# 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()

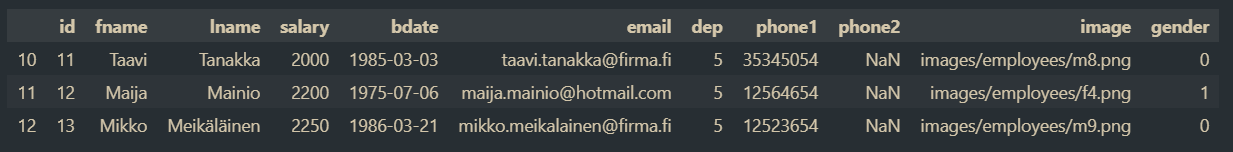
employeesDF.isnull()

employeesDF.nlargest(3, 'salary')

Kuva, joka sisältää kohteen teksti, kuvakaappaus, Fontti, numero

Kuvaus luotu automaattisesti

employeesDF.nsmallest(3, 'salary')



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

# Tehtävä 2

empDepDF.shape[0]

Kuva, joka sisältää kohteen kuvakaappaus, Fontti, symboli, logo

Kuvaus luotu automaattisesti

empDepDF['gender'].value\_counts()

Kuva, joka sisältää kohteen teksti, kuvakaappaus, Fontti, numero

Kuvaus luotu automaattisesti

empDepDF['gender'].value\_counts(normalize=True).mul(100).round(1).astype(str) + '%'

Kuva, joka sisältää kohteen teksti, Fontti, kuvakaappaus, numero

Kuvaus luotu automaattisesti

empDepDF['salary'].min()

Kuva, joka sisältää kohteen kuvakaappaus, Fontti, teksti, numero

Kuvaus luotu automaattisesti

empDepDF['salary'].max()

Kuva, joka sisältää kohteen Fontti, kuvakaappaus, Grafiikka, logo

Kuvaus luotu automaattisesti

empDepDF['salary'].mean()



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

Kuva, joka sisältää kohteen teksti, Fontti, kuvakaappaus, muotoilu

Kuvaus luotu automaattisesti

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



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:])

Kuva, joka sisältää kohteen teksti, kuvakaappaus, numero, Fontti

Kuvaus luotu automaattisesti

salAgeGenderDF = empDepDF[['salary', 'age', 'gender']]

corr = salAgeGenderDF.corr()

sb.heatmap(corr, annot=True)

Kuva, joka sisältää kohteen teksti, kuvakaappaus, neliö, muotoilu

Kuvaus luotu automaattisesti

# 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)

Kuva, joka sisältää kohteen teksti, kuvakaappaus, diagrammi, viiva

Kuvaus luotu automaattisesti

titanic\_joined\_df.shape[0]



titanic\_joined\_df['Gender'].value\_counts()

Kuva, joka sisältää kohteen teksti, kuvakaappaus, Fontti, numero

Kuvaus luotu automaattisesti

titanic\_joined\_df['Age'].mean().round()



titanic\_joined\_df[titanic\_joined\_df['Age'] == 0].shape[0]



# Tehtävä 4

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()



titanic\_joined\_df[titanic\_joined\_df['PClass'] == '\*']



titanic\_joined\_df['Survived'].value\_counts()

Kuva, joka sisältää kohteen teksti, kuvakaappaus, Fontti, numero

Kuvaus luotu automaattisesti

titanic\_joined\_df['Survived'].value\_counts(normalize=True).mul(100).round(1).astype(str) + '%

Kuva, joka sisältää kohteen teksti, kuvakaappaus, Fontti, numero

Kuvaus luotu automaattisesti