heteroskedasticity bad?

Heteroskedasticity...

- does **not** cause a bias in parameter estimates
- does not lead to major problems with OLS
- does lead to **wrong standard errors**

→ can reduce estimation uncertainty using weighted least squares (not discussed)

We can estimate the correct variance matrix Var[b], and use it:

- Step 1 `hcRobust = m.get_robustcov_results(cov_type="HC3")`
  → **H**eteroskedasticity **C**onsistent covariance matrix
- Step 2 `hcRobust.summary()`

No autocorrelation assumption [A5]

A5 *No autocorrelation:* $\mathrm{E}[\varepsilon_i \varepsilon_j] = 0$ for $i \neq j$

(No test available in the basic diagnostics!)

When is checking for no correlation needed?

- This is sometimes better justified by "nature" than a test
- Cross–sectional data: judge by "nature"
- The part *"auto–"* comes from time series data

→ This is mainly needed for time series data

Durbin Watson test [A5] (to be used for time series)

The test statistic

$$d = \frac{\sum_{t=2}^{T}(e_t - e_{t-1})^2}{\sum_{t=1}^{T} e_t^2} \approx 2(1 - \mathrm{Cor}(\varepsilon_t, \varepsilon_{t-1}))$$

Theoretical idea

- Autocorrelation: $\mathrm{Cor}(\varepsilon_t, \varepsilon_{t-1}) = r$ should be 0
- If $r = 0$ (no autocorrelation), $d \approx 2$

`sm.stats.stattools.durbin_watson(m.resid)`

- Reported autocorrelation: should be close to zero
- D-W statistic: should be close to 2

Formal tests are also available (see later courses)

Multicollinearity [A7]

For multivariate regression we have the assumption

A7 No perfect linear relationship in X

What can go wrong if there is "a strong linear relation":

- Full collinearity: model is not identified
  - If $x_{1i} = 2x_{2i}$ for all observations
  - Indifference across $y_i = x_{1i} + \varepsilon_i$, $y_i = 2x_{2i} + \varepsilon_i$, $y_i = 3x_{1i} - x_{2i} + \varepsilon_i$, etc
- Multicollinearity: close to full collinearity
  - Very unstable estimate
  - Insignificant coefficients

Check multicollinearity

The idea to check: *variance inflation factor*

- Regress one explanatory variable on the others
- R_j^2: R^2 when using X_j as the dependent variable

$$VIF_j = \frac{1}{1 - R_j^2}$$

- Rule of thumb: $VIF > 4$ (some use $VIF > 10$)
- Note: with enough data we do not need to worry about near multicollinearity

In Python: `sm.stats.outliers_influence.variance_inflation_factor(x, ind)`
→ Give VIF for variable number `ind` in the data matrix `x`
Use `x = m.model.data.exog` to get full set of variables (variable 0 is the intercept)

Assignment

In-class Assignment 5.1 – Part II

Continue with the earlier model

- Test for no autocorrelation
  $\rightarrow$ What do you conclude & does this test make sense?
- Test for homoskedasticity in the model $\rightarrow$ What is your conclusion?
- Calculate heteroskedasticity consistent standard errors
  $\rightarrow$ Do you obtain the same significance conclusions?
- Calculate the VIFs for the included variables
  $\rightarrow$ Do we need to worry about multicollinearity?

Unusual Observations

Unusual observations

"Unusual" comes in three flavors

- Outlier: bad prediction
- High-leverage points: unusual independent variables (X)
- Influential observations: severely affect model estimates

Differences and relations

- High-leverage points are **not** determined by the dependent variable
- Outliers and high-leverage points are not the same
- Influential observations are a combination of outlier and high-leverage points

Outlier detection

- Outliers
  - Definition: Large prediction error
  - The simplest way to check presence: Q-Q plot
- Testing in a formal way
  - Can we directly use a t-test on the largest studentized residual?
  - Yes, but some correction on the p-value is needed!
    $\rightarrow$ Bonferroni correction (use a stricter threshold for the test)
- In Python `m.outlier_test()`

High-leverage points

- High-leverage points
  - Definition: unusual because of "extreme" independent variables
    → The dependent variable is not used for detection
- The hat matrix
  Recall the "Most important formula":

$$b = (X'X)^{-1}X'y$$

The fitted values: $\hat{y} = Xb = \underbrace{[X(X'X)^{-1}X']}_{H}\,y$

- Leverage: values on the diagonal of H (="own weight in the prediction")
  - Property: sum to k, the number of regressors
- High-leverage: leverage higher than 2-3 times of average (k/n)
- `i.hat_matrix_diag`

Influential observations

- Influential observations
  - Definition: unusual because of the *impact on estimated coefficients*
- Influence is measured by Cook's distance

$$D_i = \frac{\text{Stud-res}_i^2}{k}\,\frac{\text{leverage}_i}{1-\text{leverage}_i}$$

- Clearly, it combines the previous two measures
- Influential observation
  - Quite influential: $D_i > 1$
  - Should be investigated: $D_i > \frac{4}{n-k}$
- In Python `i.cooks_distance[0]`
- To make a Cook's distance graph
  `i.plot_index()`

Put everything in one graph

It was quite some work to go through all these step!

- Someone has done us a favor to put them together
- Influence Plot: the silver bullet
- In Python `i.plot_influence()`
  - Hat-values (leverage) against studentized residuals
  - Reference lines for studentized resid at -2 and $+2$
  - Reference lines for leverage at $2k/n$ and $3k/n$
  - Size of bubble corresponds to Cook's distance (=influence)

In-class Assignment 5.1 – Part III

Continue with the the model you created before:

- Explain Murder using Population, Income, Frost and Illiteracy

Questions

- Use `m.outlier_test()` to (potentially) find outliers
  → Do you find any?
- Calculate the Cook's distance using `m.cooks_distance[0]`
- Which observations "should be investigated"? ($D_i > 4/(n-k)$)
  → What is special about these states? (not a statistical question, but a common knowledge one)
- Use `i.plot_index()` and `i.plot_influence()` to graphically summarize the influence measures and interpret.
- Which observation should we worry about most?

Fixing things

What can you do after diagnosis

"Cure" the model: your toolkit

- Deleting observations
- Adding or deleting variables
- Transforming dependent variables
- Add transformations of independent variables to capture non-linear relations
  - Squared terms
  - Log terms
- Use corrected (robust) standard errors
- Using an alternative regression method

Basic rule

- Do not "abuse" these methods
- Use the background information/knowledge about the data

Method 1: Deleting observations

- Easiest one after detecting outliers or influential observations
- Think twice, or three times!
  - Is there a reason to delete the outlier?
  - With that reason, are there other observations that should be deleted as well?
  - How many would you delete in total, are they really outliers?
  - Is there any interesting relation between the deleted and remaining observations?
- Once you reach the last question, quite often you get a new insight about the data!

Method 2: Transforming variables

- This usually refers to transforming the dependent variable Y
  - Logarithm: $\log Y$ (for positive variables indicating "size")
  - Logit: $\log(Y/(1-Y))$ (for variables indicating "proportion")
  - Power: Y^λ (least used)
- Be careful: can you still interpret the transformed model?

Method 3: Adding or deleting variables

Besides playing with observations (rows), one may try to play with variables (columns)!

- More freedom, more fun!
- Deleting
  - Reduces model fit, can make model "better"
  - Keep those you are interested in!
- Adding variables
  - Which subgroup of the available regressors we should use?
  - A large literature: variable selection

→ Use a clear strategy!

Variable selection

Variable selection

Finding the "best" model

- Constraints: a group of potential explanatory variables
- Goal: explain the variation of the dependent variable y (as much as possible)

Model comparison

Comparing two models, which one is "better"?

- Quantitative comparison
  - Goodness of fit measures: R^2, AdjR^2, AIC
    → Cannot tell whether the difference is *significant*
  - Out-of-sample (=hold out) forecast comparison
- Statistical (in-sample) testing: only between nested models

Nested model test

The complete model: $y = \beta_1 + \beta_2 x_2 + \cdots + \beta_{k_R} x_{k_R} + \cdots + \beta_{k_C} x_{k_C} + \varepsilon$
Nested model: $y = \beta_1 + \beta_2 x_2 + \cdots + \beta_{k_R} x_{k_R} + \varepsilon$

The nested model..

- has less independent variables: setting some of the coefficients to zero
  → E.g. set $\beta_{k_R+1} = \cdots = \beta_{k_C} = 0$
- is also called restricted model
- has a lower R^2, but *may* be more appropriate

Test whether the nested model is preferred
→ Test $H_0 : \beta_{k_R+1} = \cdots = \beta_{k_C} = 0$ in the original model ($k_C - k_R$ restrictions)

F-test for nested model

In R (requires the package "car"): `sm.stats.anova_lm(fitR,fitC)` (Restricted vs. Complete)

- Compare fit of both models using F-test (similar to before)
- Does the Explained Sum of Squares [ESS] differ *significantly*?
  - The test statistic and distribution under H_0

$$F = \frac{(ESS_C - ESS_R)/(k_C - k_R)}{RSS_C/(n-k_C)} \sim F(k_C - k_R, n - k_C)$$

  - A large F value
    - The null H_0 is rejected
    - Restrictions are not plausible
    - The nested model is significantly "worse" than the original model

In practice: after deleting a few variables

- Run the F-test
- If significant: the nested model is significantly worse
- If insignificant: the deleting is OK

Tricks for (manual) model specification

Include after y∼

- x:z: include $x \times z$
- x*z: include x, z, and $x \times z$
- x*w*z: include x, w, z, $x \times w$, $x \times z$, $w \times z$, $x \times w \times z$
- (x+w+z)**2: include interactions up to 2^{nd} degree: x, w, z, $x \times w$, $x \times z$, and $w \times z$,
- -z: remove variable z, eg. x*w*z -w:z gives x, w, z, $x \times w$, $x \times z$, $x \times w \times z$
- I(x^2): evaluate function within I() mathematically, so use: x^2

Stepwise regression

- The toolkit we have now: p-values, nested model test, or AIC
  - We can check whether deleting/adding one (or more) variable(s) is appropriate
- Backward stepwise regression
  - Start with all variables
  - Delete the worst variable and reestimate
  - Stop when there are no bad variables
- Forward stepwise regression
  - Start with no variable
  - Add the best variable and reestimate
  - Stop when adding any other variable doesn't help

Criteria:

- AIC: look at change in AIC (needs to decrease) → go for largest decrease
- p-values: want variables to be significant (below threshold) → go for smallest p-value

Implementation

- For AIC and p-value
- Implemented in `model_selection.py` see Canvas
  - `backward_elimination_pvalue(model, significance=0.05)`
  - `backward_elimination_aic(model)`
  - `forward_selection_aic(model)`
  - `forward_selection_pvalue(model, significance=0.05)`

  where `model` is an not-fitted model:

  eg `model = smf.ols(formula="y ∼ X", data=df)`

All subsets regression

Why not compare **all** possible models?

- With k potential variables, there are 2^k potential models
  - For $k = 10$, we get $2^{10} = 1024$ models!
  - A lot of computation, but who cares?
  - Still, would be quite messy to view all of the results
- In Python: see `model_selection.py`
- `allsubset(m, best=10)` → show the best (max) 10 models
- `m` is again a not-fitted model
- Limitations
  - If k is really large (say 1000), we do care about computation time!
  - Stepwise regression is preferred in this case
  - However, it may miss the best model

→ This is an ongoing field: A large part of machine learning literature is on finding the best models for regressions!

Take home assignment (see "starter code" on Canvas)

- Use the Murder rate data (Murder as dependent variable)
- Take four independent variables: Income, Population, Illiteracy, Frost
- Perform forward stepwise regression
- Test whether the optimal model obtained from forward stepwise regression is significantly different from the complete model (these are nested models)
- Also try backward selection starting from all four variables
- And try all subsets selection on AIC

Before next time

- Reread Chapter 4 if needed
- Read from Chapter 5:
  - Logistic regression
  - Evaluating Classification Models
- Reconsider the in-class assignments of this week
- Take home assignment
- Ask questions on the discussion board
- Work on final assignment (next deadline October 12)