Combining DataFrames With Pandas - Lab

Introduction

In this lab, you'll gain practice combining DataFrames through concatenation. You'll also practice executing various types of joins to selectively combine the information stored in the tables!

Objectives

In this lab you will:

- Use concatenation to combine DataFrames
- Determine which type of join is preferred for two tables of data and a task
- Use different types of joins to merge dataframes

Concatenating DataFrames

Run the cell below to create some sample DataFrames for us to work with.

```
import pandas as pd
df1 = pd.DataFrame({'A': ['A0', 'A1', 'A2', 'A3'],
                    'B': ['B0', 'B1', 'B2', 'B3'],
                    'C': ['C0', 'C1', 'C2', 'C3'],
                    'D': ['D0', 'D1', 'D2', 'D3']},
                    index=[0, 1, 2, 3])


df2 = pd.DataFrame({'A': ['A4', 'A5', 'A6', 'A7'],
                    'B': ['B4', 'B5', 'B6', 'B7'],
                    'C': ['C4', 'C5', 'C6', 'C7'],
                    'D': ['D4', 'D5', 'D6', 'D7']},
                    index=[4, 5, 6, 7])

df3 = pd.DataFrame({'A': ['A8', 'A9', 'A10', 'A11'],
                    'B': ['B8', 'B9', 'B10', 'B11'],
                    'C': ['C8', 'C9', 'C10', 'C11'],
                    'D': ['D8', 'D9', 'D10', 'D11']},
                    index=[8, 9, 10, 11])
```

Now that you have multiple DataFrames to work with, you can execute a concatenation to join them together.

In the cell below, concatenate the 3 DataFrames together using the appropriate function.

```
combined_df = pd.concat([df1, df2, df3])
combined_df
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	A	B	C	D

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7
8	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10
11	A11	B11	C11	D11

EXPECTED OUTPUT:

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7
8	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10
11	A11	B11	C11	D11

Setting join conditions with concatenation

You can also specify if the concatenation is an ***Outer Join*** or an ***Inner Join***. Next, you'll execute an inner join. Before you do, you need to create another table that contains some overlapping index values with a DataFrame that already exists.

Run the cell below to create the new DataFrame.

```
df4 = pd.DataFrame({'B': ['B2', 'B3', 'B6', 'B7'],
                    'D': ['D2', 'D3', 'D6', 'D7'],
                    'F': ['F2', 'F3', 'F6', 'F7']},
                    index=[2, 3, 6, 7])
```

Now, in the cell below, use the `pd.concat()` function to join DataFrames 1 and 4. However, this time, specify that the `join` is `'inner'`, and `axis=1`.

```
df1_and_4 = pd.concat([df1, df4], axis=1, join='inner')
df1_and_4
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	A	B	C	D	B	D	F
2	A2	B2	C2	D2	B2	D2	F2
3	A3	B3	C3	D3	B3	D3	F3

EXPECTED OUTPUT:

	A	B	C	D	B	D	F
2	A2	B2	C2	D2	B2	D2	F2
3	A3	B3	C3	D3	B3	D3	F3

You'll notice that in this case, the results contain only the rows with indexes that exist in both tables -- rows 2 and 3. The resulting table contains the values for each column in both tables for the rows.

Note that there are many, many ways that you can make full use of the `pd.concat()` function in pandas to join DataFrames together -- these are just a few of the most common examples pulled from the pandas documentation on the subject. For a full view of all the ways you can use `pd.concat()`, see the pandas documentation!

Load data

Now, it's time to move on to working with the Hearthstone cards database. This database contains information on cards from the popular game, Hearthstone! For full information on the dataset, see the Kaggle page for this dataset.

This database consists of the following tables:

- *cards*
- *dust_costs*
- *entourages*
- *mechanics*
- *play_requirements*

Many of rows in each table -- but not all -- correspond to the same cards. As such, each table contains a column called `card_id` which acts as a ***Primary Key*** for each table. You'll make use of these keys to ***join*** the different tables together into a single DataFrame. You'll also experiment with different types of joins to help us decide exactly what information you wish to combine.

Simply run the cell below to import the tables from the database as DataFrames.

```
cards_df = pd.read_csv('cards.csv')
dust_df = pd.read_csv('dust.csv')
entourages_df = pd.read_csv('entourages.csv')
mechanics_df = pd.read_csv('mechanics.csv')
play_requirements_df = pd.read_csv('play_requirements.csv')
```

Great. Now, let's set the correct column, `card_id`, as the index column for each of these tables, and then display each to ensure that everything is as expected.

For each of the DataFrames you created in the cell above, call the `.set_index()` method and pass in `card_id`. Also set `inplace=True`. Then, display the `.head()` of each respective DataFrame to ensure everything worked.

NOTE: Since you are performing this operation in place, running any cell a second time will result in pandas throwing an error. If you need to run something a second time, restart the kernel using the jupyter notebook menu at the top of the page.

```
cards_df.set_index('card_id', inplace=True)
cards_df.head()
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	player_class	type	name	set	text
card_id					
KARA_00_07	MAGE	SPELL	Astral Portal	KARA	Summon a rando <b>Legendary</ minion.
NEW1_008a	DRUID	SPELL	Ancient Teachings	EXPERT1	Draw a card.
BRM_010t2	DRUID	MINION	Druid of the Flame	BRM	NaN
AT_132	NEUTRAL	MINION	Justicar Trueheart	TGT	<b>Battlecry:</b Replace your starting Hero P...

	player_class	type	name	set	text
card_id					
OG_141	NEUTRAL	MINION	Faceless Behemoth	OG	NaN

```
dust_df.set_index('card_id', inplace=True)
dust_df.head()
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	action	cost
card_id		
BRM_010t2	CRAFTING_NORMAL	40
BRM_010t2	CRAFTING_GOLDEN	400
BRM_010t2	DISENCHANT_NORMAL	5
BRM_010t2	DISENCHANT_GOLDEN	50
AT_132	CRAFTING_NORMAL	1600

```
entourages_df.set_index('card_id', inplace=True)
entourages_df.head()
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	entourage_card_id
card_id	
KAR_A10_22	KAR_A10_09
KAR_A10_22	KAR_A10_02
KAR_A10_22	KAR_A10_08
KAR_A10_22	KAR_A10_04
KAR_A10_22	KAR_A10_05

```
mechanics_df.set_index('card_id', inplace=True)
mechanics_df.head()
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	mechanic
card_id	
AT_132	BATTLECRY

	mechanic
card_id	
GVG_011a	TAG_ONE_TURN_EFFECT
EX1_583	BATTLECRY
LOE_007t	EVIL_GLOW
LOE_007t	ImmuneToSpellpower

```
play_requirements_df.set_index('card_id', inplace=True)
play_requirements_df.head()
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

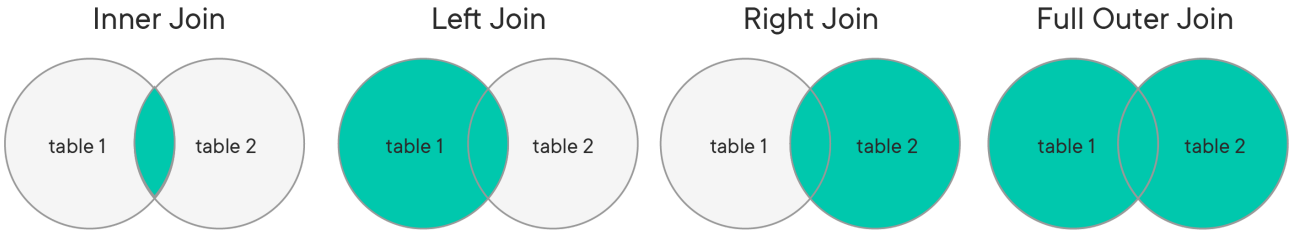
```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	play_requirement	value
card_id		
KARA_00_07	REQ_NUM_MINION_SLOTS	1
PRO_001a	REQ_NUM_MINION_SLOTS	1
NAX1_01	REQ_NUM_MINION_SLOTS	1
DS1h_292_H1	REQ_STEADY_SHOT	0
DS1h_292_H1	REQ_MINION_OR_ENEMY_HERO	0

Executing Joins

Now that you have the tables loaded correctly, we're going to execute some joins. There are four different kinds of joins, which can best be visualized with Venn diagrams:



In these diagrams, each circle represents a DataFrame or SQL Table. The left table is the table you are working with, and the right table is the table you want to join to the table you are working with. You'll start by executing the most common type of join, an ***Inner Join***.

In the cell below, join `cards_df` with `mechanics_df` using the built-in `.join()` method on the `cards_df` object.

Pass in the following parameters:

- the table you want to join with, `mechanics_df`
- The `how` parameter set to the type of join you want, `'inner'`

```
cards_with_mechanics_df = cards_df.join(mechanics_df, how='inner')
cards_with_mechanics_df
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	player_class	type	name	set
card_id				
AT_002	MAGE	SPELL	Effigy	TGT
AT_005t	NEUTRAL	MINION	Boar	TGT

	player_class	type	name	set
card_id				
AT_006	MAGE	MINION	Dalaran Aspirant	TGT
AT_007	MAGE	MINION	Spellslinger	TGT
AT_009	MAGE	MINION	Rhonin	TGT
AT_010	HUNTER	MINION	Ram Wrangler	TGT
AT_011e	NEUTRAL	ENCHANTMENT	Light's Blessing	TGT
AT_012	PRIEST	MINION	Spawn of Shadows	TGT
AT_017	NEUTRAL	MINION	Twilight Guardian	TGT

	player_class	type	name	set
card_id				
AT_018	PRIEST	MINION	Confessor Paletress	TGT
AT_019	WARLOCK	MINION	Dreadsteed	TGT
AT_022	WARLOCK	SPELL	Fist of Jaraxxus	TGT
AT_023	WARLOCK	MINION	Void Crusher	TGT
AT_028	ROGUE	MINION	Shado-Pan Rider	TGT
AT_030	ROGUE	MINION	Undercity Valiant	TGT
AT_032	ROGUE	MINION	Shady Dealer	TGT

	player_class	type	name	set
card_id				
AT_035t	ROGUE	SPELL	Ambush!	TGT
AT_036	ROGUE	MINION	Anub'arak	TGT
AT_037	DRUID	SPELL	Living Roots	TGT
AT_038	DRUID	MINION	Darnassus Aspirant	TGT
AT_038	DRUID	MINION	Darnassus Aspirant	TGT
AT_039	DRUID	MINION	Savage Combatant	TGT
AT_039e	DRUID	ENCHANTMENT	Savage	TGT
AT_040	DRUID	MINION	Wildwalker	TGT

	player_class	type	name	set
card_id				
AT_042	DRUID	MINION	Druid of the Saber	TGT
AT_042t	DRUID	MINION	Sabertooth Lion	TGT
AT_042t2	DRUID	MINION	Sabertooth Panther	TGT
AT_045	DRUID	MINION	Aviana	TGT
AT_046	SHAMAN	MINION	Tuskarr Totemic	TGT
AT_047	SHAMAN	MINION	Draenei Totemcarver	TGT
...	...	...	...	...
TB_SPT_DPromoSecret3	WARRIOR	SPELL	Visions of Valor	TB
TB_SPT_DPromoSecret4	WARRIOR	SPELL	Visions of Fate	TB
TB_SPT_DPromoSecret5	WARRIOR	SPELL	Visions of the Amazon	TB
TB_SPT_DPromoSecret6	WARRIOR	SPELL	Visions of the Sorcerer	TB

	player_class	type	name	set
card_id				
TB_SPT_DPromoSecret7	WARRIOR	SPELL	Visions of the Necromancer	TB
TB_SPT_DPromoSecret9	WARRIOR	SPELL	Visions of Knowledge	TB
TB_SPT_DPromo_Hero2	WARRIOR	HERO	The Cow King	TB
TB_SPT_DpromoPortal	NEUTRAL	MINION	Enigmatic Portal	TB
TB_SPT_DpromoPortal	NEUTRAL	MINION	Enigmatic Portal	TB
TB_SPT_Minion1	NEUTRAL	MINION	Shieldsman	TB
TB_SPT_Minion2	NEUTRAL	MINION	Battle Standard	TB
TB_SPT_Minion2	NEUTRAL	MINION	Battle Standard	TB
TU4c_003	NEUTRAL	MINION	Barrel	MISSIO
TU4c_008e	NEUTRAL	ENCHANTMENT	Might of Mukla	MISSIO
TU4f_007	NEUTRAL	MINION	Crazy Monkey	MISSIO
XXX_001	NEUTRAL	SPELL	Damage 1	CHEAT
XXX_002	NEUTRAL	SPELL	Damage 5	CHEAT
XXX_008	NEUTRAL	SPELL	Freeze	CHEAT
XXX_020	NEUTRAL	SPELL	Damage all but 1	CHEAT
XXX_060	NEUTRAL	SPELL	Damage All	CHEAT

	player_class	type	name	set
card_id				
XXX_100	NEUTRAL	MINION	Yogg-Saron Test (Manual)	CHEAT
XXX_101	NEUTRAL	SPELL	Set health to full	CHEAT
XXX_102	NEUTRAL	SPELL	Add 1 to Health.	CHEAT
XXX_103	NEUTRAL	SPELL	Add 2 to Health	CHEAT
XXX_104	NEUTRAL	SPELL	Add 4 to Health.	CHEAT
XXX_105	NEUTRAL	SPELL	Add 8 to Health.	CHEAT
XXX_107	NEUTRAL	SPELL	Set Health to 1	CHEAT
XXX_110	NEUTRAL	MINION	Yogg-Saron Test (Auto)	CHEAT
hexfrog	NEUTRAL	MINION	Frog	CORE
tt_010	MAGE	SPELL	Spellbender	EXPERT

1079 rows × 18 columns

Examine the output from the cell above and compare it to the original output of both the `cards_df` and `mechanics_df` DataFrame heads you displayed earlier. Notice how this now combines the columns from both?

Question

If you inspect the original `cards_df` DataFrame, you'll notice that it contains 2,819 records. The result of our inner join, `cards_with_mechanics_df`, contains only 1079 rows. Why?

Write your answer below this line:

```
# First performed an inner join, which only includes records that are present in bot
# Although there were 2819 records in the left table, there were only 1079 records t
# which are what you see in the resulting dataframe.
```

Other Types of Joins

By default, the `.join()` method performs a left join if no parameter is passed in for `how=`. In the cell below, perform a ***Left Join*** of `cards_with_mechanics_df` and `play_requirements_df`, with `cards_with_mechanics_df` as the left table.

Then, display `left_join_df` to inspect our results.

```
left_join_df = cards_with_mechanics_df.join(play_requirements_df)
left_join_df
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	player_class	type	name	set
card_id				
AT_002	MAGE	SPELL	Effigy	TGT
AT_005t	NEUTRAL	MINION	Boar	TGT

	player_class	type	name	set
card_id				
AT_006	MAGE	MINION	Dalaran Aspirant	TGT
AT_007	MAGE	MINION	Spellslinger	TGT
AT_009	MAGE	MINION	Rhonin	TGT
AT_010	HUNTER	MINION	Ram Wrangler	TGT
AT_011e	NEUTRAL	ENCHANTMENT	Light's Blessing	TGT
AT_012	PRIEST	MINION	Spawn of Shadows	TGT
AT_017	NEUTRAL	MINION	Twilight Guardian	TGT

	player_class	type	name	set
card_id				
AT_018	PRIEST	MINION	Confessor Paletress	TGT
AT_019	WARLOCK	MINION	Dreadsteed	TGT
AT_022	WARLOCK	SPELL	Fist of Jaraxxus	TGT
AT_023	WARLOCK	MINION	Void Crusher	TGT
AT_028	ROGUE	MINION	Shado-Pan Rider	TGT
AT_030	ROGUE	MINION	Undercity Valiant	TGT
AT_032	ROGUE	MINION	Shady Dealer	TGT

	player_class	type	name	set
card_id				
AT_035t	ROGUE	SPELL	Ambush!	TGT
AT_036	ROGUE	MINION	Anub'arak	TGT
AT_037	DRUID	SPELL	Living Roots	TGT
AT_038	DRUID	MINION	Darnassus Aspirant	TGT
AT_038	DRUID	MINION	Darnassus Aspirant	TGT
AT_039	DRUID	MINION	Savage Combatant	TGT
AT_039e	DRUID	ENCHANTMENT	Savage	TGT
AT_040	DRUID	MINION	Wildwalker	TGT

	player_class	type	name	set
card_id				
AT_040	DRUID	MINION	Wildwalker	TGT
AT_040	DRUID	MINION	Wildwalker	TGT
AT_040	DRUID	MINION	Wildwalker	TGT
AT_042	DRUID	MINION	Druid of the Saber	TGT
AT_042t	DRUID	MINION	Sabertooth Lion	TGT
AT_042t2	DRUID	MINION	Sabertooth Panther	TGT
...	...	...	...	...
TB_SPT_DPromoSecret3	WARRIOR	SPELL	Visions of Valor	TB
TB_SPT_DPromoSecret4	WARRIOR	SPELL	Visions of Fate	TB
TB_SPT_DPromoSecret5	WARRIOR	SPELL	Visions of the Amazon	TB

	player_class	type	name	set
card_id				
TB_SPT_DPromoSecret6	WARRIOR	SPELL	Visions of the Sorcerer	TB
TB_SPT_DPromoSecret7	WARRIOR	SPELL	Visions of the Necromancer	TB
TB_SPT_DPromoSecret9	WARRIOR	SPELL	Visions of Knowledge	TB
TB_SPT_DPromo_Hero2	WARRIOR	HERO	The Cow King	TB
TB_SPT_DpromoPortal	NEUTRAL	MINION	Enigmatic Portal	TB
TB_SPT_DpromoPortal	NEUTRAL	MINION	Enigmatic Portal	TB
TB_SPT_Minion1	NEUTRAL	MINION	Shieldsman	TB
TB_SPT_Minion2	NEUTRAL	MINION	Battle Standard	TB
TB_SPT_Minion2	NEUTRAL	MINION	Battle Standard	TB
TU4c_003	NEUTRAL	MINION	Barrel	MISSIO
TU4c_008e	NEUTRAL	ENCHANTMENT	Might of Mukla	MISSIO
TU4f_007	NEUTRAL	MINION	Crazy Monkey	MISSIO
XXX_001	NEUTRAL	SPELL	Damage 1	CHEAT
XXX_002	NEUTRAL	SPELL	Damage 5	CHEAT
XXX_008	NEUTRAL	SPELL	Freeze	CHEAT
XXX_020	NEUTRAL	SPELL	Damage all but 1	CHEAT

	player_class	type	name	set
card_id				
XXX_060	NEUTRAL	SPELL	Damage All	CHEAT
XXX_100	NEUTRAL	MINION	Yogg-Saron Test (Manual)	CHEAT
XXX_101	NEUTRAL	SPELL	Set health to full	CHEAT
XXX_102	NEUTRAL	SPELL	Add 1 to Health.	CHEAT
XXX_103	NEUTRAL	SPELL	Add 2 to Health	CHEAT
XXX_104	NEUTRAL	SPELL	Add 4 to Health.	CHEAT
XXX_105	NEUTRAL	SPELL	Add 8 to Health.	CHEAT
XXX_107	NEUTRAL	SPELL	Set Health to 1	CHEAT
XXX_110	NEUTRAL	MINION	Yogg-Saron Test (Auto)	CHEAT
hexfrog	NEUTRAL	MINION	Frog	CORE
tt_010	MAGE	SPELL	Spellbender	EXPERT

1226 rows × 20 columns

Note that the results of this sort of join are dependent upon the position of each table--if you were to make `cards_with_mechanics_df` the right table and `play_requirements_df` the left table and then perform a ***Right Join***, our results would be the same.

Question:

Describe what was included from each table in this join.

Write your answer below this line:

```
# Every record from cards_with_mechanics_df, as well as any records from play_requir
# with a record from the left table.
```

Outer Joins

In the cell below, perform an outer join between `cards_df` and `dust_df`. Since these tables contain columns with the same name, we'll need to specify a suffix for at least one of them, so that the column can be renamed to avoid a naming collision.

During this join, set the `rsuffix` parameter to `_dust`

```
outer_join_df = cards_df.join(dust_df, rsuffix='_dust', how='outer')
outer_join_df
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	player_class	type	name	set	
card_id					
AT_001	MAGE	SPELL	Flame Lance	TGT	De mi

	player_class	type	name	set	
card_id					
AT_001	MAGE	SPELL	Flame Lance	TGT	De mi
AT_001	MAGE	SPELL	Flame Lance	TGT	De mi
AT_001	MAGE	SPELL	Flame Lance	TGT	De mi
AT_002	MAGE	SPELL	Effigy	TGT	<b Wh mi
AT_002	MAGE	SPELL	Effigy	TGT	<b Wh mi
AT_002	MAGE	SPELL	Effigy	TGT	<b Wh mi
AT_002	MAGE	SPELL	Effigy	TGT	<b Wh mi
AT_003	MAGE	MINION	Fallen Hero	TGT	You dea dar
AT_003	MAGE	MINION	Fallen Hero	TGT	You dea dar

	player_class	type	name	set	
card_id					
AT_003	MAGE	MINION	Fallen Hero	TGT	You dea dar
AT_003	MAGE	MINION	Fallen Hero	TGT	You dea dar
AT_004	MAGE	SPELL	Arcane Blast	TGT	De mir get
AT_004	MAGE	SPELL	Arcane Blast	TGT	De mir get
AT_004	MAGE	SPELL	Arcane Blast	TGT	De mir get
AT_004	MAGE	SPELL	Arcane Blast	TGT	De mir get
AT_005	MAGE	SPELL	Polymorph: Boar	TGT	Tra int <b
AT_005	MAGE	SPELL	Polymorph: Boar	TGT	Tra int <b
AT_005	MAGE	SPELL	Polymorph: Boar	TGT	Tra int <b
AT_005	MAGE	SPELL	Polymorph: Boar	TGT	Tra int <b
AT_005t	NEUTRAL	MINION	Boar	TGT	<b

	player_class	type	name	set	
card_id					
AT_006	MAGE	MINION	Dalaran Aspirant	TGT	<b Gai Dar
AT_006	MAGE	MINION	Dalaran Aspirant	TGT	<b Gai Dar
AT_006	MAGE	MINION	Dalaran Aspirant	TGT	<b Gai Dar
AT_006	MAGE	MINION	Dalaran Aspirant	TGT	<b Gai Dar
AT_006e	MAGE	ENCHANTMENT	Power of Dalaran	TGT	Inc Dar
AT_007	MAGE	MINION	Spellslinger	TGT	<b Ad to
AT_007	MAGE	MINION	Spellslinger	TGT	<b Ad to

	player_class	type	name	set	
card_id					
AT_007	MAGE	MINION	Spellslinger	TGT	<b Ad to
AT_007	MAGE	MINION	Spellslinger	TGT	<b Ad to
...	...	...	...	...	...
XXX_110	NEUTRAL	MINION	Yogg-Saron Test (Auto)	CHEAT	<b Cas spe
XXX_111	NEUTRAL	SPELL	AI Buddy - All Charge, All Windfury!	CHEAT	Pla all <b
XXX_111e	NEUTRAL	ENCHANTMENT	All Charge, All Windfury, All The Time	CHEAT	You hav <b <b
XXX_112	NEUTRAL	SPELL	Fill Deck	CHEAT	Fill wit
XXX_113	NEUTRAL	SPELL	Again	CHEAT	Na
XXX_115	NEUTRAL	SPELL	Destroy Played Cards	CHEAT	Wh sur des
XXX_115e	NEUTRAL	ENCHANTMENT	Destroy Played Card Enchantment	CHEAT	Wh sur des

	player_class	type	name	set	
card_id					
XXX_119	NEUTRAL	SPELL	Death No Rattle	CHEAT	Die trig <b Al..
XXX_119e	NaN	ENCHANTMENT	Death No Rattle	CHEAT	Die trig <b Al..
XXX_999_Crash	NEUTRAL	SPELL	Crash the server	CHEAT	Cra
ds1_whelptoken	NEUTRAL	MINION	Whelp	EXPERT1	Na
hexfrog	NEUTRAL	MINION	Frog	CORE	<b
skele11	NEUTRAL	MINION	Skeleton	CORE	<b
skele21	NEUTRAL	MINION	Damaged Golem	EXPERT1	Na
skele21	NEUTRAL	MINION	Damaged Golem	EXPERT1	Na
skele21	NEUTRAL	MINION	Damaged Golem	EXPERT1	Na
skele21	NEUTRAL	MINION	Damaged Golem	EXPERT1	Na
tt_004	NEUTRAL	MINION	Flesheating Ghoul	EXPERT1	Wh die
tt_004	NEUTRAL	MINION	Flesheating Ghoul	EXPERT1	Wh die

	player_class	type	name	set	
card_id					
tt_004	NEUTRAL	MINION	Flesheating Ghoul	EXPERT1	Wh die
tt_004	NEUTRAL	MINION	Flesheating Ghoul	EXPERT1	Wh die
tt_004o	NEUTRAL	ENCHANTMENT	Cannibalize	EXPERT1	Inc
tt_010	MAGE	SPELL	Spellbender	EXPERT1	<b Wh cas
tt_010	MAGE	SPELL	Spellbender	EXPERT1	<b Wh cas
tt_010	MAGE	SPELL	Spellbender	EXPERT1	<b Wh cas
tt_010	MAGE	SPELL	Spellbender	EXPERT1	<b Wh cas

	player_class	type	name	set	
card_id					
tt_010a	MAGE	MINION	Spellbender	EXPERT1	Na
tt_010a	MAGE	MINION	Spellbender	EXPERT1	Na
tt_010a	MAGE	MINION	Spellbender	EXPERT1	Na
tt_010a	MAGE	MINION	Spellbender	EXPERT1	Na

5849 rows × 19 columns

Inspect the output above. Note that the naming collision has been avoided by renaming the `cost` column from the right table to `cost_dust`.

Summary

In this lab, you learned how to:

- Concatenate multiple DataFrames together into a single DataFrame
- Understand and execute the various types of joins (inner, outer, left, and right joins)

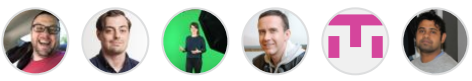
Releases

No releases published

Packages

No packages published

Contributors 6



Languages

- **Jupyter Notebook** 100.0%