	1. Skrap nettsiden og lagre resultatet i en dataramme. Hint: argumentet "on_bad_lines='skip'" kan være nyttig når du skal laste opp csv-filen
n [46]: n [47]:	<pre>from bs4 import BeautifulSoup import requests def fetch_html_tables(url): "Returns a list of tables in the html of url" page = requests.get(url) bs=BeautifulSoup(page.content) tables=bs.find_all('table') return tables</pre>
	<pre>tables=fetch_html_tables('https://www.motor.no/aktuelt/motors-store-vintertest-av-rekkevidde-pa-elbiler/217132') table_html=tables[0] #printing top print(str(table_html)[:1000])</pre>
	Modell (temp. varierte fra 0° til -10°) WLTP-tall *td> *td> *td> *td> *td> *STOPP
	Avvik
n [48]:	<pre>def html_to_table(html): "Returns the table defined in html as a list" table=[] for row in html.find_all('tr'): r=[]</pre>
	<pre>cells=row.find_all('td') if len(cells)==0: cells=row.find_all('th') for cell in cells: cell=format(cell) r.append(cell)</pre>
	<pre>table.append(r) return table def format(cell): "Returns a string after converting bs4 object cell to clean text" if cell.content is None: s=cell.text elif len(cell.content)==0: return ''</pre>
	<pre>else: s=' '.join([str(c) for c in cell.content]) s=s.replace('\xa0','') s=s.replace('\n','') return s table=html_to_table(table_html)</pre>
n [49]:	print(str(table)[:1000]) [[' Modell (temp. varierte fra 0° til -10°) ',' WLTP-tall ',' STOPP ',' Avvik '], ['Tesla Modell Y Labell Modell (temp. varierte fra 0° til -10°) ',' STOPP ',' Avvik '], ['Tesla Modell Y Labell Modell (temp. varierte fra 0° til -10°) ',' STOPP ',' Avvik '], ['Tesla Modell Y Labell Modell Modell Y Labell Modell Y Labell Modell Y Labell Modell Y Labell Mod
ıt[49]:	' Modell (temp. varierte fra 0° til -10°) ; WLTP-tall ; STOPP ; Avvik '
n [51]: ut[51]:	<pre>import pandas as pd df = pd.read_csv('rentebarometer.csv', delimiter=';', on_bad_lines='skip') df #Henter ut datasett Modell (temp. varierte fra 0° til -10°)</pre>
	1 Mercedes-Benz EQS 580 4matic 645 km/18,3 kWh 513 km -20,47 % 2 BMW iX xDrive50 591 km/21,4 kWh 503 km -14,89 % 3 Tesla Model Y LR Dual motor 507 km/16,9 kWh 451 km -11,05 % 4 Volkswagen ID.3 PRO S 539 km/16,3 kWh 435 km -19,29 % 5 Kia EV6 2WD 528 km/16,5 kWh 429 km -18,75 % 6 NIO ES8 LR 7-seter 488 km/21,5 kWh 425 km -12,91 % 7 Kia EV6 4WD 484 km/18,0 kWh 423 km -12,60 %
	8 Volkswagen ID.4 Pro 485 km/18,4 kWh 414 km -14,64 % 9 Hyundai Ioniq 5 2WD 481 km/16,8 kWh 408 km -15,18 % 10 BMW i4 M50 497 km/19,0 kWh 406 km -18,31 % 11 Skoda Enyaq iV80X 477 km/18,2 kWh 403 km -15,51 % 12 Porsche Taycan 4 Cross Turismo 456 km/22,4 kWh 402 km -11,84 % 13 Polestar 2 LR Single motor 517 km/18,6 kWh 400 km -22,63 % 14 Audi e-tron GT 463 km/21,1 kWh 392 km -15,33 %
	15
	21 Audi e-tron Q4 50 quattro
	28 Mercedes-Benz EQA 250 401 km/17,7 kWh 331 km -17,46 % 29 BMW iX xDrive40 402 km/20,7 kWh 316 km -21,39 % 30 Mercedes-Benz EQB 350 4matic 407 km/18,1 kWh 315 km -22,60 % 31 Opel Mokka-e 338 km/16,2 kWh 263 km -22,19 % 32 Peugeot e-2008 320 km/15,6 kWh 228 km -28,75 %
	2. Vask dataen til en tabell med bare relevante tall og gode overskrifter. Her er det mange veier til Rom, men en måte å gå fram på kan vøre den følgende: • Del kolonnen med WLTP-tall i to: en for kjørte km og en for energibruk. Hint: metoden "str.split()" og "drop()" kan være nyttige her. • Fjern benevninger i alle kolonner med relevante tall. Hint: kommandoen "str.replace" kan være nyttig her. • Noen rader i datarammen inneholder ikke gyldige verdier (Polestar og Hyundai), og bør fjernes. Det kan gjøres med å sørge for at alle tallene faktisk er positive av typen "float". df = df.replace('x', '0') df #bruker replace funksjonen til å endre fra x til 0
ut[52]:	Modell (temp. varierte fra 0° til -10°) WLTP-tall STOPP Avvik 0 Tesla Model 3 LR Dual motor 614 km/14.7 kWh 521 km -15.15 % 1 Mercedes-Benz EQS 580 4matic 645 km/18.3 kWh 513 km -20.47 % 2 BMW iX xDrive50 591 km/21.4 kWh 503 km -14.89 % 3 Tesla Model Y LR Dual motor 507 km/16.9 kWh 451 km -11.05 % 4 Volkswagen ID.3 PRO S 539 km/16.3 kWh 435 km -19.29 % 5 Kia EV6 2WD 528 km/16.5 kWh 429 km -18.75 %
	6 NIO ES8 LR 7-seter 488 km/21,5 kWh 425 km -12,91 % 7 Kia EV6 4WD 484 km/18,0 kWh 423 km -12,60 % 8 Volkswagen ID.4 Pro 485 km/18,4 kWh 414 km -14,64 % 9 Hyundai Ioniq 5 2WD 481 km/16,8 kWh 408 km -15,18 % 10 BMW i4 M50 497 km/19,0 kWh 406 km -18,31 % 11 Skoda Enyaq iV80X 477 km/18,2 kWh 403 km -15,51 % 12 Porsche Taycan 4 Cross Turismo 456 km/22,4 kWh 402 km -11,84 %
	13 Polestar 2 LR Single motor 517 km/18,6 kWh 400 km -22,63 % 14 Audi e-tron GT 463 km/21,1 kWh 392 km -15,33 % 15 Xpeng P7 470 km/19,4 kWh 383 km -18,51 % 16 Audi e-tron Q4 40 485 km/18,6 kWh 380 km -21,65 % 17 Hyundai loniq 5 4WD (19-tommer) 460 km/17,7 kWh 369 km -19,78 % 18 Hyundai loniq 5 LR 4WD (20-tommer) 430 km/17,7 kWh 0 0 19 BYD Tang 400 km/21,6 kWh 356 km -11,00 %
	20 Volkswagen ID.4 GTX 466 km/18,6 kWh 353 km -24,20 % 21 Audi e-tron Q4 50 quattro 459 km/19,1 kWh 349 km -23,97 % 22 Skoda Enyaq iV80 509 km/17,7 kWh 347 km -31,83 % 23 Tesla Model 3 SR 448 km/14,0 kWh 346 km -22,87 % 24 Polestar 2 LR Dual motor 476 km/20,2 kWh 340 km -28,57 % 25 Polestar 2 LR Dual motor (m bagasie) 470 km/19,5 kWh 0 0
	26 Cupra Born 395 km/15,4 kWh 339 km -14,18 % 27 Volvo C40 Recharge 437 km/21,1 kWh 333 km -23,80 % 28 Mercedes-Benz EQA 250 401 km/17,7 kWh 331 km -17,46 % 29 BMW iX xDrive40 402 km/20,7 kWh 316 km -21,39 % 30 Mercedes-Benz EQB 350 4matic 407 km/18,1 kWh 315 km -22,60 % 31 Opel Mokka-e 338 km/16,2 kWh 263 km -22,19 % 32 Peugeot e-2008 320 km/15,6 kWh 228 km -28,75 %
n [53]: n [54]: ut[54]:	df #Splitter opp WLP tall #Kode hentet fra: https://www.google.com/search?q=how+to+split+up+a+column+in+python&sxsrf=APwXEddX_YMVFYs0XVQ0sHzhWSJwpd-AtQ%3A1681979983823&ei=T_pAZJnXMdmVxc8PtP6e4Aw&ved=0ahUKEwjZutHbh7j-AhXZSvEDHTS_B8wQ4dUDCA4&ud
	0 Tesla Model 3 LR Dual motor 614 km/14,7 kWh 521 km -15,15 % 614 km 14,7 kWh 1 Mercedes-Benz EQS 580 4matic 645 km/18,3 kWh 513 km -20,47 % 645 km 18,3 kWh 2 BMW iX xDrive50 591 km/21,4 kWh 503 km -14,89 % 591 km 21,4 kWh 3 Tesla Model Y LR Dual motor 507 km/16,9 kWh 451 km -11,05 % 507 km 16,9 kWh 4 Volkswagen ID.3 PRO S 539 km/16,3 kWh 435 km -19,29 % 539 km 16,3 kWh 5 Kia EV6 2WD 528 km/16,5 kWh 429 km -18,75 % 528 km 16,5 kWh 6 NIO ES8 LR 7-seter 488 km/21,5 kWh 425 km -12,91 % 488 km 21,5 kWh
	7 Kia EV6 4WD 484 km/18,0 kWh 423 km -12,60 % 484 km 18,0 kWh 8 Volkswagen ID.4 Pro 485 km/18,4 kWh 414 km -14,64 % 485 km 18,4 kWh 9 Hyundai Ioniq 5 2WD 481 km/16,8 kWh 408 km -15,18 % 481 km 16,8 kWh 10 BMW i4 M50 497 km/19,0 kWh 406 km -18,31 % 497 km 19,0 kWh 11 Skoda Enyaq iV80X 477 km/18,2 kWh 403 km -15,51 % 477 km 18,2 kWh 12 Porsche Taycan 4 Cross Turismo 456 km/22,4 kWh 402 km -11,84 % 456 km 22,4 kWh 13 Polestar 2 LR Single motor 517 km/18,6 kWh 400 km -22,63 % 517 km 18,6 kWh
	14 Audi e-tron GT 463 km/21,1 kWh 392 km -15,33 % 463 km 21,1 kWh 15 Xpeng P7 470 km/19,4 kWh 383 km -18,51 % 470 km 19,4 kWh 16 Audi e-tron Q4 40 485 km/18,6 kWh 380 km -21,65 % 485 km 18,6 kWh 17 Hyundai Ioniq 5 4WD (19-tommer) 460 km/17,7 kWh 369 km -19,78 % 460 km 17,7 kWh 18 Hyundai Ioniq 5 LR 4WD (20-tommer) 430 km/17,7 kWh 0 0 430 km 17,7 kWh 19 BYD Tang 400 km/21,6 kWh 356 km -11,00 % 400 km 21,6 kWh
	20 Volkswagen ID.4 GTX 466 km/18,6 kWh 353 km -24,20% 466 km 18,6 kWh 21 Audi e-tron Q4 50 quattro 459 km/19,1 kWh 349 km -23,97% 459 km 19,1 kWh 22 Skoda Enyaq iV80 509 km/17,7 kWh 347 km -31,83% 509 km 17,7 kWh 23 Tesla Model 3 SR 448 km/14,0 kWh 346 km -22,87% 448 km 14,0 kWh 24 Polestar 2 LR Dual motor 476 km/20,2 kWh 340 km -28,57% 476 km 20,2 kWh 25 Polestar 2 LR Dual motor (m bagasje) 470 km/19,5 kWh 395 km -14,18% 395 km 15,4 kWh
	27 Volvo C40 Recharge 437 km/21,1 kWh 333 km -23,80 % 437 km 21,1 kWh 28 Mercedes-Benz EQA 250 401 km/17,7 kWh 331 km -17,46 % 401 km 17,7 kWh 29 BMW iX xDrive40 402 km/20,7 kWh 316 km -21,39 % 402 km 20,7 kWh 30 Mercedes-Benz EQB 350 4matic 407 km/18,1 kWh 315 km -22,60 % 407 km 18,1 kWh 31 Opel Mokka-e 338 km/16,2 kWh 263 km -22,19 % 338 km 16,2 kWh 32 Peugeot e-2008 320 km/15,6 kWh 228 km -28,75 % 320 km 15,6 kWh
n [55]: ut[55]:	
	0 Tesla Model 3 LR Dual motor 614 km/14,7 kWh 521 -15,15 % 614 147 1 Mercedes-Benz EQS 580 4matic 645 km/18,3 kWh 513 -20,47 % 645 183 2 BMW iX xDrive50 591 km/21,4 kWh 503 -14,89 % 591 214 3 Tesla Model Y LR Dual motor 507 km/16,9 kWh 451 -11,05 % 507 169 4 Volkswagen ID.3 PRO S 539 km/16,3 kWh 435 -19,29 % 539 163 5 Kia EV6 2WD 528 km/16,5 kWh 429 -18,75 % 528 165 6 NIO ESS LR 7-seter 488 km/21,5 kWh 425 -12,91 % 488 215
	7 Kia EV6 4WD 484 km/18,0 kWh 423 -12,60 % 484 180 8 Volkswagen ID.4 Pro 485 km/18,4 kWh 414 -14,64 % 485 184 9 Hyundai Ioniq 5 2WD 481 km/16,8 kWh 408 -15,18 % 481 168 10 BMW i4 M50 497 km/19,0 kWh 406 -18,31 % 497 190 11 Skoda Enyaq iV80X 477 km/18,2 kWh 403 -15,51 % 477 182 12 Porsche Taycan 4 Cross Turismo 456 km/22,4 kWh 402 -11,84 % 456 224 13 Polestar 2 LR Single motor 517 km/18,6 kWh 400 -22,63 % 517 186
	14 Audi e-tron GT 463 km/21,1 kWh 392 -15,33 % 463 211 15 Xpeng P7 470 km/19,4 kWh 383 -18,51 % 470 194 16 Audi e-tron Q4 40 485 km/18,6 kWh 380 -21,65 % 485 186 17 Hyundai Ioniq 5 4WD (19-tommer) 460 km/17,7 kWh 369 -19,78 % 460 177 18 Hyundai Ioniq 5 LR 4WD (20-tommer) 430 km/17,7 kWh 0 0 430 177 19 BYD Tang 400 km/21,6 kWh 356 -11,00 % 400 216
	20 Volkswagen ID.4 GTX 466 km/18,6 kWh 353 -24,20 % 466 186 21 Audi e-tron Q4 50 quattro 459 km/19,1 kWh 349 -23,97 % 459 191 22 Skoda Enyaq iV80 509 km/17,7 kWh 347 -31,83 % 509 177 23 Tesla Model 3 SR 448 km/14,0 kWh 346 -22,87 % 448 140 24 Polestar 2 LR Dual motor (m bagasje) 470 km/19,5 kWh 0 0 0 470 195 25 Polestar 2 LR Dual motor (m bagasje) 470 km/19,5 kWh 339 -14,18 % 395 154
	27 Volvo C40 Recharge 437 km/21,1 kWh 333 -23,80 % 437 211 28 Mercedes-Benz EQA 250 401 km/17,7 kWh 331 -17,46 % 401 177 29 BMW iX xDrive40 402 km/20,7 kWh 316 -21,39 % 402 207 30 Mercedes-Benz EQB 350 4matic 407 km/18,1 kWh 315 -22,60 % 407 181 31 Opel Mokka-e 338 km/16,2 kWh 263 -22,19 % 338 162 32 Peugeot e-2008 320 km/15,6 kWh 228 -28,75 % 320 156
n [56]:	3. Lag en regresjonsmodell av typen $f(x) = ax + b$ der WLTP-km er uavhengig variabel, og plott modellen mot leverandørens påståtte kjørelengde. Resultatet skal bli som i figuren under, men din kode skal inkludere origo i plottet. print(df.dtypes) temp object wltp object stopp object avvik object
n [57]: n [58]:	<pre>kjørte_km object energibruk object dtype: object df['stopp'] = df['stopp'].astype('int') df['kjørte_km'] = df['kjørte_km'].astype('int') #Endrer fra objekt til innt from matplotlib import pyplot as plt</pre>
4.[50]	<pre>fig, ax=plt.subplots() #Setter navn på x- og y-akse ax.set_ylabel('STOPP') ax.set_xlabel('WLT-km') #plotter inn funksjonen ax.scatter(df['kjorte_km'], df['stopp'], label='Observasjoner') ax.legend(loc='lower right', frameon=False) <matplotlib.legend.legend 0x7f8218358b20="" at=""></matplotlib.legend.legend></pre>
	500 -
	200 - 100 -
n [59]: ut[59]:	0 - Observasjoner 350 400 450 500 550 600 650 WLT-km y=df['stopp'] pd.DataFrame(y) stopp
	 0 521 1 513 2 503 3 451 4 435 5 429 6 425
	 7 423 8 414 9 408 10 406 11 403 12 402
	13 400 14 392 15 383 16 380 17 369 18 0 19 356
	 20 353 21 349 22 347 23 346 24 340 25 0 26 339
	 27 333 28 331 29 316 30 315 31 263 32 228
n [60]: ut[60]:	<pre>import matplotlib.pyplot as plt x=pd.DataFrame(np.log(df['kjørte_km'])) x['intercept']=1 x</pre>
	2 6.381816 1 3 6.228511 1 4 6.289716 1 5 6.269096 1 6 6.190315 1 7 6.182085 1 8 6.184149 1
	9 6.175867 1 10 6.208590 1 11 6.167516 1 12 6.122493 1 13 6.248043 1 14 6.137727 1
	15 6.152733 1 16 6.184149 1 17 6.131226 1 18 6.063785 1 20 6.144186 1 21 6.129050 1 22 6.323448 1
	22 6.232448 1 23 6.104793 1 24 6.165418 1 25 6.152733 1 26 5.978886 1 27 6.079933 1 28 5.993961 1
n [61]:	29 5.996452 1 30 6.008813 1 31 5.823046 1 32 5.768321 1 from statsmodels.regression.linear_model import OLS
	res=OLS(y,x).fit() print(res.summary()) OLS Regression Results
	Time: 09:54:12 Log-Likelihood: -195.61 No. Observations: 33 AIC: 395.2 Df Residuals: 31 BIC: 398.2 Df Model: 1 Covariance Type: nonrobust
n [62]:	Omnibus: 47.490 Durbin-Watson: 2.124 Prob(Omnibus): 0.600 Jarque-Bera (JB): 189.142 Skew: -3.272 Prob(JB): 8.48e-42 Kurtosis: 12.734 Cond. No. 269
ut[62]: n [63]:	kjørte_km 439.451172 intercept -2340.527283 dtype: float64 import seaborn as sns fig, ax = plt.subplots() x = np.linspace(0,750,100) a = x
	<pre>ax.scatter(df['kjørte_km'], df['stopp'], label='Observasjoner') ax.plot(x, a, color='green', label='Forventet prestasjon') ax.set_ylabel('STOPP') ax.set_xlabel('WLT-km') ax.spines['top'].set_color('none') ax.spines['right'].set_color('none') plt.ylim(-30, 700)</pre>
	plt.ylim(-30, 700) plt.title('Rekkeviddetest') ax.legend(); Rekkeviddetest Observasjoner Forventet prestasjon
	600 - Forventet prestasjon - 400 - 400 - 60 - 60 - 60 - 60 - 60 -
	200-100-100-100-100-100-100-100-100-100-
	<pre>fig, ax = plt.subplots()</pre>
n [85]:	<pre>x = np.linspace(0,700,100) a = x #plotter inn funksjon ax.scatter(df['kjørte_km'], df['stopp'], label='Observasjoner') ax.plot(x, a, color='green', label='Forventet prestasjon') sns.regplot(x='kjørte_km', y='stopp', data=df, truncate=False, scatter=False, color='red', label='Faktisk prestasjon')</pre>
n [85]:	<pre>a = x #plotter inn funksjon ax.scatter(df['kjørte_km'], df['stopp'], label='Observasjoner') ax.plot(x, a, color='green', label='Forventet prestasjon')</pre>

ARBEIDSKRAV 5