In [12]:
```
# Importing modules
import pandas as pd

# Read datasets/yearly_deaths_by_clinic.csv into yearly
yearly = pd.read_csv("datasets/yearly_deaths_by_clinic.csv")

# Print out yearly
yearly
```

Out[12]:

	year	births	deaths	clinic
0	1841	3036	237	clinic 1
1	1842	3287	518	clinic 1
2	1843	3060	274	clinic 1
3	1844	3157	260	clinic 1
4	1845	3492	241	clinic 1
5	1846	4010	459	clinic 1
6	1841	2442	86	clinic 2
7	1842	2659	202	clinic 2
8	1843	2739	164	clinic 2
9	1844	2956	68	clinic 2
10	1845	3241	66	clinic 2
11	1846	3754	105	clinic 2

In [ ]:

In [13]:
```
%%nose

import pandas as pd

def test_yearly_exists():
    assert "yearly" in globals(), \
        "The variable yearly should be defined."

def test_yearly_correctly_loaded():
    correct_yearly = pd.read_csv("datasets/yearly_deaths_by_clinic.csv")
    try:
        pd.testing.assert_frame_equal(yearly, correct_yearly)
    except AssertionError:
        assert False, "The variable yearly should contain the data in yearly_d
eaths_by_clinic.csv"
```

Out[13]: 2/2 tests passed

2. The alarming number of deaths

The table above shows the number of women giving birth at the two clinics at the Vienna General Hospital for the years 1841 to 1846. You'll notice that giving birth was very dangerous; an *alarming* number of women died as the result of childbirth, most of them from childbed fever.

We see this more clearly if we look at the *proportion of deaths* out of the number of women giving birth. Let's zoom in on the proportion of deaths at Clinic 1.

In [14]:
```
# Calculate proportion of deaths per no. births
yearly["proportion_deaths"] = yearly["deaths"] / yearly["births"]

# Extract Clinic 1 data into clinic_1 and Clinic 2 data into clinic_2
clinic_1 = yearly[yearly["clinic"] == "clinic 1"]
clinic_2 = yearly[yearly["clinic"] == "clinic 2"]

# Print out clinic_1
clinic_1
```

Out[14]:

	year	births	deaths	clinic	proportion_deaths
0	1841	3036	237	clinic 1	0.078063
1	1842	3287	518	clinic 1	0.157591
2	1843	3060	274	clinic 1	0.089542
3	1844	3157	260	clinic 1	0.082357
4	1845	3492	241	clinic 1	0.069015
5	1846	4010	459	clinic 1	0.114464

In [15]:
```
%%nose

def test_proportion_deaths_exists():
    assert 'proportion_deaths' in yearly, \
        "The DataFrame yearly should have the column proportion_deaths"

def test_proportion_deaths_is_correctly_calculated():
    assert all(yearly["proportion_deaths"] == yearly["deaths"] / yearly["birth
s"]), \
        "The column proportion_deaths should be the number of deaths divided b
y the number of births."

def test_yearly1_correct_shape():
    assert clinic_1.shape == yearly[yearly["clinic"] == "clinic 1"].shape, \
        "`clinic_1` should contain the rows in yearly from clinic 1"

def test_yearly2_correct_shape():
    assert clinic_2.shape == yearly[yearly["clinic"] == "clinic 2"].shape, \
        "`clinic_2` should contain the rows in yearly from clinic 2"
```

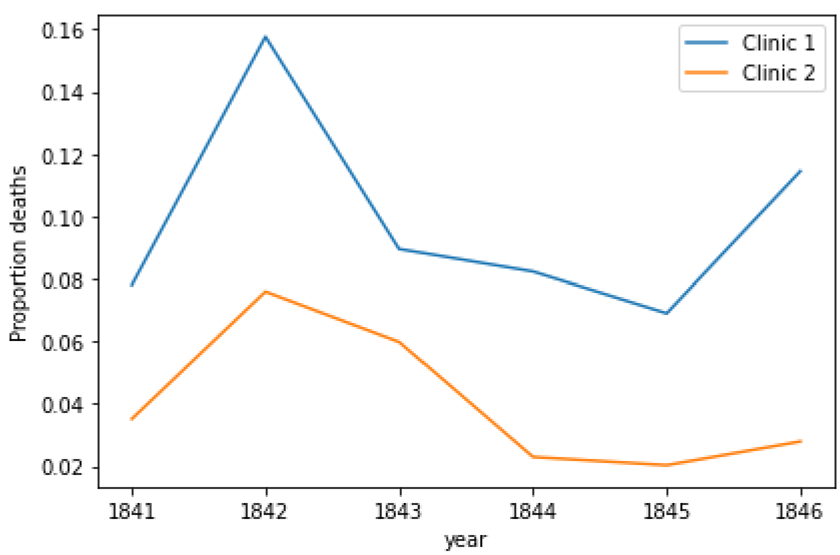
Out[15]: 4/4 tests passed

3. Death at the clinics

If we now plot the proportion of deaths at both Clinic 1 and Clinic 2 we'll see a curious pattern…

In [16]:
```
# This makes plots appear in the notebook
%matplotlib inline

# Plot yearly proportion of deaths at the two clinics
ax = clinic_1.plot(x="year", y="proportion_deaths", label="Clinic 1")
clinic_2.plot(x="year", y="proportion_deaths", label="Clinic 2", ax=ax, ylabel
="Proportion deaths")
```

Out[16]: <AxesSubplot:xlabel='year', ylabel='Proportion deaths'>



In [17]:
```
%%nose

def test_ax_exists():
    assert 'ax' in globals(), \
        "The result of the plot method should be assigned to a variable called ax"

def test_plot_plots_correct_data():
    y0 = ax.get_lines()[0].get_ydata()
    y1 = ax.get_lines()[1].get_ydata()
    assert (
        (all(clinic_1["proportion_deaths"] == y0) and
         all(clinic_2["proportion_deaths"] == y1))
        or
        (all(clinic_1["proportion_deaths"] == y1) and
         all(clinic_2["proportion_deaths"] == y0))), \
        "The data from Clinic 1 and Clinic 2 should be plotted as two separate lines."
```

Out[17]:
```
2/2 tests passed
```

4. The handwashing begins

Why is the proportion of deaths consistently so much higher in Clinic 1? Semmelweis saw the same pattern and was puzzled and distressed. The only difference between the clinics was that many medical students served at Clinic 1, while mostly midwife students served at Clinic 2. While the midwives only tended to the women giving birth, the medical students also spent time in the autopsy rooms examining corpses.

Semmelweis started to suspect that something on the corpses spread from the hands of the medical students, caused childbed fever. So in a desperate attempt to stop the high mortality rates, he decreed: *Wash your hands!* This was an unorthodox and controversial request, nobody in Vienna knew about bacteria at this point in time.

Let's load in monthly data from Clinic 1 to see if the handwashing had any effect.

In [18]:
```
# Read datasets/monthly_deaths.csv into monthly
monthly = pd.read_csv("datasets/monthly_deaths.csv", parse_dates=["date"])

# Calculate proportion of deaths per no. births
monthly["proportion_deaths"] = monthly["deaths"] / monthly["births"]

# Print out the first rows in monthly
monthly.head()
```

Out[18]:

	date	births	deaths	proportion_deaths
0	1841-01-01	254	37	0.145669
1	1841-02-01	239	18	0.075314
2	1841-03-01	277	12	0.043321
3	1841-04-01	255	4	0.015686
4	1841-05-01	255	2	0.007843

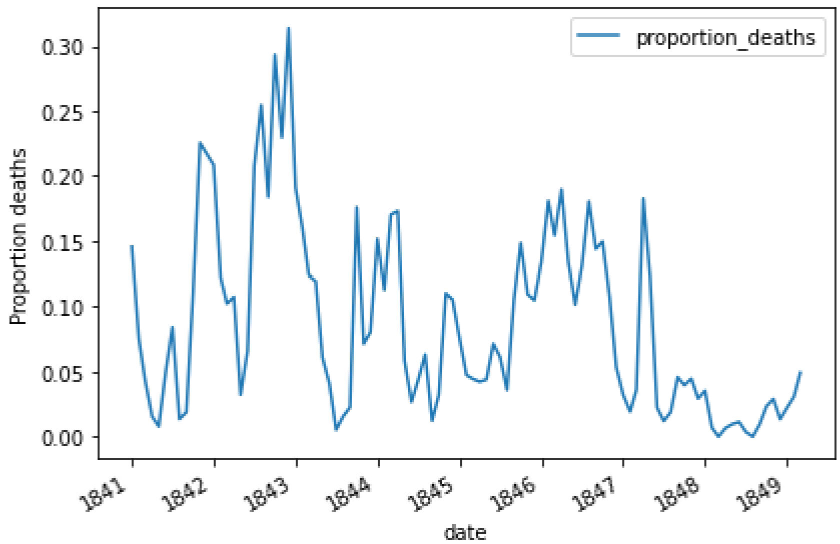
In [19]:
```
%%nose

def test_monthly_exists():
    assert "monthly" in globals(), \
        "The variable monthly should be defined."

def test_monthly_correctly_loaded():
    correct_monthly = pd.read_csv("datasets/monthly_deaths.csv")
    try:
        pd.testing.assert_series_equal(monthly["births"], correct_monthly["bir
ths"])
    except AssertionError:
        assert False, "The variable monthly should contain the data in monthly
_deaths.csv"

def test_date_correctly_converted():
    assert monthly.date.dtype == pd.to_datetime(pd.Series("1847-06-01")).dtyp
e, \
        "The column date should be converted using the pd.to_datetime() functi
on"

def test_proportion_deaths_is_correctly_calculated():
    assert all(monthly["proportion_deaths"] == monthly["deaths"] / monthly["bi
rths"]), \
        "The column proportion_deaths should be the number of deaths divided b
y the number of births."
```

Out[19]: 4/4 tests passed

5. The effect of handwashing

With the data loaded we can now look at the proportion of deaths over time. In the plot below we haven't marked where obligatory handwashing started, but it reduced the proportion of deaths to such a degree that you should be able to spot it!

In [20]:
```
# Plot monthly proportion of deaths
ax = monthly.plot(x="date", y="proportion_deaths", ylabel="Proportion deaths")
```



In [21]:
```
%%nose

def test_ax_exists():
    assert 'ax' in globals(), \
        "The result of the plot method should be assigned to a variable called
ax"

def test_plot_plots_correct_data():
    y0 = ax.get_lines()[0].get_ydata()
    assert all(monthly["proportion_deaths"] == y0), \
        "The plot should show the column 'proportion_deaths' in monthly."
```

Out[21]:
```
2/2 tests passed
```

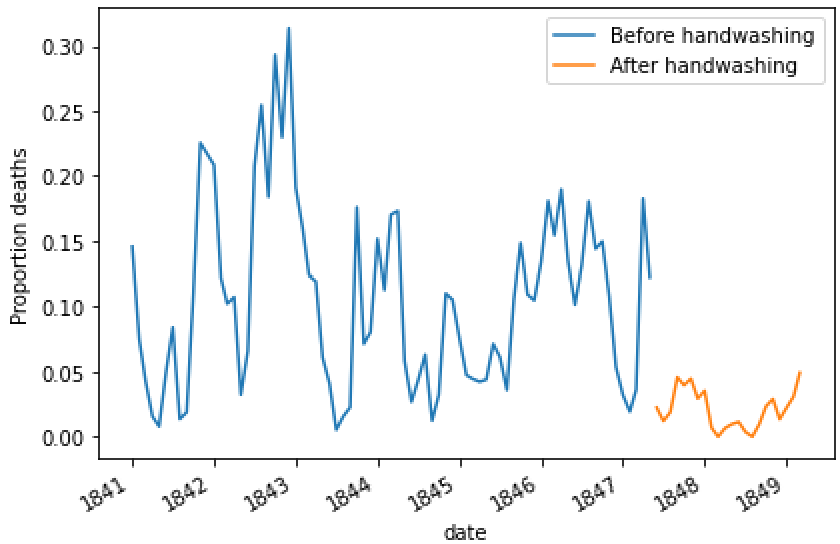
6. The effect of handwashing highlighted

Starting from the summer of 1847 the proportion of deaths is drastically reduced and, yes, this was when Semmelweis made handwashing obligatory.

The effect of handwashing is made even more clear if we highlight this in the graph.

In [22]:
```
# Date when handwashing was made mandatory
handwashing_start = pd.to_datetime('1847-06-01')

# Split monthly into before and after handwashing_start
before_washing = monthly[monthly["date"] < handwashing_start]
after_washing = monthly[monthly["date"] >= handwashing_start]

# Plot monthly proportion of deaths before and after handwashing
ax = before_washing.plot(x="date", y="proportion_deaths",
                         label="Before handwashing")
after_washing.plot(x="date", y="proportion_deaths",
                   label="After handwashing", ax=ax, ylabel="Proportion death
s")
```

Out[22]:
```
<AxesSubplot:xlabel='date', ylabel='Proportion deaths'>
```



In [23]:
```
%%nose

def test_before_washing_correct():
    correct_before_washing = monthly[monthly["date"] < handwashing_start]
    try:
        pd.testing.assert_frame_equal(before_washing, correct_before_washing)
    except AssertionError:
        assert False, "before_washing should contain the rows of monthly < han
dwashing_start"

def test_after_washing_correct():
    correct_after_washing = monthly[monthly["date"] >= handwashing_start]
    try:
        pd.testing.assert_frame_equal(after_washing, correct_after_washing)
    except AssertionError:
        assert False, "after_washing should contain the rows of monthly >= han
dwashing_start"

def test_ax_exists():
    assert 'ax' in globals(), \
        "The result of the plot method should be assigned to a variable called
ax"


def test_plot_plots_correct_data():
    y0_len = ax.get_lines()[0].get_ydata().shape[0]
    y1_len = ax.get_lines()[1].get_ydata().shape[0]
    assert (
        (before_washing["proportion_deaths"].shape[0] == y0_len and
         after_washing["proportion_deaths"].shape[0] == y1_len)
        or
        (before_washing["proportion_deaths"].shape[0] == y0_len and
         after_washing["proportion_deaths"].shape[0] == y1_len)), \
        "The data in before_washing and after_washing should be plotted as two
separate lines."
```

Out[23]: 4/4 tests passed

7. More handwashing, fewer deaths?

Again, the graph shows that handwashing had a huge effect. How much did it reduce the monthly proportion of deaths on average?

In [24]:
```
# Difference in mean monthly proportion of deaths due to handwashing
before_proportion = before_washing["proportion_deaths"]
after_proportion = after_washing["proportion_deaths"]
mean_diff = after_proportion.mean() - before_proportion.mean()
mean_diff
```

Out[24]: -0.08395660751183336

In [25]:
```
%%nose

def test_before_proportion_exists():
    assert 'before_proportion' in globals(), \
        "before_proportion should be defined"

def test_after_proportion_exists():
    assert 'after_proportion' in globals(), \
        "after_proportion should be defined"

def test_mean_diff_exists():
    assert 'mean_diff' in globals(), \
        "mean_diff should be defined"

def test_before_proportion_is_a_series():
     assert hasattr(before_proportion, '__len__') and len(before_proportion) =
= 76, \
        "before_proportion should be 76 elements long, and not a single numbe
r."

def test_correct_mean_diff():
    correct_before_proportion = before_washing["proportion_deaths"]
    correct_after_proportion = after_washing["proportion_deaths"]
    correct_mean_diff = correct_after_proportion.mean() - correct_before_propo
rtion.mean()
    assert mean_diff == correct_mean_diff, \
        "mean_diff should be calculated as the mean of after_proportion minus
 the mean of before_proportion."
```

Out[25]:
```
5/5 tests passed
```

8. A Bootstrap analysis of Semmelweis handwashing data

It reduced the proportion of deaths by around 8 percentage points! From 10% on average to just 2% (which is still a high number by modern standards).

To get a feeling for the uncertainty around how much handwashing reduces mortalities we could look at a confidence interval (here calculated using the bootstrap method).

In [26]:
```
# A bootstrap analysis of the reduction of deaths due to handwashing
boot_mean_diff = []
for i in range(3000):
    boot_before = before_proportion.sample(frac=1, replace=True)
    boot_after = after_proportion.sample(frac=1, replace=True)
    boot_mean_diff.append( boot_after.mean() - boot_before.mean() )

# Calculating a 95% confidence interval from boot_mean_diff
confidence_interval = pd.Series(boot_mean_diff).quantile([0.025, 0.975])
confidence_interval
```

Out[26]:
```
0.025   -0.101186
0.975   -0.066761
dtype: float64
```

In [27]:
```
%%nose

def test_confidence_interval_exists():
    assert 'confidence_interval' in globals(), \
        "confidence_interval should be defined"

def test_boot_before_correct_length():
    assert len(boot_before) == len(before_proportion), \
        ("boot_before have {} elements and before_proportion have {}." +
         "They should have the same number of elements."
        ).format(len(boot_before), len(before_proportion))

def test_confidence_interval_correct():
    assert ((0.09 < abs(confidence_interval).max() < 0.11) and
            (0.055 < abs(confidence_interval).min() < 0.075)) , \
        "confidence_interval should be calculated as the [0.025, 0.975] quanti
les of boot_mean_diff."
```

Out[27]:
```
3/3 tests passed
```

9. The fate of Dr. Semmelweis

So handwashing reduced the proportion of deaths by between 6.7 and 10 percentage points, according to a 95% confidence interval. All in all, it would seem that Semmelweis had solid evidence that handwashing was a simple but highly effective procedure that could save many lives.

The tragedy is that, despite the evidence, Semmelweis' theory — that childbed fever was caused by some "substance" (what we today know as *bacteria*) from autopsy room corpses — was ridiculed by contemporary scientists. The medical community largely rejected his discovery and in 1849 he was forced to leave the Vienna General Hospital for good.

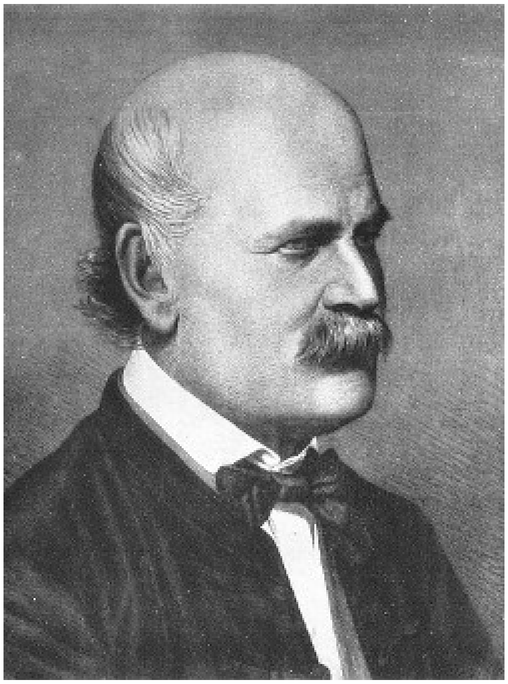
One reason for this was that statistics and statistical arguments were uncommon in medical science in the 1800s. Semmelweis only published his data as long tables of raw data, but he didn't show any graphs nor confidence intervals. If he would have had access to the analysis we've just put together he might have been more successful in getting the Viennese doctors to wash their hands.

In [28]:
```
# The data Semmelweis collected points to that:
doctors_should_wash_their_hands = True
```

In [29]:
```
%%nose

def test_doctors_should_was_their_hands():
    assert doctors_should_wash_their_hands, \
        "Semmelweis would argue that doctors_should_wash_their_hands should be
True ."
```

Out[29]:
```
1/1 tests passed
```

1. Meet Dr. Ignaz Semmelweis



This is Dr. Ignaz Semmelweis, a Hungarian physician born in 1818 and active at the Vienna General Hospital. If Dr. Semmelweis looks troubled it's probably because he's thinking about *childbed fever*: A deadly disease affecting women that just have given birth. He is thinking about it because in the early 1840s at the Vienna General Hospital as many as 10% of the women giving birth die from it. He is thinking about it because he knows the cause of childbed fever: It's the contaminated hands of the doctors delivering the babies. And they won't listen to him and *wash their hands*!

In this notebook, we're going to reanalyze the data that made Semmelweis discover the importance of *handwashing*. Let's start by looking at the data that made Semmelweis realize that something was wrong with the procedures at Vienna General Hospital.