Lab 1, Part A: US data from FRED

International Monetary Policy and Relations • AUEB
Natassa Zervou-Velez

Purpose

In Lab 2 we will estimate Taylor rules of the form

$$i_t = r^* + \pi^* + \phi_\pi(\pi_t - \pi^*) + \phi_y\, y_t,$$

where i_t is the policy rate, π_t is inflation (past/expected), π^* is the inflation target and y_t is a measure of output gap (or unemployment gap). FRED gives us none of these variables in the form the regression needs. Today you will find the raw series, learn which transformation turns each of them into a regressor, do the transformations first in the FRED interface and then in Stata. At the end you will save a quarterly US dataset, `taylor_us.dta`, which we will use in Lab 2.

Answer the questions marked **Q** in a text file or on paper as you go.

0. Set-up

a. Create a folder called `lab1` on the desktop. Save every file from today in this folder.

b. Open a browser and go to `https://fred.stlouisfed.org`. FRED is the database of the Federal Reserve Bank of St. Louis; it is free and needs no account.

c. Open Stata and set the working directory to your folder: **File** → **Change Working Directory**, or type `cd "path/to/lab1"`.

Downloading data from FRED: general rules

In several sections below you will be asked to download the series you have just looked at. Every FRED graph has a **DOWNLOAD** button at the top right, next to **EDIT GRAPH**. Choose **CSV (data)**. The file contains exactly what the graph shows: the same lines, units, frequency and date range. Set up the graph first, then download.

Follow these rules for every download today:

a. **Start date** `1984-01-01`. We need one extra year because year-on-year inflation loses the first four quarters.

b. **Frequency.** For every line, set **Modify frequency** to **Quarterly** and **Aggregation method** to **Average**. Section 5 explains why.

c. **Units.** Leave the original units (levels) unless the instructions say otherwise. We will do the transformations ourselves in Stata.

d. **No formulas.** If a line uses a formula (Section 6), download the raw series instead; the formula is reproduced in Stata.

e. **Save** the file in your `lab1` folder under the name given in the instructions. If the browser saves it to `Downloads`, move it.

Open each file in a text editor or Excel after downloading it. The first column is the date (`observation_date`, in the form `YYYY-MM-DD`, the first day of each quarter). Each further column is one series, named by its ID. Transformed series receive a suffix, for example `_PC1` for percent change from a year ago. Missing observations are blank.

1. Finding a series and reading its notes

1. Type `federal funds` in the FRED search box. Open *Federal Funds Effective Rate*, series ID `FEDFUNDS`. Every FRED series has a unique ID; it appears in brackets after the title and in the web address.
2. Read the information above the graph and the **Notes** below it: units, frequency, seasonal adjustment, source, release, and the date of the last update.
3. Search again and open `DFF`, the same rate at daily frequency.

Q1. What are the units and the frequency of `FEDFUNDS`? How is a monthly value of `FEDFUNDS` constructed from the daily rate `DFF`?

The units are percent, not seasonally adjusted; the frequency is monthly. Each monthly value is the average of the daily effective rate (`DFF`) .

Q2. Who publishes the original data, and who publishes FRED? Why does the distinction matter when you cite a series?

The original source is the Board of Governors of the Federal Reserve System (H.15 release); the New York Fed has computed the effective rate since 2016. FRED is published by the Federal Reserve Bank of St. Louis, which collects series from other sources. Definitions, revisions and copyright belong to the original source, so the citation names the source and adds "retrieved from FRED." FRED versions can also lag the source; several euro-area series, for example, still refer to a fixed 19-country composition.

2. The graph: date range, recessions

1. On the `FEDFUNDS` page, set the date range to start on `1984-01-01` using the date boxes above the graph. Grey bars are NBER recessions. Move the mouse over the line to read individual values.
2. Click **EDIT GRAPH** (top right of the graph). A panel opens with the tabs **EDIT LINE 1**, **ADD LINE** and **FORMAT**.
3. In **ADD LINE**, search for `DFEDTARU` (upper limit of the target range), select it and click **Add data series**. Repeat for `DFEDTARL` (lower limit).
4. **Download.** Following the general rules (start date `1984-01-01`; all three lines **Quarterly**, **Average**), download the graph as **CSV (data)** and save it as `fred_rate.csv`.

Q3. In which periods was the fed funds rate at the zero lower bound? Give start and end dates.

From 16 December 2008 to 16 December 2015, and from 15 March 2020 to 16 March 2022 (target range 0–0.25%). In monthly `FEDFUNDS`: December 2008 to November 2015, and April 2020 to February 2022.

Q4. From which date does the target range exist? What did the Fed announce before that date?

From 16 December 2008. Before that the FOMC set a single target for the federal funds rate, announced explicitly from February 1994; before 1994 markets inferred it from open market operations.

Q5. The Fed sets a target range, but we will use the effective rate `FEDFUNDS` as the dependent variable. Give one argument for this choice.

The effective rate is the rate at which transactions actually take place; it is one continuous series over the whole sample, whereas the target changes definition in 2008; it records movements within the range; and Taylor (1993) used it.

3. Prices are levels; inflation is a transformation

1. Open a new graph for `CPIAUCSL` (Consumer Price Index for All Urban Consumers: All Items). Look at the units: it is an *index*, with 1982–84 = 100.
2. Click **EDIT GRAPH → EDIT LINE 1**. Open the **Units** menu and choose, one at a time:
   - **Percent Change**: the change from the previous month, $100\,(P_t/P_{t-1} - 1)$;
   - **Compounded Annual Rate of Change**: the monthly change expressed as an annual rate, assuming it repeats for twelve months and compounds, $100\left[(P_t/P_{t-1})^{12} - 1\right]$ (the exponent is 4 for quarterly data);
   - **Percent Change from Year Ago**.

   Look at the graph after each choice.
3. Leave the line on **Percent Change from Year Ago**. FRED computes it as

$$\pi_t = 100\left(\frac{P_t}{P_{t-12}} - 1\right) \quad \text{(monthly data)}, \qquad \pi_t = 100\left(\frac{P_t}{P_{t-4}} - 1\right) \quad \text{(quarterly data)}.$$

Q6. Why is the level of the CPI meaningless for a Taylor rule, while its rate of change is not?

The level of an index depends on an arbitrary base year (1982–84 = 100) and trends upward without limit. The central bank targets the rate of change, so only inflation, measured against the target, can enter the rule.

Q7. The year-on-year rate does not require seasonally adjusted data, but the month-on-month rate does. Explain why.

The year-on-year rate compares the same month in consecutive years, so a stable seasonal pattern cancels. The month-on-month rate compares adjacent months with different seasonal patterns (January sales, heating fuel), which would appear as inflation.

4. Which inflation measure?

Build one graph with four lines, all with **Units** set to **Percent Change from Year Ago**, starting in 1985:

Series ID	Measure	Note
`CPIAUCSL`	CPI, headline	Bureau of Labor Statistics
`CPILFESL`	CPI, core (less food and energy)	Bureau of Labor Statistics
`PCEPI`	PCE price index, headline	Bureau of Economic Analysis; the Fed's target
`PCEPILFE`	PCE price index, core	Bureau of Economic Analysis

- **Consumer Price Index (CPI).** The price of a fixed basket of goods and services bought by urban households, collected monthly by the Bureau of Labor Statistics (BLS) from surveys of prices and of household expenditure. The basket weights are updated only periodically.
- **Personal Consumption Expenditures (PCE) price index.** The price of all consumption in the national accounts, produced by the Bureau of Economic Analysis (BEA). It covers spending by households and spending made on their behalf (for example, employer-paid health insurance and Medicare), and its weights change every period as spending shifts (a chain-weighted index).
- **Headline and core.** Headline inflation includes all items. Core inflation excludes food and energy, whose prices are volatile and driven largely by supply shocks; it is used as a guide to the underlying trend.

1. Start from `CPIAUCSL`; add the other three with **ADD LINE**. After adding each line, set its **Units** in its own **EDIT LINE** tab.
2. Add a horizontal line at the 2% target: in **ADD LINE** add `PCEPI` once more, then under **Customize data** type the formula `a*0+2` and click **Apply**.

Q8. Which measure is usually higher, CPI or PCE inflation? By roughly how much? Look up one reason in the series notes or on the BEA website.

CPI inflation is usually higher, by about 0.4 percentage points on average since 1985. Three differences between the indices explain the gap. (i) *Formula.* The CPI prices a basket whose weights are fixed for long periods. When a good becomes relatively more expensive, households buy less of it, but the CPI keeps its original weight, so goods whose prices have risen fastest are overweighted (substitution bias). The PCE index is chain-weighted: its weights are updated every period as spending shifts, so it captures this substitution and rises more slowly. This is the largest and most systematic source of the gap. (ii) *Weights.* Shelter makes up about one third of the CPI but about 15% of PCE. Shelter prices have generally risen faster than other prices, so the larger weight raises CPI inflation relative to PCE inflation. (iii) *Scope.* PCE also covers spending made on households' behalf, such as employer-paid health insurance and Medicare, so health care has a much larger weight in PCE. Because medical prices have tended to rise faster than average, this difference partly offsets (i) and (ii) rather than adding to the gap.

Q9. In 2021–23, when did headline inflation peak, and when did core inflation peak? What does the difference tell you about the source of the shock?

Headline CPI peaked at 9.1% in June 2022 and headline PCE at about 7.2% in the same month. Core CPI peaked at 6.6% in September 2022, and core PCE at about 5.6% in early 2022. Headline inflation peaked with energy and food prices after Russia's invasion of Ukraine and fell quickly when they reversed. Core inflation declined more slowly because the shock spread to goods and services more broadly. Core CPI peaked later than core PCE because of the larger weight of shelter in the CPI, where rents respond with a lag.

Q10. The Fed's 2% target refers to headline PCE inflation, but many estimated Taylor rules use core inflation. Give one argument for each choice.

Headline: it is the measure in which the target is defined and the one households experience. Core: it is a better predictor of medium-term inflation and filters out supply shocks, such as energy, that monetary policy cannot offset quickly and should look through. Using $\pi^* = 2$ with core inflation assumes that core and headline inflation share the same average over the medium term; this holds approximately because food and energy prices do not trend much relative to other prices over long periods. In an estimated rule with an intercept, π^* is absorbed by the constant, so the estimated responses ϕ_π and ϕ_y do not depend on the assumed target.

5. Changing the frequency

Our regressions will be quarterly, because GDP is quarterly. Every monthly series must therefore be converted to quarterly. FRED offers two ways to do this: the *average* of the three months in the quarter, or the value of the *last* month (end of period). We will use the average for all series.

1. On the graph from Section 4, open **EDIT LINE 1**, set **Modify frequency** to **Quarterly** and **Aggregation method** to **Average**. Repeat for every line.
2. Switch the aggregation method of one line to **End of Period** and compare. For inflation the difference is small. Switch it back to **Average**.
3. **Record FRED's values.** With the four lines still in **Percent Change from Year Ago**, **Quarterly**, **Average**, move the mouse over the graph and write down the CPI and PCE inflation

rates that FRED reports for 1990Q1, 2008Q3 and 2022Q2. We will compare them with our own calculations in Stata (Section 8.6).

4. **Download: the price levels.** On the same graph, set the **Units** of every line back to the original index. Add `GDPDEF` (GDP deflator, quarterly) with **ADD LINE**; Taylor (1993) used this measure. Download as **CSV (data)** and save as `fred_prices.csv`.

Q11. FRED first converts the monthly index to quarterly and then computes the percent change from a year ago. Write the formula for the quarterly inflation rate it reports, first with **Average** and then with **End of Period**.

With **Average**, FRED averages the three monthly index values in each quarter and then computes the change from a year ago:

$$\bar{P}_q = \tfrac{1}{3}\left(P_{m1,q} + P_{m2,q} + P_{m3,q}\right), \qquad \pi_q = 100\left(\frac{\bar{P}_q}{\bar{P}_{q-4}} - 1\right).$$

With **End of Period**, it uses the last month of each quarter, $\pi_q = 100\,(P_{m3,q}/P_{m3,q-4} - 1)$, which is the year-on-year rate of March, June, September or December.

6. The output gap with a FRED formula

The output gap is the percentage deviation of real GDP from potential GDP:

$$y_t = 100\left(\ln Y_t - \ln Y_t^*\right) \;\approx\; 100\,\frac{Y_t - Y_t^*}{Y_t^*}.$$

The difference of two logarithms approximates a percentage difference, because $\ln(1+x) \approx x$ for small x. This is the same approximation used for growth rates: the growth of GDP from one quarter to the next is $100\,(Y_t/Y_{t-1} - 1) \approx 100\,(\ln Y_t - \ln Y_{t-1})$. In the gap, actual output is compared with potential output in the same quarter instead of with actual output in the previous quarter.

Where potential output and the noncyclical unemployment rate come from. Neither is observed; both are estimated by the Congressional Budget Office (CBO).

- *Potential GDP* (`GDPPOT`) is the output the economy would produce with a high rate of use of its capital and labour, without overheating. CBO estimates it with a production function: it combines the potential labour force and hours worked, the stock of capital, and the trend growth of total factor productivity, after removing the effect of the business cycle from each. Potential output therefore reflects demographics, investment and productivity, not demand.
- *The noncyclical rate of unemployment* (`NROU`) is the unemployment rate that would prevail without fluctuations in aggregate demand. It reflects frictions in matching workers to jobs and the structure of the labour force (age, education). CBO estimates it from demographic trends and from the relation between unemployment and inflation (the Phillips curve): it is the rate consistent with stable inflation.

1. Open a new graph for `GDPC1` (real GDP, billions of chained 2017 dollars, seasonally adjusted annual rate, quarterly).
2. **EDIT GRAPH → EDIT LINE 1 → Customize data**: search for `GDPPOT` (real potential GDP, Congressional Budget Office, CBO) and click **Add**. `GDPC1` is now series `a` and `GDPPOT` series `b`.
3. Type the formula `100*(a-b)/b` and click **Apply**. This is the percentage deviation of GDP from potential, the right-hand side of the formula above. In Stata (Section 8) we will compute the exact log version and compare.

4. Add a second line: `UNRATE` (unemployment rate, monthly) with `NROU` (CBO noncyclical rate of unemployment, quarterly) as series `b`, formula `a-b`. Set **Modify frequency** to **Quarterly**, **Average**.
5. **Download.** The formula lines cannot be used in Stata, so download the raw series. Open a new graph with four lines: `GDPC1`, `GDPPOT`, `UNRATE` and `NROU`, with no formulas and the original units. Set start date `1984-01-01` and every line to **Quarterly**, **Average**. Download as **CSV (data)** and save as `fred_real.csv`. The file will include CBO projections of `GDPPOT` after the last GDP observation; we drop them in Stata.

Q12. Look at the notes of `GDPPOT`: until which year does the series run? What does this imply for the end date of our sample?

`GDPPOT` runs to 2036Q4 because it includes CBO projections (February 2026 update). The sample must end at the last quarter for which GDP is available, currently 2026Q2. CBO also revises its estimates of potential output at each update, so the series is not the gap the Fed observed at the time.

Q13. Give the output gap in 2009Q2 and 2020Q2. Why is 2020Q2 a problem for any statistical estimate of the trend?

With the FRED formula, about -5.0% in 2009Q2 and about -8.8% in 2020Q2 (current data). The log version computed in Stata gives about -5.1% and -9.2%: the two measures differ when the gap is large, because $\ln(1+x) \approx x$ only for small x. The 2020Q2 collapse was caused by lockdowns, not by an ordinary shortfall in demand. A statistical trend such as the HP filter treats it as information and bends towards it, which distorts the estimated gap for several years on both sides. CBO's production-function estimate is much less affected, because potential output depends on labour, capital and productivity, which did not collapse in one quarter.

Q14. The output gap and the unemployment gap move in opposite directions. Why? Which of the two is closer to the Fed's employment mandate?

When output is above potential, firms produce more and hire more workers, so unemployment falls below its noncyclical rate (Okun's law); one percentage point of unemployment gap corresponds to roughly two percent of output gap. The unemployment gap is closer to the Fed's mandate of "maximum employment."

7. Real-time data: ALFRED (optional)

Go to `https://alfred.stlouisfed.org` and open `GDPC1`. ALFRED stores every past *vintage* of a series, that is, the data as they were published on a given date. Add a vintage from 2008 to the graph and compare it with today's data.

Q15. By how much was GDP growth in 2008 revised? Why does this matter for estimating how the Fed reacted at the time (Orphanides, 2001)?

The advance estimate of 2008Q4 growth, published in January 2009, was -3.8% annualized; current data give about -8.5%. Annual growth for 2008 was first published as positive and is now about -0.1%. The Fed reacted to the data available at the time. Estimating its rule with revised data introduces errors in the regressors and can turn a reasonable decision into an apparent policy mistake. Orphanides (2001) shows that real-time output gaps differed markedly from revised ones and that the estimated rules change as a result.

Summary: the files we download

For each file: open the FRED page of the first series, add the others with **EDIT GRAPH** → **ADD LINE**, apply the settings to every line, click **DOWNLOAD** → **CSV (data)** and save the file in your `lab1` folder under the name in the table.

Settings for every line: start date `1984-01-01`; **Modify frequency**: **Quarterly**; **Aggregation method**: **Average**; **Units**: original units (levels); no formulas.

File	Series (IDs)	Content
`fred_rate.csv`	`FEDFUNDS`, `DFEDTARU`, `DFEDTARL`	effective federal funds rate; upper and lower limits of the target range
`fred_prices.csv`	`CPIAUCSL`, `CPILFESL`, `PCEPI`, `PCEPILFE`, `GDPDEF`	CPI, headline and core; PCE price index, headline and core; GDP deflator
`fred_real.csv`	`GDPC1`, `GDPPOT`, `UNRATE`, `NROU`	real and potential GDP; unemployment rate and noncyclical rate of unemployment

8. Into Stata

In this section we load the four files you downloaded into Stata, compute inflation and the output gap ourselves, check our numbers against FRED, and save the final dataset. You should have four files in your `lab1` folder: `fred_rate.csv`, `fred_prices.csv`, `fred_infl_pc1.csv` and `fred_real.csv`.

8.0 Stata in five minutes

The windows. When Stata opens you see five panels: the *Results* window (centre, where output appears), the *Command* window (bottom, where you can type one command), the *Variables* window (right, the list of variables in the data), the *Properties* window (below it) and the *History* window (left, the commands you have run).

One dataset at a time. Stata holds exactly one dataset in memory, a table in which rows are observations (here, quarters) and columns are variables (here, series). If you have used R, think of a single data frame that every command works on. Loading a new file replaces the one in memory, which is why we will save each file as a Stata dataset before merging them.

The do-file. We will not type commands in the Command window. Instead we write them in a *do-file*, a text file of commands that can be rerun at any time and gives an exact record of what we did. Open it with **Window** → **Do-file Editor** → **New Do-file Editor**. Save it at once in your `lab1` folder as `lab1_us.do` (**File** → **Save as**).

Running commands. In the Do-file Editor, select one or more lines with the mouse and press `Ctrl+D` (Windows) or `Cmd+Shift+D` (Mac), or click the **Execute (do)** icon at the top right. If nothing is selected, the whole file runs. Run each block below as soon as you have typed it, and look at the Results window.

Reading the code.

- A line that starts with `*` is a comment; Stata ignores it. Text after `//` on a line is also a comment. Comments explain the code to you and to others.
- `///` at the end of a line means that the command continues on the next line.
- Stata is case-sensitive: `GDP` and `gdp` are different variables.

- Output in red is an error. Stata stops at the line with the error; read the message, correct the line and run it again.
- `help` followed by a command name, for example `help generate`, opens the manual page for that command.

8.1 Start the do-file

Type the following at the top of the do-file and run it.

```
* Lab 1, Part A: US data for Taylor rules
* Author: your name        Date: 19 Oct 2026

clear all                  // remove any data from memory
cd "C:/Users/yourname/Desktop/lab1"   // change to your folder
pwd                        // show the current folder
dir                        // list the files in it
```

What it does. `clear all` empties memory so that we start from nothing. `cd` (change directory) tells Stata which folder to read files from and save files to; after this line we can refer to files by name only. To find your folder's path on Windows, open the folder in File Explorer, click the address bar and copy it; keep the quotation marks, and use forward slashes `/`. `pwd` prints the folder Stata is now in, and `dir` lists its files: check that the four `.csv` files appear. If they do not, the path in `cd` is wrong.

8.2 Import the files one at a time

We import each CSV file, convert its date into a quarterly date that Stata understands, and save it as a Stata dataset (`.dta`). Start with the first file:

```
import delimited using "fred_rate.csv", clear case(lower)
describe
browse
```

What it does. `import delimited` reads a text file whose columns are separated by commas. The option `clear` replaces whatever data are in memory; `case(lower)` turns all variable names into lower case, so `FEDFUNDS` becomes `fedfunds`. `describe` lists the variables with their type: numbers appear as `float` or `double`, text as `str`. `browse` opens the data in a spreadsheet-like window; look at it, then close it.

You will see that `observation_date` is text (for example `"1984-01-01"`). Stata cannot do time-series calculations with text, so we convert it:

```
gen qdate = qofd(date(observation_date, "YMD"))
format qdate %tq
drop observation_date
save "fred_rate.dta", replace
```

What it does. `gen` (short for `generate`) creates a new variable. Read the right-hand side from the inside out: `date(observation_date, "YMD")` turns the text into a Stata date, telling Stata that the text is in year–month–day order; `qofd()` ("quarter of date") converts that date into a quarter. Internally Stata stores quarters as numbers (1960q1 is 0, 1960q2 is 1, and so on); `format qdate %tq` makes them display as `1984q1`, `1984q2`, ... We then `drop` the text date, which we no longer need, and `save` the data as `fred_rate.dta`. The option `replace` allows Stata to overwrite the file if it already exists, so the do-file can be rerun.

Repeat exactly the same steps for the other three files:

```
import delimited using "fred_prices.csv", clear case(lower)
describe
gen qdate = qofd(date(observation_date, "YMD"))
format qdate %tq
drop observation_date
save "fred_prices.dta", replace

import delimited using "fred_real.csv", clear case(lower)
describe
gen qdate = qofd(date(observation_date, "YMD"))
format qdate %tq
drop observation_date
save "fred_real.dta", replace
```

If `describe` reports that the date column has a different name (for example `date`), use that name in the `gen` and `drop` lines. If a series is shown as `str`, a missing value was read as text; we fix this in 8.3.

8.3 Merge the four files into one dataset

Each file has one row per quarter and a common variable, `qdate`. Merging puts the columns of all four files side by side, matching rows by quarter.

```
use "fred_rate.dta", clear
merge 1:1 qdate using "fred_prices.dta"
tabulate _merge
drop _merge
```

What it does. `use` loads a Stata dataset into memory. `merge 1:1 qdate using ...` adds the variables of the second file, matching on `qdate`; `1:1` says that each quarter appears once in each file. Stata creates a variable `_merge` that records where each row came from: 3 means the quarter was found in both files, 1 only in the data in memory, 2 only in the file being merged. `tabulate _merge` counts them. A few rows with 1 or 2 at the end of the sample are normal (the series end in different quarters). We then drop `_merge`, because the next merge creates it again.

Merge the remaining two files in the same way, then declare the data a time series:

```
merge 1:1 qdate using "fred_real.dta"
tabulate _merge
drop _merge

tsset qdate
describe
```

Why `tsset`. `tsset qdate` tells Stata that the rows are consecutive quarters ordered by `qdate`. Without it, Stata does not know which row is "the previous quarter", and the lag operators we use below (`L.`, `L4.`) do not work. Stata should reply `time variable: qdate, 1984q1 to ...`, `delta: 1 quarter`.

If `describe` shows any series as `str`, convert it to a number, for example:

```
destring fedfunds, replace force
```

8.4 Short names and the sample

FRED's IDs are hard to remember. We give the variables short names:

```
rename fedfunds ff
rename dfedtaru ff_upper
rename dfedtarl ff_lower
rename cpiaucsl cpi
rename cpilfesl cpi_core
rename pcepi    pce
rename pcepilfe pce_core
rename gdpdef   defl
rename gdpc1    gdp
rename gdppot   gdp_pot
rename unrate   u
rename nrou     u_star
```

`rename old new` changes a variable's name and nothing else. The four `_pc1` variables from FRED keep their names; we use them only in 8.6.

Next, we remove the quarters for which there is no GDP:

```
drop if missing(gdp)
```

Why. In Stata a missing value is shown as a dot (`.`). `GDPPOT` contains CBO projections up to 2036, so after the merge the data run to 2036, with GDP missing after the latest published quarter. `drop if missing(gdp)` deletes every row in which `gdp` is missing. Run `browse` to check that the data now end in the last quarter with GDP.

Finally, the midpoint of the target range:

```
gen ff_mid = (ff_upper + ff_lower)/2
label var ff_mid "Midpoint of the fed funds target range (%)"
```

`label var` attaches a description to a variable; it appears in the Variables window and on graphs. `ff_mid` is missing before 2009, when there was no range.

8.5 Inflation

We compute four-quarter (year-on-year) inflation from each price index.

```
* exact formula: percent change from four quarters earlier
gen pi_cpi      = 100*(cpi/L4.cpi - 1)
gen pi_cpi_core = 100*(cpi_core/L4.cpi_core - 1)
gen pi_pce      = 100*(pce/L4.pce - 1)
gen pi_pce_core = 100*(pce_core/L4.pce_core - 1)
gen pi_defl     = 100*(defl/L4.defl - 1)

label var pi_cpi      "CPI inflation, y/y (%)"
label var pi_cpi_core "Core CPI inflation, y/y (%)"
label var pi_pce      "PCE inflation, y/y (%)"
label var pi_pce_core "Core PCE inflation, y/y (%)"
label var pi_defl     "GDP deflator inflation, y/y (%)"
```

What it does. `L4.cpi` is the value of `cpi` four quarters earlier; `L.cpi` (or `L1.cpi`) would be the previous quarter. This works only because we declared the data with `tsset`. So `100*(cpi/L4.cpi - 1)` is exactly the formula $100\,(P_t/P_{t-4}-1)$ that FRED uses. For the first four quarters (1984) there is no earlier value, so inflation is missing there; this is why we downloaded data from 1984.

Two variants for comparison:

```
* log approximation of four-quarter CPI inflation
```

```
gen pil_cpi = 100*(ln(cpi) - ln(L4.cpi))
label var pil_cpi "CPI inflation, y/y, log approximation (%)"

* quarter-on-quarter PCE inflation, annualised
gen pi_pce_qq = 400*(ln(pce) - ln(L.pce))
label var pi_pce_qq "PCE inflation, q/q annualised (%)"
```

The annualised quarterly rate multiplies the one-quarter change by four, so it is comparable in size to an annual rate.

Now plot the results:

```
tsline pi_cpi pi_pce pi_pce_core if qdate >= tq(1985q1), ///
    yline(2) name(inflation, replace)

tsline pi_pce pi_pce_qq if qdate >= tq(1985q1), ///
    yline(2) name(yy_vs_qq, replace)
```

What it does. `tsline` draws time-series lines against `qdate`. The condition `if qdate >= tq(1985q1)` restricts the graph to 1985 onward; `tq()` converts a quarter written as text into Stata's internal number. `yline(2)` draws a horizontal line at the 2% target. `name(..., replace)` gives the graph a name, so that each new graph opens in its own window instead of replacing the previous one.

Finally, compare the exact and log formulas:

```
gen d_log = pi_cpi - pil_cpi
summarize d_log
tsline d_log if qdate >= tq(1985q1), name(exact_vs_log, replace)
```

`summarize` reports the number of observations, mean, standard deviation, minimum and maximum of a variable.

Q16. Look at `d_log`, the difference between the exact formula and the log approximation. When is it largest, and why?

The exact formula is always at least as large as the log approximation, because $\ln(1+x) \leq x$. The difference is about $100\,x^2/2$ and grows with inflation: it is largest in 2022, about 0.3–0.4 percentage points, and negligible at 2% inflation.

8.6 Check against FRED

In Section 5 you wrote down FRED's year-on-year CPI and PCE inflation for three quarters. If our code is right, our numbers should match FRED's.

```
list qdate pi_cpi pi_pce if inlist(qdate, tq(1990q1), tq(2008q3), tq(2022q2))
```

What it does. `list` prints the values of the listed variables in the Results window. `inlist()` keeps only the rows whose `qdate` equals one of the quarters in brackets.

Q17. Do your numbers match the ones you wrote down from FRED? If not, how large are the differences, and what could explain them?

They should match up to rounding: FRED shows values rounded to a few decimals. Both FRED and our code average the monthly index to quarters first and compute the year-on-year change second. Averaging the monthly year-on-year rates instead gives differences of up to about 0.1 percentage points, which shows that the order of aggregation and transformation matters.

8.7 The output gap and the unemployment gap

We compute the two measures of slack from the CBO series, exactly as in the FRED formulas of Section 6:

```
gen gap = 100*(ln(gdp) - ln(gdp_pot))
label var gap "Output gap, CBO (%)"

gen ugap = u - u_star
label var ugap "Unemployment gap (percentage points)"
```

What it does. `ln()` is the natural logarithm. The first line is the percentage deviation of real GDP from potential GDP; the second is the difference between the unemployment rate and CBO's noncyclical rate.

Plot the two gaps and compute their correlation:

```
tsline gap ugap if qdate >= tq(1985q1), yline(0) name(gaps, replace)
correlate gap ugap
```

`correlate` reports the correlation coefficient between the two variables.

Q18. What is the correlation between the output gap and the unemployment gap? What does its sign tell you, and in which episodes do the two measures give the most different picture of slack?

The correlation is strongly negative: when output is above potential, unemployment is below its noncyclical rate (Okun's law). The two measures diverge most around 2020: the output gap collapsed in 2020Q2 and recovered within a few quarters, while unemployment rose by more and fell back more slowly. After 2008 the unemployment gap also closed later than the output gap. In Lab 2 we can therefore ask whether the Fed's behaviour is described better by the output gap or by the unemployment gap, which is closer to its employment mandate.

8.8 Inflation gap, lagged rate, and save

```
gen pigap = pi_pce - 2
label var pigap "PCE inflation minus 2% target (pp)"

gen ff_lag = L.ff
label var ff_lag "Fed funds rate, previous quarter (%)"
label var ff     "Effective federal funds rate (%)"

keep if qdate >= tq(1985q1)
order qdate ff ff_lag ff_mid pi_* pil_cpi pigap gap ugap
save "taylor_us.dta", replace
```

What it does. `pigap` is the inflation gap $\pi_t - \pi^*$ in the Taylor rule. `ff_lag` is the policy rate one quarter earlier; we will need it in Lab 2, because central banks adjust rates gradually (interest-rate smoothing). `keep if` keeps only the rows that satisfy the condition, here 1985 onward, since 1984 was needed only to compute inflation. `order` moves the listed variables to the front of the dataset; `pi_*` means every variable whose name starts with `pi_`. `save` writes the final dataset, which we will open in Lab 2 with `use "taylor_us.dta", clear`.

8.9 If something goes wrong

- `file fred_rate.csv not found`: the folder in `cd` is wrong, or the file has a different name. Check with `dir`.

- `variable qdate already defined` (or any `already defined`): you ran a `gen` line twice. Do not fix this line by line: run the whole do-file again from the top. `clear all` at the start removes everything, so the file always rebuilds the data from scratch.
- `variable fedfunds not found`: the name in the file is different. Type `describe` and use the name shown.
- `time variable not set`: you used `L.` or `L4.` before `tsset qdate`. Run the do-file from the top.
- `type mismatch`: a series was read as text. Use `destring varname, replace force`.
- The results look wrong after you edited a line in the middle: run the whole do-file from the top. A do-file is meant to be run in full; this is what makes your work reproducible.

Before you leave Part A

Check that your `lab1` folder contains `fred_us_transformed.csv`, `fred_us_levels.csv`, `lab1_us.do` and `taylor_us.dta`. In Part B we build the same dataset for the euro area from the ECB Data Portal.