In [235…

```python
#pip install opencv-python
from tabula import read_pdf
from tabulate import tabulate
import pandas as pd
import requests
from bs4 import BeautifulSoup
from io import StringIO
import re
```

I will be retriving Earnings data and cleaning it and making it presentable for Humans

As you see below, The data is really messy

In [36]:

```python
import tabula
pdf_path = "tsla2.pdf"
pdf = tabula.read_pdf(pdf_path, pages=7)
pdf[0].head(10)
```

Out[36]:

	O P E R A T I O N A L S U M M A R Y\r(Unaudited)\rQ4-2022Q1-2023Q2-2023Q3-2023Q4-2023YoY\rModel 3/Y production419,088421,371460,211416,800476,777\r14%\rOther models production20,61319,43719,48913,68818,212\r-12%\rTotal production439,701440,808479,700430,488494,989\r13%\rModel 3/Y deliveries388,131412,180446,915419,074461,538\r19%\rOther models deliveries17,14710,69519,22515,98522,969\r34%\rTotal deliveries405,278422,875466,140435,059484,507\r20%\rof which subject to operating lease accounting15,18422,35721,88317,42310,563-30%\rTotal end of quarter operating lease vehicle count140,667153,988168,058176,231176,56426%\rGlobal vehicle inventory (days of supply)(1)\r131516161515%\rSolar deployed (MW)10067664941-59%\rStorage deployed (MWh)2,4623,8893,6533,9803,20230%\rTesla locations9631,0001,0681,1291,20825%\rMobile service fleet1,5841,6921,7691,8461,90921%\rSupercharger stations4,6784,9475,2655,5955,95227%\rSupercharger connectors42,41945,16948,08251,10554,89229%\r(1)Days of supply is calculated by dividing new vehicle ending inventory by the relevant quarter's deliveries and using 75 trading days (aligned with Automotive News definition).\r7	Unnamed: 0	Unnamed: 1	Unnamed: 2	Unna
0	NaN	Q4-2022	Q1-2023	Q2-2023	Q3
1	Model 3/Y production	419,088	421,371	460,211	41
2	Other models production	20,613	19,437	19,489	1
3	Total production	439,701	440,808	479,700	43
4	NaN	NaN	NaN	NaN	
5	Model 3/Y deliveries	388,131	412,180	446,915	41
6	Other models deliveries	17,147	10,695	19,225	1
7	Total deliveries	405,278	422,875	466,140	43
8	of which subject to operating lease accounting	15,184	22,357	21,883	1
9	NaN	NaN	NaN	NaN	

In [ ]:

Notice how the below data is now presentable to humans after automating the cleaning process

In [23]:

```python
import tabula
pdf_path = "tsla2.pdf"
pdf = tabula.read_pdf(pdf_path, pages=7)
pdf[0].columns = pdf[0].iloc[0]
pdf = pdf[0].drop(0)
pdf = pdf.reset_index(drop=True)
pdf = pdf.fillna(" ")
pdf
```

Out[23]:

	NaN	Q4-2022	Q1-2023	Q2-2023	Q3-2023	Q4-2023	YoY
0	Model 3/Y production	419,088	421,371	460,211	416,800	476,777	14%
1	Other models production	20,613	19,437	19,489	13,688	18,212	-12%
2	Total production	439,701	440,808	479,700	430,488	494,989	13%
3							
4	Model 3/Y deliveries	388,131	412,180	446,915	419,074	461,538	19%
5	Other models deliveries	17,147	10,695	19,225	15,985	22,969	34%
6	Total deliveries	405,278	422,875	466,140	435,059	484,507	20%
7	of which subject to operating lease accounting	15,184	22,357	21,883	17,423	10,563	-30%
8							
9	Total end of quarter operating lease vehicle c...	140,667	153,988	168,058	176,231	176,564	26%
10	Global vehicle inventory (days of supply)(1)	13	15	16	16	15	15%
11							
12	Solar deployed (MW)	100	67	66	49	41	-59%
13	Storage deployed (MWh)	2,462	3,889	3,653	3,980	3,202	30%
14							
15	Tesla locations	963	1,000	1,068	1,129	1,208	25%
16	Mobile service fleet	1,584	1,692	1,769	1,846	1,909	21%
17							
18	Supercharger stations	4,678	4,947	5,265	5,595	5,952	27%
19	Supercharger connectors	42,419	45,169	48,082	51,105	54,892	29%

I made a simple algorithm to automate the cleaning process

In [34]:

```python
def find_table( filelocation  ,num):
    pdf = tabula.read_pdf(filelocation, pages=num)
    for data in pdf:
        data = data.fillna('')
        def clean_percentage(value):
            match = re.match(r'([-+]?\d*\.\d+|\d+)%', str(value))
            return float(match.group(1)) if match else None
        cleaned_data = []
        for row in data.values:
            cleaned_row = [re.sub(r'[\r\n]+', '', str(cell)) for cell in row]
            cleaned_data.append(cleaned_row)
        # Create a DataFrame
        df = pd.DataFrame(cleaned_data[1:], columns=cleaned_data[0])
        df = df.fillna('')
        df1 = df.copy()
    return df1
filelocation = re.sub(r'\\', '/', r"C:\Users\faraz\Desktop\kaggle\Earnings PDF table\tsla2.pdf")
find_table(filelocation, 5)
```

Out[34]:

	($ in millions, except percentages and per share data)	2019	2020	2021	2022	2023	YoY
0	Total automotive revenues	20,821	27,236	47,232	71,462	82,419	15%
1	Energy generation and storage revenue	1,531	1,994	2,789	3,909	6,035	54%
2	Services and other revenue	2,226	2,306	3,802	6,091	8,319	37%
3							
4	Total revenues	24,578	31,536	53,823	81,462	96,773	19%
5	Total gross profit	4,069	6,630	13,606	20,853	17,660	-15%
6	Total GAAP gross margin	16.6%	21.0%	25.3%	25.6%	18.2%	-735 bp
7							
8	Operating expenses	4,138	4,636	7,083	7,197	8,769	22%
9	(Loss) income from operations	(69)	1,994	6,523	13,656	8,891	-35%
10	Operating margin	-0.3%	6.3%	12.1%	16.8%	9.2%	-758 bp
11							
12	Adjusted EBITDA	2,985	5,817	11,621	19,186	16,631	-13%
13	Adjusted EBITDA margin	12.1%	18.4%	21.6%	23.6%	17.2%	-637 bp
14							
15	Net (loss) income attributable to common stock...	(862)	721	5,519	12,556	14,997	19%
16	Net income attributable to common stockholders...	36	2,455	7,640	14,116	10,882	-23%
17							
18	EPS attributable to common stockholders, dilut...	(0.33)	0.21	1.63	3.62	4.30	19%
19	EPS attributable to common stockholders, dilut...	0.01	0.75	2.26	4.07	3.12	-23%
20							
21	Net cash provided by operating activities	2,405	5,943	11,497	14,724	13,256	-10%
22	Capital expenditures	(1,327)	(3,157)	(6,482)	(7,158)	(8,898)	24%
23	Free cash flow	1,078	2,786	5,015	7,566	4,358	-42%
24	Cash, cash equivalents and investments	6,268	19,384	17,707	22,185	29,094	31%

This code prints saves all of the tables as Excel file for later analytics

In [ ]:
```
import tabula
import pandas as pd
pdf_path = 'tsla2.pdf'
tables = tabula.read_pdf(pdf_path, pages='all')

for i, table in enumerate(tables):
    table.fillna(" ")
    cleaned_table = table.applymap(lambda x: str(x).replace('\r', '').strip())
    csv_filename = f'table_{i + 1}.csv'
    cleaned_table.to_csv(csv_filename, index=False)
    for j in range(5)
        print(f"Table {i + 1} cleaned and saved as {csv_filename}")
```

Here I am trying to clean a really really messy table for analytics

In [39]:
```
import PyPDF2
with open('tsla2.pdf', 'rb') as file:
    pdf_reader = PyPDF2.PdfReader(file)
    page_num = 25
    page = pdf_reader.pages[page_num - 1]  # Adjusting for 0-based indexing
    text = page.extract_text()
    text
lines = text.strip().split('\n')
lines
```

```
Out[39]: ['In millions of USD or shares as applicable, except per share data Q4-2022 Q1-2023 Q2-2023 Q3-2
023 Q4-2023',
 'REVENUES',
 'Automotive sales 20,241 18,878 20,419 18,582 20,630 ',
 'Automotive regulatory credits 467 521 282 554 433 ',
 'Automotive leasing 599 564 567 489 500 ',
 'Total automotive revenues 21,307 19,963 21,268 19,625 21,563 ',
 'Energy generation and storage 1,310 1,529 1,509 1,559 1,438 ',
 'Services and other 1,701 1,837 2,150 2,166 2,166 ',
 'Total revenues 24,318 23,329 24,927 23,350 25,167  ',
 'COST OF REVENUES',
 'Automotive sales 15,433 15,422 16,841 15,656 17,202 ',
 'Automotive leasing 352 333 338 301 296 ',
 'Total automotive cost of revenues 15,785 15,755 17,179 15,957 17,498 ',
 'Energy generation and storage 1,151 1,361 1,231 1,178 1,124 ',
 'Services and other 1,605 1,702 1,984 2,037 2,107 ',
 'Total cost of revenues 18,541 18,818 20,394 19,172 20,729 ',
 'Gross profit 5,777 4,511 4,533 4,178 4,438  ',
 'OPERATING EXPENSES',
 'Research and development 810 771 943 1,161 1,094 ',
 'Selling, general and administrative 1,032 1,076 1,191 1,253 1,280 ',
 'Restructuring and other 34 —   —   — —   ',
 'Total operating expenses 1,876 1,847 2,134 2,414 2,374 ',
 'INCOME FROM OPERATIONS 3,901 2,664 2,399 1,764 2,064  ',
 'Interest income 157 213 238 282 333 ',
 'Interest expense (33) (29) (28) (38) (61)',
 'Other (expense) income, net (42) (48) 328 37 (145)',
 'INCOME BEFORE INCOME TAXES 3,983 2,800 2,937 2,045 2,191 ',
 'Provision for (benefit from) income taxes 276 261 323 167 (5,752)',
 'NET INCOME 3,707 2,539 2,614 1,878 7,943 ',
 'Net income (loss) attributable to noncontrolling interests and redeemable noncontrolling inter
ests in subsidiaries 20 26 (89) 25 15 ',
 'NET INCOME ATTRIBUTABLE TO COMMON STOCKHOLDERS 3,687 2,513 2,703 1,853 7,928 ',
 'Net income per share of common stock attributable to common stockholders',
 'Basic $        1.18 $        0.80 $        0.85 $        0.58 $        2.49 ',
 'Diluted $        1.07 $        0.73 $        0.78 $        0.53 $        2.27 ',
 'Weighted average shares used in computing net income per share of common stock',
 'Basic 3,160 3,166 3,171 3,176 3,181',
 'Diluted 3,471 3,468 3,478 3,493 3,492',
 'S T A T E M E N T   O F   O P E R A T I O N S',
 '(Unaudited)',
 '25']
```

```
In [40]: import numpy as np
columns = ['Word',  'num1', 'num2', 'num3','num4','num5']
df = pd.DataFrame(columns=columns)

results = []
pattern = re.compile(r'[[a-zA-Z]{2,}')
npattern = re.compile(r'\S*\d\S*')

for line in lines:

    wordmatch = pattern.findall(line)
    result = " ".join(wordmatch)
    nummatch = npattern.findall(line)
    if len(nummatch) != 5:
        nummatch = [np.nan, np.nan, np.nan, np.nan, np.nan]
    nummatch.insert(0, result)
    new_row_index = len(df)
    df.loc[new_row_index] = nummatch

df.columns = df.iloc[0]
df = df.drop(0)
df = df.reset_index(drop=True)
df = df.fillna("")
df
```

```
C:\Users\faraz\AppData\Local\Temp\ipykernel_7464\1356744486.py:6: FutureWarning: Possible nested
set at position 1
  pattern = re.compile(r'[[a-zA-Z]{2,}')
```

Out[40]:

	In millions of USD or shares as applicable except per share data	Q4-2022	Q1-2023	Q2-2023	Q3-2023	Q4-2023
0	REVENUES					
1	Automotive sales	20,241	18,878	20,419	18,582	20,630
2	Automotive regulatory credits	467	521	282	554	433
3	Automotive leasing	599	564	567	489	500
4	Total automotive revenues	21,307	19,963	21,268	19,625	21,563
5	Energy generation and storage	1,310	1,529	1,509	1,559	1,438
6	Services and other	1,701	1,837	2,150	2,166	2,166
7	Total revenues	24,318	23,329	24,927	23,350	25,167
8	COST OF REVENUES					
9	Automotive sales	15,433	15,422	16,841	15,656	17,202
10	Automotive leasing	352	333	338	301	296
11	Total automotive cost of revenues	15,785	15,755	17,179	15,957	17,498
12	Energy generation and storage	1,151	1,361	1,231	1,178	1,124
13	Services and other	1,605	1,702	1,984	2,037	2,107
14	Total cost of revenues	18,541	18,818	20,394	19,172	20,729
15	Gross profit	5,777	4,511	4,533	4,178	4,438
16	OPERATING EXPENSES					
17	Research and development	810	771	943	1,161	1,094
18	Selling general and administrative	1,032	1,076	1,191	1,253	1,280
19	Restructuring and other					
20	Total operating expenses	1,876	1,847	2,134	2,414	2,374
21	INCOME FROM OPERATIONS	3,901	2,664	2,399	1,764	2,064
22	Interest income	157	213	238	282	333
23	Interest expense	(33)	(29)	(28)	(38)	(61)
24	Other expense income net	(42)	(48)	328	37	(145)
25	INCOME BEFORE INCOME TAXES	3,983	2,800	2,937	2,045	2,191
26	Provision for benefit from income taxes	276	261	323	167	(5,752)
27	NET INCOME	3,707	2,539	2,614	1,878	7,943
28	Net income loss attributable to noncontrolling...	20	26	(89)	25	15
29	NET INCOME ATTRIBUTABLE TO COMMON STOCKHOLDERS	3,687	2,513	2,703	1,853	7,928
30	Net income per share of common stock attributa...					
31	Basic	1.18	0.80	0.85	0.58	2.49
32	Diluted	1.07	0.73	0.78	0.53	2.27
33	Weighted average shares used in computing net ...					
34	Basic	3,160	3,166	3,171	3,176	3,181
35	Diluted	3,471	3,468	3,478	3,493	3,492
36						
37	Unaudited					
38						

Quickly running analytics on Web Data/Tables Examples used Tesla earning

```
In [24]: forthq = 'https://ir.tesla.com/press-release/tesla-vehicle-production-deliveries-and-date-financ
         response = requests.get(forthq)

         if response.status_code == 200:
             soup = BeautifulSoup(response.text, 'html.parser')
             tables = soup.find_all('table')
             dfs = []
             for i, table in enumerate(tables):
                 html_string = str(table)
                 df = pd.read_html(StringIO(html_string), header=0)[0]
                 dfs.append(df)
         dfs[0]
```

```
Out[24]:
     Unnamed: 0  Production  Deliveries  Subject to operating lease accounting
0     Model 3/Y      476777      461538                                     2%
1  Other Models       18212       22969                                     3%
2         Total      494989      484507                                     2%
```

```
In [25]: dfs[1]
```

```
Out[25]:
     Unnamed: 0  Production  Deliveries  Unnamed: 3
0     Model 3/Y     1775159     1739707         NaN
1  Other Models       70826       68874         NaN
2         Total     1845985     1808581         NaN
```