

```
In [1]: import pandas as pd
import numpy as np
from forex_python.converter import CurrencyRates
from datetime import datetime
from pycbrf.toolbox import ExchangeRates
import warnings

warnings.filterwarnings('ignore')

pd.set_option('display.max_columns', None)
pd.options.display.float_format= '{:,.2f}'.format

data = pd.read_excel('Initial data.csv')
data
```

Out[1]:

	DT	DEAL_ID	BALMAIN	BALPRMN	PRODUCT	CURCODENUM	CURCODE	DOCBEGINDATE	ENDDATEWOPROLONG	EN
0	31.03.2020	6943416	nan	45,815.00	Пенсионеры	810	Рубли	24.05.2011	24.05.2013	
1	30.04.2020	6943416	nan	45,815.00	Пенсионеры	810	Рубли	24.05.2011	24.05.2013	
2	31.05.2020	6943416	nan	45,815.00	Пенсионеры	810	Рубли	24.05.2011	24.05.2013	
3	30.06.2020	6943416	nan	45,815.00	Пенсионеры	810	Рубли	24.05.2011	24.05.2013	
4	31.07.2020	6943416	nan	45,815.00	Пенсионеры	810	Рубли	24.05.2011	24.05.2013	
...
6035	30.04.2020	20757944533	nan	45,815.00	Ипотека	810	Рубли	07.10.2019	07.10.2019	
6036	31.05.2020	20757944533	nan	45,815.00	Ипотека	810	Рубли	07.10.2019	07.10.2019	
6037	30.06.2020	20757944533	nan	45,815.00	Ипотека	810	Рубли	07.10.2019	07.10.2019	
6038	31.07.2020	20757944533	nan	45,815.00	Ипотека	810	Рубли	07.10.2019	07.10.2019	
6039	31.08.2020	20757944533	nan	45,815.00	Ипотека	810	Рубли	07.10.2019	07.10.2019	

6040 rows × 11 columns

Finding the exchange rate by parcing CB site

```
In [2]: def get_currency(df, currency):
    if currency == 'RUB':
        df['CUR_RATE'] = 1
    elif currency == 'USD' or currency == 'EUR':
        df['CUR_RATE'] = df.apply(lambda x: float(str(ExchangeRates(x['DT'])[currency].rate).split())[0])
    return df

def count_provision(df):
    df['PROVISION'] = 0
    for column in ['SUMRESERVEMAINRPBU', 'SUMRESERVEPRMNRPBU', 'SUMRESERVEPRMN', 'SUMRESERVEPRPR']:
        df[column] = df[column].fillna(0)
        df['PROVISION'] += df[column]

    return df

def count_delta_provision(df):
    res_df = pd.DataFrame()
    credit_ids = df['DEAL_ID'].unique()
    for credit_id in credit_ids:
        tmp_df = df[df['DEAL_ID'] == credit_id]
        tmp_df = tmp_df.sort_values(by = ['DT'])
        tmp_df.reset_index().drop(columns = ['index'])
        deltas = [tmp_df.iloc[0]['PROVISION']]
        for i in range(1, tmp_df.shape[0]):
            prev_row = tmp_df.iloc[i-1]
            init_row = tmp_df.iloc[i]
            deltas.append(init_row['PROVISION'] - prev_row['PROVISION'])
        tmp_df['DELTA_PROVISION'] = deltas
        res_df = res_df.append(tmp_df)
    return res_df

def count_dA(df):
    res_df = pd.DataFrame()
    credit_ids = df['DEAL_ID'].unique()
    for credit id in credit ids:
```

```

    tmp_df = df[df['DEAL_ID'] == credit_id]
    tmp_df = tmp_df.sort_values(by = ['DT'])
    tmp_df.reset_index().drop(columns = ['index'])
    for column in ['MAINRESTCUR', 'PRMNRESTCUR']:
        tmp_df[column] = tmp_df[column].fillna(0)
    dA_s = [tmp_df.iloc[0]['MAINRESTCUR'] + tmp_df.iloc[0]['PRMNRESTCUR']]
    prev_value = dA_s[0]
    for i in range(1, tmp_df.shape[0]):
        prev_row = tmp_df.iloc[i-1]
        init_row = tmp_df.iloc[i]
        init_value = tmp_df.iloc[i]['MAINRESTCUR'] + tmp_df.iloc[i]['PRMNRESTCUR']
        dA_s.append(init_value - prev_value)
        prev_value = init_value
    tmp_df['dA'] = dA_s
    res_df = res_df.append(tmp_df)
    return res_df

def count_dC(df):
    res_df = pd.DataFrame()
    credit_ids = df['DEAL_ID'].unique()

    for credit_id in credit_ids:
        tmp_df = df[df['DEAL_ID'] == credit_id]
        tmp_df = tmp_df.sort_values(by = ['DT'])
        tmp_df.reset_index().drop(columns = ['index'])
        dC_s = [0]
        for i in range(1, tmp_df.shape[0]):
            prev_row = tmp_df.iloc[i-1]
            init_row = tmp_df.iloc[i]
            dC_s.append(init_row['CUR_RATE'] - prev_row['CUR_RATE'])
        tmp_df['dC'] = dC_s
        res_df = res_df.append(tmp_df)
    return res_df

def count_dR(df):
    res_df = pd.DataFrame()
    credit_ids = df['DEAL_ID'].unique()
    for credit_id in credit_ids:
        tmp_df = df[df['DEAL_ID'] == credit_id]
        tmp_df = tmp_df.sort_values(by = ['DT'])
        tmp_df.reset_index().drop(columns = ['index'])
        dR_s = [0]
        for i in range(1, tmp_df.shape[0]):
            prev_row = tmp_df.iloc[i-1]
            init_row = tmp_df.iloc[i]
            dR_s.append(init_row['RESERVERATERPBU'] - prev_row['RESERVERATERPBU'])
        tmp_df['dR'] = dR_s
        res_df = res_df.append(tmp_df)
    return res_df

def count_dPA(df):
    res_df = pd.DataFrame()
    credit_ids = df['DEAL_ID'].unique()
    for credit_id in credit_ids:
        tmp_df = df[df['DEAL_ID'] == credit_id]
        tmp_df = tmp_df.sort_values(by = ['DT'])
        tmp_df.reset_index().drop(columns = ['index'])

        dPA_s = [tmp_df.iloc[0]['dA'] * (tmp_df.iloc[0]['RESERVERATERPBU'] / 100) * tmp_df.iloc[0]['CUR_
        for i in range(tmp_df.shape[0]-1):
            next_row = tmp_df.iloc[i+1]
            init_row = tmp_df.iloc[i]
            dPA_s.append(next_row['dA'] * (init_row['RESERVERATERPBU'] / 100) * init_row['CUR_RATE'])

        tmp_df['dPA'] = dPA_s
        res_df = res_df.append(tmp_df)
    return res_df

df['dPA'] = df['dA'] * (df['RESERVERATERPBU'] / 100) * df['CUR_RATE']
return df

def count_dPR(df):
    res_df = pd.DataFrame()
    credit_ids = df['DEAL_ID'].unique()
    for credit_id in credit_ids:
        tmp_df = df[df['DEAL_ID'] == credit_id]

```

```

    tmp_dt = tmp_dt.sort_values(by = ['DT'])
    tmp_df.reset_index().drop(columns = ['index'])
    dPR_s = [0]

    for column in ['MAINRESTCUR', 'PRMNRESTCUR']:
        tmp_df[column] = tmp_df[column].fillna(0)

    for i in range(1, tmp_df.shape[0]):
        init_row = tmp_df.iloc[i]
        dPR_s.append((init_row['MAINRESTCUR'] + init_row['PRMNRESTCUR']) * (init_row['dR'] / 100) *

    tmp_df['dPR'] = dPR_s
    res_df = res_df.append(tmp_df)
    return res_df

def count_dPC(df):
    res_df = pd.DataFrame()
    credit_ids = df['DEAL_ID'].unique()
    for credit_id in credit_ids:
        tmp_df = df[df['DEAL_ID'] == credit_id]
        tmp_df = tmp_df.sort_values(by = ['DT'])
        tmp_df.reset_index().drop(columns = ['index'])
        dPC_s = [0]

        for column in ['RESERVERATERPBU', 'MAINRESTCUR', 'PRMNRESTCUR']:
            tmp_df[column] = tmp_df[column].fillna(0)

        for i in range(1, tmp_df.shape[0]):
            init_row = tmp_df.iloc[i]
            dPC_s.append(((init_row['MAINRESTCUR'] + init_row['PRMNRESTCUR']) * init_row['RESERVERATERPBU

        tmp_df['dPC'] = dPC_s
        res_df = res_df.append(tmp_df)
    return res_df

def count_dXR(df):
    df['dXR'] = df['DELTA_PROVISION'].copy()
    for column in ['dPA', 'dPR', 'dPC']:
        df['dXR'] -= df[column]
    return df

def calculate_deltas(df):
    df = count_provision(df)
    df = count_delta_provision(df)
    df = count_dA(df)
    df = count_dC(df)
    df = count_dR(df)
    df = count_dPA(df)
    df = count_dPR(df)
    df = count_dPC(df)
    df = count_dXR(df)
    return df

```

```
In [3]: cur_codes = {'RUB': 810,
                    'EUR': 978,
                    'USD': 840
                    }

data['DT'] = pd.to_datetime(data['DT'])

rub_df = get_currency(data[data['CURCODENUM'] == cur_codes['RUB']], 'RUB')
usd_df = get_currency(data[data['CURCODENUM'] == cur_codes['USD']], 'USD')
eur_df = get_currency(data[data['CURCODENUM'] == cur_codes['EUR']], 'EUR')

df = rub_df.copy()
df = df.append(usd_df).append(eur_df)
df
```

Out[3]:

	DT	DEAL_ID	BALMAIN	BALPRMN	PRODUCT	CURCODENUM	CURCODE	DOCBEGINDATE	ENDDATEWOPROLONG	ENDDATEW
0	2020-03-31	6943416	nan	45,815.00	Пенсионеры	810	Рубли	24.05.2011	24.05.2013	
1	2020-04-30	6943416	nan	45,815.00	Пенсионеры	810	Рубли	24.05.2011	24.05.2013	
2	2020-05-31	6943416	nan	45,815.00	Пенсионеры	810	Рубли	24.05.2011	24.05.2013	
3	2020-06-30	6943416	nan	45,815.00	Пенсионеры	810	Рубли	24.05.2011	24.05.2013	
4	2020-07-31	6943416	nan	45,815.00	Пенсионеры	810	Рубли	24.05.2011	24.05.2013	
...
2689	2020-04-30	63764213	nan	45,815.00	Ипотека	978	Евро	24.01.2013	11.01.2016	
2690	2020-05-31	63764213	nan	45,815.00	Ипотека	978	Евро	24.01.2013	11.01.2016	
2691	2020-06-30	63764213	nan	45,815.00	Ипотека	978	Евро	24.01.2013	11.01.2016	
2692	2020-07-31	63764213	nan	45,815.00	Ипотека	978	Евро	24.01.2013	11.01.2016	
2693	2020-08-31	63764213	nan	45,815.00	Ипотека	978	Евро	24.01.2013	11.01.2016	

6040 rows × 32 columns

Breakdown by number of credits

```
In [4]: grouped_by_deal_id = df[['DT', 'DEAL_ID']].groupby(by = ['DEAL_ID']).count()
grouped_by_deal_id = grouped_by_deal_id.reset_index()

credit_types = {}
credits_dfs = []

for i in range(1, 7):
    tmp_df = grouped_by_deal_id[grouped_by_deal_id['DT'] == i]
    credit_types[i] = tmp_df['DEAL_ID'].unique()

for i in range(1, 7):
    credits_dfs.append(df[df['DEAL_ID'].isin(credit_types[i])].reset_index().drop(columns = ['index']))

In [5]: credits_dfs_new = []
```

1 credit

```
In [6]: cr_1 = calculate_deltas(credits_dfs[0])
credits_dfs_new.append(cr_1)
cr_1
```

Out[6]:

	DT	DEAL_ID	BALMAIN	BALPRMN	PRODUCT	CURCODENUM	CURCODE	DOCBEGINDATE	ENDDATEWOPROLONG	ENDDATEWITHI
0	2020-03-31	62944933	47,801.00	nan	Ипотека	810	Рубли	29.05.2013	29.05.2020	4

1	2020-03-31	63049299	47,801.00	nan	Ипотека	810	Рубли	09.11.2006	08.11.2021	(
2	2020-03-31	63060942	nan	nan	Потреб	810	Рубли	25.12.2013	23.12.2016	;
3	2020-03-31	63245372	nan	45,815.00	Ипотека	810	Рубли	21.02.2012	21.02.2037	;
4	2020-03-31	63246064	45,507.00	45,815.00	Ипотека	810	Рубли	12.03.2014	12.03.2034	.
5	2020-03-31	63901471	45,507.00	nan	Ипотека	810	Рубли	13.07.2012	13.07.2032	.
6	2020-03-31	63979552	47,801.00	nan	Ипотека	810	Рубли	16.10.2013	16.10.2033	.
7	2020-04-30	63992398	nan	nan	Потреб	810	Рубли	16.05.2013	16.05.2023	.
8	2020-03-31	63997211	47,801.00	nan	Ипотека	810	Рубли	11.12.2013	11.12.2028	.
9	2020-03-31	64032722	45,507.00	nan	Ипотека	810	Рубли	25.07.2016	28.05.2028	;
10	2020-03-31	64038240	47,801.00	45,815.00	Ипотека	810	Рубли	21.02.2012	31.01.2027	;
11	2020-03-31	64055617	47,801.00	nan	Ипотека	810	Рубли	06.05.2014	04.05.2034	(
12	2020-03-31	64143881	47,801.00	nan	Ипотека	810	Рубли	03.05.2012	03.05.2032	(
13	2020-03-31	64223718	47,801.00	nan	Ипотека	810	Рубли	11.08.2008	10.08.2023	.
14	2020-03-31	64274905	47,801.00	nan	Ипотека	810	Рубли	22.07.2014	21.07.2034	;
15	2020-03-31	64298430	nan	45,815.00	Ипотека	810	Рубли	08.05.2008	07.05.2020	(
16	2020-03-31	64421403	nan	45,815.00	Ипотека	810	Рубли	16.02.2007	15.02.2026	.
17	2020-03-31	64879050	45,507.00	nan	Ипотека	810	Рубли	29.05.2014	29.05.2034	;
18	2020-03-31	64894641	47,801.00	nan	Ипотека	810	Рубли	27.04.2012	31.03.2022	;
19	2020-03-31	64932762	45,507.00	nan	Ипотека	810	Рубли	17.06.2016	17.05.2021	.
20	2020-03-31	64976938	47,801.00	nan	Ипотека	810	Рубли	22.11.2013	22.11.2028	;
21	2020-03-31	65049762	47,801.00	nan	Ипотека	810	Рубли	27.02.2014	27.02.2044	;
22	2020-03-31	65246786	nan	45,815.00	Ипотека	810	Рубли	13.05.2008	12.05.2026	.
23	2020-03-31	65299419	47,801.00	nan	Ипотека	810	Рубли	04.07.2013	04.07.2033	(
24	2020-03-31	65302136	45,507.00	nan	Ипотека	810	Рубли	28.02.2014	21.02.2029	;
25	2020-03-31	65380583	45,507.00	nan	Ипотека	810	Рубли	24.06.2015	21.02.2029	;
26	2020-03-31	65400870	nan	45,815.00	Ипотека	810	Рубли	23.04.2012	23.04.2032	;
27	2020-03-31	65404591	47,801.00	nan	Ипотека	810	Рубли	28.12.2011	28.12.2041	;
28	2020-03-31	65465034	47,801.00	nan	Ипотека	810	Рубли	03.08.2012	03.08.2027	(
29	2020-03-31	62918321	nan	45,815.00	Ипотека	840	Доллары США	21.12.2007	20.12.2022	;
30	2020-03-31	63795784	nan	45,815.00	Ипотека	840	Доллары США	03.04.2008	02.04.2018	(
31	2020-03-31	64823346	nan	45,815.00	Ипотека	840	Доллары США	14.02.2008	13.02.2033	.
32	2020-03-31	65472777	nan	45,815.00	Ипотека	840	Доллары США	12.12.2007	11.12.2027	.

2 credits

```
In [7]: cr_2 = calculate_deltas(credits_dfs[1])
credits_dfs_new.append(cr_2)
cr_2
```

Out[7]:

	DT	DEAL_ID	BALMAIN	BALPRMN	PRODUCT	CURCODENUM	CURCODE	DOCBEGINDATE	ENDDATEWOPROLONG	ENDDATEWIT
0	2020-03-31	28220635	nan	45,815.00	Пенсионеры	810	Рубли	10.06.2014	20.06.2017	
1	2020-04-30	28220635	nan	45,815.00	Пенсионеры	810	Рубли	10.06.2014	20.06.2017	
2	2020-06-30	62909499	nan	nan	Ипотека	810	Рубли	29.08.2012	29.08.2027	
3	2020-08-31	62909499	nan	nan	Ипотека	810	Рубли	29.08.2012	29.08.2027	
4	2020-03-31	63043550	47,801.00	nan	Ипотека	810	Рубли	29.10.2013	29.10.2038	
5	2020-04-30	63043550	47,801.00	nan	Ипотека	810	Рубли	29.10.2013	29.10.2038	
6	2020-03-31	64106332	47,801.00	nan	Ипотека	810	Рубли	04.10.2011	04.10.2026	
7	2020-04-30	64106332	47,801.00	nan	Ипотека	810	Рубли	04.10.2011	04.10.2026	
8	2020-03-31	64219497	47,801.00	nan	Ипотека	810	Рубли	19.03.2007	18.03.2022	
9	2020-04-30	64219497	47,801.00	nan	Ипотека	810	Рубли	19.03.2007	18.03.2022	
10	2020-03-31	64739949	47,801.00	nan	Потреб	810	Рубли	15.12.2006	14.12.2021	
11	2020-04-30	64739949	47,801.00	nan	Потреб	810	Рубли	15.12.2006	14.12.2021	

3 credits

```
In [8]: cr_3 = calculate_deltas(credits_dfs[2])
credits_dfs_new.append(cr_3)
cr_3
```

Out[8]:

	DT	DEAL_ID	BALMAIN	BALPRMN	PRODUCT	CURCODENUM	CURCODE	DOCBEGINDATE	ENDDATEWOPROLONG	ENDDATEWITHI
0	2020-03-31	62873186	47,801.00	nan	Ипотека	810	Рубли	14.12.2012	14.12.2032	
1	2020-04-30	62873186	47,801.00	nan	Ипотека	810	Рубли	14.12.2012	14.12.2032	
2	2020-05-31	62873186	47,801.00	nan	Ипотека	810	Рубли	14.12.2012	14.12.2032	
3	2020-03-31	62995490	47,801.00	45,815.00	Ипотека	810	Рубли	26.09.2011	31.08.2026	;
4	2020-04-30	62995490	47,801.00	45,815.00	Ипотека	810	Рубли	26.09.2011	31.08.2026	;
5	2020-05-31	62995490	47,801.00	nan	Ипотека	810	Рубли	26.09.2011	31.08.2026	;
6	2020-03-31	63093260	47,801.00	nan	Ипотека	810	Рубли	22.08.2012	22.08.2020	;
7	2020-04-30	63093260	47,801.00	nan	Ипотека	810	Рубли	22.08.2012	22.08.2020	;
8	2020-05-31	63093260	47,801.00	nan	Ипотека	810	Рубли	22.08.2012	22.08.2020	;
9	2020-03-31	63414135	47,801.00	nan	Ипотека	810	Рубли	08.08.2013	08.08.2033	(
10	2020-04-30	63414135	47,801.00	nan	Ипотека	810	Рубли	08.08.2013	08.08.2033	(
11	2020-05-31	63414135	47,801.00	nan	Ипотека	810	Рубли	08.08.2013	08.08.2033	(
12	2020-03-31	63550695	47,801.00	nan	Ипотека	810	Рубли	13.11.2012	13.11.2032	
13	2020-04-30	63550695	47,801.00	nan	Ипотека	810	Рубли	13.11.2012	13.11.2032	
14	2020-05-31	63550695	47,801.00	nan	Ипотека	810	Рубли	13.11.2012	13.11.2032	

15	2020-03-31	63650405	47,801.00	nan	Ипотека	810	Рубли	13.06.2013	13.06.2020	.
16	2020-04-30	63650405	47,801.00	nan	Ипотека	810	Рубли	13.06.2013	13.06.2020	.
17	2020-05-31	63650405	47,801.00	nan	Ипотека	810	Рубли	13.06.2013	13.06.2020	.
18	2020-03-31	63690247	47,801.00	nan	Ипотека	810	Рубли	01.06.2012	01.06.2027	(
19	2020-04-30	63690247	47,801.00	nan	Ипотека	810	Рубли	01.06.2012	01.06.2027	(
20	2020-05-31	63690247	47,801.00	nan	Ипотека	810	Рубли	01.06.2012	01.06.2027	(
21	2020-03-31	63805313	47,801.00	nan	Ипотека	810	Рубли	16.08.2011	16.08.2022	.
22	2020-04-30	63805313	47,801.00	nan	Ипотека	810	Рубли	16.08.2011	16.08.2022	.
23	2020-05-31	63805313	47,801.00	nan	Ипотека	810	Рубли	16.08.2011	16.08.2022	.
24	2020-03-31	64063344	45,507.00	nan	Ипотека	810	Рубли	06.07.2011	06.07.2026	(
25	2020-04-30	64063344	45,507.00	45,815.00	Ипотека	810	Рубли	06.07.2011	06.07.2026	(
26	2020-05-31	64063344	45,507.00	nan	Ипотека	810	Рубли	06.07.2011	06.07.2026	(
27	2020-03-31	64293721	45,507.00	nan	Ипотека	810	Рубли	18.03.2014	18.03.2029	.
28	2020-04-30	64293721	45,507.00	nan	Ипотека	810	Рубли	18.03.2014	18.03.2029	.
29	2020-05-31	64293721	45,507.00	nan	Ипотека	810	Рубли	18.03.2014	18.03.2029	.
30	2020-03-31	64882141	45,507.00	nan	Ипотека	810	Рубли	04.06.2014	04.06.2029	(
31	2020-04-30	64882141	45,507.00	nan	Ипотека	810	Рубли	04.06.2014	04.06.2029	(
32	2020-05-31	64882141	45,507.00	nan	Ипотека	810	Рубли	04.06.2014	04.06.2029	(
33	2020-03-31	65250417	47,801.00	nan	Ипотека	810	Рубли	04.05.2010	30.04.2030	;
34	2020-04-30	65250417	47,801.00	nan	Ипотека	810	Рубли	04.05.2010	30.04.2030	;
35	2020-05-31	65250417	47,801.00	nan	Ипотека	810	Рубли	04.05.2010	30.04.2030	;
36	2020-03-31	65427988	47,801.00	nan	Ипотека	810	Рубли	24.07.2013	24.07.2033	;
37	2020-04-30	65427988	47,801.00	nan	Ипотека	810	Рубли	24.07.2013	24.07.2033	;
38	2020-05-31	65427988	47,801.00	nan	Ипотека	810	Рубли	24.07.2013	24.07.2033	;
39	2020-03-31	65470962	47,801.00	nan	Ипотека	810	Рубли	09.04.2014	09.04.2024	(
40	2020-04-30	65470962	47,801.00	nan	Ипотека	810	Рубли	09.04.2014	09.04.2024	(
41	2020-05-31	65470962	47,801.00	nan	Ипотека	810	Рубли	09.04.2014	09.04.2024	(

4 credits

In [9]:

```
cr_4 = calculate_deltas(credits_dfs[3])
credits_dfs_new.append(cr_4)
cr_4.head(60)
```

9	2020-04-30	63242168	47,801.00	nan	Ипотека	810	Рубли	14.09.2012	14.09.2027	:
10	2020-05-31	63242168	47,801.00	nan	Ипотека	810	Рубли	14.09.2012	14.09.2027	:
11	2020-06-30	63242168	47,801.00	nan	Ипотека	810	Рубли	14.09.2012	14.09.2027	:
12	2020-03-31	63309893	45,507.00	nan	Ипотека	810	Рубли	30.12.2015	20.12.2030	:
13	2020-04-30	63309893	45,507.00	nan	Ипотека	810	Рубли	30.12.2015	20.12.2030	:
14	2020-05-31	63309893	45,507.00	nan	Ипотека	810	Рубли	30.12.2015	20.12.2030	:
15	2020-06-30	63309893	45,507.00	nan	Ипотека	810	Рубли	30.12.2015	20.12.2030	:
16	2020-03-31	63446305	45,507.00	45,815.00	Ипотека	810	Рубли	30.05.2012	30.05.2038	:
17	2020-04-30	63446305	45,507.00	45,815.00	Ипотека	810	Рубли	30.05.2012	30.05.2038	:

5 credits

In [10]:

```
cr_5 = calculate_deltas(credits_dfs[4])
credits_dfs_new.append(cr_5)
cr_5
```

Out[10]:

	DT	DEAL_ID	BALMAIN	BALPRMN	PRODUCT	CURCODENUM	CURCODE	DOCBEGINDATE	ENDDATEWOPROLONG	ENDDATEWITHI
0	2020-03-31	62837355	45,507.00	nan	Ипотека	810	Рубли	24.05.2012	24.05.2022	:
1	2020-04-30	62837355	45,507.00	nan	Ипотека	810	Рубли	24.05.2012	24.05.2022	:
2	2020-05-31	62837355	45,507.00	nan	Ипотека	810	Рубли	24.05.2012	24.05.2022	:
3	2020-06-30	62837355	45,507.00	nan	Ипотека	810	Рубли	24.05.2012	24.05.2022	:
4	2020-07-31	62837355	45,507.00	nan	Ипотека	810	Рубли	24.05.2012	24.05.2022	:
...
80	2020-03-31	65445195	nan	45,815.00	Потреб	810	Рубли	05.03.2013	28.02.2018	:
81	2020-04-30	65445195	nan	45,815.00	Потреб	810	Рубли	05.03.2013	28.02.2018	:
82	2020-05-31	65445195	nan	45,815.00	Потреб	810	Рубли	05.03.2013	28.02.2018	:
83	2020-06-30	65445195	nan	45,815.00	Потреб	810	Рубли	05.03.2013	28.02.2018	:
84	2020-07-31	65445195	nan	45,815.00	Потреб	810	Рубли	05.03.2013	28.02.2018	:

85 rows × 41 columns

In [11]:

```
cr_5[cr_5['DEAL_ID'] == 65445195]
```

Out[11]:

	DT	DEAL_ID	BALMAIN	BALPRMN	PRODUCT	CURCODENUM	CURCODE	DOCBEGINDATE	ENDDATEWOPROLONG	ENDDATEWITHI
80	2020-03-31	65445195	nan	45,815.00	Потреб	810	Рубли	05.03.2013	28.02.2018	:
81	2020-04-30	65445195	nan	45,815.00	Потреб	810	Рубли	05.03.2013	28.02.2018	:
82	2020-05-31	65445195	nan	45,815.00	Потреб	810	Рубли	05.03.2013	28.02.2018	:
83	2020-06-30	65445195	nan	45,815.00	Потреб	810	Рубли	05.03.2013	28.02.2018	:
84	2020-07-31	65445195	nan	45,815.00	Потреб	810	Рубли	05.03.2013	28.02.2018	:

6 credits

In [12]:

cr_6 = calculate_deltas(credits_dfs[5])
credits_dfs_new.append(cr_6)
cr_6

Out [12]:

	DT	DEAL_ID	BALMAIN	BALPRMN	PRODUCT	CURCODENUM	CURCODE	DOCBEGINDATE	ENDDATEWOPROLONG	ENDDATEWIT
0	2020-03-31	6943416	nan	45,815.00	Пенсионеры	810	Рубли	24.05.2011	24.05.2013	
1	2020-04-30	6943416	nan	45,815.00	Пенсионеры	810	Рубли	24.05.2011	24.05.2013	
2	2020-05-31	6943416	nan	45,815.00	Пенсионеры	810	Рубли	24.05.2011	24.05.2013	
3	2020-06-30	6943416	nan	45,815.00	Пенсионеры	810	Рубли	24.05.2011	24.05.2013	
4	2020-07-31	6943416	nan	45,815.00	Пенсионеры	810	Рубли	24.05.2011	24.05.2013	
...
5767	2020-04-30	63764213	nan	45,815.00	Ипотека	978	Евро	24.01.2013	11.01.2016	
5768	2020-05-31	63764213	nan	45,815.00	Ипотека	978	Евро	24.01.2013	11.01.2016	
5769	2020-06-30	63764213	nan	45,815.00	Ипотека	978	Евро	24.01.2013	11.01.2016	
5770	2020-07-31	63764213	nan	45,815.00	Ипотека	978	Евро	24.01.2013	11.01.2016	
5771	2020-08-31	63764213	nan	45,815.00	Ипотека	978	Евро	24.01.2013	11.01.2016	

5772 rows × 41 columns

Checking single credits for correctness

In [13]:

cr_6[cr_6['DEAL_ID'] == 62824709]

Out [13]:

	DT	DEAL_ID	BALMAIN	BALPRMN	PRODUCT	CURCODENUM	CURCODE	DOCBEGINDATE	ENDDATEWOPROLONG	ENDDATEWIT
1254	2020-03-31	62824709	47,801.00	nan	Ипотека	810	Рубли	20.08.2013	20.08.2028	
1255	2020-04-30	62824709	47,801.00	nan	Ипотека	810	Рубли	20.08.2013	20.08.2028	
1256	2020-05-31	62824709	47,801.00	nan	Ипотека	810	Рубли	20.08.2013	20.08.2028	
1257	2020-06-30	62824709	47,801.00	45,815.00	Ипотека	810	Рубли	20.08.2013	20.08.2028	
1258	2020-07-31	62824709	47,801.00	nan	Ипотека	810	Рубли	20.08.2013	20.08.2028	
1259	2020-08-31	62824709	47,801.00	45,815.00	Ипотека	810	Рубли	20.08.2013	20.08.2028	

Combine 6 datasets into one

```
In [14]: res_df = pd.DataFrame()
for i in range(len(credits_dfs_new)):
    res_df = res_df.append(credits_dfs_new[i])
res_df
```

Out[14]:

	DT	DEAL_ID	BALMAIN	BALPRMN	PRODUCT	CURCODENUM	CURCODE	DOCBEGINDATE	ENDDATEWOPROLONG	ENDDATEWIT
0	2020-03-31	62944933	47,801.00	nan	Ипотека	810	Рубли	29.05.2013	29.05.2020	
1	2020-03-31	63049299	47,801.00	nan	Ипотека	810	Рубли	09.11.2006	08.11.2021	
2	2020-03-31	63060942	nan	nan	Потреб	810	Рубли	25.12.2013	23.12.2016	
3	2020-03-31	63245372	nan	45,815.00	Ипотека	810	Рубли	21.02.2012	21.02.2037	
4	2020-03-31	63246064	45,507.00	45,815.00	Ипотека	810	Рубли	12.03.2014	12.03.2034	
...
5767	2020-04-30	63764213	nan	45,815.00	Ипотека	978	Евро	24.01.2013	11.01.2016	
5768	2020-05-31	63764213	nan	45,815.00	Ипотека	978	Евро	24.01.2013	11.01.2016	
5769	2020-06-30	63764213	nan	45,815.00	Ипотека	978	Евро	24.01.2013	11.01.2016	
5770	2020-07-31	63764213	nan	45,815.00	Ипотека	978	Евро	24.01.2013	11.01.2016	
5771	2020-08-31	63764213	nan	45,815.00	Ипотека	978	Евро	24.01.2013	11.01.2016	

6040 rows × 41 columns

```
In [15]: res_df['Month'] = res_df['DT'].dt.month
res_df
```

Out[15]:

	DT	DEAL_ID	BALMAIN	BALPRMN	PRODUCT	CURCODENUM	CURCODE	DOCBEGINDATE	ENDDATEWOPROLONG	ENDDATEWIT
0	2020-03-31	62944933	47,801.00	nan	Ипотека	810	Рубли	29.05.2013	29.05.2020	
1	2020-03-31	63049299	47,801.00	nan	Ипотека	810	Рубли	09.11.2006	08.11.2021	
2	2020-03-31	63060942	nan	nan	Потреб	810	Рубли	25.12.2013	23.12.2016	
3	2020-03-31	63245372	nan	45,815.00	Ипотека	810	Рубли	21.02.2012	21.02.2037	
4	2020-03-31	63246064	45,507.00	45,815.00	Ипотека	810	Рубли	12.03.2014	12.03.2034	
...
5767	2020-04-30	63764213	nan	45,815.00	Ипотека	978	Евро	24.01.2013	11.01.2016	
5768	2020-05-31	63764213	nan	45,815.00	Ипотека	978	Евро	24.01.2013	11.01.2016	
5769	2020-06-30	63764213	nan	45,815.00	Ипотека	978	Евро	24.01.2013	11.01.2016	
5770	2020-07-31	63764213	nan	45,815.00	Ипотека	978	Евро	24.01.2013	11.01.2016	
5771	2020-08-31	63764213	nan	45,815.00	Ипотека	978	Евро	24.01.2013	11.01.2016	

6040 rows × 42 columns

Grouping and aggregating by date

```
In [19]: cols = ['PROVISION', 'DELTA_PROVISION', 'dA', 'dC', 'dR', 'dPA', 'dPR', 'dPC', 'dXR', 'Month']
agg = res_df[cols].groupby(by = ['Month']).sum()
```

```
In [21]: agg.to_csv('Итог_месяцы.csv', index = False, sep=';', encoding='utf-8-sig')
```

