Data analysis tasks with python

Electricity consumption analysis

- Implementation: in python 3.9, environment according to https://github.com/DiLTAnalytics/default_model_environment/
- **steps**
  - write a python script for a standardized load profile analysis of electricity meter measurement data (kilowatt, 15-minute intervals)
  - Different grip operators provide load profiles of big buildings in different formats (3 examples attached) - prepare a standardized, modular analysis using python for the raw data including proper documentation (e.g. jupyter notebook, readme) for all the steps.
  - The results should be available as well as CSV and the plots as JPGs.

  - **First step: think about meaningful plausibility checks, e.g. create scatter plot for outlier detection, etc.**
    - **Make a suggestion of 2 plausibility checks and present it to me**

14.08.2023: plausibility checks NOT DONE, see updates on pages 10 & 11

Screenshots of raw data (3 csv files)

MEC	AT0080000870000000000000000097058
UNIT	kWh
TGRID	15
Zeitbereich	01.01.2019 00:01 - 01.01.2020 00:00
Summe im Zeitraum	50.743,050 kWh
Max im Zeitraum	21,920 kW
01.01.2019 00:15	1,3
01.01.2019 00:30	0,77
01.01.2019 00:45	0,8
01.01.2019 01:00	0,78
01.01.2019 01:15	0,78
01.01.2019 01:30	0,76
01.01.2019 01:45	0,82
01.01.2019 02:00	0,8
01.01.2019 02:15	0,73
01.01.2019 02:30	0,83
01.01.2019 02:45	0,77
01.01.2019 03:00	0,79
01.01.2019 03:15	0,77
01.01.2019 03:30	0,75
01.01.2019 03:45	0,76
01.01.2019 04:00	0,89

Anlage			
Vertragsbeze			
Anlagenadre			
GP-Nummer			
Geschäftspar			
GP-Adresse			
VP-Nummer			
Vertriebsspa			
Zählpunkt			
Netzebene			6
Netzgebiet			N08170
Service			NE
Branche			
Profil			10000308
Profilbezeich			OP_L_AT0081700860500000000000000
Profilrolle			1001
Profiltyp/ME			01/kW
Maximum			89,52
Summe kW			453.685,20
Summe kWh			113.421,30
Profilwert	01.01.2020	00:00	6,72
Profilwert	01.01.2020	00:15	6,8
Profilwert	01.01.2020	00:30	6,64
Profilwert	01.01.2020	00:45	6,72
Profilwert	01.01.2020	01:00	6,72
Profilwert	01.01.2020	01:15	6,96
Profilwert	01.01.2020	01:30	7,6
Profilwert	01.01.2020	01:45	7,92
Profilwert	01.01.2020	02:00	7,28
Profilwert	01.01.2020	02:15	6,56
Profilwert	01.01.2020	02:30	6,96
Profilwert	01.01.2020	02:45	6,8
Profilwert	01.01.2020	03:00	7,6

	A	B	C	D
1	Date	Time	1-1:1.5.0	
2	01.01.2019	00:15	21,76	
3	01.01.2019	00:30	20,64	
4	01.01.2019	00:45	21,84	
5	01.01.2019	01:00	20,96	
6	01.01.2019	01:15	21,12	
7	01.01.2019	01:30	20,32	
8	01.01.2019	01:45	21,84	
9	01.01.2019	02:00	22,24	
10	01.01.2019	02:15	21,6	
11	01.01.2019	02:30	19,04	
12	01.01.2019	02:45	20,32	
13	01.01.2019	03:00	22,16	
14	01.01.2019	03:15	21,68	
15	01.01.2019	03:30	21,84	

Create a report per dataset

14.08.2023: 90% DONE, see updates on page 10

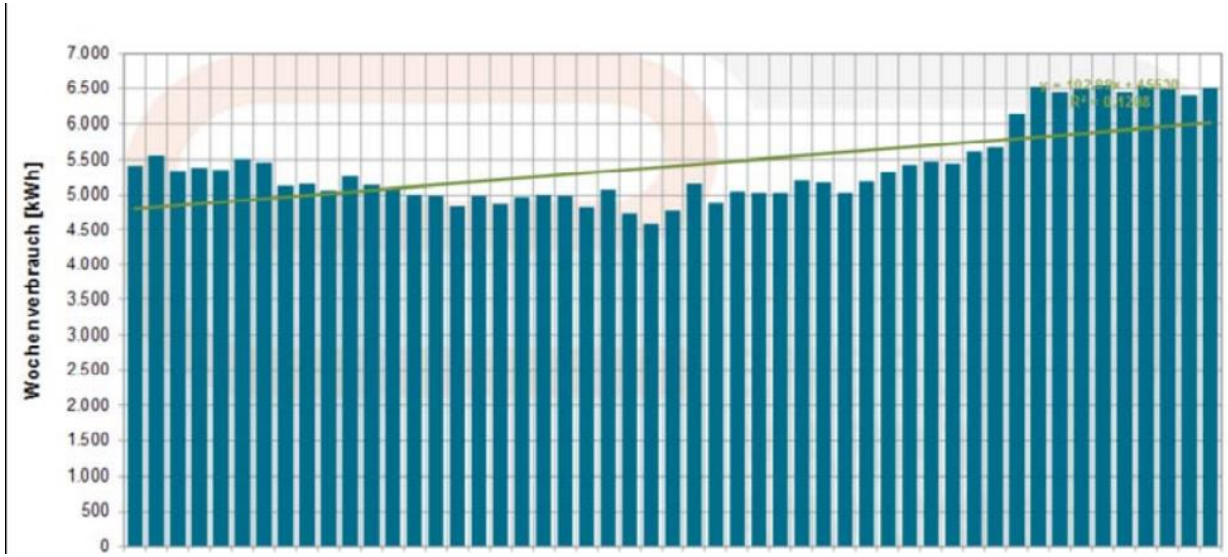
Description of analysis	Result of analysis (write all results in a CSV)
Name of dataset	Name of analysed CSV
Time period of data set	dd.mm.yyyy-dd.mm.yyyy
Count amount of kWh values	1 integer (should be ~35000)
Total energy consumption (=sum of all values)	1 Decimal (kWh)
Maximum value in dataset	1 Decimal value (kWh) and corresponding timestamp
Minimum value in dataset	1 Decimal value (kWh) and corresponding timestamp
Average and median of daily sum in dataset	2 Decimal values (kWh) and standard deviation
Average and median of daily sum on weekdays	2 Decimal values (kWh) and standard deviation
Average and median of daily sum on saturdays and sundays	2 Decimal values (kWh) and standard deviation
Average and median of on sundays at 03:00 in the morning	2 Decimal values (kWh) and standard deviation
Average daily sum between 08:00-17:00 on weekdays	1 Decimal value (kWh) and standard deviation
Average daily sum between 08:00-17:00 on Saturdays and sundays	1 Decimal value (kWh) and standard deviation
Average daily sum between 17:00-08:00 on weekdays	1 Decimal value (kWh) and standard deviation
Average daily sum between 17:00-08:00 on saturdays and sundays	1 Decimal value (kWh) and standard deviation

Create 3 bar charts

14.08.2023: 90% DONE; comments about plots see pages 13 & 14

- **Bar chart 1:** daily sums in kWh (y-axis), date on x-axis
- **Bar chart 2:** weekly sums in kWh (y-axis), number of calendar week on x-axis
- **Bar chart 3:** Monthly sums in kWh (y-axis), month (mm.yy) on x-axis

Example how bar chart 2 could look like

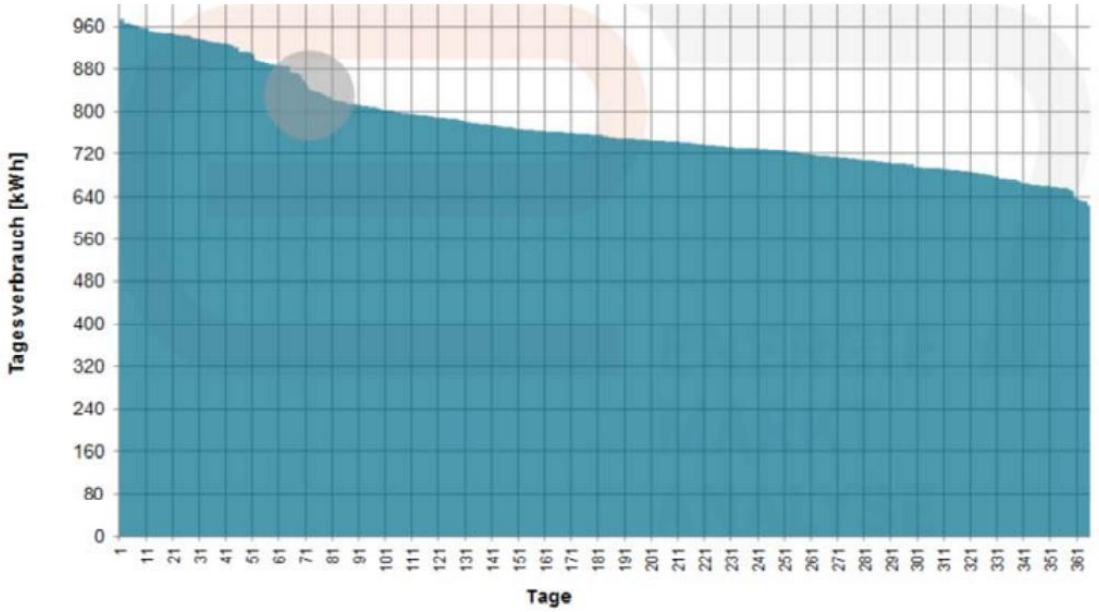


Create 2 bar charts

14.08.2023: NOT DONE yet

- **Sort values of bar chart 1 (slide before) by size (descending):** daily sums in kWh (y-axis), number of days on x-axis
- **Sort values of bar chart 2 (slide before) by size (descending):** weekly sums in kWh (y-axis), first day of related week on x-axis

Example how bar chart 1 could look like

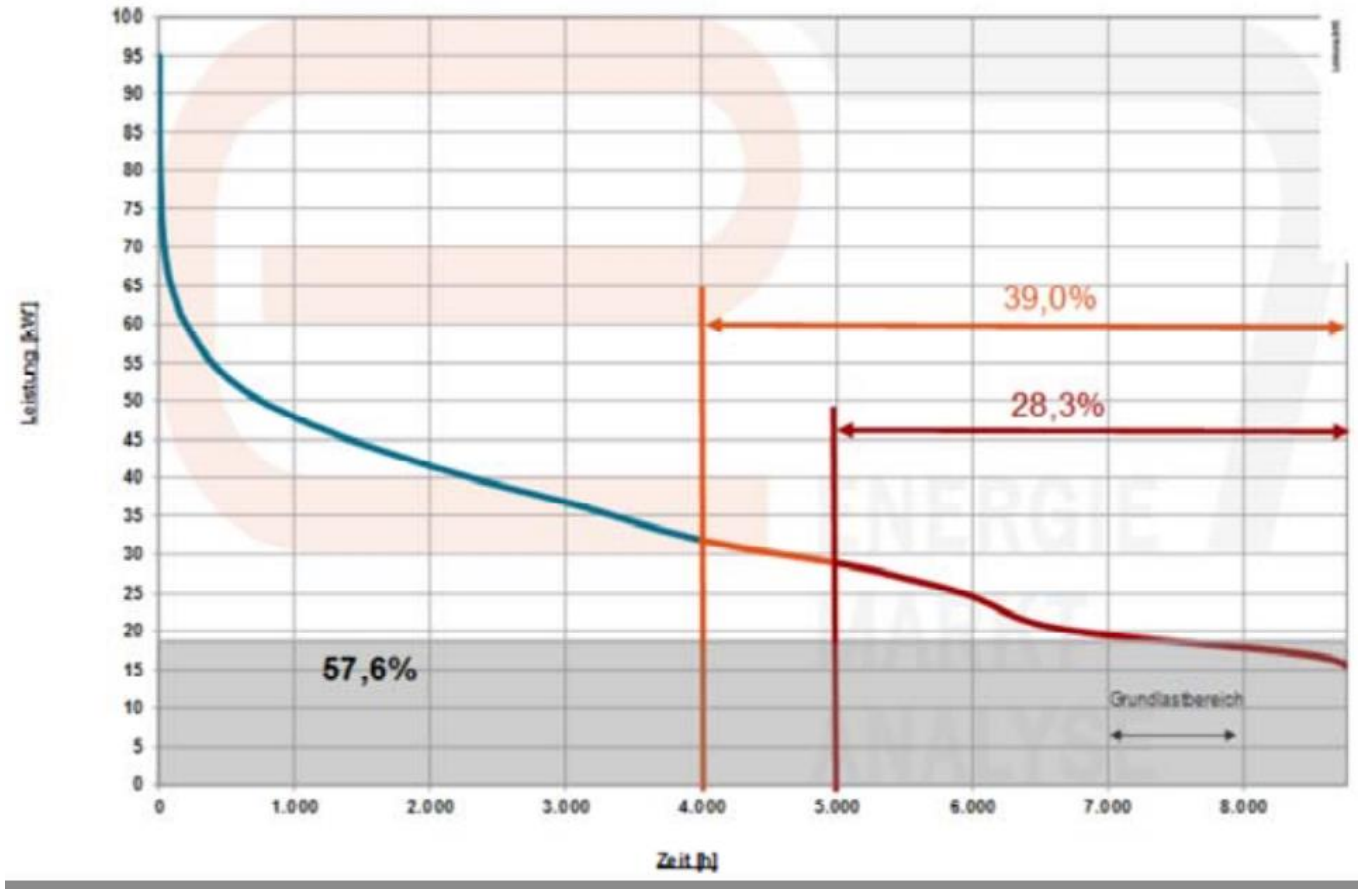


14.08.2023: NOT DONE yet

Create a line diagram

- Multiply each 15-min-value with 4 (converts kWh in kW)
- Create a plot which arranges the kW values by size (y-axis, descending) and time in hours on x-axis

Example plot how it could look like (blue, orange & red line, don‘t take care about the labels in the plot)

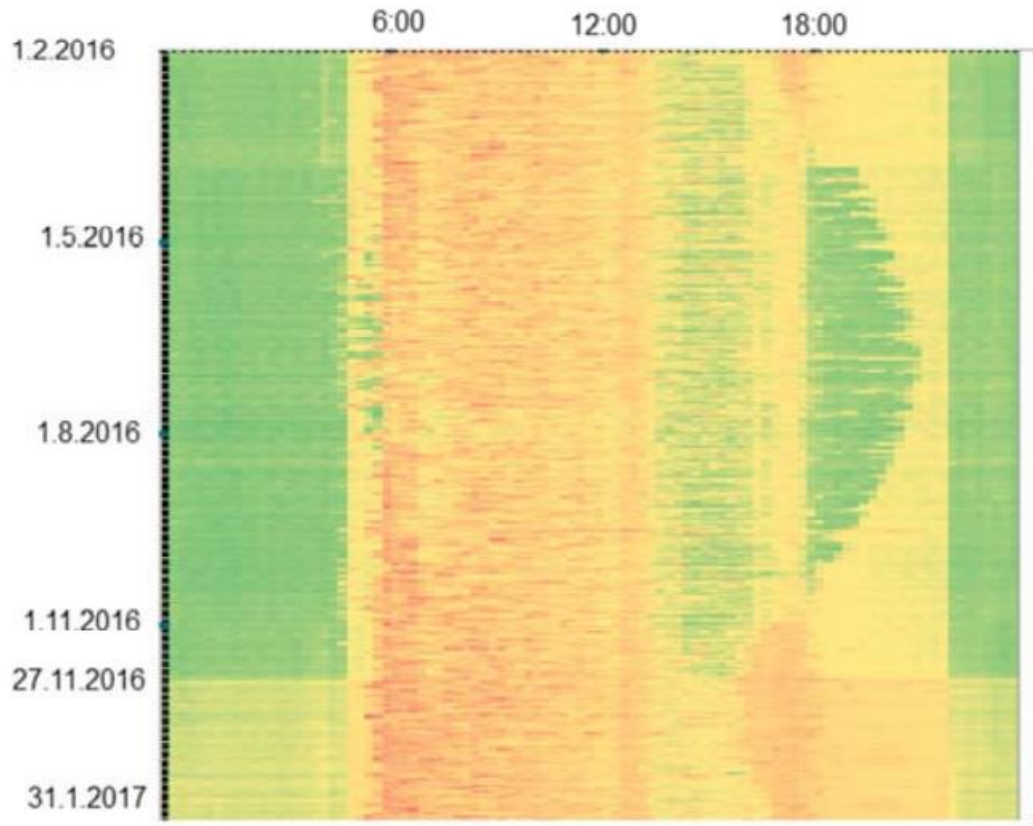


Create a carpet plot

14.08.2023: NOT DONE yet

- Create a carpet plot or heat map (kWh values on y-axis, hour of the day (e.g. 06:00, 12:00, 18:00) on x-axis

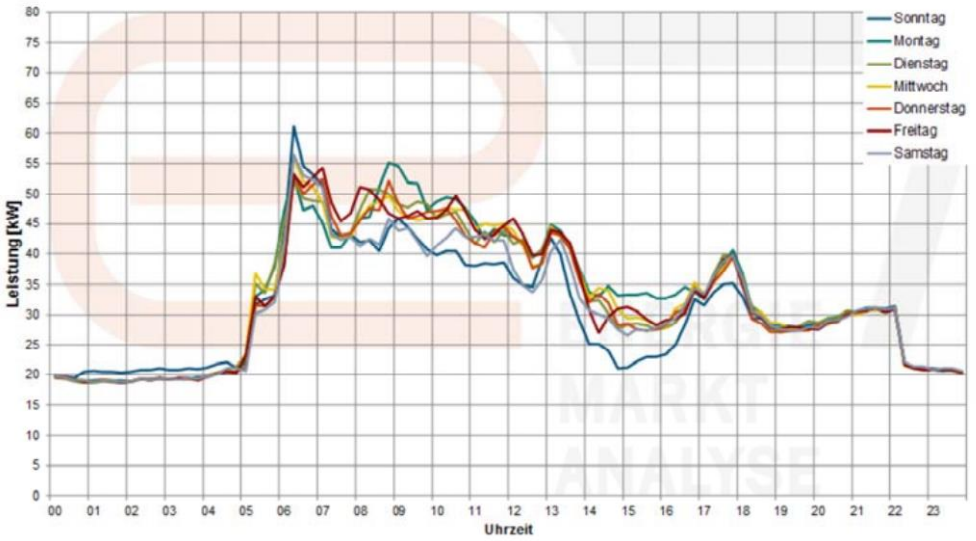
Example how the plot should look like



Create a line diagram of the average daily profile per day of the week

- Multiply each 15-min-value with 4 (converts kWh in kW)
- Calculate the average 15-minute value per weekday and time of the day (*e.g. on all mondays at 08:00 the average value is 45 kW*)
- Create a line diagram out of the calculated values (kWh values on y-axis, hour of the day (e.g. 06:00, 12:00, 18:00) on x-axis which shows the a line diagram per day

Example how the plot could look like



Legend:
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday

Hour of the day (on y-axis)

14.08.2023: NOT DONE yet

Updates 14.08.2023

- Include data sanity checks (you will learn more on this step in upcoming machine learning tasks with Hirschi)
  - Create a scatter plot: each datapoint (kWh) on y-axis, time (00:00) of occurence on x-axis
  - Create a scatter plot, where each data point (kWh) represents either the daily sum of a weekday (blue points) or daily sum of a weekend day (green points)
  - Create a histogram of all data points
  - New analysis for sanity checks see updated table on next page (in green)
- Merge two CSV„basic_info“ an „dataset_info“ in one CSV
- Include in the name of the csv export, if kWh or kW was chosen
- Include in the csv file itself, if kWh or kW was chosen (e.g. in the first row)
- Include the code for the plots in the same jupyter notebook

Description of analysis	Result of analysis (write all results in a CSV)
Name of dataset	Name of analysed CSV
Time period of data set	dd.mm.yyyy-dd.mm.yyyy
Count number of datapoints and perform a target/actual comparison	1 integer (should be ~35000), result of comparison (e.g. 99%)
Are there missing values? Missing timeperiod?	Timpestamp, timeperiod
Count zero values	Integers and and corresponding timestamp
Count negative values	Decimal values (kWh) and corresponding timestamp
Total energy consumption (=sum of all values)	1 Decimal (kWh)
Maximum value in dataset	1 Decimal value (kWh) and corresponding timestamp
Minimum value in dataset	1 Decimal value (kWh) and corresponding timestamp
Average and median of daily sum in dataset	2 Decimal values (kWh) and standard deviation
Average and median of daily sum on weekdays	2 Decimal values (kWh) and standard deviation
Average and median of daily sum on saturdays and sundays	2 Decimal values (kWh) and standard deviation
Average and median of on sundays at 03:00 in the morning	2 Decimal values (kWh) and standard deviation
Average daily sum between 08:00-17:00 on weekdays	1 Decimal value (kWh) and standard deviation
Average daily sum between 08:00-17:00 on Saturdays and sundays	1 Decimal value (kWh) and standard deviation
Average daily sum between 17:00-08:00 on weekdays	1 Decimal value (kWh) and standard deviation
Average daily sum between 17:00-08:00 on saturdays and sundays	1 Decimal value (kWh) and standard deviation

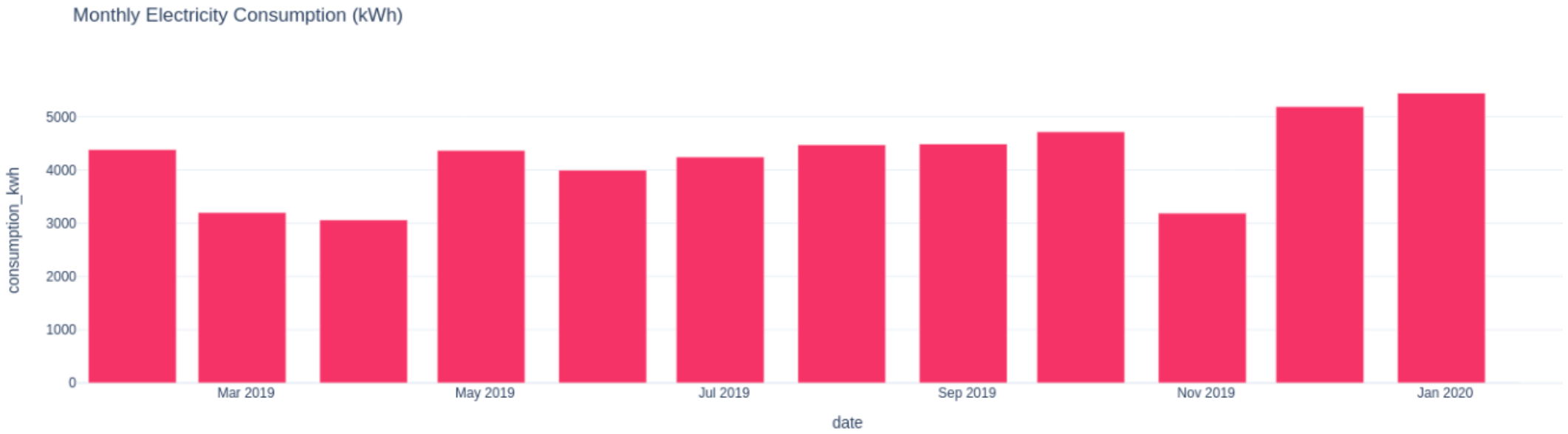
- Skip the following outputs in the CSV export, if kW was chosen:

Metric,Value
Name of dataset,C:\Users\there\Desktop\Github\Energy_consumption\data\raw\LP_AT0080000870000000000000000097058_2019.csv
Time period of dataset,2019-01-01 00:30:00 to 2020-01-01 00:00:00
Count of kWh values,35039
~~Total energy consumption (kWh),202942.75~~
Maximum value in dataset,21.92
Maximum value date,2019-09-06 11:00:00
Minimum value in dataset,0.63
Minimum value date,2019-03-31 03:15:00

Metric,Mean,Median,Standard Deviation
~~Daily sum,554.49,525.58,188.66~~
~~Weekday sum,431.31,499.18,312.91~~
~~Weekend sum,125.58,0.0,211.64~~
Sunday at 03:00,4.5,3.42,2.53
Weekdays 08:00-17:00,206.24,244.0,151.25
Weekends 08:00-17:00,44.67,0.0,76.32
Weekdays 17:00-08:00,225.63,269.66,166.81
Weekends 17:00-08:00,80.91,0.0,135.96

X-axis: add number of calendar week (1-52) and year

X-axis: add names of all months (MM.YY) to the bars



Do you have any ideas for creating a better x-axis-labeling? E.g. can you give the first day of each new month a different colour?