2025

4Geeks Academy: data science cohort 12

DAY 3: PREWORK REVIEW & INTRO TO PANDAS

TOPICS

01 TODO

02 PANDAS

03 PANDAS: ACCESSING DATA

04 PANDAS: APPLYING FUNCTIONS

TODO

PANDAS Cover basics of indexing and applying functions to dataframes

PREWORK Submit assignments (if you haven’t already) - it’s time to move on!

DATA CLEANUP Start data cleanup assignment (Pandas module) plan to finish before class Monday.

PANDAS

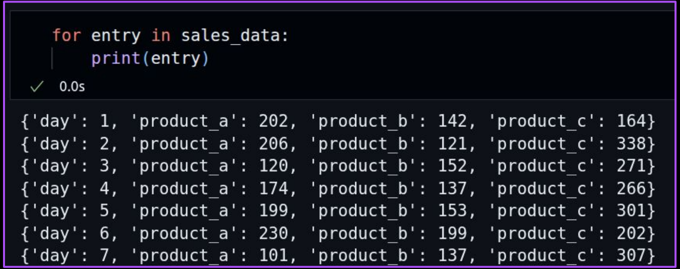
WHAT

Python package implementing structures for data analysis (DataFrame)

WHERE

Documentation: pandas.pydata.org/docs
GitHub: pandas-dev/pandas
PyPI: pandas

WHY

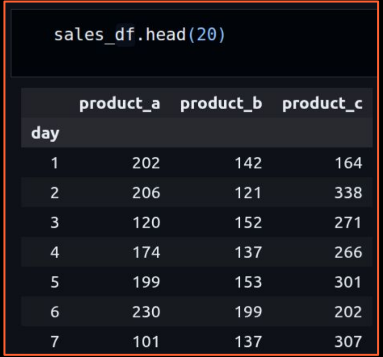


```
for entry in sales_data:
    print(entry)
```

✓ 0.0s

```
{'day': 1, 'product_a': 202, 'product_b': 142, 'product_c': 164}
{'day': 2, 'product_a': 206, 'product_b': 121, 'product_c': 338}
{'day': 3, 'product_a': 120, 'product_b': 152, 'product_c': 271}
{'day': 4, 'product_a': 174, 'product_b': 137, 'product_c': 266}
{'day': 5, 'product_a': 199, 'product_b': 153, 'product_c': 301}
{'day': 6, 'product_a': 230, 'product_b': 199, 'product_c': 202}
{'day': 7, 'product_a': 101, 'product_b': 137, 'product_c': 307}
```

VS



```
sales_df.head(20)
```

	product_a	product_b	product_c
day			
1	202	142	164
2	206	121	338
3	120	152	271
4	174	137	266
5	199	153	301
6	230	199	202
7	101	137	307

PANDAS: ACCESSING DATA

COLUM `sales_df[‘product_a’]`

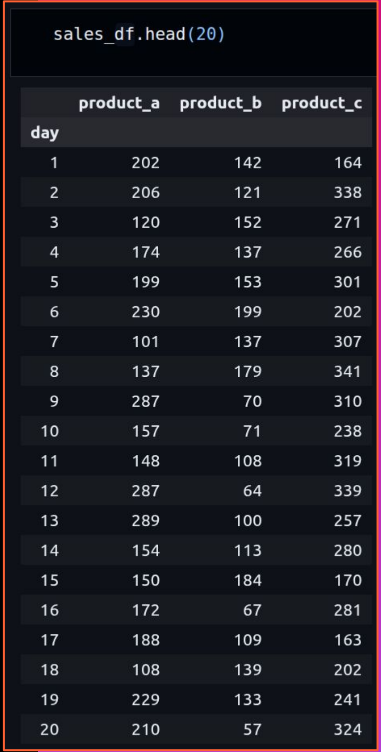
COLUMNS `sales_df[[‘product_a’, ‘product_b]]`

ROW `sales_df.loc[1]`

ROWS `sales_df.loc[1:10]`

CONDITION `sales_df[sales_df[‘product_a’] > 200]`

MORE INFO [How do I access a subset of a DataFrame?]()



```
sales_df.head(20)
```

	product_a	product_b	product_c
day			
1	202	142	164
2	206	121	338
3	120	152	271
4	174	137	266
5	199	153	301
6	230	199	202
7	101	137	307
8	137	179	341
9	287	70	310
10	157	71	238
11	148	108	319
12	287	64	339
13	289	100	257
14	154	113	280
15	150	184	170
16	172	67	281
17	188	109	163
18	108	139	202
19	229	133	241
20	210	57	324

PANDAS: APPLYING FUNCTIONS

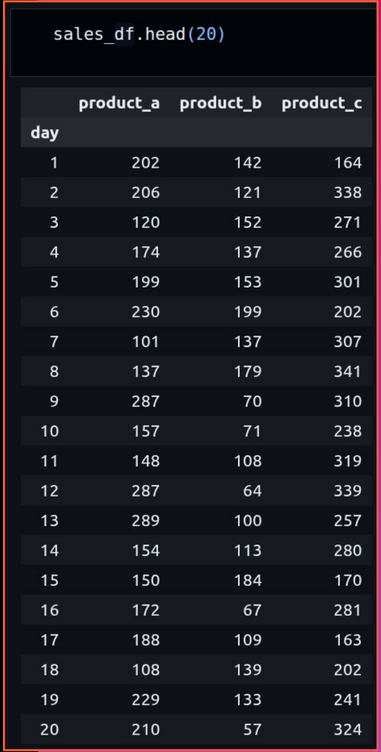
ARITHMETIC `sales_df[‘product_a’] + sales_df[‘product_b’]`

BUILTIN `sales_df[‘product_a’].sum()`

CUSTOM `sales_df[‘product_a’].apply(my_function)`

SPECIALIZED `sales_df.rolling(window=7).mean()`

MORE INFO [PyData DataFrame documentation]



```
sales_df.head(20)
```

	product_a	product_b	product_c
day			
1	202	142	164
2	206	121	338
3	120	152	271
4	174	137	266
5	199	153	301
6	230	199	202
7	101	137	307
8	137	179	341
9	287	70	310
10	157	71	238
11	148	108	319
12	287	64	339
13	289	100	257
14	154	113	280
15	150	184	170
16	172	67	281
17	188	109	163
18	108	139	202
19	229	133	241
20	210	57	324