```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Dec 28 01:14:58 2019

@author: LeliaSofinezHaouaya
"""
#Lelia Sofinez Haouaya
#Lxk384
#6 January 2020
#--------------------
print("Hello World")

#Spørgsmål 1)
#Det første jeg kan se, når jeg åbner filen er en lang liste af tal og navne. Men efter nærmere inspektion, kan jeg se at data typerne vi har med at gøre blandt andet er tekst ”strings”, som navne på personer og koloner.
#Vi har også en kæmpe mængde af tal typer der siger noget om alder, priser, mængder etc. Tallene fremstår både som heltal ”interger” og komatal ”float” fra priser.
#Steder hvor der står enten ”1 eller 0” som under ”Survived”, ”1” svarer til et ja som betyder at de er overlevet. Hvorimod ”0” svarer til et nej og betyder at de er døde.
#Jeg kan se at hvert eneste felt er fyldt op, så det ligner ikke at der skulle mangle noget data som sådan, men det vil kommandoerne vise.

#Spørgsmål 2)
import pandas as pd
titanic_data = pd.read_csv("titanic.csv")
#Jeg bruger først følgende kommando, for at importere min fil som jeg har kaldt for "data".

titanic_data.head()
#Her tjekker jeg de første 5 kolonner på mit datasæt.

pd.set_option("display.max_columns",8)
#Siden jeg gerne vil se lidt tydligere hvad jeg har stående som navne på kollonerne, så bruger jeg denne kommando.
#Den viser maximum antallet af navnene på alle kollonerne. Jeg har valgt tallet "8" fordi jeg har 8 kolonne navne.

titanic_data.head()
#Så når jeg så bruger følgende samme kommando igen, så kan jeg se alle kolonne navnene, dog stadig kun de første 5 rækker.

titanic_data.dtypes
#Ud fra denne kommando kan vi se de data-typer vi har med at gøre, som både er "objects, integers og floats" udfra hver beskrevet kategori.
#Det giver mening at de koloner i datasættet der indeholder tal kaldes
```

```
"int/float" og at dem der indeholder tekst kaldes "object".

titanic_data.select_dtypes(include=['float64', 'int64'])
#Med denne kommando kan man specificere hvilket data typer man vil se
noget om fra sit datasæt.

print("her vises antal celler:", titanic_data.size)
#Her vises antal af celler som er "7096".
#Grunden til at jeg har brugt sætningen i "..." er for at gøre det
tydeligere for mig hvor mit svar ligger i konsollen.

print("her vises navnet på kolonnerne:", titanic_data.columns)
#Hvis man ikke allerede regnede det ud fra "data.head" kommandoen
tidligere, så er det her en anden kommando til at vise hvilket navnene
vi har på kollonerne i datasættet.

print("her vises størrelsen af mit df:", titanic_data.shape)
#Her ser vi på størrelsen af data framen, både antallet af rækker som
er "887" og antallet af kolonne navne som er "8".
#Igen det er noget jeg kan se på min variabel explorer, under "size"
men kunne man ikke se det, så er kommandoen god til at vise det.

#Spørgsmål 3)
titanic_data.describe()
#Kommandoen her bruges til at beskrive de forskellige deskriptive
datatyper, og finde ting som maximum, minimum, midertallet osv.
#Den bruges dog kun til at beskrive tal som "floats, integers" som vi
har i datasættet.
#De to kommandoer nedenunder kan bruges til at specificere præcist
hvad og hvilket data man vil fokusere sin data beskrive på.

titanic_data[["Age"]].mean()
#Midter tallet af den samlede alder af folket var næsten ca. "30"
årige.

titanic_data[["Fare"]].max()
#Maximum prisen som folket betalte var "512$"

titanic_data.groupby(['Sex']).count().Name
titanic_data.groupby(['Pclass']).count().Name
#Her får jeg først basis informationer om antallet af hvert køn, samt
antallet af overlevende og i hvilken klasse de var i.

titanic_data.groupby(['Survived','Sex']).count().Name
#Med denne kommando har jeg fundet uf af hvor mange der døde og
overlede, og jeg har specificeret fordelingen ud fra køn.
titanic_data.groupby(['Survived','Pclass']).count().Name
#Jeg tænkte det ville være interresant at se på antallet af
overlevende, samt hvilken klasse de var i.
#Ud fra det kan man konkludere at de fleste der overlevede, var folk
```

```
fra første klasse, dernæst folk fra anden klasse.
#Dog kan man også se at de fleste der døde var fra 3-klasse, og det
kan måske sige noget om hvor gode sikkerheds forholdene er i forhold
til hvilken klasse man var i dengang.

titanic_data.groupby(['Survived','Sex']).count().Name.plot(kind='bar')
titanic_data.groupby(['Survived','Pclass']).count().Name.plot(kind='ba
r')
#Jeg har valgt at visualisere følgende 2 forrige kommandoer i et
søjlediagram.

#Spørgsmål 4)
new_df = titanic_data['Name'].str.rsplit(n=1, expand=True)
print (new_df)
new_df[1].value_counts()
#Først laver jeg en ny data frame som jeg kalder "new_df" også vælger
jeg at den skal tage data fra mit "Name" kollonefelt.
#Derefter printer jeg det så jeg kan se hvad jeg har med at gøre, men
min variabel explorer visualiserer også fint mit nye data frame.
#Derefter laver jeg en "value_count" kommando som før og tæller
antallet af folk med samme efternavn.
#Vi kan se at efternavnet "Anderson" har hele 9 personer med det samme
efternavn.
#Der næst har vi efternavnet "Skage" med hele 7 personer med det samme
efternavn.
#Så altså der er en heel del flere folk ombord med det samme efter
navn.

#Spørgsmål 5)
pd.pivot_table(titanic_data, values= 'Survived', index= 'Pclass',
aggfunc='count')
#Her får jeg at vide hvor mange passagere der er inddelt i hvert
klasse.

pd.pivot_table(titanic_data, values='Survived', index='Pclass',
aggfunc='sum')
#Her har jeg valgt at bruge følgende kommando for at få svaret på hvor
mange der overlevede i hvert klasse. 3 klasse havde flest omkomne og
det kan som jeg tidligere nævnte, give os informationer som i jo
lavere klasse man befandt sig i, jo mindre var chancen nok for at
overleve.
```