Reading csv files with pandas

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

This will show us the first 5 rows of our dataframe

```
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

To see the number of rows and columns we can use the shape attribute

```
df.shape
```

```
(51, 6)
```

We can access the column names directly with the columns attribute

```
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')
```

We can look for NaN (Not A Number) values in our dataframe. NAN values are blank cells or cells that contain strings instead of numbers. With the use of the .isna() method you can spot if there's a problem somewhere.

```
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
3	False	False	False	False	False	False
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
25	False	False	False	False	False	False
26	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

Check the last couple of rows in the dataframe with the command tail

```
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:	NaN	NaN	NaN	NaN	NaN

Use the .dropna() method from pandas to delete the rows containing NaN values.

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

	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

To access a particular column from a data frame we can use the square bracket notation

```
clean_df["Starting Median Salary"].head()
```

```
0    46000.0
1    57700.0
2    42600.0
3    36800.0
4    41600.0
Name: Starting Median Salary, dtype: float64
```

To find the highest starting salary we can use .max() method.

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

```
74300.0
```

The .idxmax() method will give us index for the row with the largest value.

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

```
43
```

To get a particular row, we can use the .loc (location) property.

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

```
'Physician Assistant'
```

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

```
Undergraduate Major                  Physician Assistant
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
```

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
```

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()]
```

```
Undergraduate Major                  Spanish
Starting Median Salary               34000.0
Mid-Career Median Salary             53100.0
Mid-Career 10th Percentile Salary    31000.0
Mid-Career 90th Percentile Salary    96400.0
```

```
Group                              HASS
Name: 49, dtype: object
```

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
```

Pandas allows us to do simple arithmetic with entire columns

```
clean_df["Mid-Career 90th Percentile Salary"] - clean_df["Mid-Career 10th Percentile Salary"]
# alternatively
spread_col = clean_df["Mid-Career 90th Percentile Salary"].subtract(clean_df["Mid-Career 10th Percentile Salary"])
```

Inserting a new column

```
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

Sorting values is done by sort_values() methode

```
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

Top 5 degrees with the highest values in the 90th percentile

```
clean_df.sort_values("Mid-Career 90th Percentile Salary", ascending=False).head()
```

	Undergraduate Major	Spread	Starting Median Salary	Mid-Career Median Salary	Mid-Career 10th Percentile Salary	Mid-Career 90th Percentile Salary	Group
17	Economics	159400.0	50100.0	98600.0	50600.0	210000.0	Business
22	Finance	147800.0	47900.0	88300.0	47200.0	195000.0	Business
8	Chemical Engineering	122100.0	63200.0	107000.0	71900.0	194000.0	STEM
37	Math	137800.0	45400.0	92400.0	45200.0	183000.0	STEM

degrees with the greatest spread in salaries

```
clean_df.sort_values("Spread", ascending=False).head()
```

	Undergraduate Major	Spread	Starting Median Salary	Mid-Career Median Salary	Mid-Career 10th Percentile Salary	Mid-Career 90th Percentile Salary	Group
17	Economics	159400.0	50100.0	98600.0	50600.0	210000.0	Business
22	Finance	147800.0	47900.0	88300.0	47200.0	195000.0	Business
37	Math	137800.0	45400.0	92400.0	45200.0	183000.0	STEM
36	Marketing	132900.0	40800.0	79600.0	42100.0	175000.0	Business

count how many majors we have in each category

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

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

the average salary by group

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

```
C:\Users\AS-Computer\AppData\Local\Temp\ipykernel_13476\3608719275.py:1: FutureWarning: The default val
  clean_df.groupby("Group").mean()
```

	Spread	Starting Median Salary	Mid-Career Median Salary	Mid-Career 10th Percentile Salary	Mid-Career 90th Percentile Salary
Group					
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

Number formats in the Output

```
pd.options.display.float_format = '{:,.2f}'.format
```

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

```
C:\Users\AS-Computer\AppData\Local\Temp\ipykernel_13476\3608719275.py:1: FutureWarning: The default val
  clean_df.groupby("Group").mean()
```

	Spread	Starting Median Salary	Mid-Career Median Salary	Mid-Career 10th Percentile Salary	Mid-Career 90th Percentile Salary
Group					
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