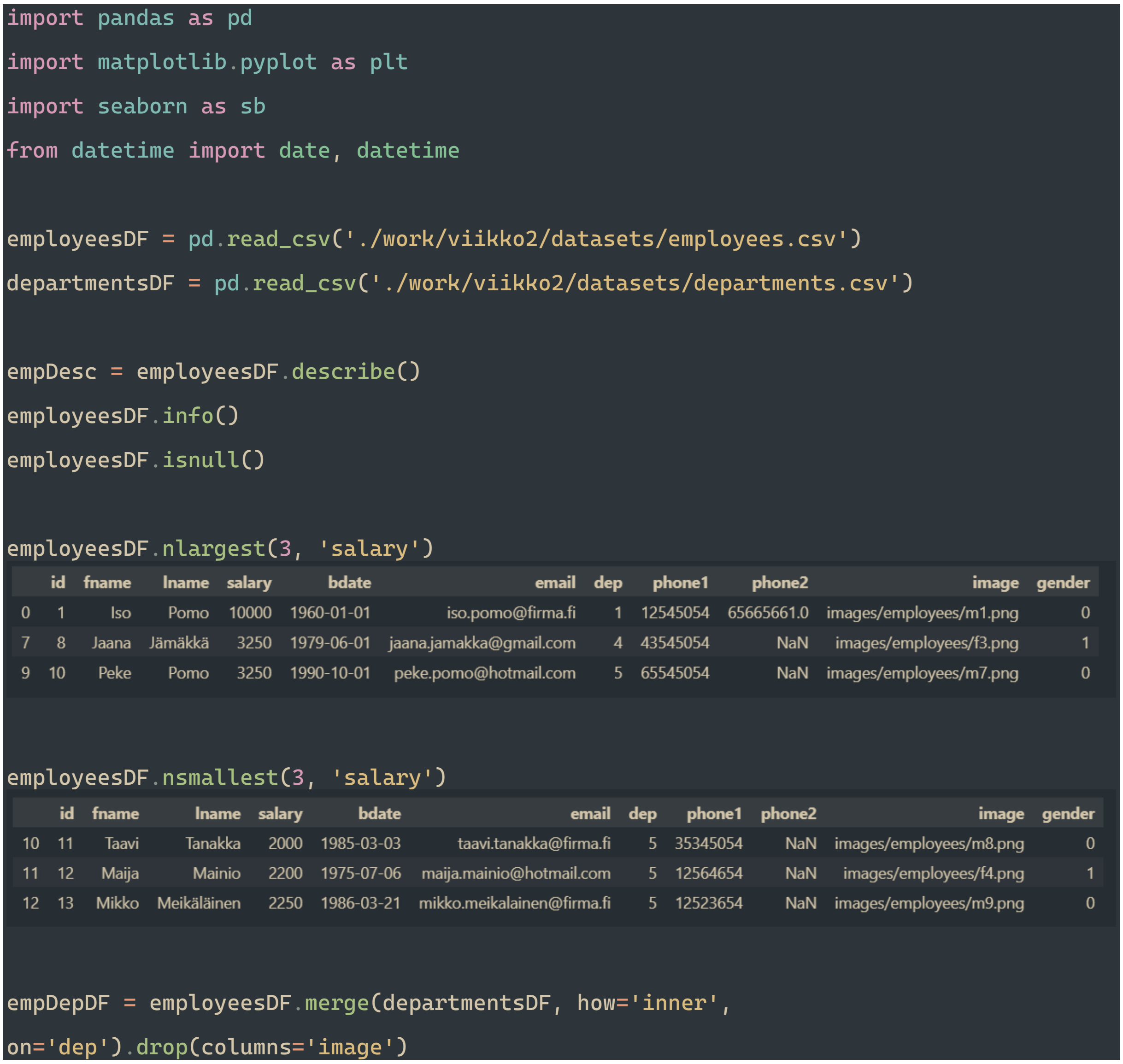
Viikko 36 -tehtävät

Tehtävä 1



```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
from datetime import date, datetime

employeesDF = pd.read_csv('./work/viikko2/datasets/employees.csv')
departmentsDF = pd.read_csv('./work/viikko2/datasets/departments.csv')

empDesc = employeesDF.describe()
employeesDF.info()
employeesDF.isnull()

employeesDF.nlargest(3, 'salary')
```

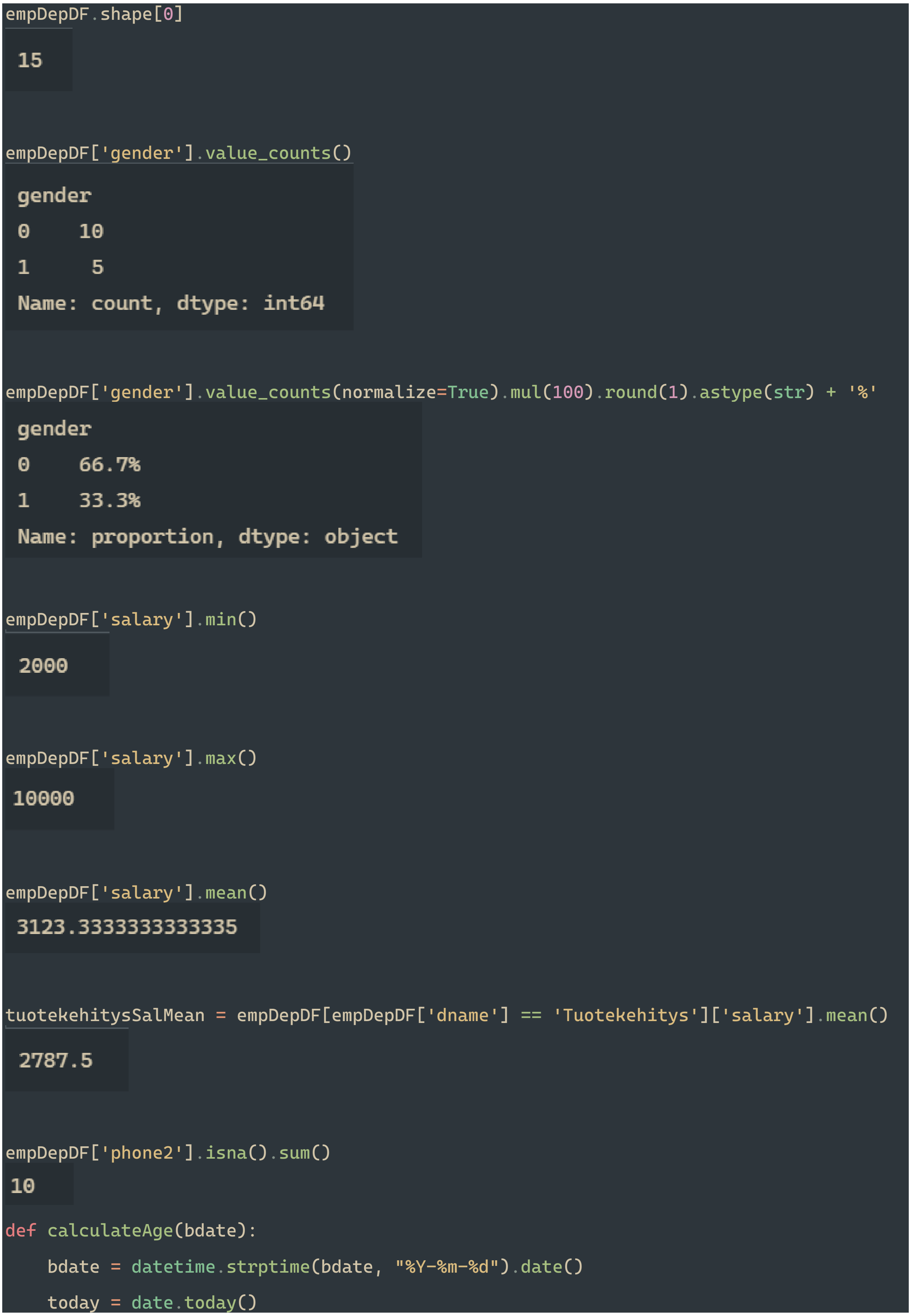
	id	fname	lname	salary	bdate	email	dep	phone1	phone2	image	gender
0	1	Iso	Pomo	10000	1960-01-01	iso.pomo@firma.fi	1	12545054	65665661.0	images/employees/m1.png	0
7	8	Jaana	Jämäkkä	3250	1979-06-01	jaana.jamakka@gmail.com	4	43545054	NaN	images/employees/f3.png	1
9	10	Peke	Pomo	3250	1990-10-01	peke.pomo@hotmail.com	5	65545054	NaN	images/employees/m7.png	0

```
employeesDF.nsmallest(3, 'salary')
```

	id	fname	lname	salary	bdate	email	dep	phone1	phone2	image	gender
10	11	Taavi	Tanakka	2000	1985-03-03	taavi.tanakka@firma.fi	5	35345054	NaN	images/employees/m8.png	0
11	12	Maija	Mainio	2200	1975-07-06	maija.mainio@hotmail.com	5	12564654	NaN	images/employees/f4.png	1
12	13	Mikko	Meikäläinen	2250	1986-03-21	mikko.meikalainen@firma.fi	5	12523654	NaN	images/employees/m9.png	0

```
empDepDF = employeesDF.merge(departmentsDF, how='inner',
on='dep').drop(columns='image')
```

Tehtävä 2



```
empDepDF.shape[0]
```

```
15
```

```
empDepDF['gender'].value_counts()
```

```
gender
0    10
1     5
Name: count, dtype: int64
```

```
empDepDF['gender'].value_counts(normalize=True).mul(100).round(1).astype(str) + '%'
```

```
gender
0    66.7%
1    33.3%
Name: proportion, dtype: object
```

```
empDepDF['salary'].min()
```

```
2000
```

```
empDepDF['salary'].max()
```

```
10000
```

```
empDepDF['salary'].mean()
```

```
3123.3333333333335
```

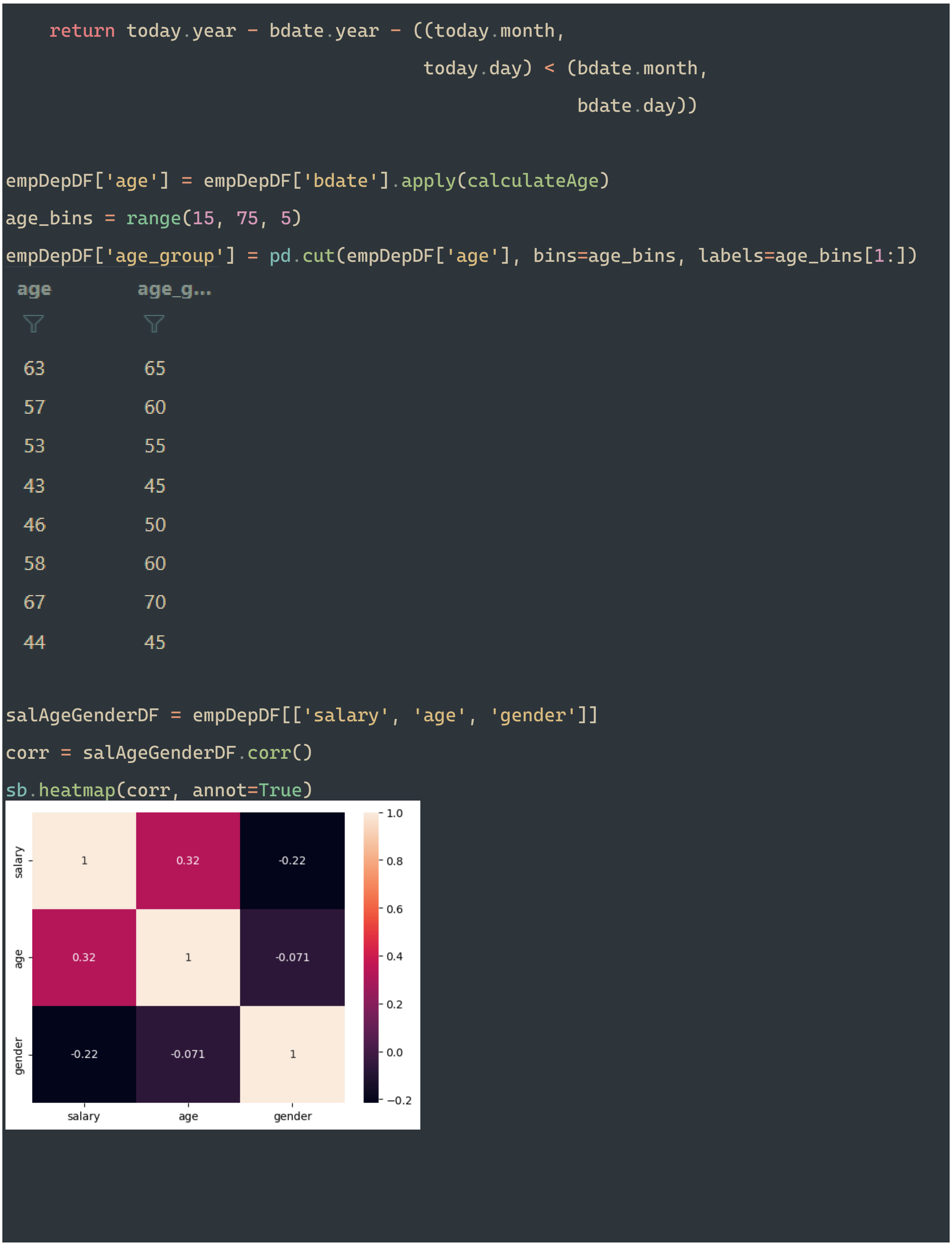
```
tuotekehitysSalMean = empDepDF[empDepDF['dname'] == 'Tuotekehitys']['salary'].mean()
```

```
2787.5
```

```
empDepDF['phone2'].isna().sum()
```

```
10
```

```
def calculateAge(bdate):
    bdate = datetime.strptime(bdate, "%Y-%m-%d").date()
    today = date.today()
```



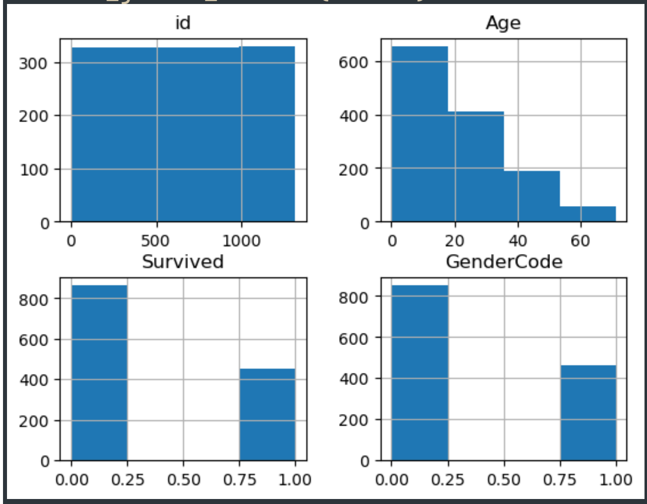
```
    return today.year - bdate.year - ((today.month,
                                      today.day) < (bdate.month,
                                                    bdate.day))

empDepDF['age'] = empDepDF['bdate'].apply(calculateAge)
age_bins = range(15, 75, 5)
empDepDF['age_group'] = pd.cut(empDepDF['age'], bins=age_bins, labels=age_bins[1:])
```

```
 age         age_g...

 63          65
 57          60
 53          55
 43          45
 46          50
 58          60
 67          70
 44          45
```

```
salAgeGenderDF = empDepDF[['salary', 'age', 'gender']]
corr = salAgeGenderDF.corr()
sb.heatmap(corr, annot=True)
```

Tehtävä 3

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sb


titanic_data_df = pd.read_csv('./work/viikko2/datasets/Titanic_data.csv')

titanic_names_df = pd.read_csv('./work/viikko2/datasets/Titanic_names.csv')


titanic_joined_df = titanic_data_df.merge(titanic_names_df, how='inner', on='id')


titanic_joined_df.info()

titanic_joined_df.describe()

titanic_joined_df.hist(bins=4)
```



```
titanic_joined_df.shape[0]
```

```
1313
```

```
titanic_joined_df['Gender'].value_counts()
```

```
Gender
male      851
female    462
Name: count, dtype: int64
```

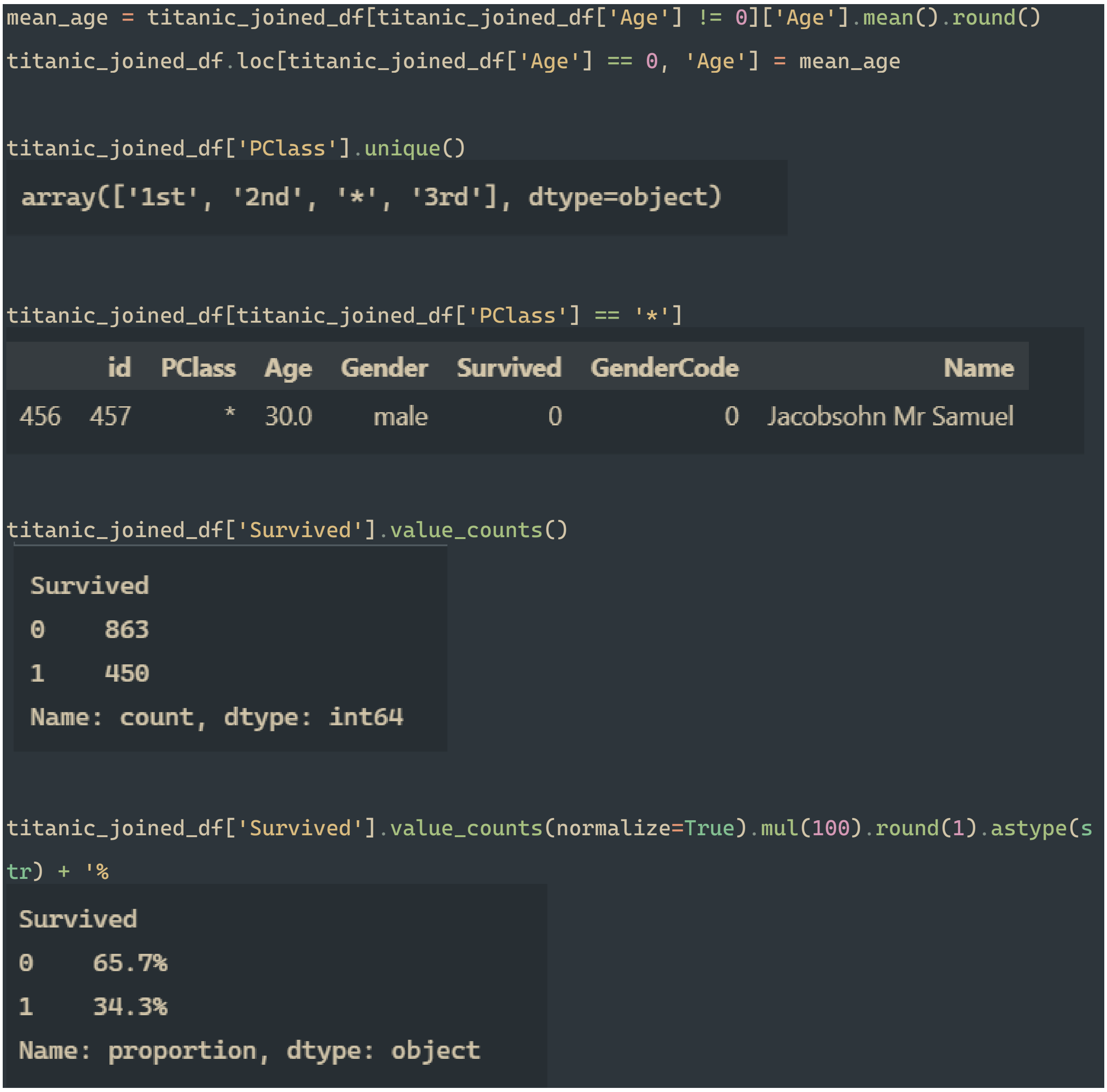
```
titanic_joined_df['Age'].mean().round()
```

```
18.0
```

```
titanic_joined_df[titanic_joined_df['Age'] == 0].shape[0]
```

```
557
```

Tehtävä 4



```python
mean_age = titanic_joined_df[titanic_joined_df['Age'] != 0]['Age'].mean().round()
titanic_joined_df.loc[titanic_joined_df['Age'] == 0, 'Age'] = mean_age

titanic_joined_df['PClass'].unique()
```

```
array(['1st', '2nd', '*', '3rd'], dtype=object)
```

```python
titanic_joined_df[titanic_joined_df['PClass'] == '*']
```

	id	PClass	Age	Gender	Survived	GenderCode	Name
456	457	*	30.0	male	0	0	Jacobsohn Mr Samuel

```python
titanic_joined_df['Survived'].value_counts()
```

```
Survived
0    863
1    450
Name: count, dtype: int64
```

```python
titanic_joined_df['Survived'].value_counts(normalize=True).mul(100).round(1).astype(str) + '%
```

```
Survived
0    65.7%
1    34.3%
Name: proportion, dtype: object
```