NOTE! Much of this code is notes to demonstrate Pandas. Some areas were Exercises and the Exercise text has been inserted above the Exercise.

There is a "Sync to Github option, and I have no idea how this will appear over there.

```
60*5
```

```
300
```

Load the Table of Data and display the Top 5 Rows

```
import pandas as pd
df = pd.read_csv('salaries_by_college_major.csv')\

df.head()
```

	Undergraduate Major	Starting Median Salary	Mid-Career Median Salary	Mid-Career 10th Percentile Salary	Mid-Career 90th Percentile Salary	Group
0	Accounting	46000.0	77100.0	42200.0	152000.0	Business
1	Aerospace Engineering	57700.0	101000.0	64300.0	161000.0	STEM
2	Agriculture	42600.0	71900.0	36300.0	150000.0	Business
3	Anthropology	36800.0	61500.0	33800.0	138000.0	HASS

Display Dimensions of the Table of Data

```
df.shape
```


Display Column Names

```
df.columns
```

```
Index(['Undergraduate Major', 'Starting Median Salary',
       'Mid-Career Median Salary', 'Mid-Career 10th Percentile Salary',
       'Mid-Career 90th Percentile Salary', 'Group'],
      dtype='object')
```

Display Values with "Not a Number" values

```
df.isna()
```

	Undergraduate Major	Starting Median Salary	Mid-Career Median Salary	Mid-Career 10th Percentile Salary	Mid-Career 90th Percentile Salary	Group
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False

✓ 0s completed at 4:06 PM

4	False	False	False	False	False	False
5	False	False	False	False	False	False
6	False	False	False	False	False	False
7	False	False	False	False	False	False
8	False	False	False	False	False	False
9	False	False	False	False	False	False
10	False	False	False	False	False	False
11	False	False	False	False	False	False
12	False	False	False	False	False	False
13	False	False	False	False	False	False
14	False	False	False	False	False	False
15	False	False	False	False	False	False
16	False	False	False	False	False	False
17	False	False	False	False	False	False
18	False	False	False	False	False	False
19	False	False	False	False	False	False
20	False	False	False	False	False	False
21	False	False	False	False	False	False
22	False	False	False	False	False	False
23	False	False	False	False	False	False
24	False	False	False	False	False	False
		False	False	False	False	False
		False	False	False	False	False
27	False	False	False	False	False	False
28	False	False	False	False	False	False
29	False	False	False	False	False	False
30	False	False	False	False	False	False
31	False	False	False	False	False	False
32	False	False	False	False	False	False
33	False	False	False	False	False	False
34	False	False	False	False	False	False
35	False	False	False	False	False	False
36	False	False	False	False	False	False
37	False	False	False	False	False	False
38	False	False	False	False	False	False
39	False	False	False	False	False	False
40	False	False	False	False	False	False
41	False	False	False	False	False	False

Saved successfully!

41	False	False	False	False	False	False
42	False	False	False	False	False	False
43	False	False	False	False	False	False
44	False	False	False	False	False	False
45	False	False	False	False	False	False
46	False	False	False	False	False	False
47	False	False	False	False	False	False
48	False	False	False	False	False	False
49	False	False	False	False	False	False
50	False	True	True	True	True	True

Display the last 5 values

```
df.tail()
```

	Undergraduate Major	Starting Median Salary	Mid-Career Median Salary	Mid-Career 10th Percentile Salary	Mid-Career 90th Percentile Salary	Group
46	Psychology	35900.0	60400.0	31600.0	127000.0	HASS
47	Religion	34100.0	52000.0	29700.0	96400.0	HASS
48	Sociology	36500.0	58200.0	30700.0	118000.0	HASS
49	Spanish	34000.0	53100.0	31000.0	96400.0	HASS
50	Source: PayScale	NaN	NaN	NaN	NaN	NaN

Drop the bottom row of data since it's irrelevant and store it in a new Data Set Variable

```
clean_df = df.dropna()
```

Saved successfully!

	Undergraduate Major	Starting Median Salary	Mid-Career Median Salary	Mid-Career 10th Percentile Salary	Mid-Career 90th Percentile Salary	Group
45	Political Science	40800.0	78200.0	41200.0	168000.0	HASS
46	Psychology	35900.0	60400.0	31600.0	127000.0	HASS
47	Religion	34100.0	52000.0	29700.0	96400.0	HASS
48	Sociology	36500.0	58200.0	30700.0	118000.0	HASS
49	Spanish	34000.0	53100.0	31000.0	96400.0	HASS

```
clean_df['Starting Median Salary']
```

```
0     46000.0
1     57700.0
2     42600.0
3     36800.0
4     41600.0
5     35800.0
6     38800.0
7     43000.0
8     63200.0
9     42600.0
10    53900.0
11    38100.0
```

```
12    61400.0
13    55900.0
14    53700.0
15    35000.0
16    35900.0
17    50100.0
18    34900.0
19    60900.0
20    38000.0
21    37900.0
22    47900.0
23    39100.0
24    41200.0
25    43500.0
26    35700.0
27    38800.0
28    39200.0
29    37800.0
30    57700.0
31    49100.0
32    36100.0
33    40900.0
34    35600.0
35    49200.0
36    40800.0
37    45400.0
38    57900.0
39    35900.0
40    54200.0
41    39900.0
42    39900.0
43    74300.0
44    50300.0
45    40800.0
46    35900.0
47    34100.0
48    36500.0
49    34000.0
Name: Starting Median Salary, dtype: float64
```

Saved successfully!

```
clean_df['Starting Median Salary'].max()
```

```
74300.0
```

Get the ID of the Max of a columm

```
clean_df['Starting Median Salary'].idxmax()
```

```
43
```

Both of the next two do the same thing in different ways, and displays data for a particular column in a row.

```
clean_df['Undergraduate Major'].loc[43]
```

```
'Physician Assistant'
```

```
clean_df['Undergraduate Major'][43]
```

```
'Physician Assistant'
```

OR

Display all the data on a Row

```
clean_df.loc[43]
```

```
Undergraduate Major                 Physician Assistant
Spread                                          57600.0
Starting Median Salary                          74300.0
Mid-Career Median Salary                        91700.0
Mid-Career 10th Percentile Salary               66400.0
Mid-Career 90th Percentile Salary              124000.0
Group                                              STEM
Name: 43, dtype: object
```

Exercise - What college major has the highest mid-career salary? How much do graduates with this major earn? (Mid-career is defined as having 10+ years of experience).

```
clean_df.loc[clean_df['Mid-Career Median Salary'].idxmax()]
```

```
Undergraduate Major                 Chemical Engineering
Starting Median Salary                           63200.0
Mid-Career Median Salary                        107000.0
Mid-Career 10th Percentile Salary                71900.0
Mid-Career 90th Percentile Salary               194000.0
Group                                              STEM
Name: 8, dtype: object
```

Exercise - Which college major has the lowest starting salary and how much do graduates earn after university?

```
clean_df.loc[clean_df['Starting Median Salary'].idxmin()]
```

Saved successfully!

```
                                    Spanish
                                    34000.0
                                    53100.0
Mid-Career 10th Percentile Salary   31000.0
Mid-Career 90th Percentile Salary   96400.0
Group                                  HASS
Name: 49, dtype: object
```

Exercise - Which college major has the lowest mid-career salary and how much can people expect to earn with this degree?

```
clean_df.loc[clean_df['Mid-Career Median Salary'].idxmin()]
```

```
Undergraduate Major                 Education
Starting Median Salary                34900.0
Mid-Career Median Salary              52000.0
Mid-Career 10th Percentile Salary     29300.0
Mid-Career 90th Percentile Salary    102000.0
Group                                   HASS
Name: 18, dtype: object
```

Two Different ways to subtracke one column from another.

```
clean_df['Mid-Career 90th Percentile Salary'] - clean_df['Mid-Career 10th Percentile Salary']
```

```
0     109800.0
1      96700.0
2     113700.0
3     104200.0
4      85400.0
5      96200.0
6      98100.0
7     108200.0
8     122100.0
9     102700.0
10     84600.0
11    105500.0
12     95900.0
13     98000.0
14    114700.0
15     74800.0
16    116300.0
17    159400.0
18     72700.0
19     98700.0
20     99600.0
21    102100.0
22    147800.0
23     70000.0
24     92000.0
25    111000.0
26     76000.0
27     66400.0
28    112000.0
29     88500.0
30    115900.0
31     84500.0
32     71300.0
33    118800.0
34    106600.0
35    100700.0
36    132900.0
37    137800.0
38     99300.0
```

Saved successfully!

```
42    132500.0
43     57600.0
44    122000.0
45    126800.0
46     95400.0
47     66700.0
48     87300.0
49     65400.0
dtype: float64
```

```
clean_df['Mid-Career 90th Percentile Salary'].subtract(clean_df['Mid-Career 10th Percentile Salary'])
```

```
0     109800.0
1      96700.0
2     113700.0
3     104200.0
4      85400.0
5      96200.0
6      98100.0
7     108200.0
8     122100.0
9     102700.0
10     84600.0
11    105500.0
12     95900.0
13     98000.0
```

```
14    114700.0
15     74800.0
16    116300.0
17    159400.0
18     72700.0
19     98700.0
20     99600.0
21    102100.0
22    147800.0
23     70000.0
24     92000.0
25    111000.0
26     76000.0
27     66400.0
28    112000.0
29     88500.0
30    115900.0
31     84500.0
32     71300.0
33    118800.0
34    106600.0
35    100700.0
36    132900.0
37    137800.0
38     99300.0
39    107300.0
40     50700.0
41     65300.0
42    132500.0
43     57600.0
44    122000.0
45    126800.0
46     95400.0
47     66700.0
48     87300.0
49     65400.0
dtype: float64
```

Insert the Subtraction Value as a column in the data (Called Spread)

Saved successfully!

```
spread_col = clean_df['Mid-Career 90th Percentile Salary'] - clean_df['Mid-Career 10th Percentile Salary']
clean_df.insert(1, 'Spread', spread_col)
clean_df.head()
```

	Undergraduate Major	Spread	Starting Median Salary	Mid-Career Median Salary	Mid-Career 10th Percentile Salary	Mid-Career 90th Percentile Salary	Group
0	Accounting	109800.0	46000.0	77100.0	42200.0	152000.0	Business
1	Aerospace Engineering	96700.0	57700.0	101000.0	64300.0	161000.0	STEM
2	Agriculture	113700.0	42600.0	71900.0	36300.0	150000.0	Business
3	Anthropology	104200.0	36800.0	61500.0	33800.0	138000.0	HASS

Get the Majors with the Lowest risk (Lowest difference between 10th Percentile and 90th Percentile) and sort them to show the lowest risk.

```
····low_risk·=·clean_df.sort_values('Spread')
····low_risk[['Undergraduate·Major',·'Spread']].head()
```

	Undergraduate Major	Spread

40	Nursing	50700.0
43	Physician Assistant	57600.0
41	Nutrition	65300.0
49	Spanish	65400.0
27	Health Care Administration	66400.0

Exercise - Find the highest potential (Highest 90th Percentile)

```
high_potential = clean_df.sort_values('Mid-Career 90th Percentile Salary', ascending=False)
high_potential[['Undergraduate Major', 'Mid-Career 90th Percentile Salary']].head()
```

	Undergraduate Major	Mid-Career 90th Percentile Salary
17	Economics	210000.0
22	Finance	195000.0
8	Chemical Engineering	194000.0
37	Math	183000.0
44	Physics	178000.0

Exercise - Find the Highest Risk (Largest Spread)

```
high_risk = clean_df.sort_values('Spread', ascending=False)
high_risk[['Undergraduate Major', 'Spread']].head()
```

	Undergraduate Major	Spread
17	Economics	159400.0
		0.0
37	Math	137800.0
36	Marketing	132900.0
42	Philosophy	132500.0

Saved successfully!

Values can be grouped by catwegories (in columns), This Table has STEM, HASS, and Business .

```
clean_df.groupby('Group').count()
```

Group	Undergraduate Major	Spread	Starting Median Salary	Mid-Career Median Salary	Mid-Career 10th Percentile Salary	Mid-Career 90th Percentile Salary
Business	12	12	12	12	12	12
HASS	22	22	22	22	22	22
STEM	16	16	16	16	16	16

Exercise - Find the Mean Salary of each Group

```
clean_df.groupby('Group').mean()
```

```
clean_df.groupby('Group').mean()
```

Group	Spread	Starting Median Salary	Mid-Career Median Salary	Mid-Career 10th Percentile Salary	Mid-Career 90th Percentile Salary
Business	103958.333333	44633.333333	75083.333333	43566.666667	147525.000000
HASS	95218.181818	37186.363636	62968.181818	34145.454545	129363.636364
STEM	101600.000000	53862.500000	90812.500000	56025.000000	157625.000000

The numbers can also be formatted to be more useful.

```
pd.options.display.float_format = '{:,.2f}'.format
clean_df.groupby('Group').mean()
```

Group	Spread	Starting Median Salary	Mid-Career Median Salary	Mid-Career 10th Percentile Salary	Mid-Career 90th Percentile Salary
Business	103,958.33	44,633.33	75,083.33	43,566.67	147,525.00
HASS	95,218.18	37,186.36	62,968.18	34,145.45	129,363.64
STEM	101,600.00	53,862.50	90,812.50	56,025.00	157,625.00

NOTE! I did not write this code, it came from someone in a comment, but I am making a copy of it here as a note because well, that's how useful programming works. Code is code, knowing how to use more code is better and best.

ψ

Fresh Data for Scraping AT https://www.payscale.com/college-salary-report/majors-that-pay-you-back/bachelors

Fresh Data for Scraping AT https://www.payscale.com/college-salary-report/majors-that-pay-you-back/bachelors

Saved successfully!

```
# Main dataframe to collect all data
table_from_html = pd.read_html("https://www.payscale.com/college-salary-report/majors-that-pay-you-back/bachel
df = table_from_html[0].copy()
df.columns = ["Rank", "Major", "Type", "EarlyCareerPay", "MidCareerPay", "HighMeaning"]

# Add tables from other pages to main dataframe
for page_no in range(2, 35):
    table_from_html = pd.read_html(f"https://www.payscale.com/college-salary-report/majors-that-pay-you-back/b
    page_df = table_from_html[0].copy()
    page_df.columns = ["Rank", "Major", "Type", "EarlyCareerPay", "MidCareerPay", "HighMeaning"]
    df = df.append(page_df, ignore_index=True)

# Select necessary columns only
df = df[["Major", "EarlyCareerPay", "MidCareerPay"]]

# Clean columns
df.replace({"^Major:": "", "^Early Career Pay:\$": "", "^Mid-Career Pay:\$": "", ",": ""}, regex=True, inplace

# Change datatype of numeric columns
df[["EarlyCareerPay", "MidCareerPay"]] = df[["EarlyCareerPay", "MidCareerPay"]].apply(pd.to_numeric)
```

```
df.head()
```

	Major	EarlyCareerPay	MidCareerPay

0	Petroleum Engineering	93200	187300
1	Operations Research & Industrial Engineering	84800	170400
2	Electrical Engineering & Computer Science (EECS)	108500	159300
3	Interaction Design	68300	155800
4	Public Accounting	59800	147700

```
lowest_2021 = df.sort_values('MidCareerPay', ascending=True)
lowest_2021[['Major', 'MidCareerPay']].head()
```

	Major	MidCareerPay
826	Metalsmithing	40300
825	Medical Assisting	44800
824	Mental Health	45000
823	Early Childhood Education	45400
822	Outdoor Education	46300

Saved successfully!