Bericht

June 9, 2023

[1]:
```
!export PATH=/Library/TeX/texbin:$PATH
```

[2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[3]:
```
# bereits geputzten und gemergten Datensatz einlesen (siehe python Code)
df = pd.read_csv('final_df.csv')
```

[4]:
```
# a) Metadaten:
# Datebezeichnung: Energieverbauchswerte von Gebäuden zusammen mit stündlichen␣
 ↪Wettermessungen und Informationen zu den Gebäuden
# Quelle: Vorlesung/Kaggle(https://www.kaggle.com/competitions/
 ↪ashrae-energy-prediction/overview)
#Lizenz: Kaggle exclusive license for academic research and education
#Anzahl Attribute: 6, Anzahl Instanzen: 5629287

#meter                          int8
#timestamp              datetime64[ns]
#meter_reading                 float64
#primary_use                    object
#square_feet                     int32
#air_temperature               float32
```

[5]:
```
########################## b) ########################
#Forschungsfragen und gewählte Methoden zur Beantwortung:
#Frage 1: Zusammenhang zwischen meter_reading und air_temperature? Methode:␣
 ↪Korrelation zwischen den beiden Attributen berechnet und visuell dargestellt␣
 ↪durch binnieren nach air temperature.
#Frage 2: Zusammenhang zwischen square footage und meter_reading?  Methode:␣
 ↪Korrelation zwischen den beiden Attributen berechnet und visuell dargestellt␣
 ↪durch binnieren nach air temperature.
#Frage 3: Durchschnittlicher Energieverbrauch pro Gebäudenutzungsart? Methode:␣
 ↪groupby verwendet und Durchschnitt für jede Art von Gebäude berechnet,␣
 ↪danach Werte sortiert und als Barplot dargestellt
#Frage 4: Saisoneller Einfluss auf Energieverbrauch und dazu folgende Hypothese␣
 ↪getestet: Im Winter ist Energieverbrauch höher als im Sommer.
```

```
#Methode: Zeitreihenanalyse mit timestamp als primary key und dazu rollende␣
 ↪Mittelwerte sowie totale tägliche Verbräuche berechnet.
```

[6]:
```
##### c) #####
```

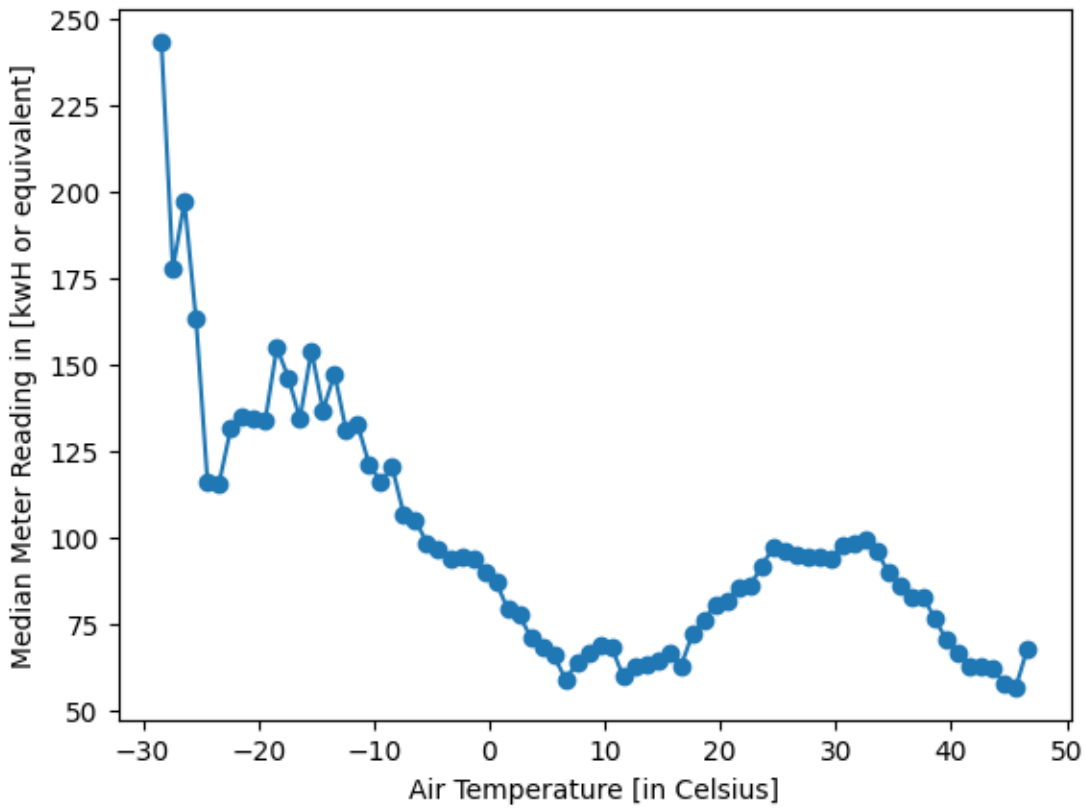
[7]:
```
## Erkenntnisse Frage 1:
correlation = df['meter_reading'].corr(df['air_temperature'])
print('Correlation:', correlation)
```

```
Correlation: -0.003980445350054368
```

[8]:
```
df['temp_bin'] = pd.cut(df['air_temperature'], bins=np.
 ↪arange(df['air_temperature'].min(), df['air_temperature'].max(), 1))

# Berechnen des Median 'meter_reading' für jedes Bin
median_meter_reading = df.groupby('temp_bin')['meter_reading'].median()

# Plot
plt.plot(median_meter_reading.index.categories.mid, median_meter_reading,␣
 ↪marker='o')
plt.xlabel('Air Temperature [in Celsius]')
plt.ylabel('Median Meter Reading in [kwH or equivalent]')
plt.show()
```



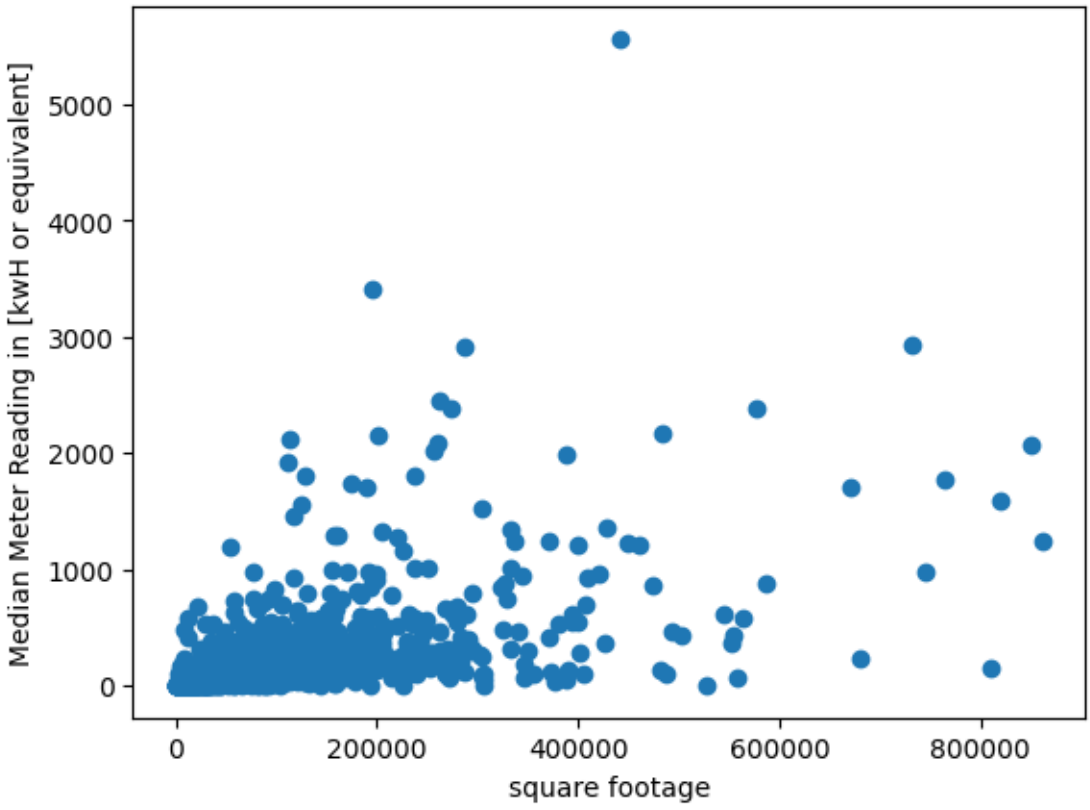
[9]:
```
# Es besteht keine signifikante Korrelation zwischen Energieverbrauch und␣
 ↪Lufttemperatur, trotzdem sind einige Trends erkennbar.
# Wie zu erwarten ist der Enegieverbrauch bei kälteren Temperaturen signifikant␣
 ↪größer als in moderat warmen,
# jedoch ist auch erkennbar, dass ab 20 Grad bis 40 Grad auch ein relativ hoher␣
 ↪Energieverbrauch zu beobachten ist
```

[10]:
```
### Frage 2:
correlation = df['meter_reading'].corr(df['square_feet'])
print('Correlation:', correlation)
# visuell:
# Binnieren der Daten nach 'square_footage'
df['square_bin'] = pd.cut(df['square_feet'], bins=np.arange(df['square_feet'].
 ↪min(), df['square_feet'].max(), 1))

# Berechnen des Median 'meter_reading' für jedes Bin
median_meter_reading = df.groupby('square_bin')['meter_reading'].median()

# Plot
plt.plot(median_meter_reading.index.categories.mid, median_meter_reading,␣
 ↪marker='o')
plt.xlabel('square footage')
plt.ylabel('Median Meter Reading in [kwH or equivalent]')
plt.show()
```

```
Correlation: 0.02436351032270559
```



[11]:
```
# Korrelation hier etwas höher als bei der letzten Frage, aber immer noch
 ↪ziemlich gering
# Also gibt es keinen wirklichen klaren Zusammenhang zwischen der Größe eines
 ↪Gebäudes und dem Energieverbrauch
## Aus Frage 1 und 2 mitgenommen: Lufttemperatur und Square Footage an und für
 ↪sich haben keinen direkten Einfluss auf den Energieverbrauch
```

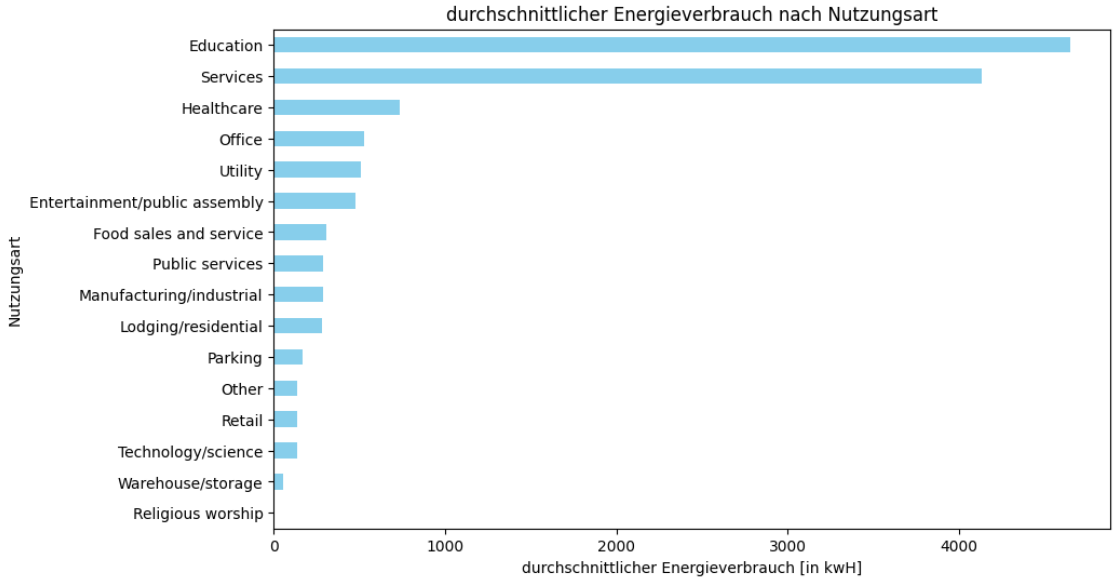
[12]:
```
##  Frage 3:
average_energy_use = df.groupby('primary_use')['meter_reading'].mean()
print(average_energy_use)
```

```
primary_use
Education                       4653.455082
Entertainment/public assembly    474.733586
Food sales and service           304.948675
Healthcare                       737.526621
Lodging/residential              279.567960
Manufacturing/industrial         285.697048
Office                           526.668556
Other                            139.129002
Parking                          169.070372
```

```
Public services                290.142992
Religious worship                5.377399
Retail                         138.846315
Services                      4131.665972
Technology/science             138.660728
Utility                        510.534650
Warehouse/storage               54.225427
Name: meter_reading, dtype: float64
```

[13]:
```
# sortieren der Daten
average_energy_use_sorted = average_energy_use.sort_values()

# visuell als Barplot darstellen
plt.figure(figsize=(10, 6))
average_energy_use_sorted.plot(kind='barh', color='skyblue')
plt.xlabel('durchschnittlicher Energieverbrauch [in kwH]')
plt.ylabel('Nutzungsart')
plt.title('durchschnittlicher Energieverbrauch nach Nutzungsart')
plt.show()
```

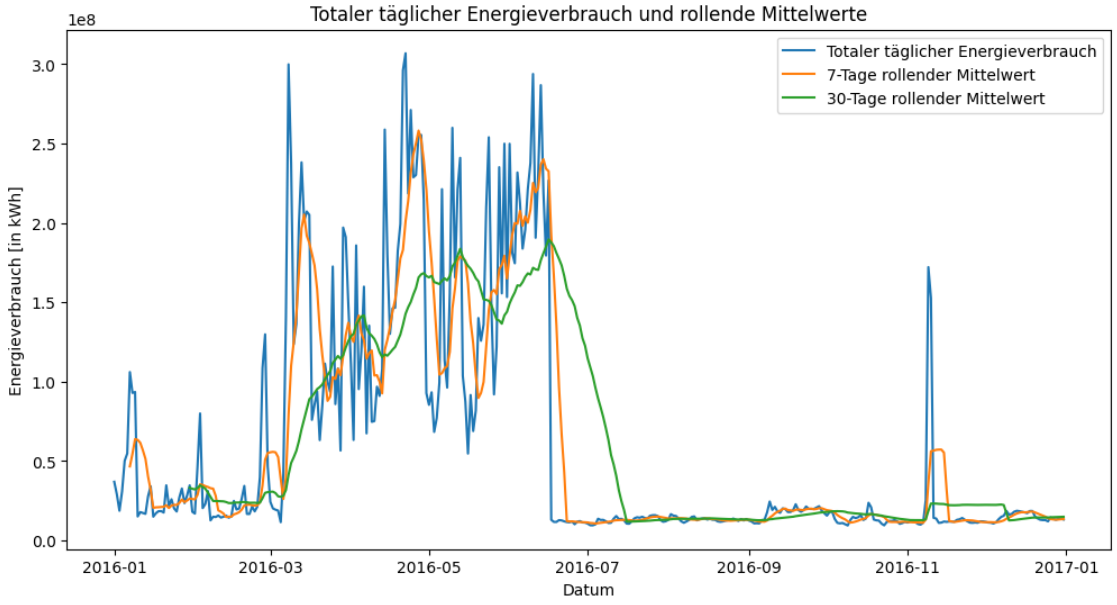


[14]:
```
# In diesem Sample haben Gebäude für Bildung zusammen mit Dienstleistungen den␣
↪mit am Abstand größten Energieverbrauch.
# Also könnte man nun untersuchen, woran das genau liegen könnte, um den␣
↪Energievebrauch möglicherweise sogar senken zu können.
```

[15]:
```
###Frage 4:
##Zeitreihenanalyse um saisonale Muster im Energieverbrauch zu erkennen
```

```
# timestamp Attribut zu DateTimeIndex konvertieren, damit ich timestamp als␣
 ↪index setzen kann
df['timestamp'] = pd.to_datetime(df['timestamp'])
# timestamp als 'primary key' setzen, um resample ausführen zu können
df.set_index('timestamp', inplace=True)
#täglichen totalen Energieverbrauch
daily_energy = df['meter_reading'].resample('D').sum()
#rollenden 7-Tage und 30-Tage Mittelwert berechnen
daily_energy_7d = daily_energy.rolling(window=7).mean()
daily_energy_30d = daily_energy.rolling(window=30).mean()
```

[16]:
```
##Visualisierung der Daten
plt.figure(figsize=(12, 6))
plt.plot(daily_energy, label='Totaler täglicher Energieverbrauch')
plt.plot(daily_energy_7d, label='7-Tage rollender Mittelwert')
plt.plot(daily_energy_30d, label='30-Tage rollender Mittelwert')
plt.title('Totaler täglicher Energieverbrauch und rollende Mittelwerte')
plt.xlabel('Datum')
plt.ylabel('Energieverbrauch [in kWh]')
plt.legend()
plt.show()
```

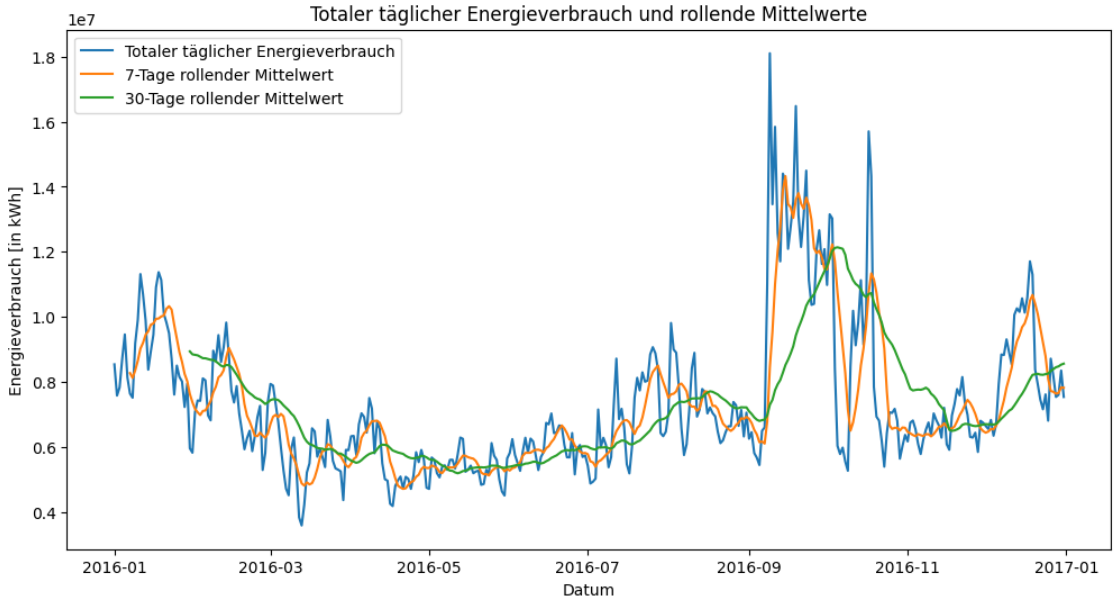


[17]:
```
# man sieht ziemlich klar, dass die Energieverbauchswerte zwischen März und␣
 ↪Juli am höchsten sind, mit kleinen Peaks im Januar und November. Die␣
 ↪Interpretation dieser Daten erweist sich uns als ziemlich
```

```
# kompliziert, vor allem da die Untersuchung zu Frage 2 und der damit␣
 ↪einhergehende Plot zumindestens gefühlt einen gegensätzlichen Trend aufweist␣
 ↪(Im Winter sollte deutlich höherer Energieverbrauch
# erkennbar sein als im Sommer) . Dies lässt sich uns entweder durch ein␣
 ↪ungewöhnliches Klima am Standort der Gebäude oder (deutlich␣
 ↪wahrscheinlicher) durch eine fehlerhafte Repräsentation der Daten
# unsererseits erklären)
```

[21]:
```
# wir haben die Vermutung dass im Sommerzeit Schulen und weitere Gebäude für
# Education schliessen und deshalb der Verbrauch so sinkt
# wenn man vor dem plotten diese Operation durchführt und dfohneEd plottet wird␣
 ↪unsere Vermutung bestärkt
df_ohne_Ed = df[df['primary_use'] != 'Education']
daily_energy = df_ohne_Ed['meter_reading'].resample('D').sum()
#rollenden 7-Tage und 30-Tage Mittelwert berechnen
daily_energy_7d = daily_energy.rolling(window=7).mean()
daily_energy_30d = daily_energy.rolling(window=30).mean()


##Visualisierung der Daten ohne Education
plt.figure(figsize=(12, 6))
plt.plot(daily_energy, label='Totaler täglicher Energieverbrauch')
plt.plot(daily_energy_7d, label='7-Tage rollender Mittelwert')
plt.plot(daily_energy_30d, label='30-Tage rollender Mittelwert')
plt.title('Totaler täglicher Energieverbrauch und rollende Mittelwerte')
plt.xlabel('Datum')
plt.ylabel('Energieverbrauch [in kWh]')
plt.legend()
plt.show()
```



[19]:
```
#### d) ####
#Im Kontext der untersuchten Fragestellungen wäre ein Indiz dafür,  wofür genau␣
 ↪die verwendete Energie genutzt wird äußerst hilfreich , um etwas klarere und␣
 ↪vor allem nützlichere Interpretationen
#formulieren zu können. Nicht verkehrt wäre noch Daten zu den Standorten der␣
 ↪jeweiligen Gebäude und vielleicht auch ungefähre regelmäßige Menschenanzahl␣
 ↪innerhalb. Mithilfe dieser und den bereits vorhandenen  Daten sowie␣
 ↪irgendwelchen Machine-Learning Modellen (die wir hoffentlich bald in der␣
 ↪Vorlesung kennen lernen werden) ließen sich bestimmt einige nützliche␣
 ↪Predictions erstellen lassen.
```

[20]:
```
############### Aufgabe 6 ####################
# Die Größe des Datensatzes sowie die automatische Konvertierung der Datenytpen␣
 ↪von pandas haben die größten Probleme bereitet. Dicht daran kommt die␣
 ↪Unwissenheit,
# wie man genau mit Ausreißern umgegangen werden soll sowie auch die␣
 ↪Unfähigkeit klare Schlüsse aus den untersuchten Fragestellungen zu ziehen.
# Das Mergen der Datensätze hat sich auch als deutlich schwieriger␣
 ↪herausgestellt als gedacht, vor allem im Kontext von Jupyter Notebooks,
# da diese beim Versuch des mergen und auch Einlesen der Datensätze bestimmt␣
 ↪über 100 mal abgestürzt ist, weswegen das Putzen und mergen der Datensätze␣
 ↪auf eien seperate Pyhton Datei verlegt wurde.
# Trotzdem war die Anfertigung dieses 'Berichtes' eine sehr willkommene␣
 ↪Herausforderung und wir haben mit Sicherheit den Umgang mit Pandas sowie␣
 ↪auch mit sehr großen Datensätzen im Allgemeinen trainieren können.
```