

## Asunto-osakeyhtiön asuntojen lämmitykseen käytetty energiankulutus. Lineaarisen regression avulla haettu yhtälö

Rakennusten lämmitys on lineaarisesti riippuvainen ulkolämpötilasta. Tämä riippuvuus voidaan kuvata yhtälöllä  $y=ax+b$ , missä  $a \neq 0$ . Sklearnilla  $a$  ja  $b$  voidaan selvittää, jolloin yhtälö on valmis. Tässä harjoituksessa selvitetään yhtälö yhden As Oy:n osalta. Yhtälön selvittämiseen tarvitaan seuraavia dataa: Taloyhtiön lämmitykseen käyttämä energiankulutus, ulkolämpötila ja vedenkulutus sikäli kun käyttövesi lämmitetään samalla energialähteellä kuin rakennukset. Kulutustiedot poimitaan Afry Oy:n websivustolta. Lämpötilatiedot puolestaan haetaan Ilmatieteenlaitoksen sivustolta.

In [ ]:

```
import numpy as np
import pandas as pd

# Mallinnuksessa käytetään sklearn, joten tuodaan sen vaatimat modulit.
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import train_test_split

#Tuodaan myös visualisointiin käytettävät modulit
import matplotlib.pyplot as plt
import seaborn as sns

# Estetään häiritsevät varoitukset
import warnings
warnings.filterwarnings("ignore", category=FutureWarning)
warnings.filterwarnings("ignore", category=DeprecationWarning)
```

### Keskivirheen laskeminen

In [ ]:

```
def mean_absolute_percentage_error(y_true, y_pred):
    y_true, y_pred = np.array(y_true), np.array(y_pred)
    return np.mean(np.abs((y_true - y_pred) / y_true)) * 100
```

Tuodaan data, joka on poimittu Afry Oy:n tarjoaman rajapinnan kautta, sekä käsitelty lämpötila ja tuulidata ja yhdistetään tiedostot samaan dataframeen.

In [ ]:

```
df = pd.read_excel('Data.xlsx') #Tuodaan kulutustiedot"
df_temp = pd.read_excel('C.xlsx') #Tuodaan lämpötilatiedot
df_temp['DS'] = pd.to_datetime(df_temp['DS'])

df_C = pd.merge(df, df_temp, on='DS', how='inner') #Yhdistetään dataframet.
df_C.replace(r'^-$', np.nan, regex=True, inplace=True) #muuttaa ne solut, joissa
NULL arvo on merkitty merkillä '.' arvoksi NaN. Huomioidaan lämpötilan
etumerkki!
```

In [ ]:

```
# Katsotaan vielä miltä data näyttää
print(df_C.describe())
```

		DS	MWh	M3	\
count		41102	41102.000000	41102.000000	
mean	2021-09-26 20:05:18.028319488		0.015255	0.178416	
min	2019-05-23 00:00:00		0.000000	0.000000	
25%	2020-07-24 20:15:00		0.005390	0.070000	
50%	2021-09-27 09:30:00		0.015105	0.150000	
75%	2022-11-29 12:45:00		0.022946	0.260000	
max	2024-01-31 23:00:00		0.064635	1.230000	
std		NaN	0.010747	0.137908	

	C	wind	hPa	moist%	Wperm2
count	41078.000000	41075.000000	41081.000000	40840.000000	40223.000000
mean	6.848384	4.333867	1011.339963	79.015720	113.119959
min	-26.000000	0.200000	961.700000	17.000000	-9.800000
25%	0.300000	2.800000	1004.400000	67.000000	-1.100000
50%	6.200000	4.100000	1012.400000	85.000000	3.300000
75%	14.100000	5.600000	1019.000000	94.000000	143.600000
max	32.400000	15.300000	1047.100000	100.000000	901.100000
std	9.386286	2.145019	11.538608	18.997075	194.687557

On yllättävää havaita, että energiankulutus saa myös nolla-arvoja. Tuon ei pitäisi olla mahdollista, koska veden kiertäessä putkistossa tapahtuu aina lämpöhäviöitä vaikka ulkolämpötila olisi raja-arvon yläpuolella ja vedenkulutus olisi nolla.

Katsotaan siis miltä data näyttää. Tulostetaan aluksi hajontakuvio. (Scatterplot)

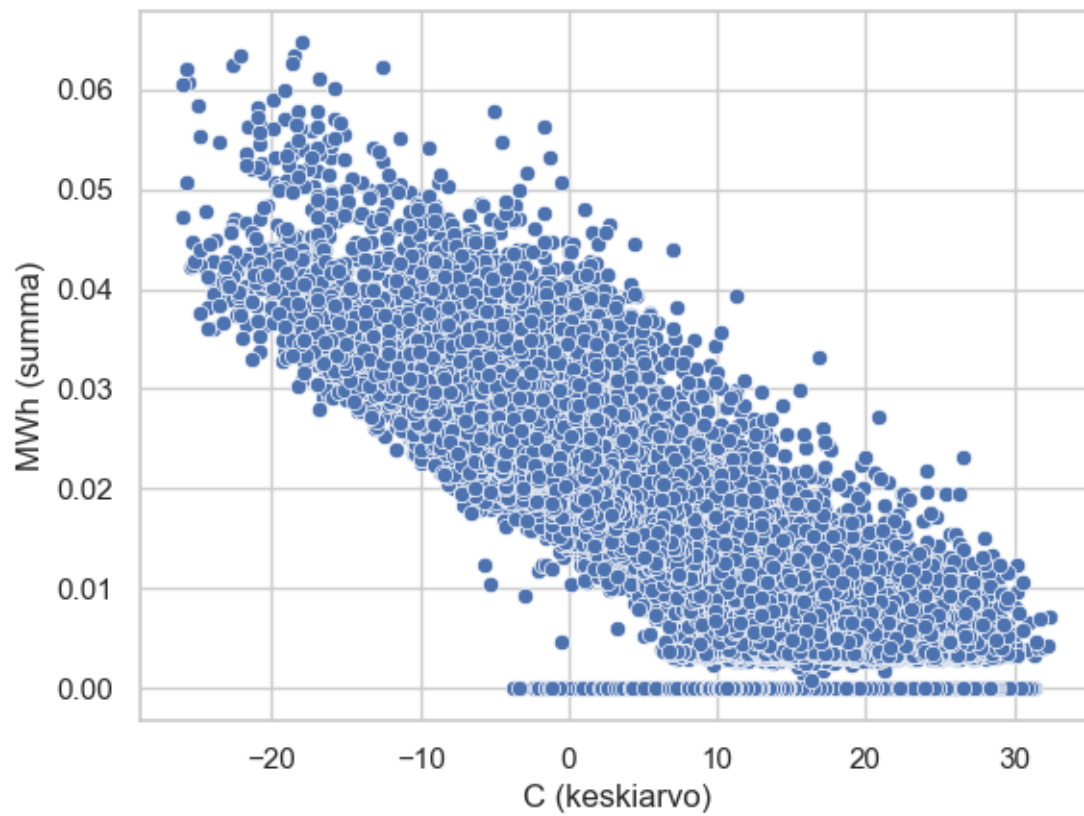
In [ ]:

```
# Aseta Seaborn-tyyli
sns.set(style="whitegrid")

# Luodaan pistekaavio
sns.scatterplot(x="C", y="MWh", data=df_C)

# Määrittele x- ja y-akseleiden otsikot
plt.xlabel('C (keskiarvo)')
plt.ylabel('MWh (summa)')

# Näytä kaavio
plt.show()
```



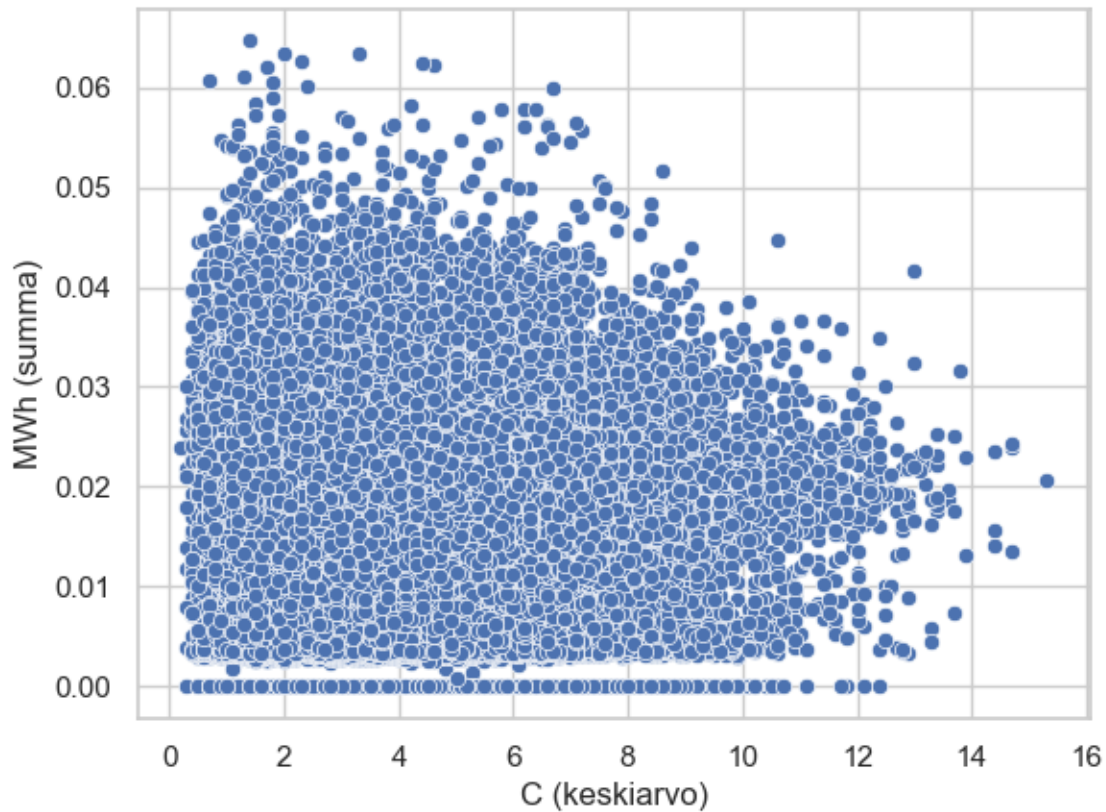
In [ ]:

```
# Aseta Seaborn-tyyli
sns.set(style="whitegrid")

# Luodaan pistekaavio
sns.scatterplot(x="wind", y="MWh", data=df_C)

# Määrittele x- ja y-akseleiden otsikot
plt.xlabel('C (keskiarvo)')
plt.ylabel('MWh (summa)')

# Näytä kaavio
plt.show()
```



Tuulen vaikutus ei näyttäisi olevan lainkaan niin selkeästi lineaarinen kuin ulkolämpötilan. Pidetään se kuitenkin vielä mukana laskennassa. Sillä näyttäisi kuitenkin olevan lievästi positiivinen vaikutus, kuten seuraavasta korrelaatiomatriisista voidaan päätellä.

Kuvaajissa näkyy paljon nollaenergiaa, joka on hieman yllättävää. Kyseessä on kirjausongelma eli energiaa on kulunut, mutta se ei ole kuitenkaan kirjautunut. Nämä nolla-arvot pitää poistaa.

Kuten kuvaajista näkyy, on energiankulutus joillakin riveillä todella nolla, joka käytännössä ei ole oikein mahdollista. Tarkastellaan vielä miltä jakauma näyttää histogrammissa. Sinänsä hajontakuviot näyttäisi siltä, että lineaarisuus on selkeä, mutta arvot hajaantuvat voimakkaasti. Tämä puolestaan johtuu siitä, että energiaa käytetään myös veden lämmitukseen ja vedenkulutus vaihtelee voimakkaasti ajallisesti ja määrällisesti.

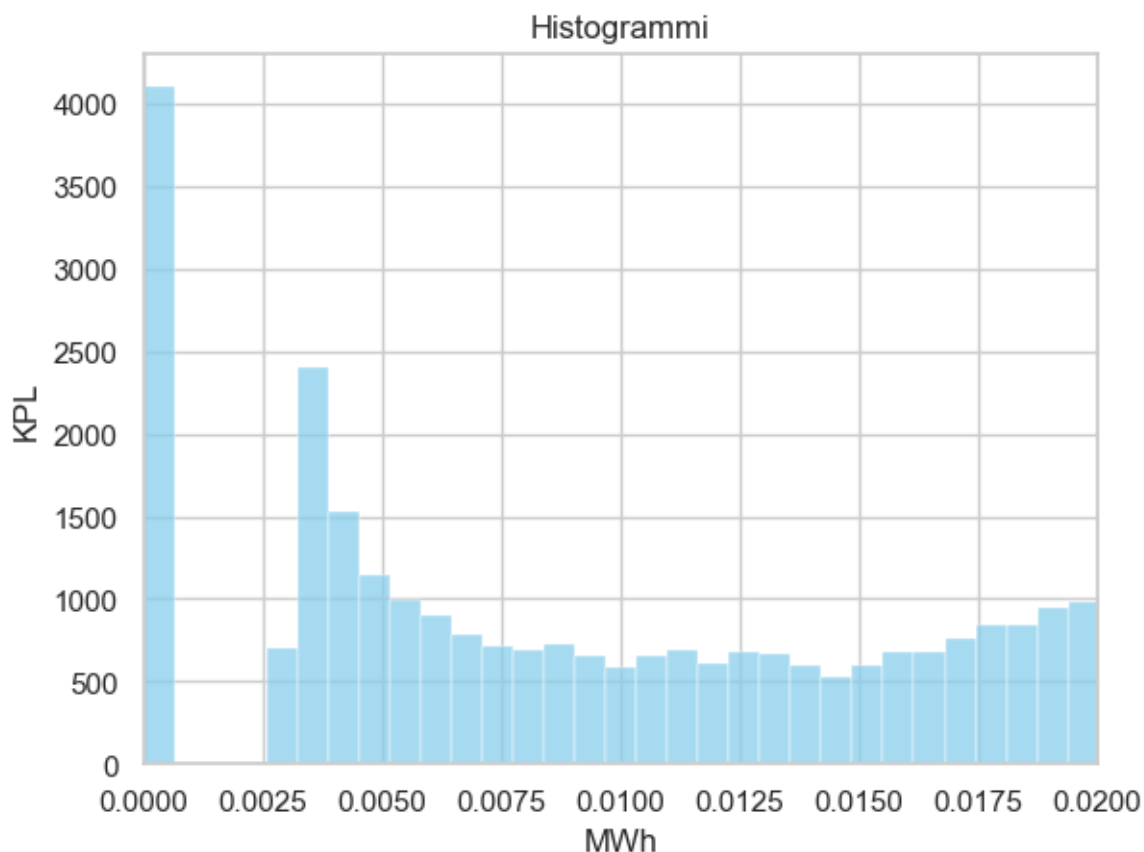
In [ ]:

```
sns.set(style="whitegrid")

# Piirretään histogrammi
sns.histplot(df_C['MWh'], kde=False, color="skyblue", bins=100)
plt.xlim(0, 0.02)
# Lisätään otsikko ja akselien nimet
plt.title('Histogrammi')
plt.xlabel('MWh')
plt.ylabel('KPL')

# Näytetään kuva
```

```
plt.show()
```



Histogrammin avulla voimme päätellä, että lämpöhäviöt ovat ehkä n. 0.0025 MW/h eli n. 25 KW/h. Tämä voidaan vielä varmistaa/tarkentaa, jähka saamme ensin laskettua asuntojen lämmittämisen yhtälön.

Jotta saamme laskettua paljonko rakennusten lämmittäminen vie energiaa ulkolämpötilan funktiona, on energian kulutuksesta pystyttävä erottelemaan sekä veden lämmittämiseen kuluva energia, että lämpöhäviöiden osuus. Jos datasta löytyy sellaisia rivejä, jolloin vettä ei käytetä, tai käytetään vain kylmää vettä ja nuo rivit poistetaan, niin silloin saamme laskettua rakennuksen lämmittämisen vaativan energian.

Poistetaan kuitenkin alkajaisiksi virhelähteet eli ne rivit, joissa energian kulutus on alle 0.0025 MWh.

In [ ]:

```
# Poistetaan rivit, joissa MWh saa arvon alle 0.0025 MWh / h joka tarkoittaa,  
että poistetaan n. 10% aineistosta  
dfMWH0025 = df_C[df_C["MWh"] >= 0.002]  
dfMWH0025.count()
```

Out[ ]:

DS	36996
MWh	36996
M3	36996
C	36972
wind	36969

```
hPa      36975
moist%    36774
Wperm2    36975
dtype: int64
```

Kaikissa sarakkeissa ei ilmeisesti ole lämpötilan ja tuulen arvoja. Koska niitä on selkeästi vähän, niin poistetaan ne.

```
dfMWH0 = dfMWH0025.dropna()
```

In [ ]:

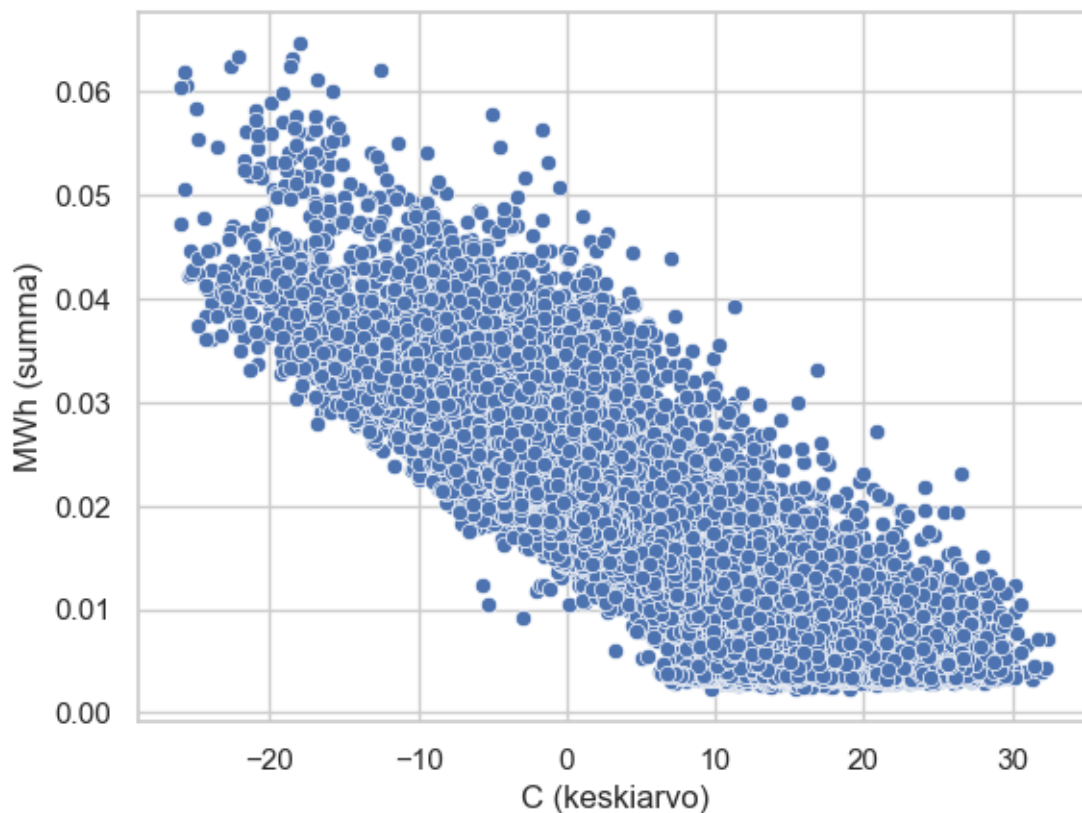
```
# Aseta Seaborn-tyyli
sns.set(style="whitegrid")

# Luodaan pistekaavio
sns.scatterplot(x="C", y="MWh", data=dfMWH0)

# Määrittele x- ja y-akselien otsikot
plt.xlabel('C (keskiarvo)')
plt.ylabel('MWh (summa)')

# Näytä kaavio
plt.show()
```

In [ ]:



Tarkastellaan vielä korrelaatioita.

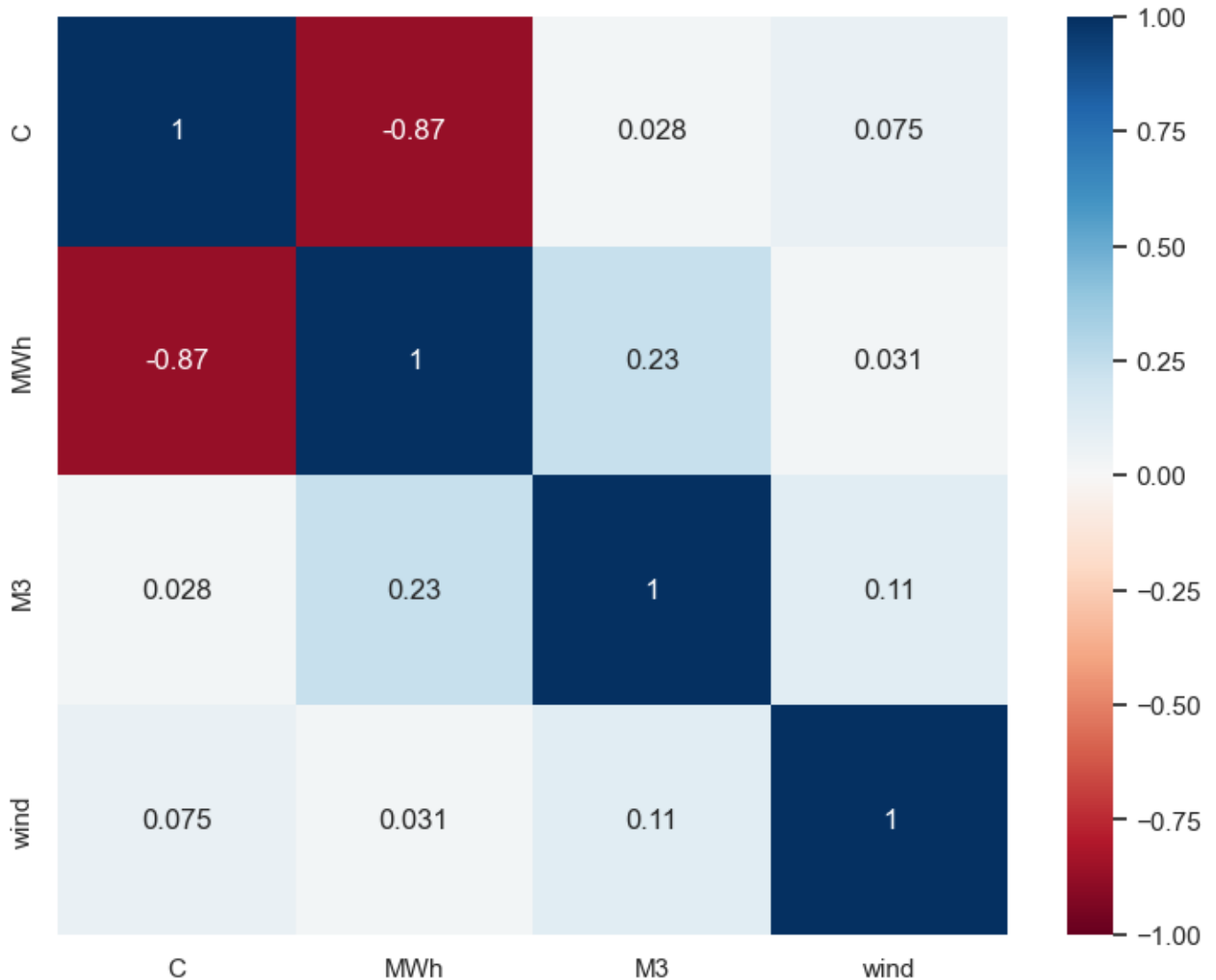
In [ ]:

```
df_corr = dfMWH0[['C', 'MWh', 'M3', 'wind']]

corr_matrix = df_corr.corr()
plt.figure(figsize=(9,7))
sns.heatmap(corr_matrix, vmin=-1.0, vmax=1.0, cmap='RdBu', annot=True)
```

<Axes: >

Out[ ]:



Ymmärrettävästi ulkolämpötilan ja energiankulutuksen välillä on vahva negatiivinen korrelaatio. Toisaalta veden- ja energiankulutuksen välillä on heikko positiivinen korrelaatio. Ymmärrettävää sikäli, että ulkolämpötila dominoi energiankulutusta ja toisaalta kylmää vettä käytettäessä energiaa ei kulu.

### Selvitetään rakennuksen lämmittämiseen kuluva energia.

Jotta saamme selville minkälainen yhtälö rakennuksen lämmittämiseksi syntyy, on poistettava veden lämmitykseen käytettävä energia sekä lämpöhäviöiden viemä energia. Aloitetaan niin, että poistetaan aineistosta ne rivit, joissa vedenkulutus on 0.0 l/t. Näin varmistetaan, että energiankulutus sisältää vain lämpöhäviöt ja rakennusten lämmitysenergian.

```
dfAs = dfMWH0[dfMWH0["M3"] <= 0.00]
lampohavio = dfMWH0['MWh'].min() # poimitaan minimilämpöhäviö
dfAs.describe()
```

In [ ]:

Out[ ]:

	DS	MWh	M3	C	wind	hPa	moist%	Wperm.
count	757	757.000000	757.0	757.000000	757.000000	757.000000	757.000000	757.000000
mean	2021-06-07 14:24:38.996037120	0.012759	0.0	5.942272	3.597622	1012.093659	86.677675	-1.249406
min	2019-05-23 01:00:00	0.002555	0.0	-24.000000	0.400000	965.900000	43.000000	-8.200000
25%	2020-02-03 02:00:00	0.004499	0.0	0.500000	2.300000	1005.900000	80.000000	-2.000000
50%	2020-12-10 02:00:00	0.011783	0.0	6.400000	3.300000	1013.400000	90.000000	-1.500000
75%	2022-09-12 01:00:00	0.018818	0.0	12.300000	4.600000	1019.200000	97.000000	-0.700000
max	2024-01-31 04:00:00	0.053119	0.0	23.700000	11.800000	1040.300000	100.000000	179.900000
std	NaN	0.008361	0.0	7.814392	1.886935	11.141081	12.663813	7.385771

In [ ]:

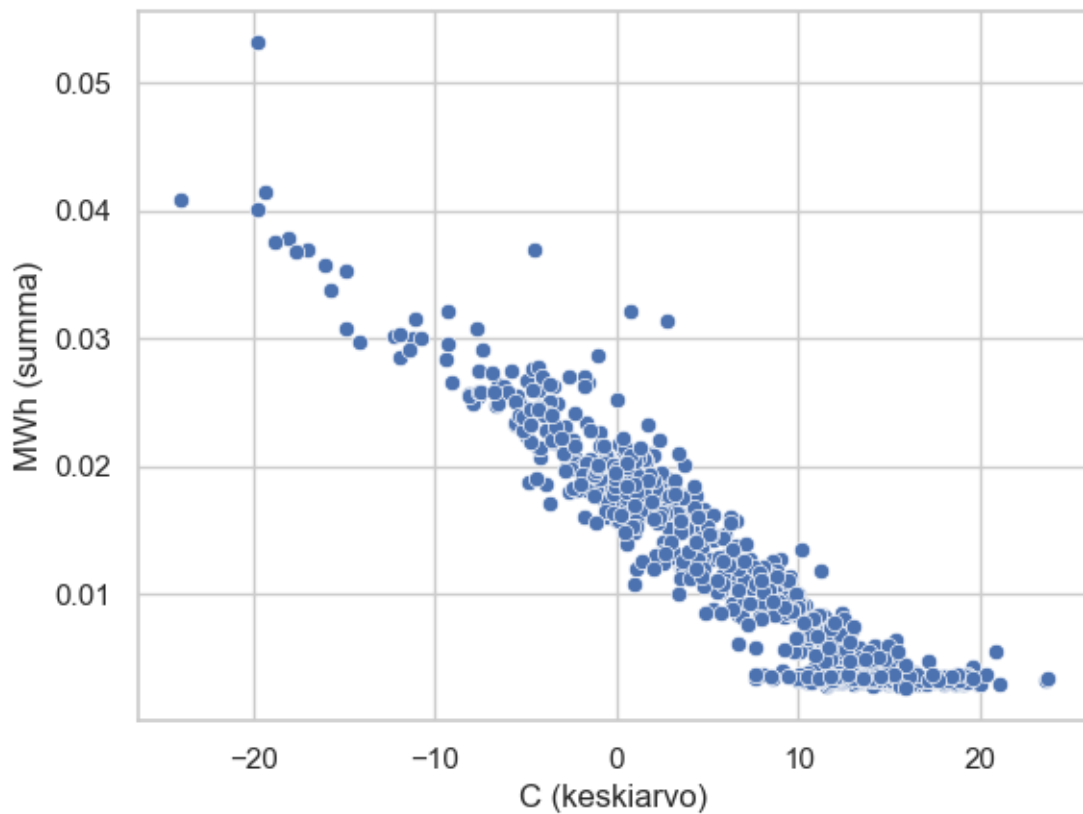
```
#Katsotaan vielä miltä hajontakuviio näyttää
sns.set(style="whitegrid")

# Luodaan pistekaavio
sns.scatterplot(x="C", y="MWh", data=dfAs)

# Määrittele x- ja y-akseleiden otsikot
plt.xlabel('C (keskiarvo)')
plt.ylabel('MWh (summa)')

# Näytä kaavio
plt.show()
```





Kuvaajassa näkyy jonkin verran outliereita, jonka lisäksi kuvio on lievästä jääkiekkomailan muotoinen. Yksi mahdollinen selitys tuohon on, että lämpöhäviöt eivät käyttäydy vakioisti.

Seuraavaksi vähennetään poistetaan lämpöhäviöt.

In [ ]:

```
# Aiemmin päädyimme siihen, että lämpöhäviöiden osuus energiankulutuksesta
minimissään luokkaa 0.025 MW/h. Vähennetään tämä sarakkeesta MWh
dfAs_2 = dfAs.copy()

# lasketaan vain lämmitysenergia
dfAs_2['MWh_025'] = dfAs_2['MWh']-lampohavio
dfAs_2 = dfAs_2.drop('MWh',axis=1)
# Katsotaan, vielä, ettei MWh mene alle nollan.
dfAs_2.describe()
```

Out[ ]:

	DS	M3	C	wind	hPa	moist%	Wperm2	MWh_025
<b>count</b>	757	757.0	757.000000	757.000000	757.000000	757.000000	757.000000	757.000000
<b>mean</b>	2021-06-07 14:24:38.996037120	0.0	5.942272	3.597622	1012.093659	86.677675	-1.249406	0.010522
<b>min</b>	2019-05-23 01:00:00	0.0	-24.000000	0.400000	965.900000	43.000000	-8.200000	0.000317
<b>25%</b>	2020-02-03 02:00:00	0.0	0.500000	2.300000	1005.900000	80.000000	-2.000000	0.002262

	DS	M3	C	wind	hPa	moist%	Wperm2	MWh_025
50%	2020-12-10 02:00:00	0.0	6.400000	3.300000	1013.400000	90.000000	-1.500000	0.009546
75%	2022-09-12 01:00:00	0.0	12.300000	4.600000	1019.200000	97.000000	-0.700000	0.016581
max	2024-01-31 04:00:00	0.0	23.700000	11.800000	1040.300000	100.000000	179.900000	0.050882
std	NaN	0.0	7.814392	1.886935	11.141081	12.663813	7.385771	0.008361

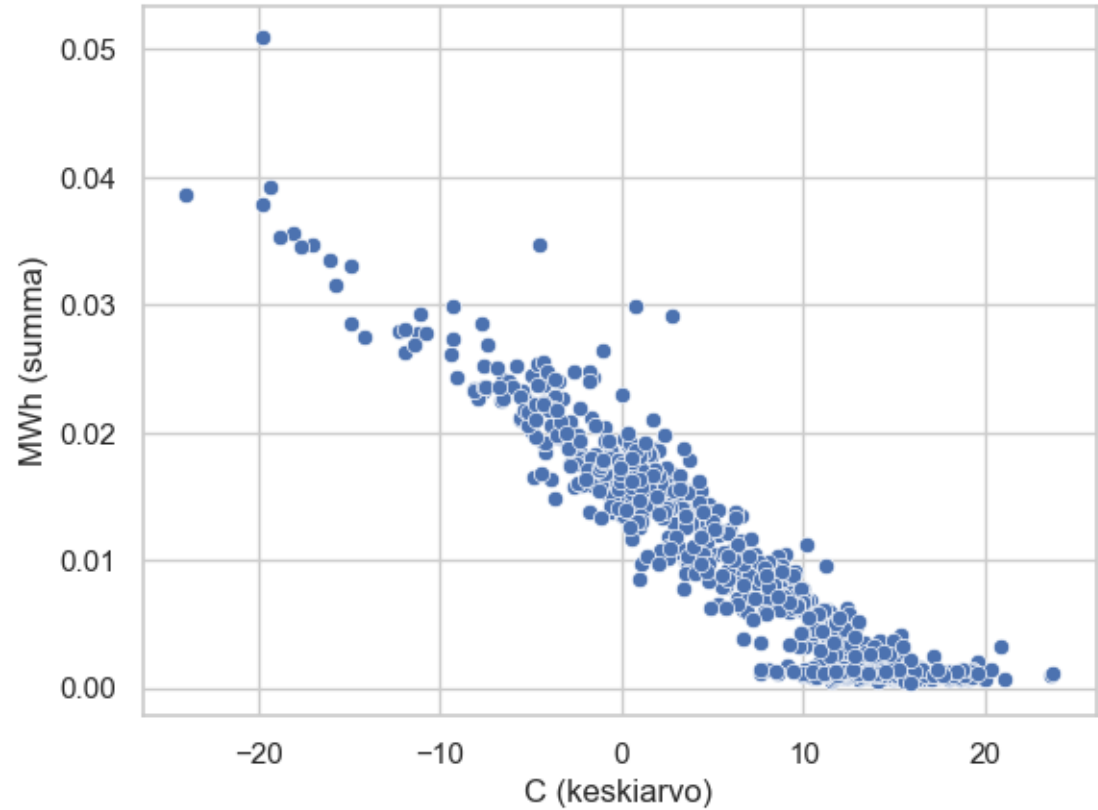
In [ ]:

```
#Katsotaan vielä miltä hajontakuviio näyttää
sns.set(style="whitegrid")

# Luodaan pistekaavio
sns.scatterplot(x="C", y="MWh_025", data=dfAs_2)

# Määrittele x- ja y-akseleiden otsikot
plt.xlabel('C (keskiarvo)')
plt.ylabel('MWh (summa)')

# Näytä kaavio
plt.show()
```



## Suoritetaan mallinnus

Jaetaan muuttujat selittäviin ja vastemuuttujaan

In [ ]:

```
# Selittävät muuttujat
x = dfAs_2[['C', 'wind']]
# Selitettävä muuttuja
y = dfAs_2['MWh_025']
```

Jaetaan aineisto koulutusdataan ja testidataan

In [ ]:

```
# jaetaan aineisto koulutus- ja testidataan. Käytetään 70/30 jakoa.
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3)
np.shape(x_train), np.shape(x_test), np.shape(y_train), np.shape(y_test)
```

Out [ ]:

```
((529, 2), (228, 2), (529,), (228,))
```

Sovitetaan malli.

In [ ]:

```
model = LinearRegression()
model.fit(x_train, y_train)
```

Out [ ]:

```
LinearRegression()
```

**In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.**

**On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.**

☒ [LinearRegression?Documentation for LinearRegression](#)  
LinearRegression()

Etsitään muuttujien kerroimet sekä leikkauspiste

In [ ]:

```
coef = pd.DataFrame(model.coef_, x_train.columns, columns = ['Coefficients'])
coef['Coefficients'].sort_values()
leikkauspiste = model.intercept_
rajalampo = -leikkauspiste/coef['Coefficients'][0] # tämä toteutuu kun y saa
arvon 0. Käytetään lisäksi x2:lle arvoa 0 (tuulen kerroin)
print(f"Rakennuksen lämmityksen yhtälö on siten seuraava: y =
{coef['Coefficients'][0]:.6f}x1 + {coef['Coefficients'][1]:.6f}x2 +
{leikkauspiste:.6f}")
print("Yhtälössä y on energiankulutus, x1 ulkolämpötila ja x2 tuulen
voimakkuus")
print(f"Käytännössä tämä tarkoittaa sitä, että rakennuksen lämmittämiseen ei
käytetä energiaa kun ulkolämpötila ylittää {rajalampo:.1f} astetta.")
Rakennuksen lämmityksen yhtälö on siten seuraava: y = -0.001009x1 + 0.000596x2 +
0.014410
Yhtälössä y on energiankulutus, x1 ulkolämpötila ja x2 tuulen voimakkuus
Käytännössä tämä tarkoittaa sitä, että rakennuksen lämmittämiseen ei käytetä
energiaa kun ulkolämpötila ylittää 14.3 astetta.
```

## Testataan mallin toimivuus

Regressiomallin toimivuutta arvioitaessa voidaan käyttää residuaaleja. Niille pätevät seuraavat yleisesti säännöt:

1. Nolla-keskiarvo: Residuaalien keskiarvon tulisi lähestyä nollaa. Tämä tarkoittaa, että malli ei yleisesti yli- tai aliarvioi riippuvaa muuttujaa.
2. Normaalius: Residuaalien tulisi noudattaa normaalijakaumaa. Tämä tarkoittaa sitä, että suurin osa residuaaleista sijoittuu nollan ympärille, ja niiden hajonta on symmetrinen.
3. Satunnaisuus: Residuaalien tulisi olla satunnaisesti hajautettuja ympäri nollaa. Tämä tarkoittaa sitä, että ei tulisi olla selkeää kuvioita residuaalien hajonnassa tietyissä ennustearvojen väleissä.
4. Varianssin vakaus: Residuaalien hajonnan pitäisi pysyä suhteellisen vakiona ennustettujen arvojen yli. Tätä kutsutaan homoskedastisuudeksi. Epävakaa hajonta voi viitata siihen, että malli ei pysty selittämään kaikkea tietoa tai että jotkin ennustajamuuttujat eivät ole sopivia.
5. Ei-autokorrelaatio: Residuaalien ei tulisi näyttää mitään järjestelmällistä kaavaa tai mallia, mikä viittaisi niihin liittyvään aikasarja-autokorrelaatioon. Tämä tarkoittaa sitä, että aikasarjaan liittyvät havainnot eivät ole toistensa kanssa korreloivia. (tässä ei testata aikasarjakorrelaatiota, vaikka aikasarjasta onkin kyse.)

In [ ]:

```
#Luodaan ennuste käyttämällä laskentaan ulkolämpötilan ja tuulen testiarvoja
y_pred= pd.DataFrame(model.predict(x_test))

#Jotta saadaan laskettua residuaalit niin määritellään y_pred indeksi samaksi
kuin y_test.
y_pred['index'] = x_test.index
y_pred.set_index('index', inplace=True, drop=True)
```

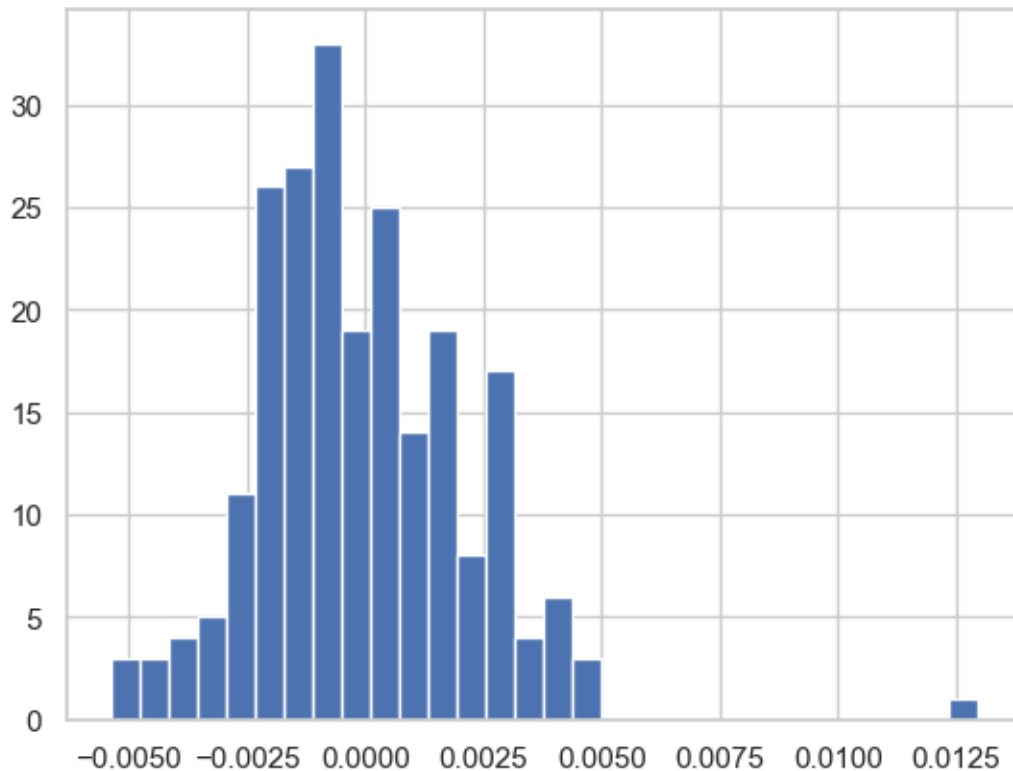
### Testataan residuaalien keskiarvo (1) ja jakauma (2)

In [ ]:

```
# Lasketaan testiaineistolle jäännösarvot.
residuals = pd.DataFrame(data = y_test)['MWh_025']-y_pred[0] #'MWH' ja ovat
sarakeotsikoita
# Katsotaan miten residuaalit käyttäytyvät.
print(f"Jäännösarvojen keskiarvo ({residuals.mean():.5f})")
residuals.hist(bins=30)
Jäännösarvojen keskiarvo (-0.00011)
```

Out[ ]:

<Axes: >



In [ ]:

```
print(residuals.head())
35135    -0.001913
37822     0.000158
36095    -0.001679
37868    -0.001217
2877     0.002233
dtype: float64
```

Jäännösarvojen keskiarvo on hyvin lähellä nollaa, joka on hyvä. Jäännösarvojen jakauma näyttää muistuttavan normaalijakaumaa. Suoritetaan kuitenkin vielä Shapiro-Wilkinin testi, jolla normaalijakauma voidaan testata.

In [ ]:

```
from scipy.stats import shapiro

# Suorita Shapiro-Wilkinin testi
stat, p_value = shapiro(residuals)

# Tulosta testin tulos
print("Testisuure:", stat)
print("p-arvo:", p_value)

# Tulkinta p-arvon perusteella
alpha = 0.05
if p_value > alpha:
    print("Data näyttää normaalijakautuneelta (hyväksytään nollahypoteesi)")
else:
```

```
print("Data ei näytä normaalijakautuneelta (hylätään nollahypoteesi)")
Testisuure: 0.9517172734967343
p-arvo: 6.555754046508354e-07
Data ei näytä normaalijakautuneelta (hylätään nollahypoteesi)
```

Shapiro-Wilkinin testin mukaan residuaalit eivät ole normaalijakautuneita.

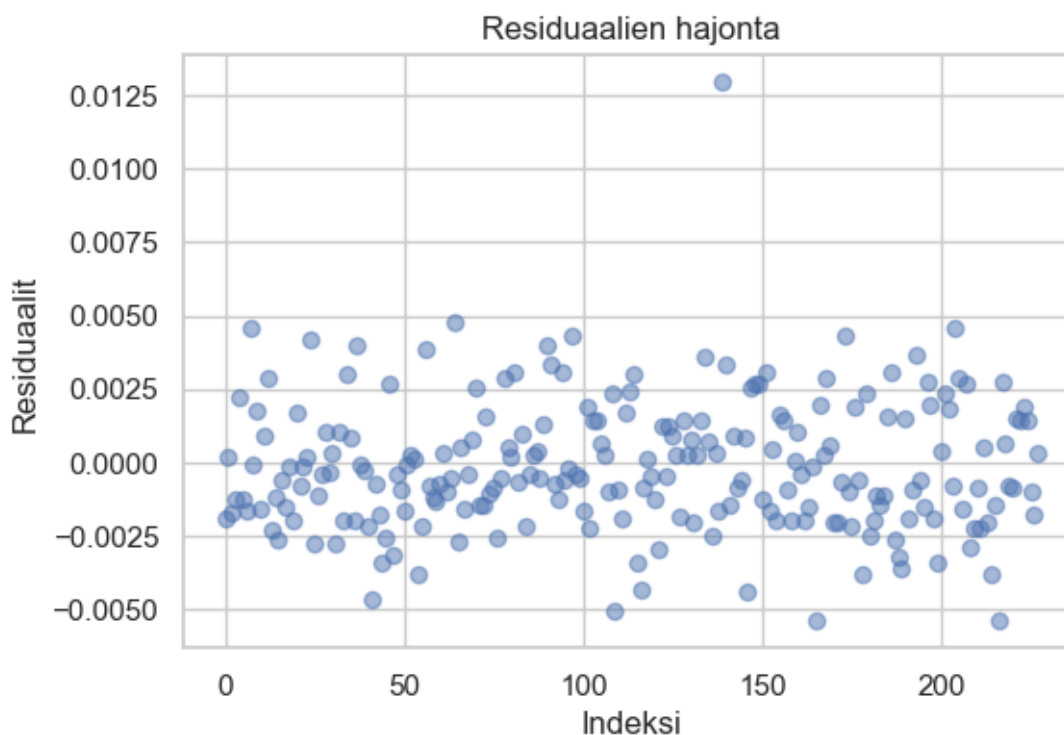
3. Tarkastetaan näyttävätkö residuaalit jakautuvan tasaisesti indeksin numeroavaruuteen hajontakuviossa. Jos näin on, niin se viittaa yleensä siihen, että residuaalien varianssi on suhteellisen vakaa.

In [ ]:

```
indeksi = range(len(residuals))
arvot = residuals
# Luo hajontakuvi
plt.figure(figsize=(6, 4)) # Aseta kuvion koko tarvittaessa
plt.scatter(indeksi, arvot, alpha=0.5) # alpha-parametri säätelee pisteen
läpinäkyvyyttä

# Aseta otsikko ja nimet akselit
plt.title('Residuaalien hajonta')
plt.xlabel('Indeksi')
plt.ylabel('Residuaalit')

# Näytä hajontakuvi
plt.show()
```



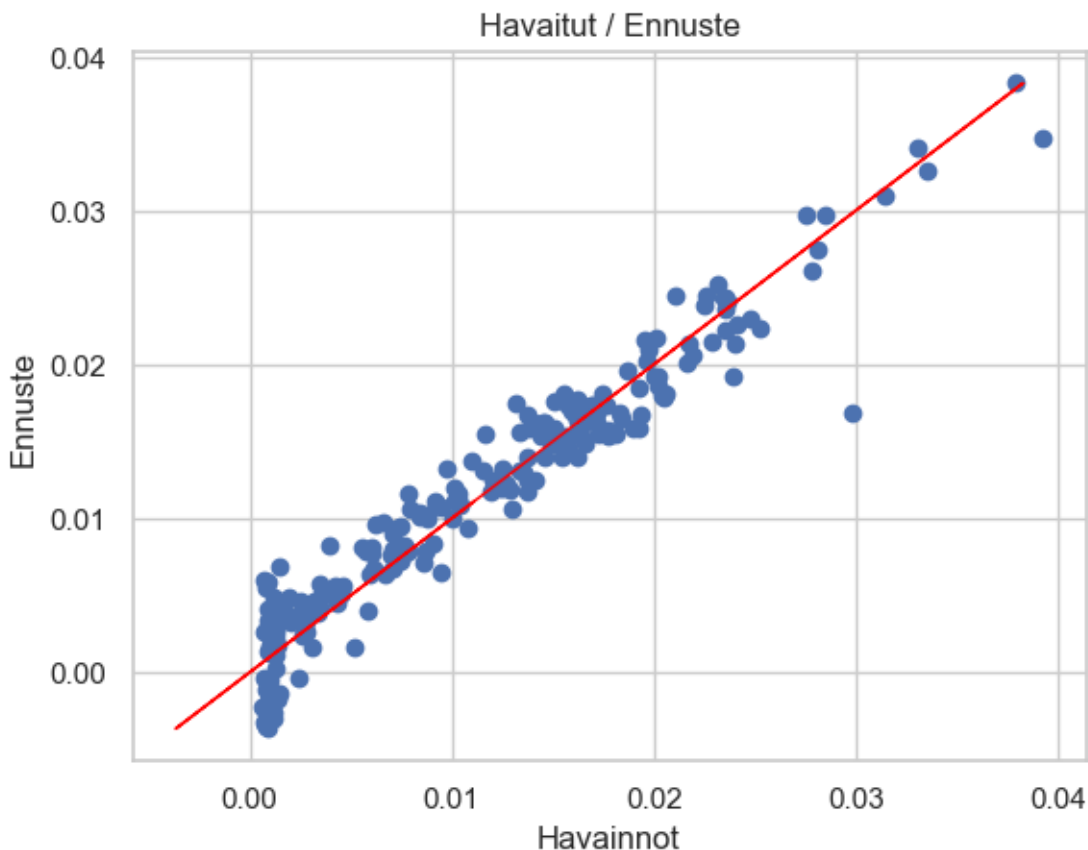
Testi näyttäisi residuaaliennäyttävät jakaantuvan tasaisesti indeksin numeroavaruuden yli, joten tämäkin on OK.

Katsotaan vielä miten residuaalit käyttäytyvät hajontakuviossa. 4)

In [ ]:

```
df2 = pd.DataFrame({'havainnot': y_test, 'ennuste': y_pred[0]})

plt.figure()
plt.scatter(x=y_test, y=y_pred)
plt.plot(y_pred, y_pred, color='red', linewidth=1) # Lisätään viiva
ennustettujen ja havaittujen arvojen väliin
plt.xlabel('Havainnot')
plt.ylabel('Ennuste')
plt.title('Havaitut / Ennuste')
plt.show()
```



Yllä olevassa hajontakuviossa ala-arvoilla hajonta on ehkä hieman suurempaa kuin muilla arvoilla, joskaan ei merkittävästi, joten siinä mielessä tulos on OK.

Lopuksi lasketaan Lasketaan vielä MSE ja R2

In [ ]:

```
mse = mean_squared_error(pd.DataFrame(data = y_test)['MWh_025'], y_pred[0])
r2s = r2_score(pd.DataFrame(data = y_test)['MWh_025'], y_pred[0])
print("MSE = ", mse)
print("R2s = ", r2s)
print()
print("MSE on erittäin pieni, jonka vuoksi voimme olla tyytyväisiä.")
```

```
print(f"Laskelman mukaan ulkolämpötilalla ja tuulen nopeudella voidaan selittää  
{r2s*100:.2f}% energiankulutuksesta")  
MSE = 5.033195131294758e-06  
R2s = 0.9319984440304977
```

MSE on erittäin pieni, jonka vuoksi voimme olla tyytyväisiä.  
Laskelman mukaan ulkolämpötilalla ja tuulen nopeudella voidaan selittää 93.20%  
energiankulutuksesta

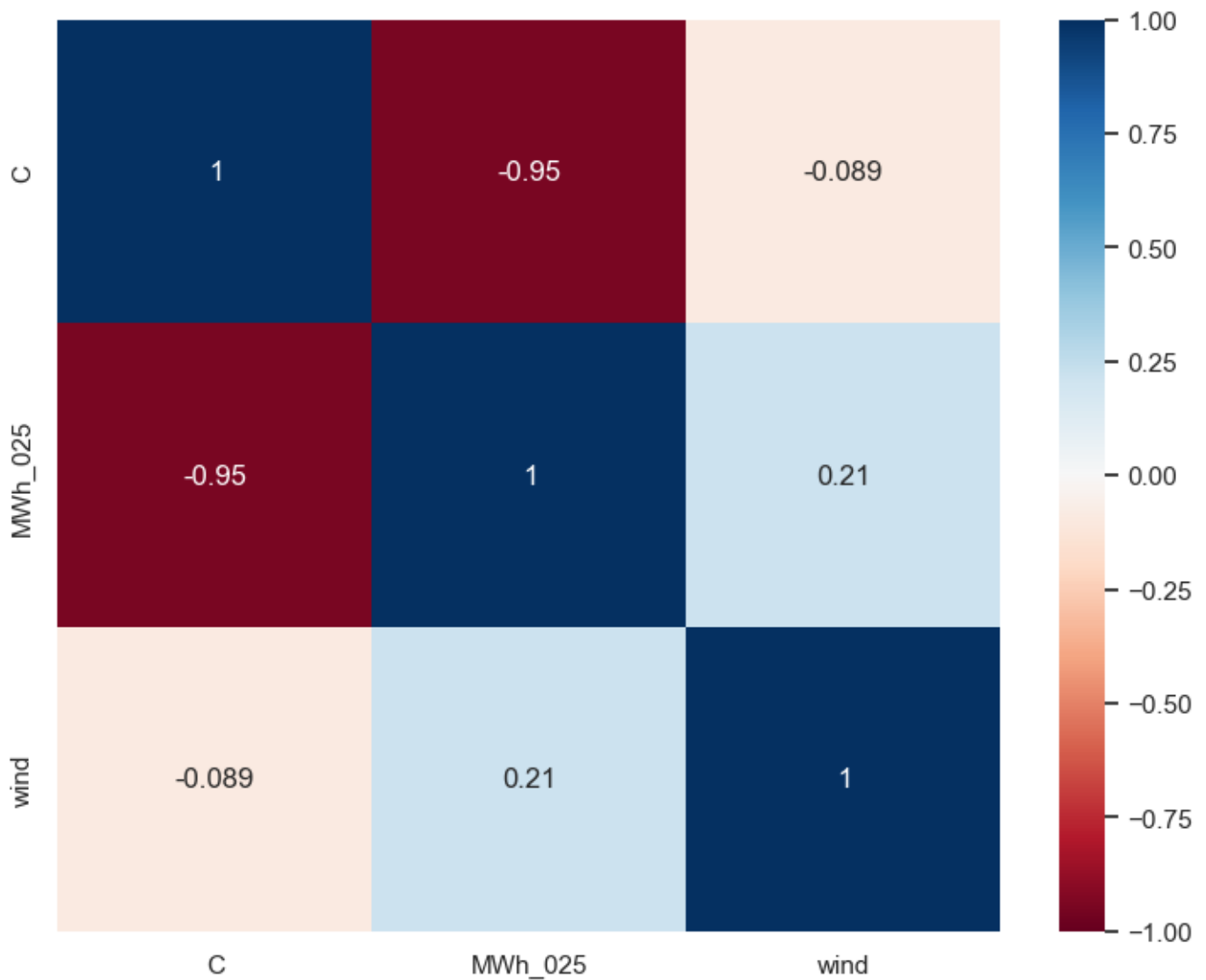
**Korrelaatio lämpötilan ja energiankulutuksen välillä on erittäin vahva.**

In [ ]:

```
df_corr = dfAs_2[['C', 'MWh_025', 'wind']]  
  
corr_matrix = df_corr.corr()  
plt.figure(figsize=(9,7))  
sns.heatmap(corr_matrix, vmin=-1.0, vmax=1.0, cmap='RdBu', annot=True)
```

Out[ ]:

<Axes: >





Tässä vaiheessa voimmekin sitten vielä varmistaa, että lämpöhäviöt todellakin ovat luokkaa 0.025 MWh. Poistetaan alkuperäisestä aineistosta virhedata (nolla energia), rivit, joissa ulkolämpötila ylittää arvon rajalämpötilan ja vedenkulutus alittaa 0.02 m<sup>3</sup>/h

In [ ]:

```
dfAs_3 = df_C[((df_C["M3"] <= 0.0) & (df_C['C']>=rajalampo)) &
(df_C['MWh']>=0.0015)]

waste = dfAs_3['MWh'].mean()/2 # Lämpöhäviöiden keskiarvo
print(f"Tämän perusteella saamme itse asiassa tarkennettua lämpöhäviöiden määrän. Se on itse asiassa hieman alunperin arvioitua suurempi.")
print(f"Keskimäärin lämpöhäviöitä syntyy ({waste:.4f}) MW/h.")
Tämän perusteella saamme itse asiassa tarkennettua lämpöhäviöiden määrän. Se on itse asiassa hieman alunperin arvioitua suurempi.
Keskimäärin lämpöhäviöitä syntyy (0.0018) MW/h.
```

In [ ]:

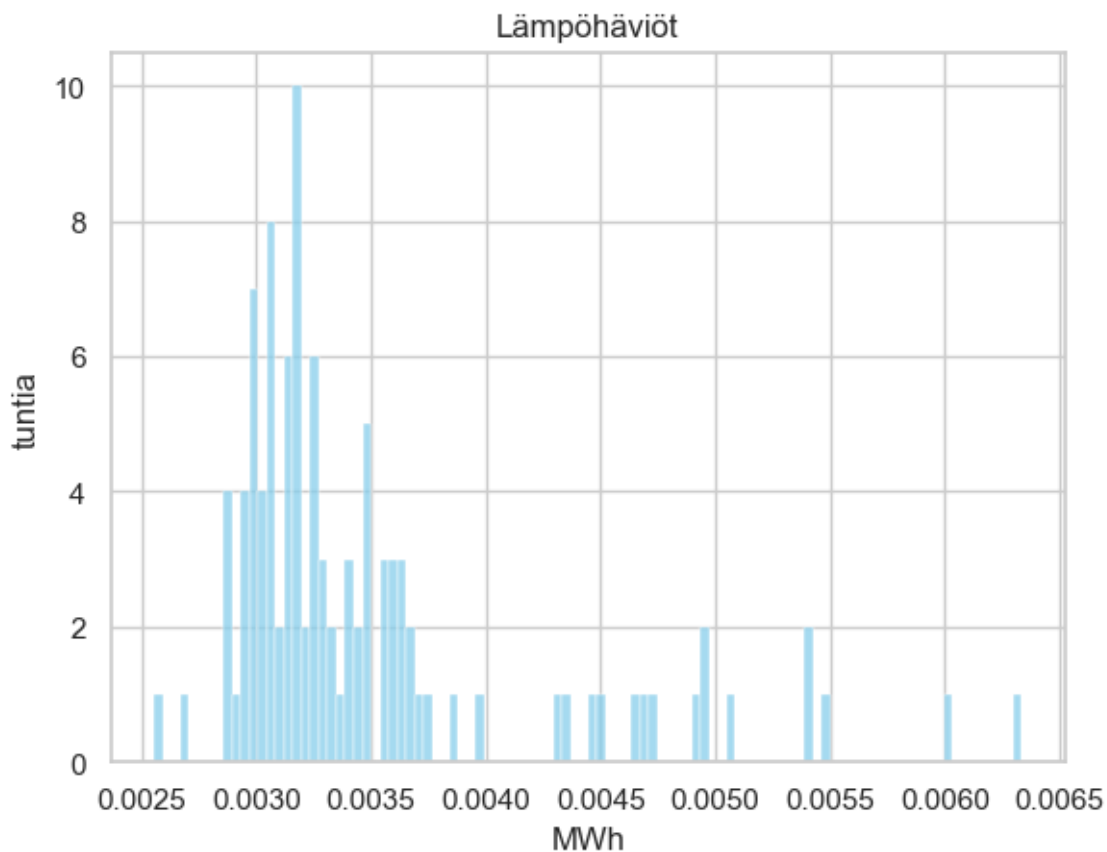
```
dfAs_3.info()
<class 'pandas.core.frame.DataFrame'>
Index: 103 entries, 1 to 38085
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   DS          103 non-null   datetime64[ns]
 1   MWh         103 non-null   float64
 2   M3          103 non-null   float64
 3   C           103 non-null   float64
 4   wind        103 non-null   float64
 5   hPa         103 non-null   float64
 6   moist%      103 non-null   float64
 7   Wperm2      103 non-null   float64
dtypes: datetime64[ns](1), float64(7)
memory usage: 7.2 KB
```

In [ ]:

```
sns.set(style="whitegrid")

# Piirretään histogrammi
sns.histplot(dfAs_3['MWh'], kde=False, color="skyblue", bins=100)
# Lisätään otsikko ja akselien nimet
plt.title('Lämpöhäviöt')
plt.xlabel('MWh')
plt.ylabel('tuntia')

# Näytetään kuva
plt.show()
```



**Lasketaan kuinka paljon kaukolämpökustannuksista olisi mahdollista säästää suljettaessa patteriverkoston kierto, jos lämpötila ylittää määritellyn rajalämmön.**

In [ ]:

```
# Kuukausihinnat ovat seuraavat:
Kuukausihinnat = {
    'kk': [1,2,3,4,5,6,7,8,9,10,11,12],

    'hinta':[82.36,82.36,51.29,51.29,31.47,26.25,26.25,26.25,32,51.83,62.55,82.36]}

df_hinnasto = pd.DataFrame(Kuukausihinnat)
```

In [ ]:

```
#Poimitaan mukaan ne rivit, jolloin lämpötila ylittää rajalämmön.
df_C = df_C[df_C["C"] >= rajalampo]

# Lisätään uudet sarakkeet 'Year' ja 'kk, koska kaukolämpö hinnoitellaan
kuukausittain.
df_C['Year'] = df_C['DS'].dt.year
df_C['kk'] = df_C['DS'].dt.month

#katsotaan mikä oli tilanne viime vuonna koska hinnasto oli viime vuoden
hinnasto
vuosi = df_C[df_C['DS'].dt.year == 2023]['Year'].max()
```

```

# Lasketaan kuinka monta tuntia per kk viime oli vuonna sellaisia, jolloin
lämpötila ylitti rajalämmön
df_count_per_month = df_C['kk'].value_counts().sort_index()

# Yhdistetään hinnasto ja rajalämmön ylittävät tuntimäärät
df_yhdistetty = pd.merge(df_hinnasto, df_count_per_month, on='kk')

# Luodaan uusi sarake 'Kustannus' kertomalla hinta, kulutus ja puolikas
lämpöhukasta
df_yhdistetty['Säästö'] = df_yhdistetty['hinta'] * df_yhdistetty['count'] *
waste/2

kokonaissäästö = df_yhdistetty['Säästö'].sum()
print(df_yhdistetty)
print()
print(f"Mikäli patteriverkoston kierto olisi sammutettu lämpötilan ylittäessä
rajalämmön ({rajalampo:.1f}) Celcius-astetta olisi kokonaissäästö vuodelle
{vuosi:.0f} ollut {kokonaissäästö:.2f} EUR")
df_yhdistetty.to_excel("saasto.xlsx")

```

	kk	hinta	count	Säästö
0	4	51.29	79	3.553914
1	5	31.47	662	18.272663
2	6	26.25	2586	59.539468
3	7	26.25	2873	66.147290
4	8	26.25	2751	63.338390
5	9	32.00	1071	30.059833
6	10	51.83	41	1.863855

Mikäli patteriverkoston kierto olisi sammutettu lämpötilan ylittäessä rajalämmön (14.3) Celcius-astetta olisi kokonaissäästö vuodelle 2023 ollut 242.78 EUR