Lab 3 – Indexing and Reindexing in pandas dataframes

```
import numpy as np
import pandas as pd
```

Question 1 — label vs position selection

1. Load nba.csv with Name as index

```
df = pd.read_csv('/content/nba - nba.csv', index_col='Name')
```

```
df
```

Name	Team	Number	Position	Age	Height	Weight	College	Salary
Avery Bradley	Boston Celtics	0	PG	25	6-2	180	Texas	7730337.0
Jae Crowder	Boston Celtics	99	SF	25	6-6	235	Marquette	6796117.0
John Holland	Boston Celtics	30	SG	27	6-5	205	Boston University	NaN
R.J. Hunter	Boston Celtics	28	SG	22	6-5	185	Georgia State	1148640.0
Jonas Jerebko	Boston Celtics	8	PF	29	6-10	231	NaN	5000000.0
...	...	...	...	...	...	...	...	...
Trey Lyles	Utah Jazz	41	PF	20	6-10	234	Kentucky	2239800.0
Shelvin Mack	Utah Jazz	8	PG	26	6-3	203	Butler	2433333.0
Raul Neto	Utah Jazz	25	PG	24	6-1	179	NaN	900000.0
Tibor Pleiss	Utah Jazz	21	C	26	7-3	256	NaN	2900000.0
Jeff Withey	Utah Jazz	24	C	26	7-0	231	Kansas	947276.0

457 rows × 8 columns

Next steps: Generate code with df | View recommended plots | New interactive sheet

2. Print the row for player "Avery Bradley" using .loc, then print the 4th row in the DataFrame using .iloc.

- df.loc['Avery Bradley'] retrieves the row by the label (player's name), returning a Series.

```
# Print the row for "Avery Bradley" using loc (label-based)
print("Row for Avery Bradley using loc:\n")
print(df.loc['Avery Bradley'])
```

```
Row for Avery Bradley using loc:

Team        Boston Celtics
Number                   0
Position                PG
Age                     25
Height                 6-2
Weight                 180
College              Texas
Salary           7730337.0
Name: Avery Bradley, dtype: object
```

3. Print the 4th row using iloc

```
# Print the 4th row using iloc (position-based, 0-indexed so 3 is the 4th row)
print("4th row using iloc:\n")
print(df.iloc[3])
```

```
4th row using iloc:

Team        Boston Celtics
Number                  28
Position                SG
Age                     22
Height                 6-5
Weight                 185
College      Georgia State
Salary           1148640.0
Name: R.J. Hunter, dtype: object
```

4. Extract "Team" and "Position" for "R.J. Hunter" using loc

- selects specific columns for that row, returning a Series or small DataFrame.

```
# Extract "Team" and "Position" for "R.J. Hunter" using loc
print("Team and Position for R.J. Hunter:\n")
print(df.loc['R.J. Hunter', ['Team', 'Position']])
```

```
Team and Position for R.J. Hunter:

Team        Boston Celtics
Position                SG
Name: R.J. Hunter, dtype: object
```

Question 2 — set_index / reset_index and at/iat

1. Read nba.csv (no index_col this time).

```
import pandas as pd
data = pd.read_csv('/content/nba - nba.csv')
data
```

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	6-2	180	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99	SF	25	6-6	235	Marquette	6796117.0
2	John Holland	Boston Celtics	30	SG	27	6-5	205	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28	SG	22	6-5	185	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8	PF	29	6-10	231	NaN	5000000.0
...	...	...	...	...	...	...	...	...	...
452	Trey Lyles	Utah Jazz	41	PF	20	6-10	234	Kentucky	2239800.0
453	Shelvin Mack	Utah Jazz	8	PG	26	6-3	203	Butler	2433333.0
454	Raul Neto	Utah Jazz	25	PG	24	6-1	179	NaN	900000.0
455	Tibor Pleiss	Utah Jazz	21	C	26	7-3	256	NaN	2900000.0
456	Jeff Withey	Utah Jazz	24	C	26	7-0	231	Kansas	947276.0

457 rows × 9 columns

Next steps: Generate code with data | View recommended plots | New interactive sheet

2.Set the DataFrame index to the "Team" column and then reset it back to default index.

- Set the DataFrame index to "Team"

```
data.set_index('Team', inplace=True)
data
```

Team	Name	Number	Position	Age	Height	Weight	College	Salary
Boston Celtics	Avery Bradley	0	PG	25	6-2	180	Texas	7730337.0
Boston Celtics	Jae Crowder	99	SF	25	6-6	235	Marquette	6796117.0
Boston Celtics	John Holland	30	SG	27	6-5	205	Boston University	NaN
Boston Celtics	R.J. Hunter	28	SG	22	6-5	185	Georgia State	1148640.0
Boston Celtics	Jonas Jerebko	8	PF	29	6-10	231	NaN	5000000.0
...	...	...	...	...	...	...	...	...
Utah Jazz	Trey Lyles	41	PF	20	6-10	234	Kentucky	2239800.0
Utah Jazz	Shelvin Mack	8	PG	26	6-3	203	Butler	2433333.0
Utah Jazz	Raul Neto	25	PG	24	6-1	179	NaN	900000.0
Utah Jazz	Tibor Pleiss	21	C	26	7-3	256	NaN	2900000.0
Utah Jazz	Jeff Withey	24	C	26	7-0	231	Kansas	947276.0

457 rows × 8 columns

Next steps: Generate code with data | View recommended plots | New interactive sheet

- reset it back to default index.

```
data.reset_index(inplace=True)
data
```

	Team	Name	Number	Position	Age	Height	Weight	College	Salary
0	Boston Celtics	Avery Bradley	0	PG	25	6-2	180	Texas	7730337.0
1	Boston Celtics	Jae Crowder	99	SF	25	6-6	235	Marquette	6796117.0
2	Boston Celtics	John Holland	30	SG	27	6-5	205	Boston University	NaN
3	Boston Celtics	R.J. Hunter	28	SG	22	6-5	185	Georgia State	1148640.0
4	Boston Celtics	Jonas Jerebko	8	PF	29	6-10	231	NaN	5000000.0
...	...	...	...	...	...	...	...	...	...
452	Utah Jazz	Trey Lyles	41	PF	20	6-10	234	Kentucky	2239800.0
453	Utah Jazz	Shelvin Mack	8	PG	26	6-3	203	Butler	2433333.0
454	Utah Jazz	Raul Neto	25	PG	24	6-1	179	NaN	900000.0
455	Utah Jazz	Tibor Pleiss	21	C	26	7-3	256	NaN	2900000.0
456	Utah Jazz	Jeff Withey	24	C	26	7-0	231	Kansas	947276.0

457 rows × 9 columns

Next steps: Generate code with data | View recommended plots | New interactive sheet

3. Using the original DataFrame (with Name as index), retrieve the Salary of "Avery Bradley" using both .at and .iat (use .at with label and .iat with integer position).

- Using .at (label-based lookup)

```
salary_at = df.at["Avery Bradley", "Salary"]
print(salary_at)
```

```
7730337.0
```

- Using .iat (position-based lookup)

```
row = df.index.get_loc("Avery Bradley")        # find row number
```

```
col = df.columns.get_loc("Salary")            # find column number
salary_iat = df.iat[row, col]
print("Salary using .iat:", salary_iat)
```

```
Salary using .iat: 7730337.0
```

- By using .get_loc

```
# Load data with Name as index
nba = pd.read_csv('/content/nba - nba.csv', index_col='Name')

# Find integer row index of "Avery Bradley"
row_pos = nba.index.get_loc("Avery Bradley")

# Find integer column index of "Salary"
col_pos = nba.columns.get_loc("Salary")

# Access salary using .iat with integer positions
salary_value = nba.iat[row_pos, col_pos]

print("Salary using .iat:", salary_value)
```

```
Salary using .iat: 7730337.0
```

Question 3 — reindex rows to add players and fill values

Application of Reindexing: Reindex to add missing labels and fill missing data.

1. Load nba.csv with index_col="Name".

```
df = pd.read_csv('/content/nba - nba.csv', index_col='Name')
df
```

Name	Team	Number	Position	Age	Height	Weight	College	Salary
Avery Bradley	Boston Celtics	0	PG	25	6-2	180	Texas	7730337.0
Jae Crowder	Boston Celtics	99	SF	25	6-6	235	Marquette	6796117.0
John Holland	Boston Celtics	30	SG	27	6-5	205	Boston University	NaN
R.J. Hunter	Boston Celtics	28	SG	22	6-5	185	Georgia State	1148640.0
Jonas Jerebko	Boston Celtics	8	PF	29	6-10	231	NaN	5000000.0
...	...	...	...	...	...	...	...	...
Trey Lyles	Utah Jazz	41	PF	20	6-10	234	Kentucky	2239800.0
Shelvin Mack	Utah Jazz	8	PG	26	6-3	203	Butler	2433333.0
Raul Neto	Utah Jazz	25	PG	24	6-1	179	NaN	900000.0
Tibor Pleiss	Utah Jazz	21	C	26	7-3	256	NaN	2900000.0
Jeff Withey	Utah Jazz	24	C	26	7-0	231	Kansas	947276.0

457 rows × 8 columns

Next steps: Generate code with df | View recommended plots | New interactive sheet

2. Create a new index list that includes the existing first 10 player names plus two new names ["New Player A","New Player B"].

```
new_index = list(df.index[:10]) + ["New Player A", "New Player B"]
display(new_index)
```

```
['Avery Bradley',
 'Jae Crowder',
 'John Holland',
 'R.J. Hunter',
 'Jonas Jerebko',
 'Amir Johnson',
 'Jordan Mickey',
 'Kelly Olynyk',
 'Terry Rozier',
 'Marcus Smart',
 'New Player A',
 'New Player B']
```

3. Reindex the DataFrame to that list and fill missing numeric fields with 0 and missing string fields with "Unknown".

```
reindexed = df.reindex(new_index)
reindexed = reindexed.fillna({
    "Team": "Unknown",
    "Number": 0,
    "Position": "Unknown",
    "Age": 0,
    "Height": "Unknown",
    "Weight": 0,
    "College": "Unknown",
    "Salary": 0
    })
```

4. Show the resulting rows for the two new players.

```
print(reindexed.loc[["New Player A", "New Player B"]])
```

Name	Team	Number	Position	Age	Height	Weight	College	Salary
New Player A	Unknown	0.0	Unknown	0.0	Unknown	0.0	Unknown	0.0
New Player B	Unknown	0.0	Unknown	0.0	Unknown	0.0	Unknown	0.0

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 457 entries, Avery Bradley to Jeff Withey
Data columns (total 8 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Team      457 non-null    object
 1   Number    457 non-null    int64
 2   Position  457 non-null    object
 3   Age       457 non-null    int64
 4   Height    457 non-null    object
 5   Weight    457 non-null    int64
 6   College   373 non-null    object
 7   Salary    446 non-null    float64
```

```
dtypes: float64(1), int64(3), object(4)
memory usage: 48.3+ KB
```

```
# another method for above question


reindexed_1 = df.reindex(new_index)
for col in df.columns:
  if reindexed_1[col].dtype == np.int64 or reindexed_1[col].dtype == np.float64
    reindexed_1[col] = reindexed_1[col].fillna(0)
  else:
    reindexed_1[col] = reindexed_1[col].fillna("Unknown")
```

```
print(reindexed_1.loc[["New Player A", "New Player B"]])
```

```
                Team  Number Position  Age   Height  Weight  College  Salary
Name
New Player A  Unknown     0.0  Unknown  0.0  Unknown     0.0  Unknown     0.0
New Player B  Unknown     0.0  Unknown  0.0  Unknown     0.0  Unknown     0.0
```

Question 4 — reorder / add columns with reindex (columns axis)

Reindex columns to reorder and introduce a new column.

```
dt = pd.read_csv('/content/nba - nba.csv', index_col='Name')
dt
```

Name	Team	Number	Position	Age	Height	Weight	College	Salary
Avery Bradley	Boston Celtics	0	PG	25	6-2	180	Texas	7730337.0
Jae Crowder	Boston Celtics	99	SF	25	6-6	235	Marquette	6796117.0
John Holland	Boston Celtics	30	SG	27	6-5	205	Boston University	NaN
R.J. Hunter	Boston Celtics	28	SG	22	6-5	185	Georgia State	1148640.0
Jonas Jerebko	Boston Celtics	8	PF	29	6-10	231	NaN	5000000.0
...	...	...	...	...	...	...	...	...
Trey Lyles	Utah Jazz	41	PF	20	6-10	234	Kentucky	2239800.0
Shelvin Mack	Utah Jazz	8	PG	26	6-3	203	Butler	2433333.0
Raul Neto	Utah Jazz	25	PG	24	6-1	179	NaN	900000.0
Tibor Pleiss	Utah Jazz	21	C	26	7-3	256	NaN	2900000.0
Jeff Withey	Utah Jazz	24	C	26	7-0	231	Kansas	947276.0

457 rows × 8 columns

Next steps: Generate code with dt | View recommended plots | New interactive sheet

1. Using nba.csv (index by Name), reindex the columns to: ["Team", "Position", "Salary", "College", "Height"].
2. Ensure the reindexed DataFrame shows the requested column order and that any missing column is created with -1.

```
dt = dt.reindex(columns=["Team", "Position", "Salary","College", "Height"], fil
dt
```

Name	Team	Position	Salary	College	Height
Avery Bradley	Boston Celtics	PG	7730337.0	Texas	6-2
Jae Crowder	Boston Celtics	SF	6796117.0	Marquette	6-6
John Holland	Boston Celtics	SG	NaN	Boston University	6-5
R.J. Hunter	Boston Celtics	SG	1148640.0	Georgia State	6-5
Jonas Jerebko	Boston Celtics	PF	5000000.0	NaN	6-10
...	...	...	...	...	...
Trey Lyles	Utah Jazz	PF	2239800.0	Kentucky	6-10
Shelvin Mack	Utah Jazz	PG	2433333.0	Butler	6-3