

#T1-T2.py

```
#!/usr/bin/python3
from datetime import datetime, timedelta
from numpy import NaN
import pandas as pd
```

```
df_emp = pd.read_csv('./employees.csv', header=0)
df_dep = pd.read_csv('./departments.csv', header=0)
```

```
# Yhdistä kaksi dataframea yhdeksi käyttäen pandas merge ominaisuutta. Käytä
inner-liitosta (how='inner'), ja tee liitos dep:n perusteella (on='dep').
Tämä on dataframe, jota jatkossa käytetään.
df =pd.merge(df_emp, df_dep, how='left', on='dep')
# Poista image sarake dataframesta
df.drop('image', inplace=True, axis=1)
```

```
#laske montako työntekijää on dataframessa
emp_count= df.shape[0]
rows, cols = df.shape
```

```
#laske montako miestä (gender=0) ja naista (gender=1) on työntekijöissä
m_count = sum(df['gender'] == 0)
f_count = sum(df.gender==1)
```

```
#laske myös em. prosenttiosuudet yhdellä desimaalilla
m_percentage = round(m_count / emp_count * 100,1)
f_percentage = round(f_count / emp_count * 100,1)
```

```
#laske työntekijöiden minimi, maksimi ja keskipalkka
min_salary = min(df.salary)
max_salary = max(df.salary)
avg_salary = df.salary.mean()
```

```
#laske 'Tuotekehitys' osaston keskipalkka
rd_avg_salary = df[df.dname == 'Tuotekehitys'].salary.mean()
```

```
#laske monellako työntekijällä ei ole kakkospuhelinta (phone2)
no_phone2 = sum(df.phone2.isna())
```

```
#lisää dataframeen uusi sarake: 'age', ja laske siihen jokaisen työntekijän
ikä vuosina
df['age'] = pd.to_datetime(df.bdate).map(lambda x: (datetime.now() - x) //
timedelta(days=365.2425))
```

#T3-T4.py

```
#!/usr/bin/python3
import pandas as pd
```

```
#Lataa Titanic csv tiedostot pythoniin pandas dataframeen
(Titanic_data.csv, Titanic_names.csv)
df_data = pd.read_csv('./Titanic_data.csv', header=0)
df_names = pd.read_csv('./Titanic_names.csv', header=0)
```

```
#Tulosta ipython consoliin dataframien info ja describe. Tee
histogrammi df_titanic_data ( bins=4) dataframesta.
#https://pandas.pydata.org/docs/reference/api/
pandas.DataFrame.info.html?highlight=info#pandas.DataFrame.info
#https://pandas.pydata.org/docs/reference/api/
pandas.DataFrame.describe.html?
highlight=describe#pandas.DataFrame.describe
#https://pandas.pydata.org/docs/reference/api/
pandas.DataFrame.hist.html?highlight=hist#pandas.DataFrame.hist
```

```
#Yhdistä dataframet (df_titanic_data, df_titanic_names) uudeksi
dataframeksi df. Käytä how='inner' on='id'.
df = pd.merge(df_data, df_names, how='inner', on='id')
```

```
#montako henkilö on aineistossa?
persons_count = df.shape[0]
```

```
#montako miestä ja naista on aineistossa?
m_count = sum(df.GenderCode == 0)
f_count = sum(df.GenderCode==1)
```

```
#matkustajien keski-ikä?
avg_age = df.Age.mean()
```

```
#kuinka monta nollan ikäistä matkustajaa?
zero_age_amount = sum(df.Age == 0)
```

```
#Huomataan, että on paljon nollan ikäisiä matkustajia. Lasketaan
ei-nollan ikäisten matkustajien keski-ikä, ja päivitetään nollan
ikäisten matkustajien iäksi tämä laskettu keskiarvo
df_babies = df[df.Age == 0]
df_nobabies = df[df.Age != 0]
```

```
avg_above_zero = df_nobabies.Age.mean()
```

```
#Tulosta yksikäsitteiset PClasses arvot. Output: ['1st' '2nd' '*'
'3rd']
```

```

print(df.apply(lambda col: col.unique()).PClass)

#Etsi, kenen PClass on: *
print(df.loc[df.PClass == '*'])

#Laske selviytyneiden ja ei-selviytyneiden lukumäärä. Tulosta
arvot myös prosentteina
notSurvived = df.loc[df.Survived == 0]
survived = df.loc[df.Survived == 1]

notSurvivedCount = notSurvived.shape[0]
survivedCount = survived.shape[0]
print("not survived count ", notSurvivedCount)
print("survived count ", survivedCount)

notSurvivedPercentage = round(notSurvivedCount / persons_count *
100,1)
survivedPercentage = round(survivedCount / persons_count * 100,1)

print("percentage not survived", notSurvivedPercentage)
print("percentage survived", survivedPercentage)

#Laske selviytyminen ja ei-selviytyminen miesten ja naisten
kohdalla

not_survived_m_count = sum(notSurvived.GenderCode == 0)
survived_m_count = sum(survived.GenderCode == 0)
print("male survivors", survived_m_count, "male not survived",
not_survived_m_count)

not_survived_f_count = sum(notSurvived.GenderCode==1)
survived_f_count = sum(survived.GenderCode==1)
print("female survivors", survived_f_count, "female not survived",
not_survived_f_count)

```