```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Dec 21 17:35:57 2019

@author: LeliaSofinezHaouaya
"""
#Spørgamål 2 og 3-A) [i,ii,iii]
#Først downloader jeg vigtige "libraries" jeg højt sansyligt skal
bruge:
import numpy as np
import requests
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.ticker as mtick
from pandas import DataFrame
from pandas.io.json import json_normalize

#Derefter laver jeg en tom "variable" kaldet API_URL.
api_url = "https://api.smk.dk/api/v1/art/search"
#Jeg bruger de passende URL kommandoer fra SMK API (Swagger UI).

#Så laver jeg en "dictionary" og giver den nogle parametre fra (SMK),
som den skal tage forbehold for.
params = {
    'keys': 'maleri',
    'rows': 200,
    "n": 0,
    'encoding': "json"}
#Så har jeg valgt at vi skal se på "maleri" og har sat den til 200
rækker.
#Her sørger jeg for at dataen kommer ud som json.
#Desuden har jeg sat "n" til 0, fordi vi ikke har brug for "records"

#Also i add "n" to be "0" because we dont need records.

result = requests.get(api_url, params=params)
#Det er så her vi får sat alle de forudstående "params" sammen med den
#første (api_url) variabel, i en kommando hvor vi beder den creere
vores URL
#samt hente søgninger på de former for informationer vi har defineret
i "params" osv.

print("My SMK API URL is:\n{}\n".format(result.url))
#Her printer/udskriver jeg mit URL, samt jeg har tilføjet nyt linje
skift så det ser bedre ud.

print(result)
#Når jeg kører denne kommando kan jeg se at alt er gjort godt fordi vi
får (Response [200]) i outputet.
```

```
#Mit Url er følgende:
#https://api.smk.dk/api/v1/art/search?
keys=maleri&rows=200&n=0&encoding=json

json = result.json()
#Her laves en "dictionary" med variablerne fra resultats-url.

df = json_normalize(json['items'])
print(df)
#Her har jeg lavet variablen "df" der indeholder min tabel der har
størrelsen (200 rækker og 52 kolloner.)
#Det kan jeg blandt andet se både på min variabel explorer eller når
jeg printer (df).

df.dtypes
#Med følgene kommando kan jeg se alle de slags data typer jeg har.

df.drop(["iiif_manifest","colors","frame_notes","parts","notes","objec
t_history_note","part_of","production_dates_notes","alternative_images
","number_of_parts","collection","content_person","content_description
","current_location_date","current_location_name","distinguishing_feat
ures","image_cropped","image_iiif_id","image_iiif_info","image_mime_ty
pe","image_native","image_thumbnail","object_names","object_url","rela
ted_objects","work_status","documentation"], inplace=True, axis=1)
#Mange af data typerne er unødvendige, så jeg kører en "df.drop"
kommando, for at fjerne dem jeg ikke skal bruge.
#Det gør så at jeg nu har renset datasætet og har (200 rækker og 26
kolloner.)


#Spørgsmål 4)
df.dtypes
#Her kører jeg kommandoen igen for at se de datatyper vi nu står med.

df.rename(columns={'production_date':'date_created',"labels":"text","r
esponsible_department":"department"}, inplace=True)
#Her har jeg valgt at ændre nogle af navnene kolonerne til noget lidt
mere simpelt/forståeligt.

print(df.head())
print(df.head)
#Jeg tager lige et kig på data. Den ene giver et kort overblik med
titler, den anden giver årstal, forfatter og lidt mere til.

print(df.mean())
#Her får vi informationer på medianen af forskellige typer deskriptiv
"tal" data.

print(df.describe())
```

```
#Følgende kommando er bedre, fordi den på engang beskriver interesante
informationer om datasættet til sammen.

df.isnull().sum()
#Her kan jeg se summen/antalet af alle mine elementer.
#Ud fra det har jeg valgt at det ville være interesant at se på
"department".
#Så nedenstående kommando er dataframen for "department" dataet.


#Spørgsmål 5)
maleri_lokation =
df["department"].value_counts(normalize=True).mul(100).round(2)
maleri_df_lokation = pd.DataFrame(maleri_lokation)
print(maleri_lokation)
#Her har jeg som sagt lavet en dataframe over "department" som nu
hedder (maleri_df_lokation.)
#Når jeg printer dataframen kan jeg se resultaterne.
#Jeg har valgt at den skulle rundes til 2 decimaler, så jeg kan se
lidt mere informationer.


#Spørgsmål 6)
x = maleri_df_lokation.department
y = maleri_df_lokation.department
plt.scatter(x, y)
plt.show()
#Her har jeg valgt at lave en graf ved hjælp af "matplotlib library".
#Mit dataframe er måske ikke den største og mest nuancerede form for
data, men det virker stadig, og vi får en korrekt graf.

ax = maleri_df_lokation['department'].plot(kind='bar')
ax.yaxis.set_major_formatter(mtick.PercentFormatter())
#Efter at have stirrede på grafen i lang tid besluttede jeg mig for
også at bruge en anden slags/model af graffer.
#Følgende er en kommando til et søjlediagram, der visualisere
dataframen meget godt.
```