# Viikko 37 -tehtävät

## Tehtävä 1

empDepDF = pd.read\_csv('./work/viikko3/datasets/emp-dep.csv')

empDepDF.plot.scatter(x='age', y='salary', s=100)

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

Kuvaus luotu automaattisesti

empDepDF['dname'].value\_counts().sort\_values().plot.bar()

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

Kuvaus luotu automaattisesti

empDepDF['dname'].value\_counts().sort\_values().plot.barh()

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

Kuvaus luotu automaattisesti

## Tehtävä 2

empDepDF['age\_group'].value\_counts().sort\_index().plot.bar()

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

Kuvaus luotu automaattisesti

## Tehtävä 3

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

gender\_mapping = {0: 'miehet', 1: 'naiset'}

gender\_percentages = (gender\_counts / gender\_counts.sum()) \* 100

gender\_percentages.index = gender\_percentages.index.map(gender\_mapping)

plt.pie(gender\_percentages, labels=gender\_percentages.index, autopct='%1.1f%%')

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

Kuvaus luotu automaattisesti

male\_age\_counts = empDepDF[empDepDF['gender'] == 0]['age\_group'].value\_counts()

female\_age\_counts = empDepDF[empDepDF['gender'] == 1]['age\_group'].value\_counts()

gender\_counts\_df = pd.DataFrame({'Miehet': male\_age\_counts, 'Naiset': female\_age\_counts}).fillna(0)

gender\_counts\_df.plot.bar(xlabel='Ikäryhmä', ylabel='Lukumäärä')

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

Kuvaus luotu automaattisesti

## Tehtävä 4

titanicDF = pd.read\_csv('./work/viikko3/datasets/titanic.csv')

age\_bins = range(0, 95, 5)

titanicDF['Age\_group'] = pd.cut(titanicDF['Age'], bins=age\_bins, labels=age\_bins[1:])

titanicDF['Age\_group'].value\_counts().sort\_index().plot.bar()

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

Kuvaus luotu automaattisesti

male\_survived\_counts = titanicDF[titanicDF['Gender'] == 'male']['Survived'].value\_counts()

female\_survived\_counts = titanicDF[titanicDF['Gender'] == 'female']['Survived'].value\_counts()

gender\_survived\_counts\_df = pd.DataFrame({'miehet': male\_survived\_counts, 'naiset': female\_survived\_counts})

gender\_survived\_percentages = (gender\_survived\_counts\_df.iloc[1] / gender\_survived\_counts\_df.iloc[1].sum()) \* 100

gender\_survived\_percentages.plot.pie(ylabel='Selviytyneet',

                                    autopct='%1.1f%%',

                                    title=f'matkustajia: {titanicDF.shape[0]}\nselviytyneet miehet: {gender\_survived\_counts\_df.iloc[1, 0]}\nselviytyneet naiset: {gender\_survived\_counts\_df.iloc[1, 1]}'

                                    )

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

Kuvaus luotu automaattisesti

titanicDF = titanicDF[titanicDF['PClass'] != '\*']

saved\_mapping = {0: 'no', 1: 'yes'}

titanicDF['Saved'] = titanicDF['Survived'].map(saved\_mapping)

sb.boxplot(data=titanicDF['Saved'], x=titanicDF['PClass'], y=titanicDF['Age'], hue=titanicDF['Saved'])

Kuva, joka sisältää kohteen diagrammi, Suunnitelma, Suorakaide, Tekninen piirros

Kuvaus luotu automaattisesti

sb.stripplot(data=titanicDF[titanicDF['Gender'] == 'female']['Saved'],  x=titanicDF['PClass'], y=titanicDF['Age'], hue=titanicDF['Saved'])

plt.title("females")

Kuva, joka sisältää kohteen kuvakaappaus, teksti, diagrammi

Kuvaus luotu automaattisesti

sb.stripplot(data=titanicDF[titanicDF['Gender'] == 'male']['Saved'],  x=titanicDF['PClass'], y=titanicDF['Age'], hue=titanicDF['Saved'])

plt.title("males")

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

Kuvaus luotu automaattisesti