```
data_lifetime.loc[:,'EAD_time'] = data_lifetime.loc[:,'balance_time_0'] * (1+data_lifetime.
    loc[:,'interest_rate_time']/(100*4))**(data_lifetime.loc[:,'TTM_0'] - data_lifetime.loc
    [:,'TTM']) - data_lifetime.loc[:,'annuity'] * ((1+data_lifetime.loc[:,'
    interest_rate_time']/(100*4))**(data_lifetime.loc[:,'TTM_0'] - data_lifetime.loc[:,'TTM
    '])-1) / (data_lifetime.loc[:,'interest_rate_time']/(100*4))
```

We now compute the survival probabilities from the PDs and PPs of prior periods. In a first step, we keep the variables `id`, `time`, `PD_time`, `PP_time` and create a running id variable (`id2`). We also create an empty dataframe (`results`) to append the outputs for the next step.

```
data_lifetime2 = data_lifetime.loc[:, ['id2', 'time', 'PD_time', 'PP_time']]
data_lifetime2['SP_time'] = 1
data_lifetime2['SP_time_tmp'] = 1

results = pd.DataFrame(columns=data_lifetime2.columns.values)
```

In a second step, we run a triple for-loop with the following steps:

1. Process every ID (`for k in range(0,20,1):`) separately. Note, we analyze a sample of 20 loans. We create a data subset for every ID: `data6.loc[data6.id2 == k, :]`;
2. Process every time period separately (`for j in range(1,len(data7),1):`). Note, we start with index 1 (rather than index 0) as the first period has a survival probability of one;
3. Compute the survival probability as 1-PD-PP (`1-data7.iloc[j-i,2]-data7.iloc[j-i,3]`) for every id-time combination `j` by computing the survival probability of prior years and multiplying them to the product of of prior survival probabilities. This requires the computation of prior survival probabilities up to `j` (`for i in range(1,j,1):`). Individual survival probabilities are multiplied to the product of prior survival probabilities: `data7.iloc[j, 4] = data7.iloc[j,4] * data7.iloc[j,5]`. Again, note that we start with index 1 (rather than index 0) as the first period has a survival probability of one;
4. Concatenate row vector with survival probability with the expanding results array using `results = pd.concat([results, data7])`.

This triple for-loop takes a bit of time, which is why we reduced the sample to 20 loans. It is straightforward to extend this process to all loans.

```
for k in range(0,20,1):
    tmp = data_lifetime2.loc[data_lifetime2.id2 == k, :].copy()
    tmp = tmp.reset_index(drop=True)

    for j in range(0,len(tmp),1):

        for i in range(0,j,1):
            tmp.iloc[j, 5] = 1-tmp.iloc[j-i,2]-tmp.iloc[j-i,3]
            tmp.iloc[j, 4] = tmp.iloc[j,4] * tmp.iloc[j,5]
    results = pd.concat([results, tmp], sort=False)
```

```
results = results.drop(['PD_time', 'PP_time', 'SP_time_tmp'], axis=1)
data_lifetime = pd.merge(data_lifetime, results, on=['id2','time'])
```


18.6.1 Expected Losses

For the one-month expected loss we aggregate the losses for the next four periods (quarters) without discounting. Note that this expected loss is based on PIT models and would be used under IFRS 9 for loan loss provisioning of Stage 1 assets. We consider survival in the last observation period 60.

```
data_lifetime2 = data_lifetime.loc[data_lifetime['time'] <= 64, :].copy()

data_lifetime2.loc[:,'EL'] = data_lifetime2.loc[:,'PD_time']*data_lifetime2.loc[:,'SP_time'
    ]*data_lifetime2.loc[:,'LGD_time']*data_lifetime2.loc[:,'EAD_time']

EL = data_lifetime2.loc[:,'EL'].sum()

print(round(EL, 2))
```

```
148038.25
```

18.6.2 Expected Lifetime Losses

In the following we use the loan rate `interest_rate_time/(100*4)` as the discount rate. We divide by 100 as the rate is in percentage points and by 4 as one time period is a quarter.

Note that this expected loss is based on PIT models and would be used under IFRS 9 for loan loss provisioning of Stage 2 assets. For Stage 3 assets, we would use the expected loss (i.e., the product of LGD and EAD) prior to the end of the resolution period.

```
data_lifetime.loc[:,'PVEL'] = data_lifetime.loc[:,'PD_time']*data_lifetime.loc[:,'SP_time'
    ]*data_lifetime.loc[:,'LGD_time']*data_lifetime.loc[:,'EAD_time']/(1+data_lifetime.loc
    [:,'interest_rate_time']/(100*4))**(data_lifetime.loc[:,'time']-60)

CECL = data_lifetime.loc[:,'PVEL'].sum()

print(round(CECL, 2))
```

```
539904.42
```

The lifetime loss is approximately 3.2 times greater than the one-year loss. This is less than the average remaining maturity of loans due to the default and payoff of loans.

18.7 IFRS 9 Significant Increase in Credit Risk

IFRS 9 differs from CECL as IFRS 9 implements a three-stage approach. Under Stage 1, performing assets are provisioned for using the 12-month EL. Under Stage 2, assets that have experienced a significant increase in credit risk (SICR) are provisioned for using the LEL. Under Stage 3, impaired assets are provisioned for using the workout EL.

The criteria for Stage 1 (all assets that do not fall in Stage 2 or Stage 3) as well as Stage 3 (all paired assets) are straightforward. However, the Stage 2 criterion is more complicated as it requires the identification of assets that have a significant increase in credit risk (SICR).

An asset may experience an increase in credit risk for loan exposures, when features deteriorate over time, including:

- Change in age;
- Deterioration of the macro economy;
- Increase of LTV or other features.

SICR may be defined by means of comparing the annual probability of default of an asset i from the year of initial recognition t_0 and the reporting year $t \geq t_0$.

Based on the periodic PD, instrument i is classified as Stage 2 if the estimated periodic PD at reporting date t is significantly higher than the periodic PD at initial recognition t_0:

$$\frac{\widehat{PD_{it}}(t)}{\widehat{PD_{it}}(t_0)} - 1 \geq \alpha$$

where $\alpha > 0$ is a threshold, e.g., 100%, 200%.

Alternatively, we may choose the Lifetime PD (LPD) for the period starting at the reporting date t from the point of view $\tilde{t} \in \{t_0, t\}$ where t_0 is the initial recognition time and t is the current time. LPD is one minus the lifetime survival probability:

$$\widehat{LPD}_{it}(\tilde{t}) = 1 - \prod_{\Delta t=0}^{TTM_{it}} \left(1 - \widehat{P}(D_{it+\Delta t} = 1 | x_{i\tilde{t}-1})\right)$$

Instrument i is classified Stage 2 if the estimated LPD at reporting date t is significantly higher than the estimated LPD at initial recognition t_0:

$$\frac{\widehat{LPD_{it}}(t)}{\widehat{LPD_{it}}(t_0)} - 1 \geq \alpha$$

IFRS 9 does not define α. However, the European Banking Authority has suggested in relation to the EBA 2018 EU-Wide Stress Test: "[. . .] banks shall, as a backstop, assume that Stage 1 assets which experience a threefold increase of annual point-in-time PD compared to the corresponding value at initial recognition (i.e., a 200% relative increase) undergo a significant increase in credit risk (SICR) and hence become Stage 2. Notably, the backstop is defined with reference to the annual PD, not the lifetime PD."

We define initial recognition as the first observation time (`first_time`) and compute the data at initial recognition (`data_initial`) and the start of our assessment period, i.e., period 61 (`data_61`) for our 20 sample loans.

```
listid = data_lifetime.id.unique()
data_inital = data[data['id'].isin(listid)]
data_inital = data_inital.loc[data_inital.time == data_inital.first_time, :].sort_values(['
    id']).reset_index(drop=True)

data_61 = data_lifetime.query('time == 61').copy().sort_values(['id']).reset_index(drop=
    True)
```

We now compute the periodic PD for the 20 analyzed loans for the information set at initial recognition and at the start of our assessment period, i.e., period 61. We then compute the annual PD using

data from the initial recognition and data from period 60. We compute the ratio for the increase and show a scatter plot of the initial and current annual PDs.

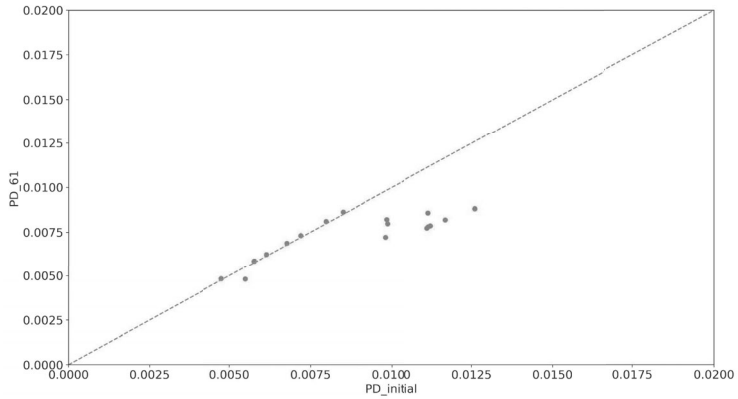
```
PD_initial = pd.DataFrame(model_lr.predict(data_inital), columns = ['PD_initial'])

PD_61 = pd.DataFrame(model_lr.predict(data_61), columns = ['PD_61'])

SICR = PD_initial.merge(PD_61, left_index=True, right_index=True)

SICR['ratio'] = (SICR.PD_61/SICR.PD_initial)-1

plt.scatter('PD_initial', 'PD_61', data=SICR)
plt.ylabel('PD_61')
plt.xlabel('PD_initial')
plt.plot([0,0.02],[0,0.02], color='gray', linestyle='dashed')
plt.xlim((0,0.02))
plt.ylim((0,0.02))
plt.show()
```



The scatters above the diagonal line are loans that have experienced an increase in credit risk as risk factors (e.g., by riding down the term structure as age increases). The scatters below the diagonal line are loans that have decreased in credit risk due to changes of non-age features.

The data reveal that all other ratios are equal or less than 1.6%. If we choose a SICR threshold of 200% no loan would have significantly increased in risk and we would calculate the provision for all loans using a 12-month expected loss definition for IFRS 9. However, in the US we calculate loan provisions for all loans using CECL, i.e., losses during lifetime.