Open in Colab

Homework 2: Pandas Introduction

Complete all exercises below. Show your work and include comments explaining your approach.

```
! curl -LsSf https://astral.sh/uv/install.sh | sh && \
  uv pip install -q --system "s26-06642 @ git+https://github.com/jkitchin/s26-06642.git"
from pycse.colab import pdf
```

```
downloading uv 0.9.26 x86_64-unknown-linux-gnu
no checksums to verify
installing to /usr/local/bin
  uv
  uvx
everything's installed!
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Problem 1: DataFrame Basics

Load and explore experiment data from a catalytic reactor study.

```
# Load experiment dataset from URL
url = "https://raw.githubusercontent.com/jkitchin/s26-06642/main/dsmles/data/hw02_experiments.csv"
experiments = pd.read_csv(url)

print(f"Loaded {len(experiments)} experiments")
experiments.head(10)
```

```
Loaded 50 experiments
```

	exp_id	temperature	pressure	catalyst	conversion	selectivity
0	EXP001	400	8.1	Ni	0.567	0.895
1	EXP002	450	2.8	Ni	0.791	0.921
2	EXP003	300	5.6	Pt	0.449	0.769
3	EXP004	400	6.3	Ni	0.350	0.911
4	EXP005	400	1.4	Pt	0.488	0.807
5	EXP006	450	6.5	Pd	NaN	0.883
6	EXP007	300	2.5	Ni	0.904	0.884
7	EXP008	300	1.6	Pd	0.825	0.855
8	EXP009	400	9.5	Pt	0.712	0.726
9	EXP010	350	9.7	Ni	0.866	0.942

Next steps: Generate code with `experiments` New interactive sheet

1a. How many experiments are missing conversion data? Which experiment IDs are missing this data?

```
# Your code here
print(f"There are {experiments.isna().sum().sum()} missing conversion data")
print(experiments[experiments.isna().any(axis=1)])
```

```
There are 3 missing conversion data
    exp_id  temperature  pressure catalyst  conversion  selectivity
5   EXP006          450       6.5       Pd         NaN        0.883
15  EXP016          300       2.1       Pd         NaN        0.705
25  EXP026          350       9.7       Pd         NaN        0.733
```

1b. Use `.describe()` to get summary statistics for the numeric columns. What is the median pressure?

```
# Your code here
print(experiments.describe())
print(f"The median is {experiments.describe()["pressure"]["50%"]}")
```

```
       temperature   pressure  conversion  selectivity
count    50.000000  50.000000   47.000000    50.000000
mean    385.000000   5.362000    0.618043     0.846500
std      56.469244   2.867046    0.195655     0.084832
min     300.000000   1.000000    0.305000     0.703000
25%     350.000000   2.800000    0.460500     0.769250
50%     400.000000   5.550000    0.578000     0.866000
75%     450.000000   8.075000    0.823500     0.911000
max     450.000000   9.900000    0.941000     0.972000
The median is 5.55
```

1c. Filter the DataFrame to show only experiments using Pt catalyst with temperature >= 400 K.

```
# Your code here
experiments[experiments.catalyst=="Pt"][experiments.temperature>=400]
```

```
/tmp/ipython-input-1701206427.py:2: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  experiments[experiments.catalyst=="Pt"][experiments.temperature>=400]
```

	exp_id	temperature	pressure	catalyst	conversion	selectivity
4	EXP005	400	1.4	Pt	0.488	0.807
8	EXP009	400	9.5	Pt	0.712	0.726
10	EXP011	400	8.3	Pt	0.822	0.793
12	EXP013	400	1.9	Pt	0.880	0.712
16	EXP017	450	5.5	Pt	0.507	0.849
23	EXP024	450	5.9	Pt	0.632	0.740
28	EXP029	450	9.1	Pt	0.913	0.775
40	EXP041	400	5.9	Pt	0.696	0.911
44	EXP045	450	9.9	Pt	0.890	0.886
45	EXP046	450	8.0	Pt	0.456	0.724
47	EXP048	400	1.0	Pt	0.618	0.961

Problem 2: Data Manipulation

Continue working with the experiments DataFrame.

2a. Calculate the average conversion and selectivity for each catalyst type using `.groupby()`.

```
# Your code here
experiments.groupby("catalyst")[["conversion","selectivity"]].mean()
#experiments.groupby("catalyst").agg({"conversion":"mean","selectivity":"mean"})
```

catalyst	conversion	selectivity
Ni	0.633333	0.876583
Pd	0.492875	0.809818
Pt	0.660333	0.825267

2b. Add a new column called 'yield' calculated as conversion × selectivity.

```
# Your code here

# I filled the NaN values with the mean, and have checked that this doesn't affect the result
experiments=experiments.fillna({"conversion":experiments.conversion.mean()})
experiments["yield"]=experiments.conversion*experiments.selectivity
experiments.head(10)
```

	exp_id	temperature	pressure	catalyst	conversion	selectivity	yield
0	EXP001	400	8.1	Ni	0.567000	0.895	0.507465
1	EXP002	450	2.8	Ni	0.791000	0.921	0.728511
2	EXP003	300	5.6	Pt	0.449000	0.769	0.345281
3	EXP004	400	6.3	Ni	0.350000	0.911	0.318850
4	EXP005	400	1.4	Pt	0.488000	0.807	0.393816
5	EXP006	450	6.5	Pd	0.618043	0.883	0.545732
6	EXP007	300	2.5	Ni	0.904000	0.884	0.799136
7	EXP008	300	1.6	Pd	0.825000	0.855	0.705375
8	EXP009	400	9.5	Pt	0.712000	0.726	0.516912
9	EXP010	350	9.7	Ni	0.866000	0.942	0.815772

Next steps: Generate code with experiments | New interactive sheet

2c. Find the top 5 experiments by yield. Display exp_id, catalyst, temperature, and yield.

```
# Your code here
experiments[["exp_id","catalyst","temperature","yield"]]
experiments.sort_values("yield",ascending=False).head(5)
```

	exp_id	temperature	pressure	catalyst	conversion	selectivity	yield
48	EXP049	350	8.3	Ni	0.941	0.876	0.824316
9	EXP010	350	9.7	Ni	0.866	0.942	0.815772
6	EXP007	300	2.5	Ni	0.904	0.884	0.799136
44	EXP045	450	9.9	Pt	0.890	0.886	0.788540
20	EXP021	350	7.0	Ni	0.832	0.900	0.748800

2d. What is the average yield for each combination of catalyst and temperature? Use `.groupby()` with multiple columns.

```
# Your code here
experiments.groupby(["catalyst","temperature"])["yield"].mean()
```

		yield
catalyst	temperature	
Ni	300	0.621562
	350	0.564672
	400	0.485867
	450	0.542167
Pd	300	0.509375
	350	0.453025
	400	0.288479
	450	0.429916
Pt	300	0.496076
	350	0.495126
	400	0.569515
	450	0.544876

dtype: float64

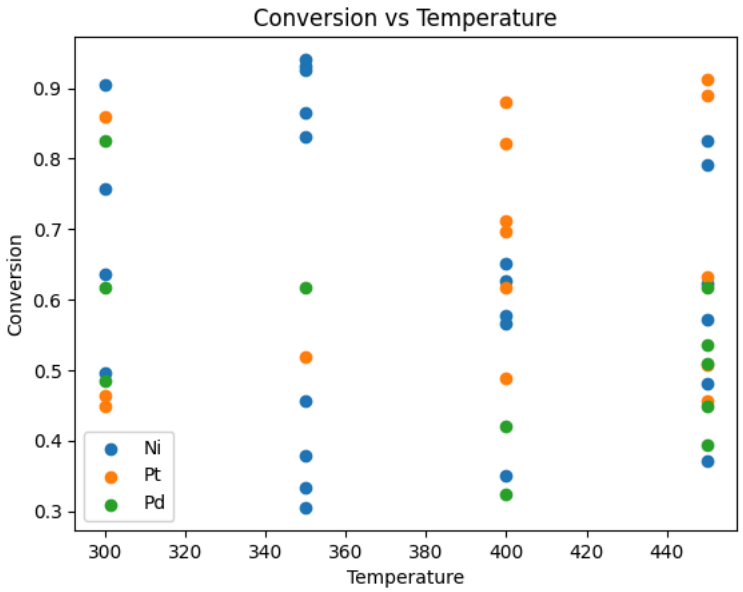
Problem 3: Visualization

Create visualizations to explore the data.

3a. Create a scatter plot of conversion vs temperature, colored by catalyst type.

Hint: You can loop through catalyst types and plot each separately with different colors, or use pandas plotting.

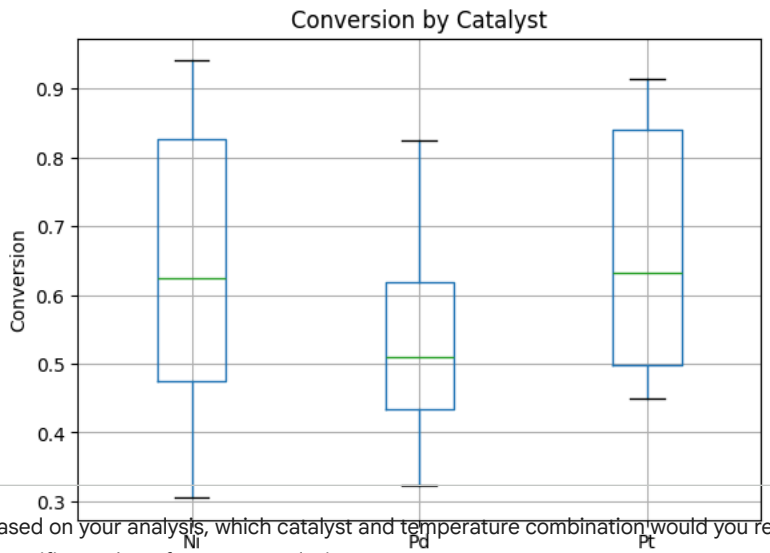
```
# Your code here

'''
catalysts= experiments["catalyst"].unique()
for c in catalysts:
  experiments[experiments.catalyst==c].plot.scatter(x="temperature",y="conversion")
'''

'''
i used AI here to figure out how to all catalysts' data in the same table
If directly plot with experiments[experiments.catalyst=c].plt
pandas will automatically create a new plot each time
but by creating and assigning the new pair data to a separate variable
panda assumes that the plotting will continue on the same plot
'''
catalysts= experiments["catalyst"].unique()
for c in catalysts:
  pair=experiments[experiments.catalyst==c]
  plt.scatter(pair["temperature"],pair["conversion"])

plt.xlabel("Temperature")
plt.ylabel("Conversion")
plt.title(f"Conversion vs Temperature")
plt.legend(catalysts)
plt.show()
```



3b. Create a box plot comparing conversion distributions across different catalysts.

Hint: Use `df.boxplot(column='conversion', by='catalyst')` *or matplotlib.*

```
# Your code here
experiments.boxplot(column='conversion', by='catalyst')
plt.ylabel('Conversion')
plt.title('Conversion by Catalyst')
plt.suptitle('')
plt.show()
```



3c. Based on your analysis, which catalyst and temperature combination would you recommend for maximizing yield? Justify your answer with specific numbers from your analysis.

Your answer here: Nickel catalyst at 350K has the highest yield as indicated in 2c where its yield is the highest (0.824316). Also shown in the scatter plot in 3a, the conversion rate is the highest with Nickel catalyst at 350K, which indirectly informs that the yield is the highest.