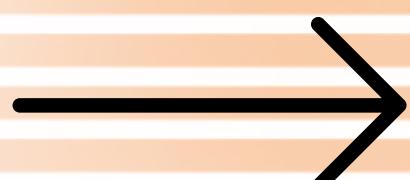


Data Engineering 101

Pandas vs SQL



Shwetank Singh
GritSetGrow - GSGLearn.com



Select all rows

Pandas

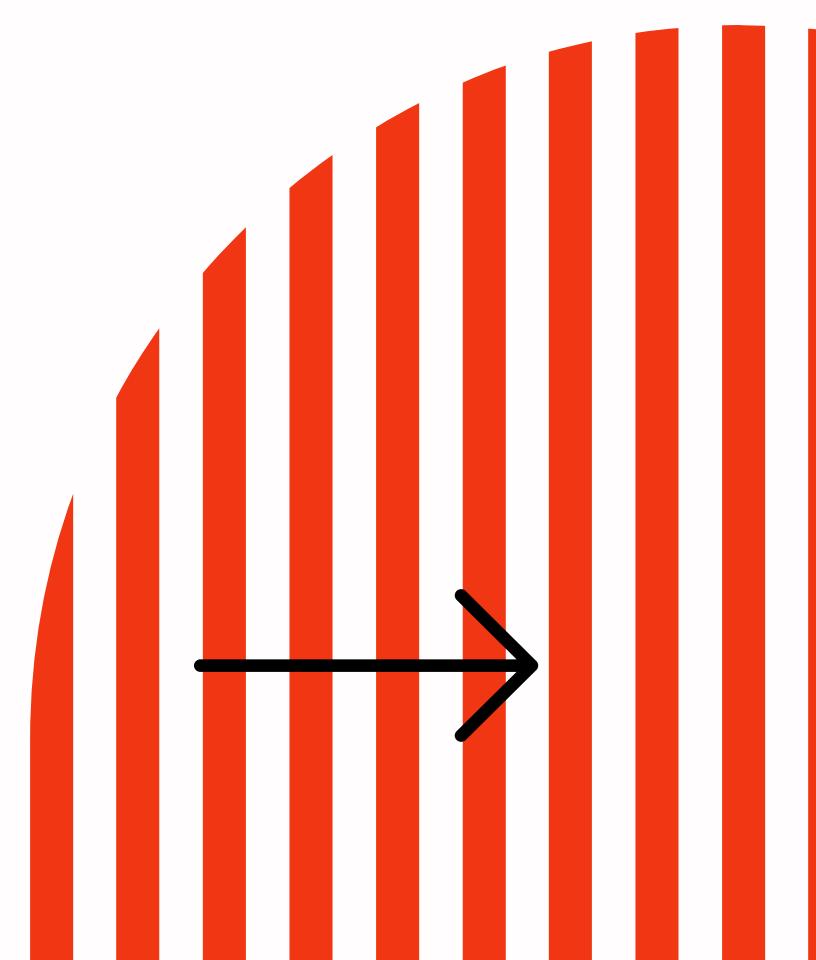
df

SQL

SELECT * FROM table;



Shwetank Singh
GritSetGrow - GSGLearn.com



Select specific columns

Pandas

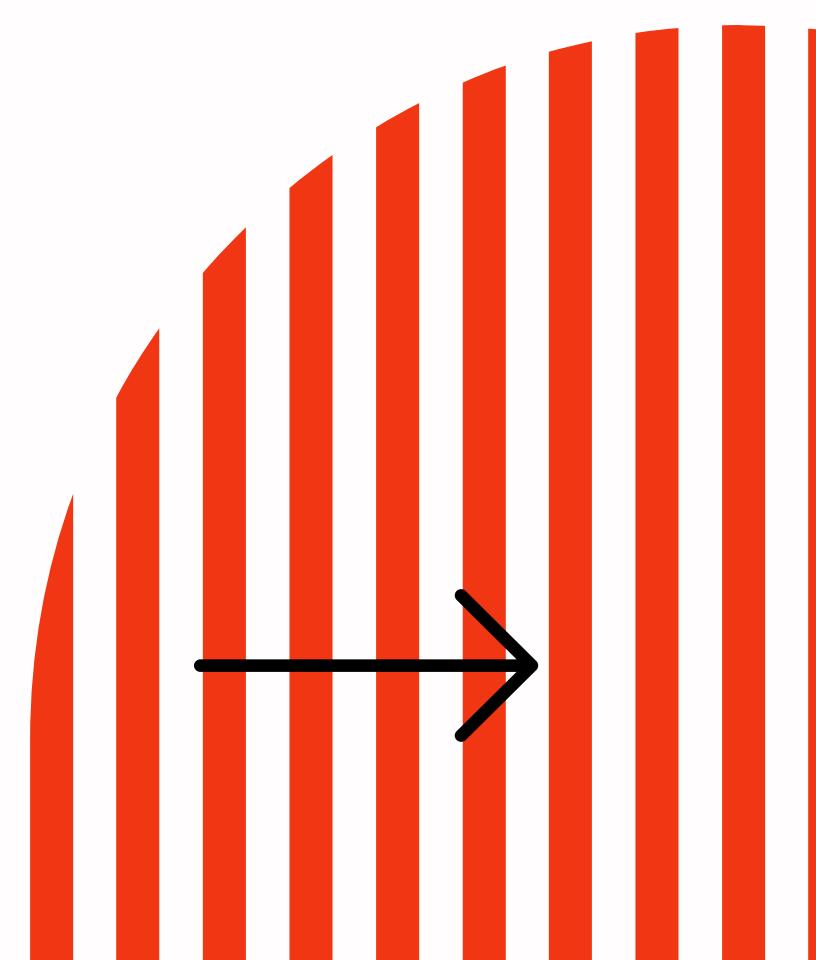
```
df[['col1', 'col2']]
```

SQL

```
SELECT col1, col2  
FROM table;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Filter rows based on a condition

Pandas

```
df[df['col1'] > 5]
```

SQL

```
SELECT *  
FROM table  
WHERE col1 > 5;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Filter rows based on multiple conditions

Pandas

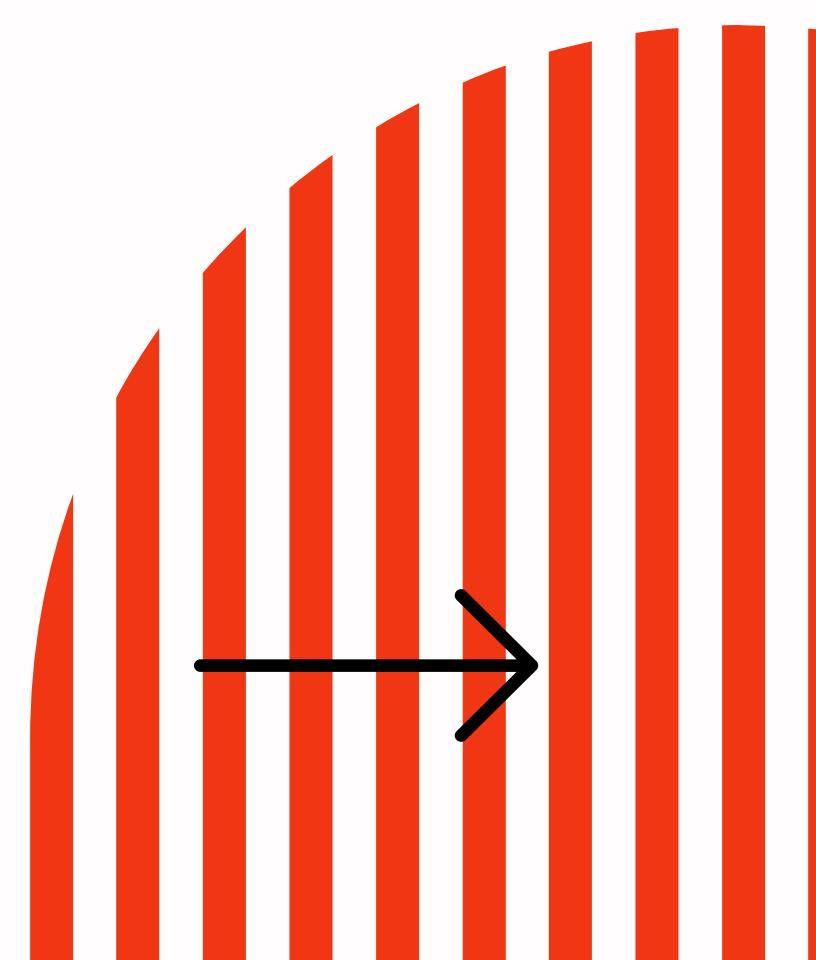
```
df[(df['col1'] > 5) & (df['col2'] < 10)]
```

SQL

```
SELECT *  
FROM table  
WHERE col1 > 5 AND  
col2 < 10;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Group by a column and count

h

Pandas

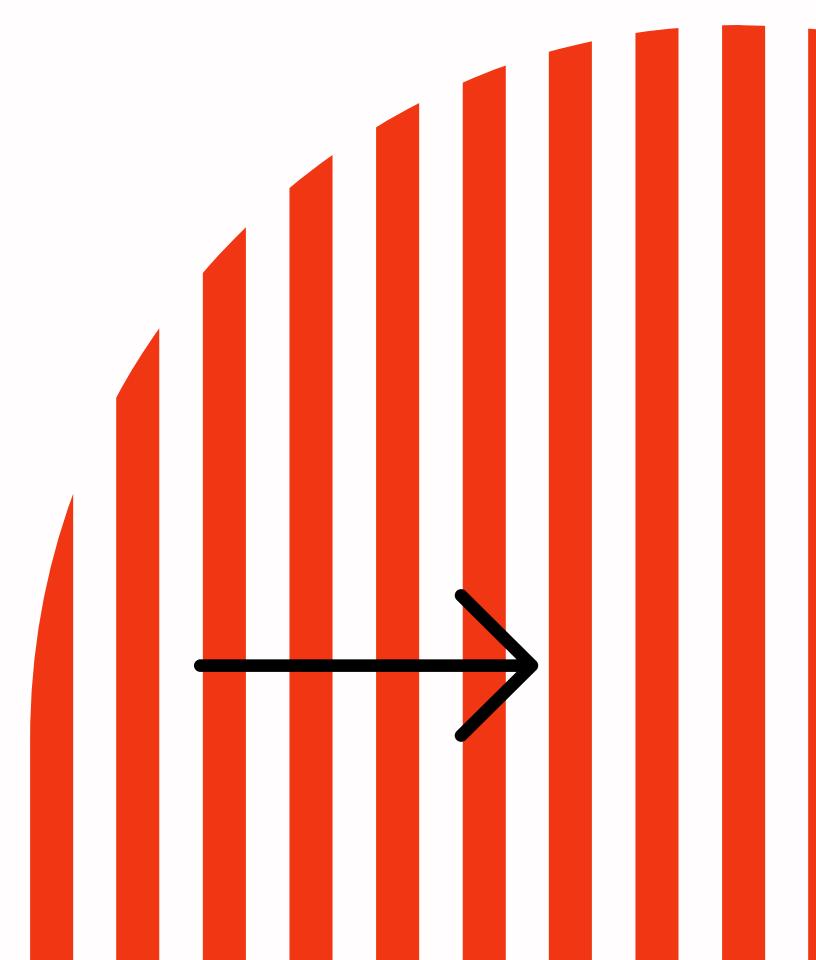
```
df.groupby('col1').size()
```

SQL

```
SELECT col1, COUNT(*)  
FROM table  
GROUP BY col1;
```



Shwetank Singh
GritSetGrow - GSGLearn.com





Group by a column and sum another column

Pandas

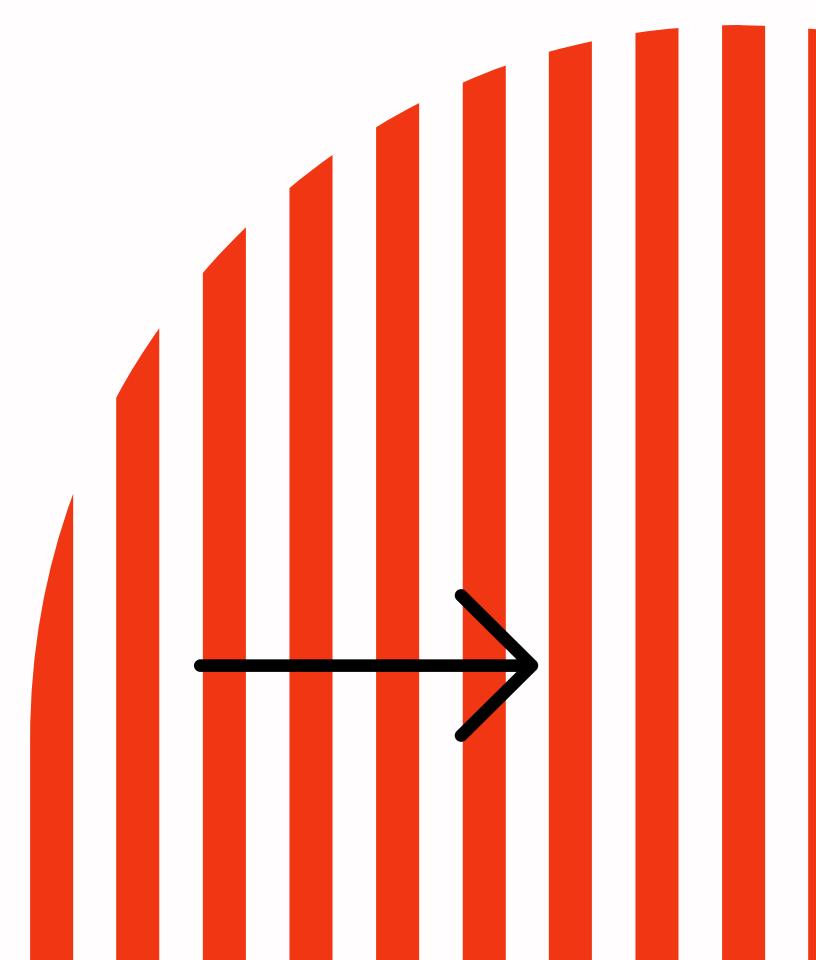
```
df.groupby('col1')['col2'].sum()
```

SQL

```
SELECT col1, SUM(col2)  
FROM table  
GROUP BY col1;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Order by a column

Pandas

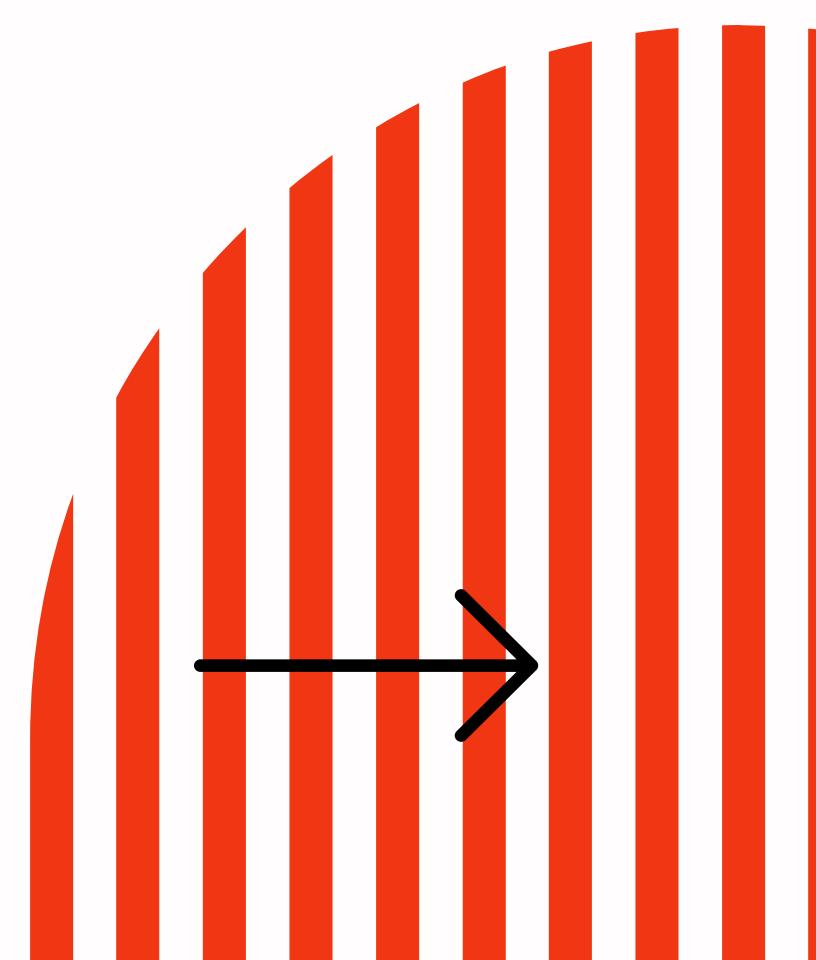
```
df.sort_values(by='col1')
```

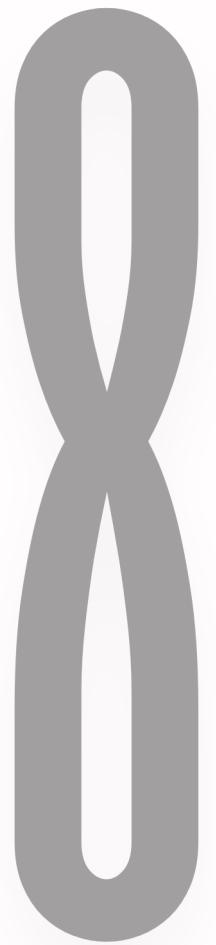
SQL

```
SELECT *  
FROM table  
ORDER BY col1;
```



Shwetank Singh
GritSetGrow - GSGLearn.com





Order by a column descending

Pandas

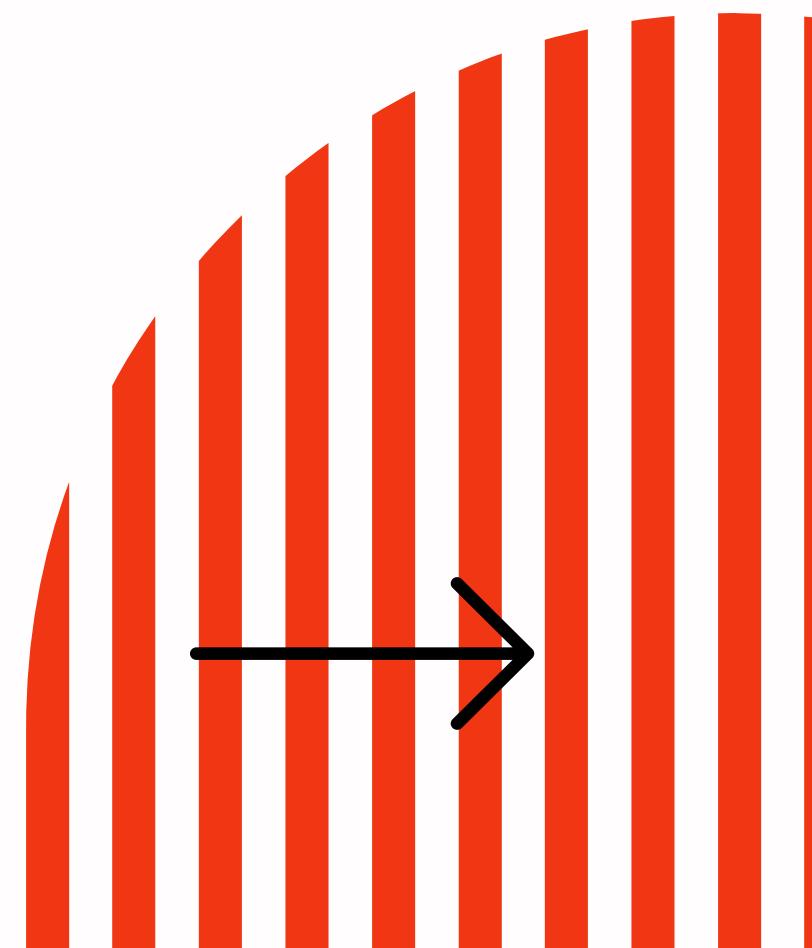
```
df.sort_values(by='col1',  
               ascending=False)
```

SQL

```
SELECT * FROM table  
ORDER BY col1 DESC;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Join two tables

Pandas

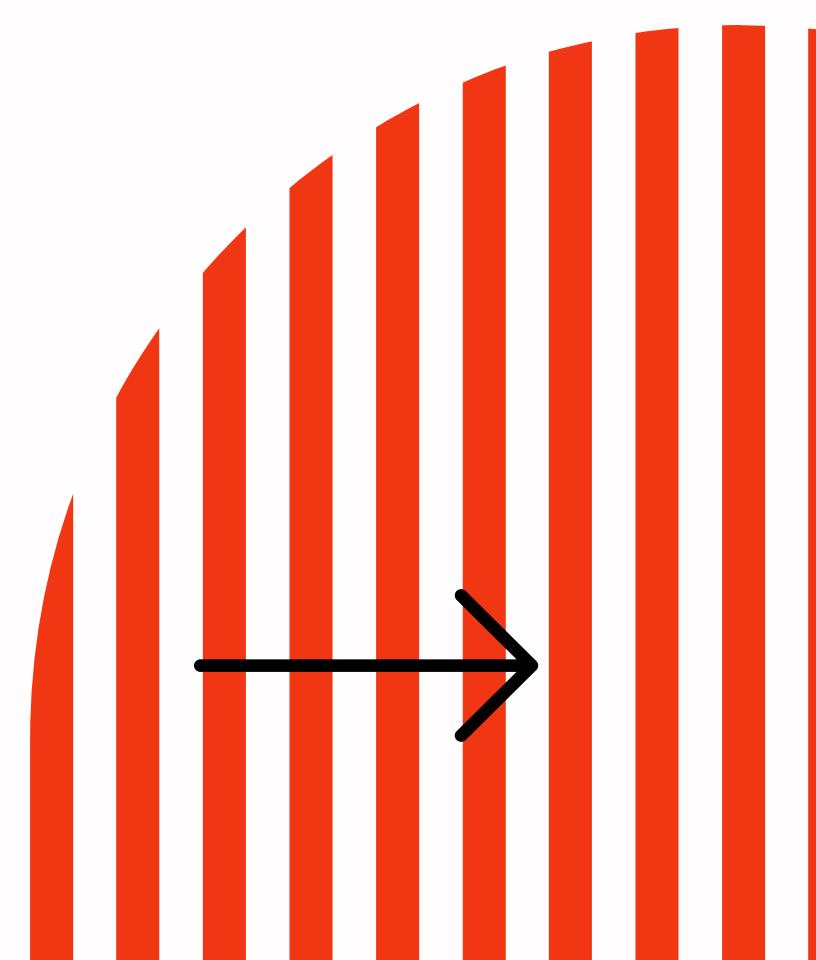
```
pd.merge(df1, df2, on='id')
```

SQL

```
SELECT * FROM table1
JOIN table2
ON table1.id = table2.id;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



10

Left join two tables

Pandas

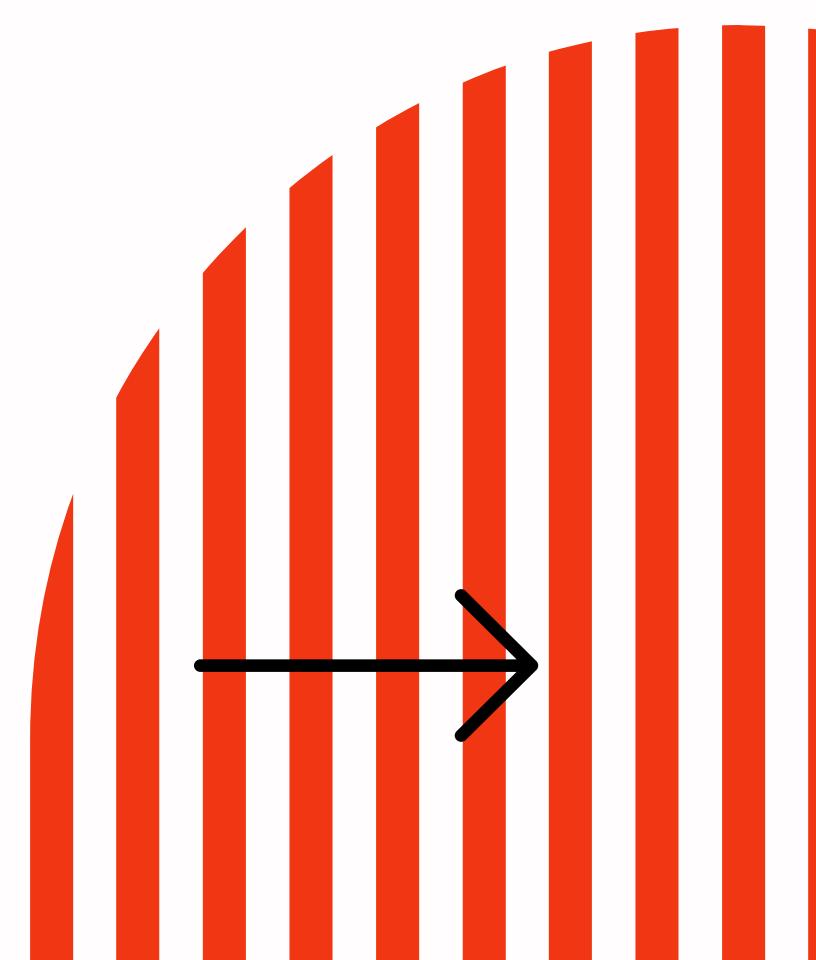
```
pd.merge(df1, df2, on='id', how='left')
```

SQL

```
SELECT * FROM table1  
LEFT JOIN table2  
ON table1.id = table2.id;
```



Shwetank Singh
GritSetGrow - GSGLearn.com





Right join two tables

Pandas

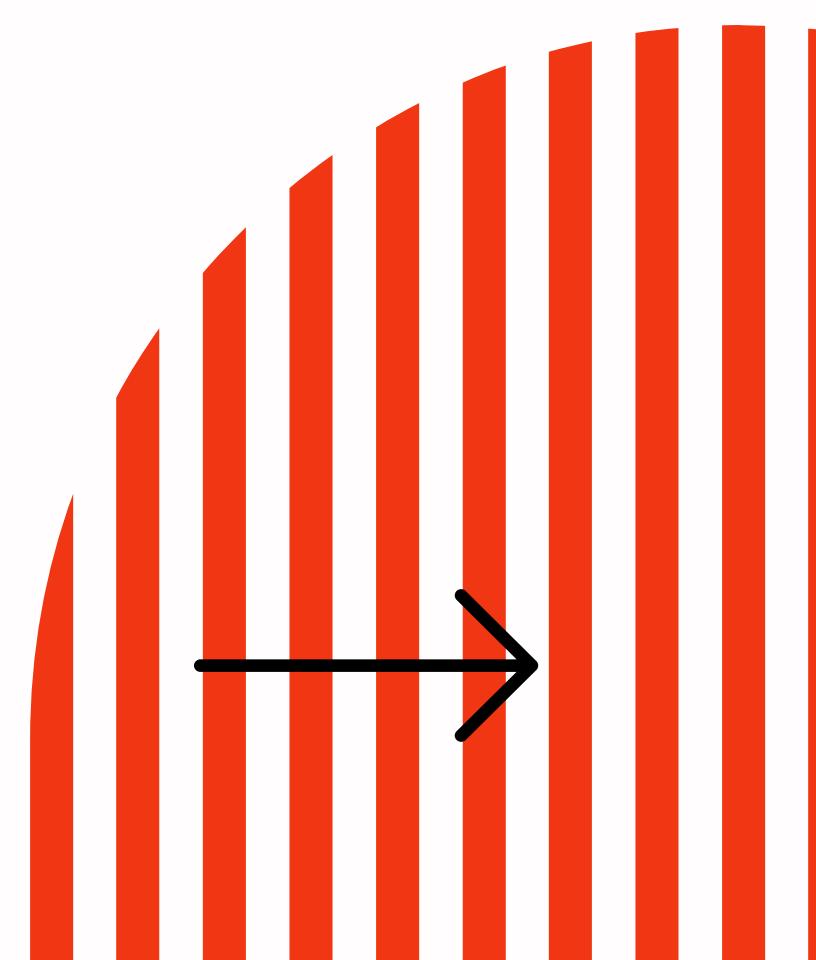
```
pd.merge(df1, df2, on='id', how='right')
```

SQL

```
SELECT * FROM table1  
RIGHT JOIN table2  
ON table1.id = table2.id;
```



Shwetank Singh
GritSetGrow - GSGLearn.com





Full outer join two tables

Pandas

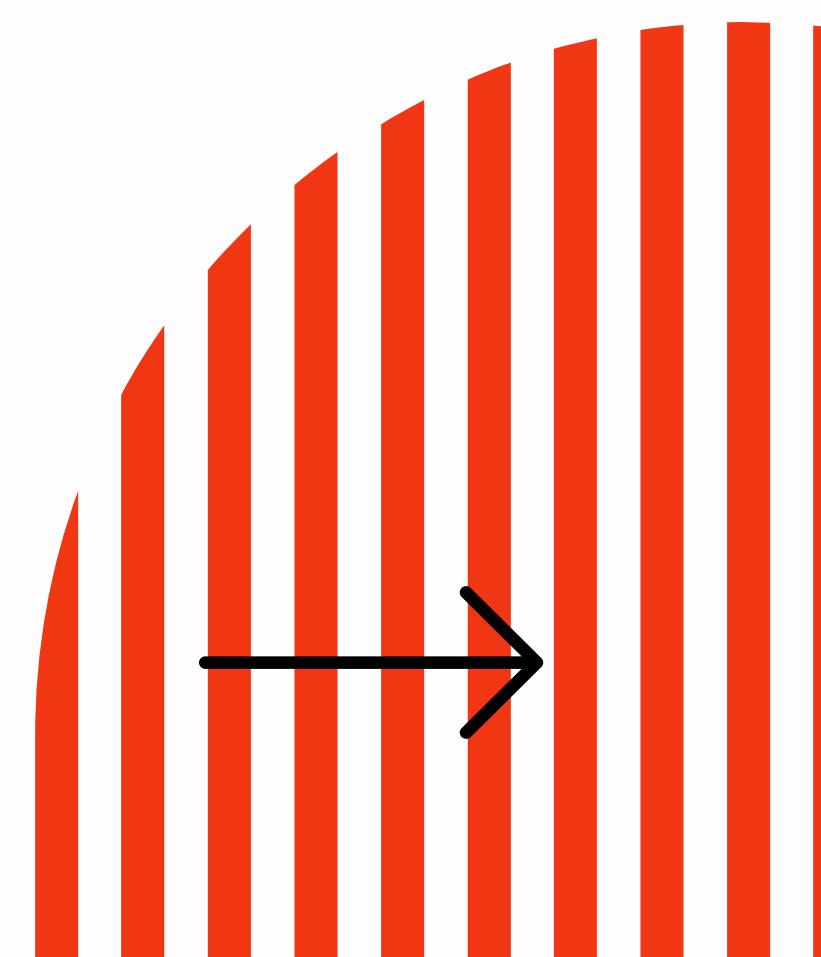
```
pd.merge(df1, df2, on='id', how='outer')
```

SQL

```
SELECT * FROM table1
FULL OUTER JOIN table2
ON table1.id = table2.id;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



10
13

Calculate the average of a column

Pandas

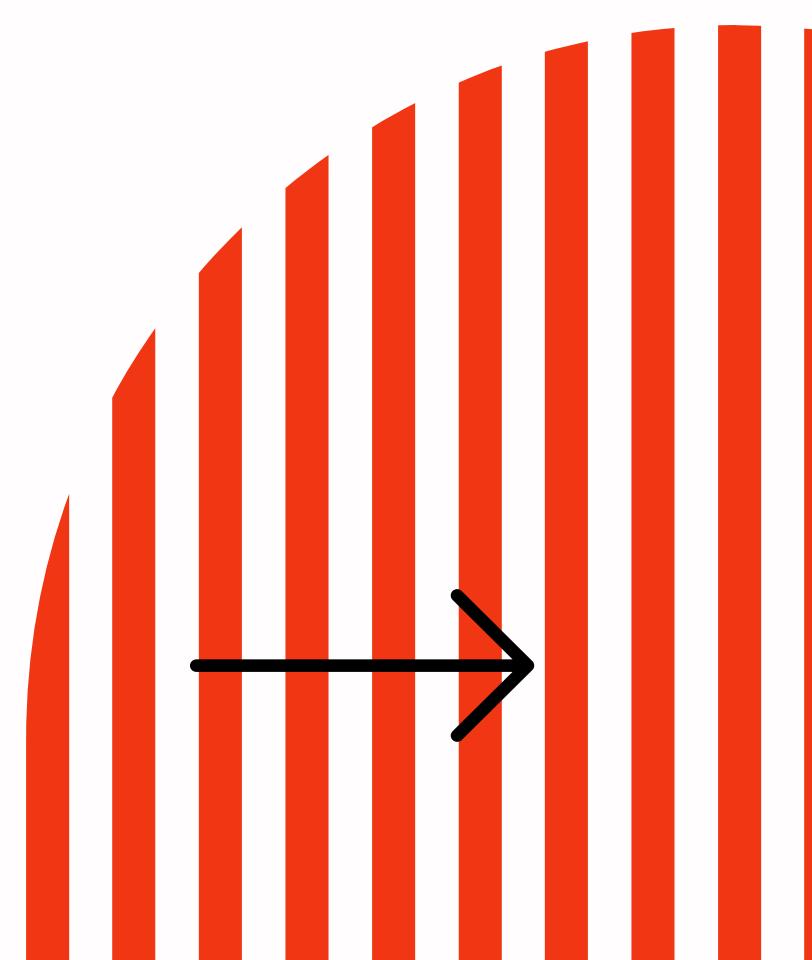
```
df['col1'].mean()
```

SQL

```
SELECT AVG(col1)  
FROM table;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Calculate the sum of a column

Pandas

```
df['col1'].sum()
```

SQL

```
SELECT SUM(col1)  
FROM table;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



15

Calculate the maximum of a column

Pandas

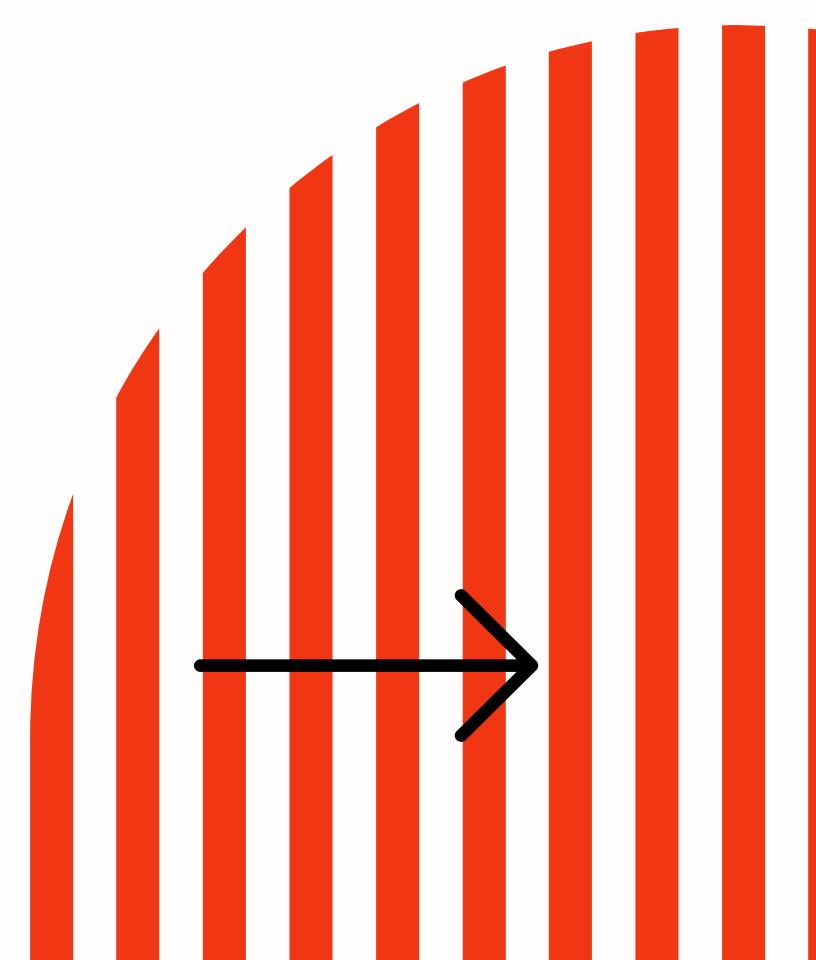
```
df['col1'].max()
```

SQL

```
SELECT MAX(col1)  
FROM table;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



16

Calculate the minimum of a column

Pandas

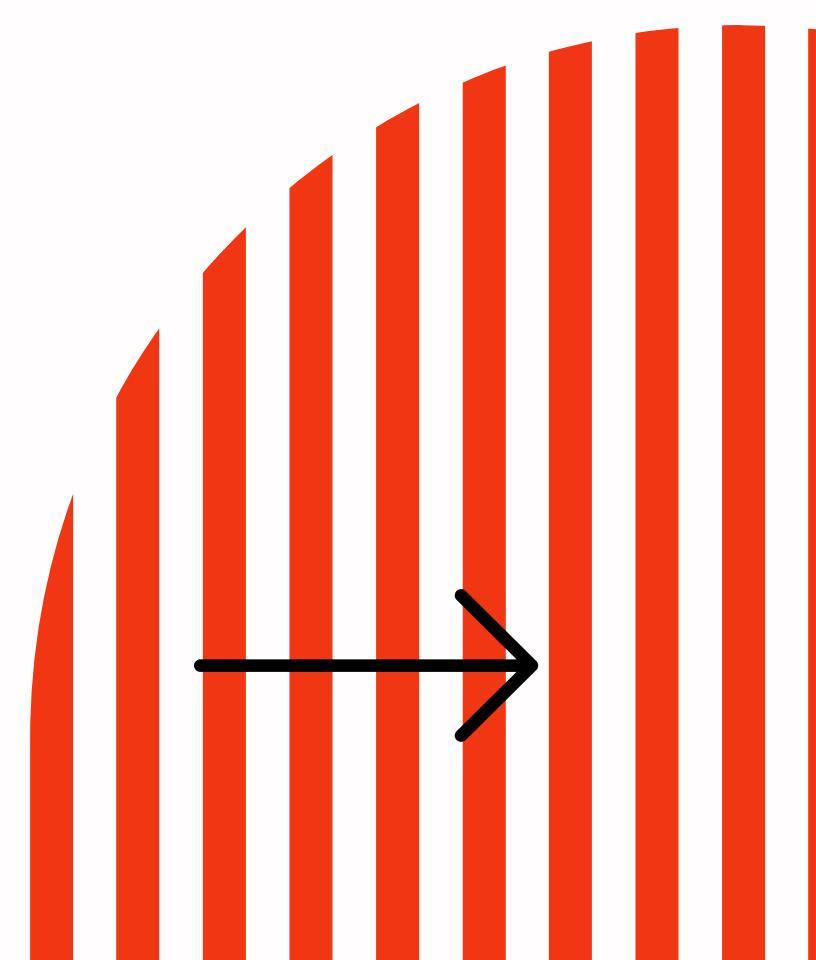
```
df['col1'].min()
```

SQL

```
SELECT MIN(col1)  
FROM table;
```



Shwetank Singh
GritSetGrow - GSGLearn.com





Count distinct values in a column

Pandas

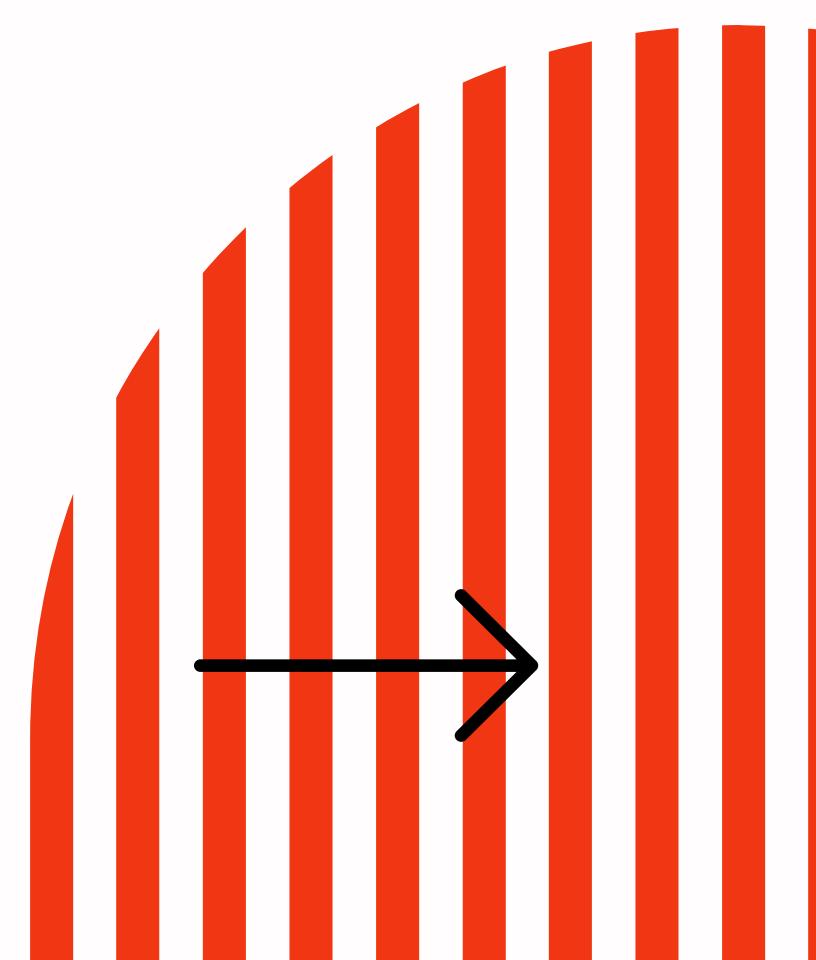
```
df['col1'].nunique()
```

SQL

```
SELECT  
COUNT(DISTINCT col1)  
FROM table;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



18

Rename a column

Pandas

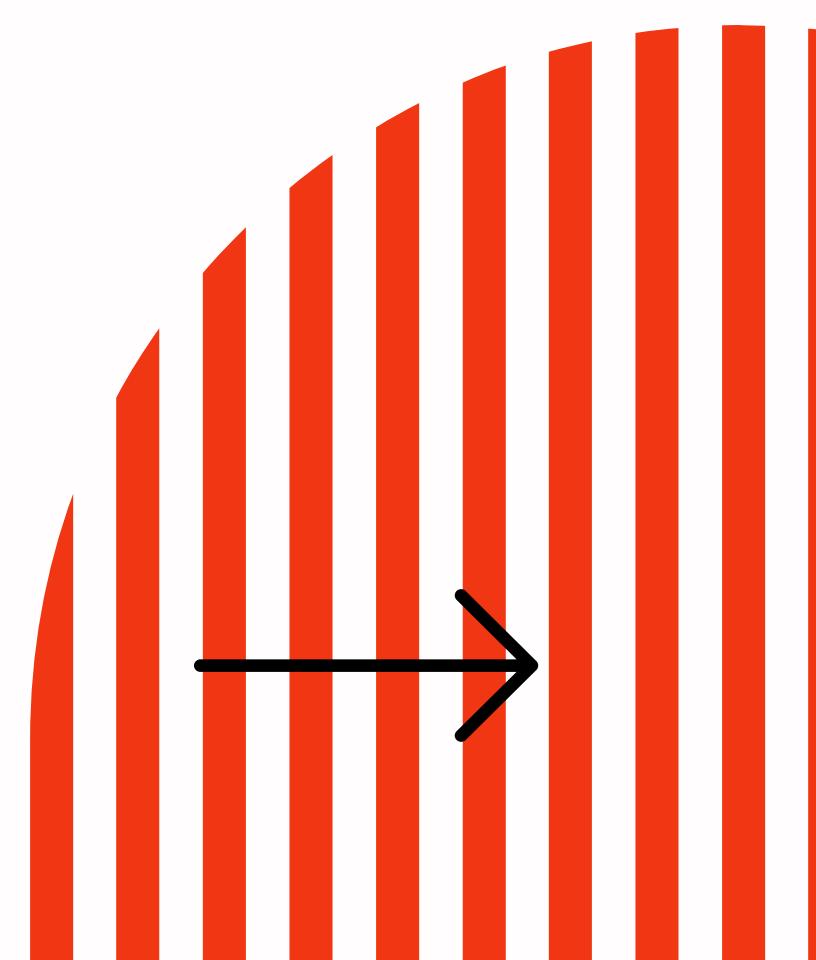
```
df.rename(columns= {  
    'old_name': 'new_name'})
```

SQL

```
ALTER TABLE table  
RENAME COLUMN  
old_name TO new_name;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



10
9
8
7
6
5
4
3
2
1

Add a new column

Pandas

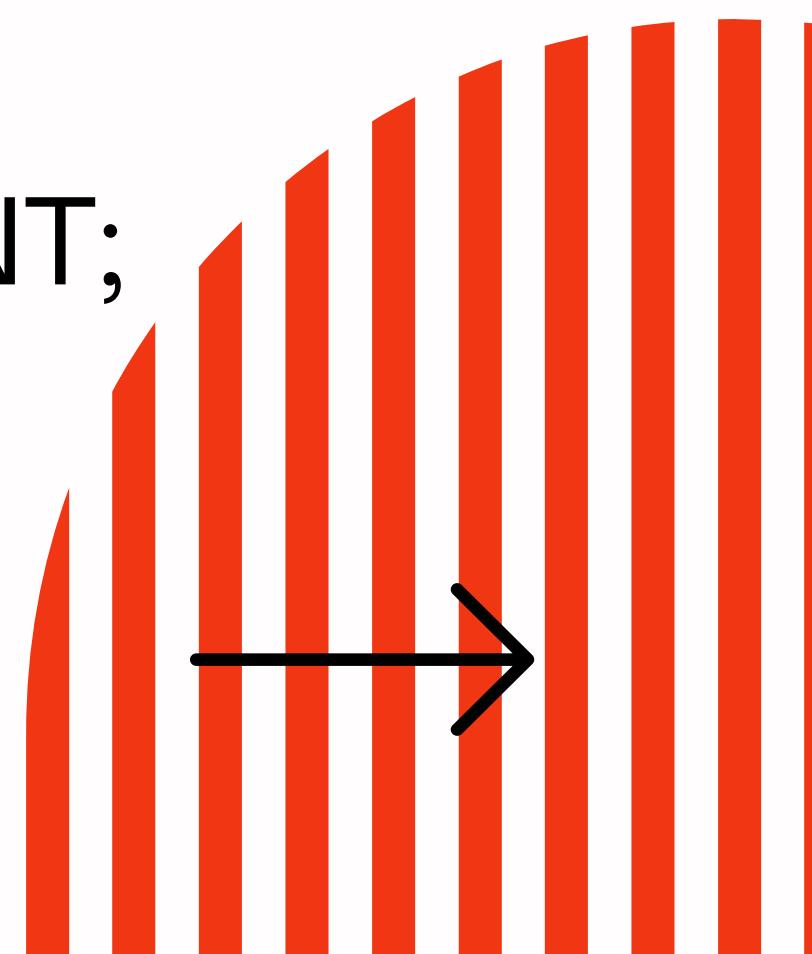
```
df['new_col'] = value
```

SQL

```
ALTER TABLE table  
ADD COLUMN new_col INT;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



20

Drop a column

Pandas

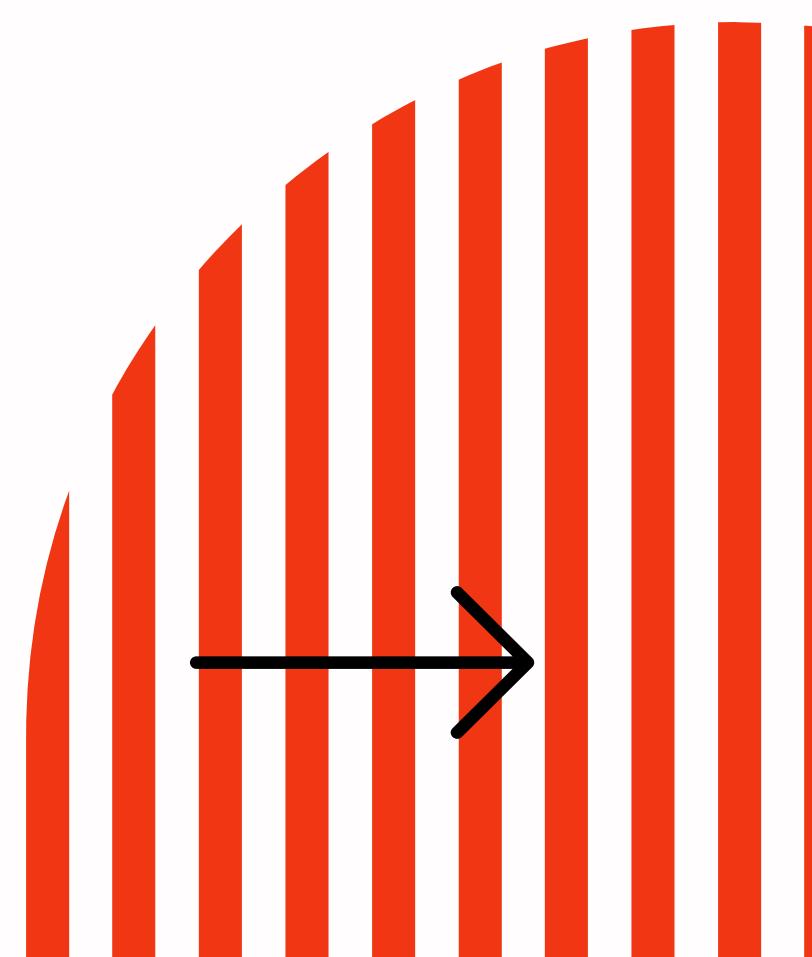
```
df.drop(columns=['col1'])
```

SQL

```
ALTER TABLE table  
DROP COLUMN col1;
```



Shwetank Singh
GritSetGrow - GSGLearn.com





Replace null values

Pandas

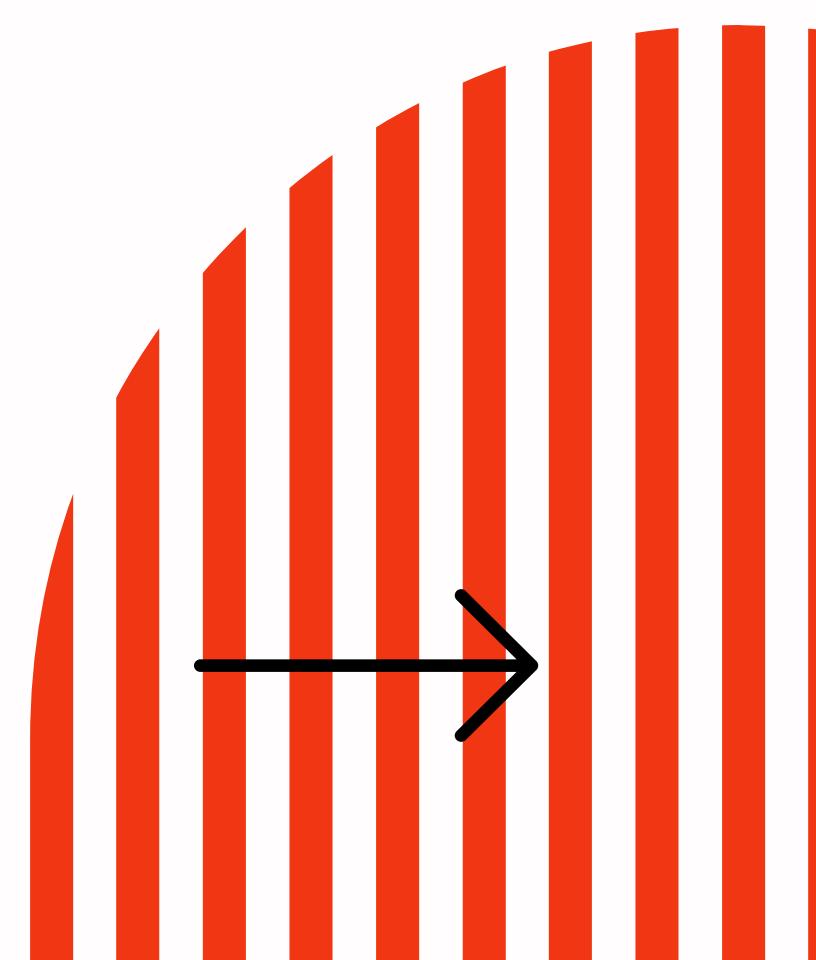
```
df['col1'].fillna(0)
```

SQL

```
SELECT  
COALESCE(col1, 0)  
FROM table;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Calculate the difference between two columns

Pandas

```
df['col1'] - df['col2']
```

SQL

```
SELECT col1 - col2  
FROM table;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



20
21
22
23

Concatenate two columns

Pandas

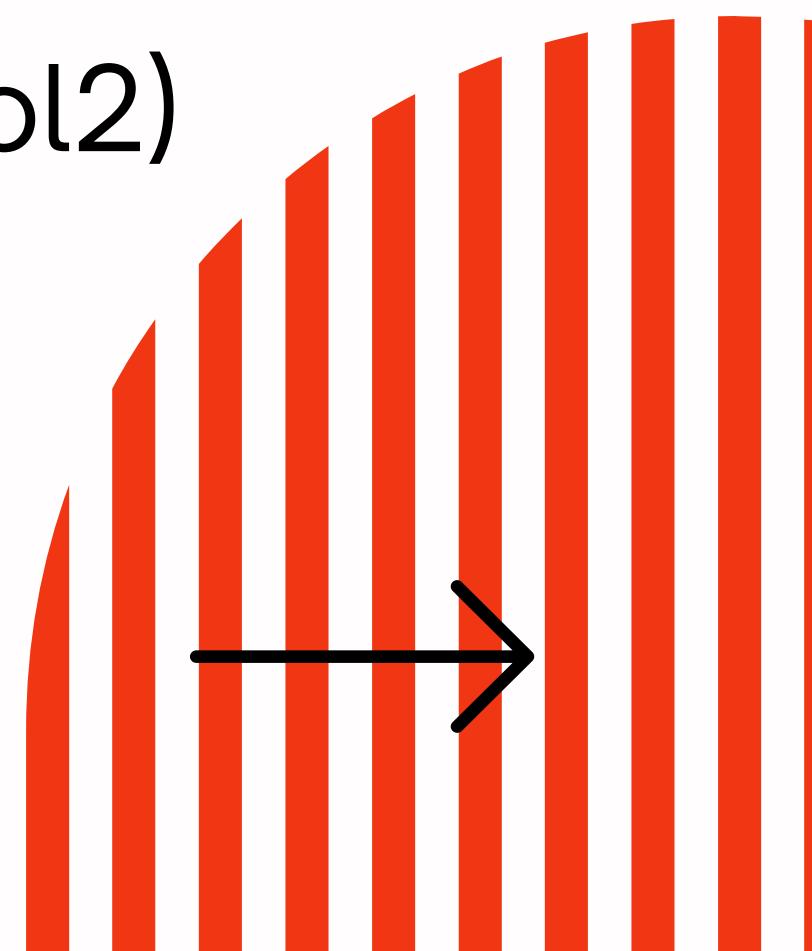
```
df['col1'] + df['col2']
```

SQL

```
SELECT CONCAT(col1, col2)  
FROM table;
```



Shwetank Singh
GritSetGrow - GSGLearn.com





Extract year from date

Pandas

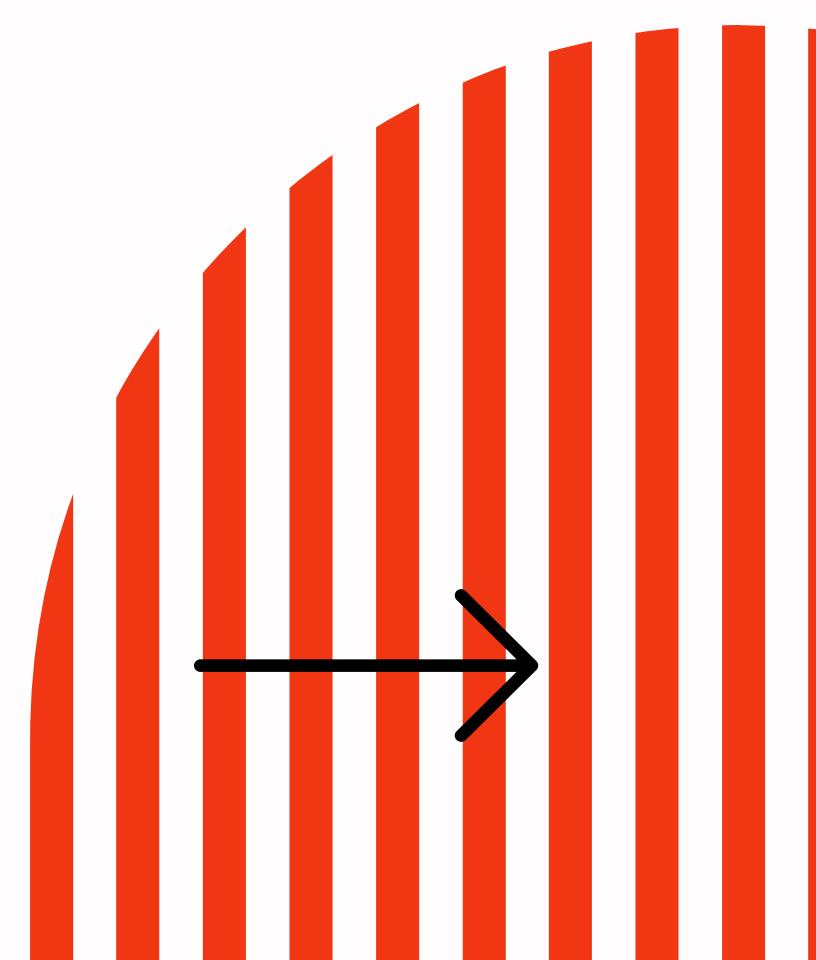
```
df['date_col'].dt.year
```

SQL

```
SELECT YEAR(date_col)  
FROM table;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Extract month from date

SQL

Pandas

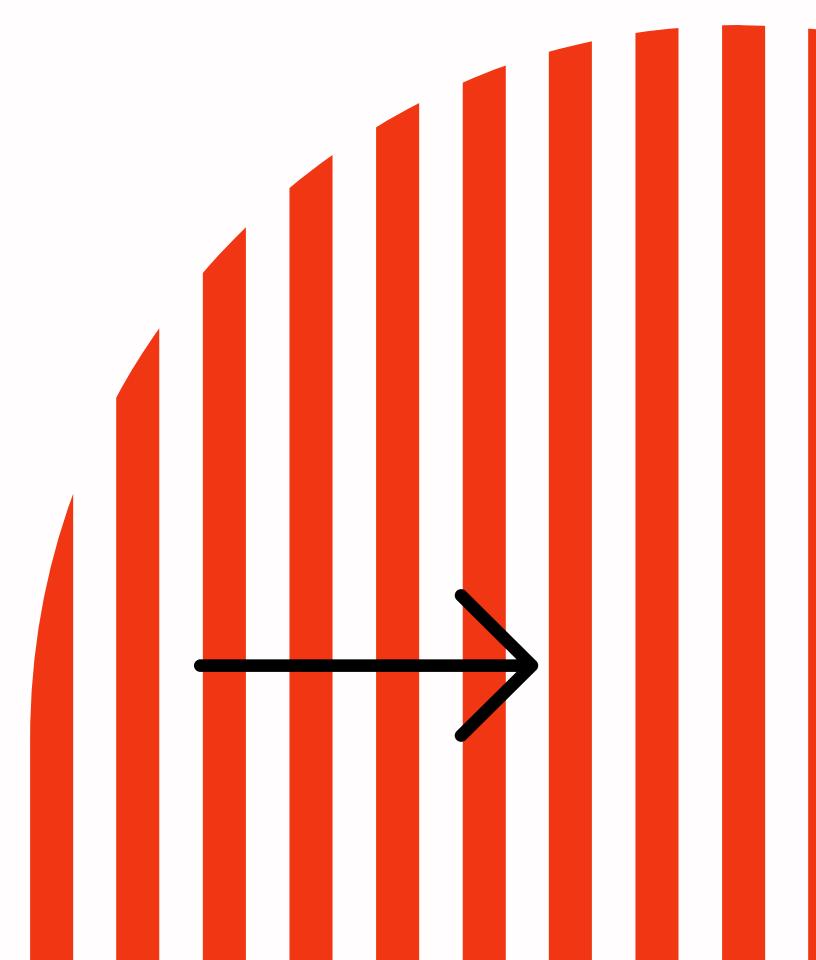
```
df['date_col'].dt.month
```

SQL

```
SELECT  
MONTH(date_col) FROM  
table;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



20

Extract day from date

Pandas

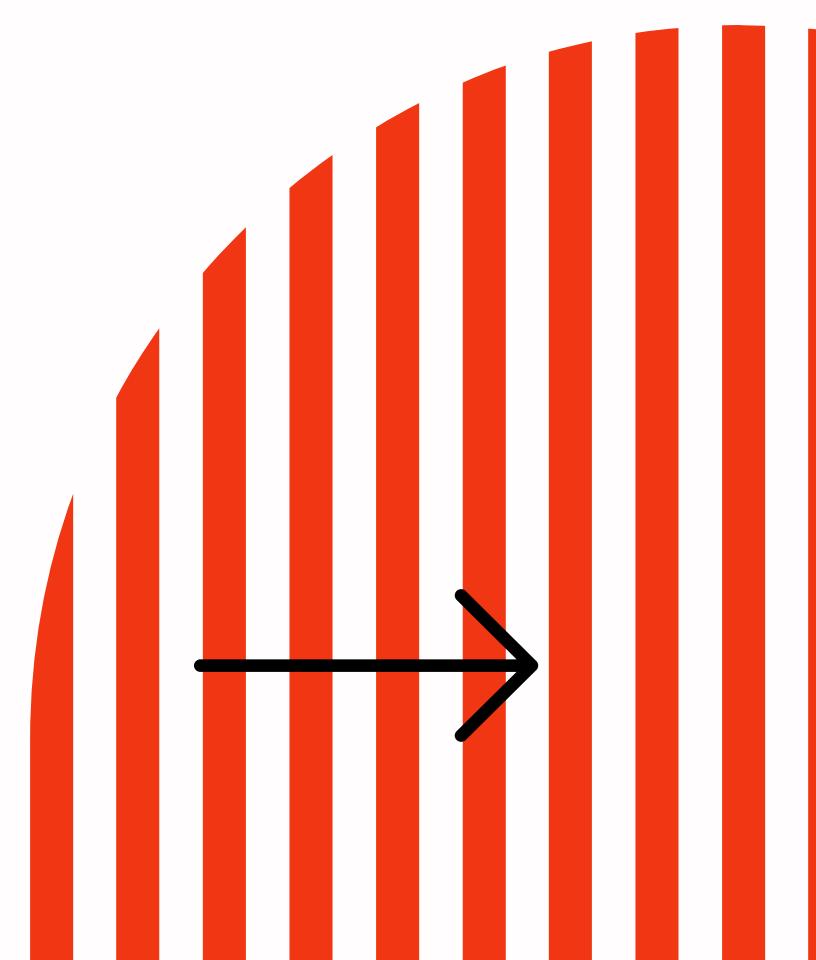
```
df['date_col'].dt.day
```

SQL

```
SELECT DAY(date_col)  
FROM table;
```



Shwetank Singh
GritSetGrow - GSGLearn.com





Filter rows based on string matching

Pandas

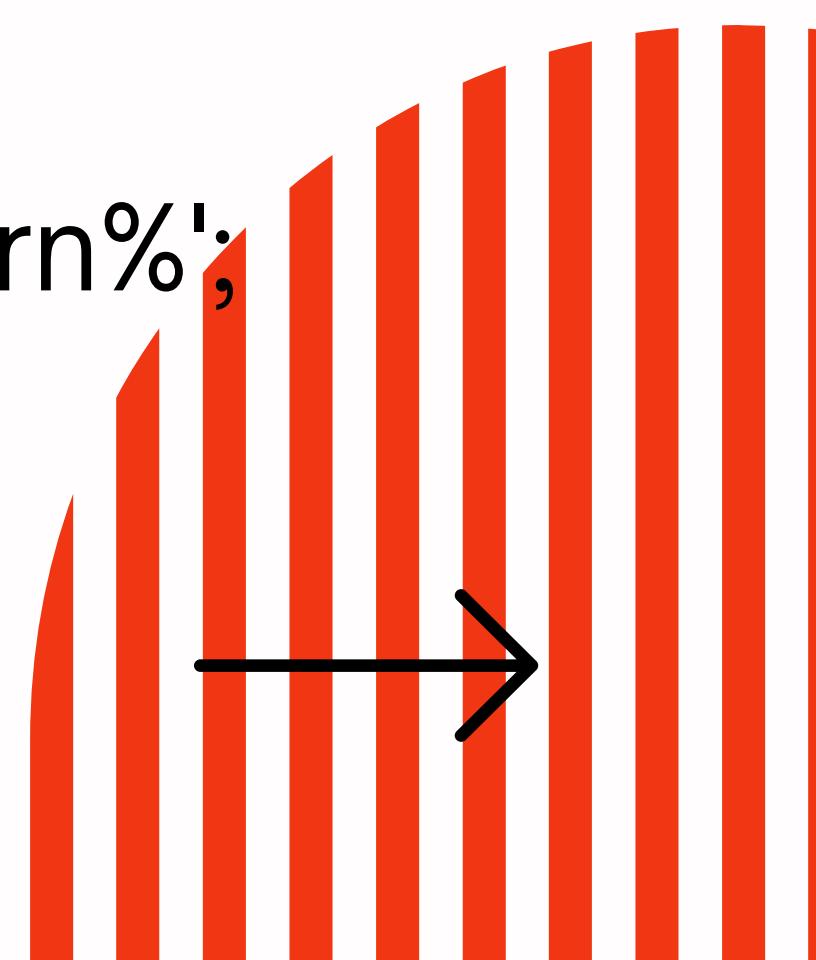
```
df[df['col1'].str.contains('pattern')]
```

SQL

```
SELECT * FROM table  
WHERE col1 LIKE '%pattern%';
```



Shwetank Singh
GritSetGrow - GSGLearn.com



28

Aggregate functions with group by

Pandas

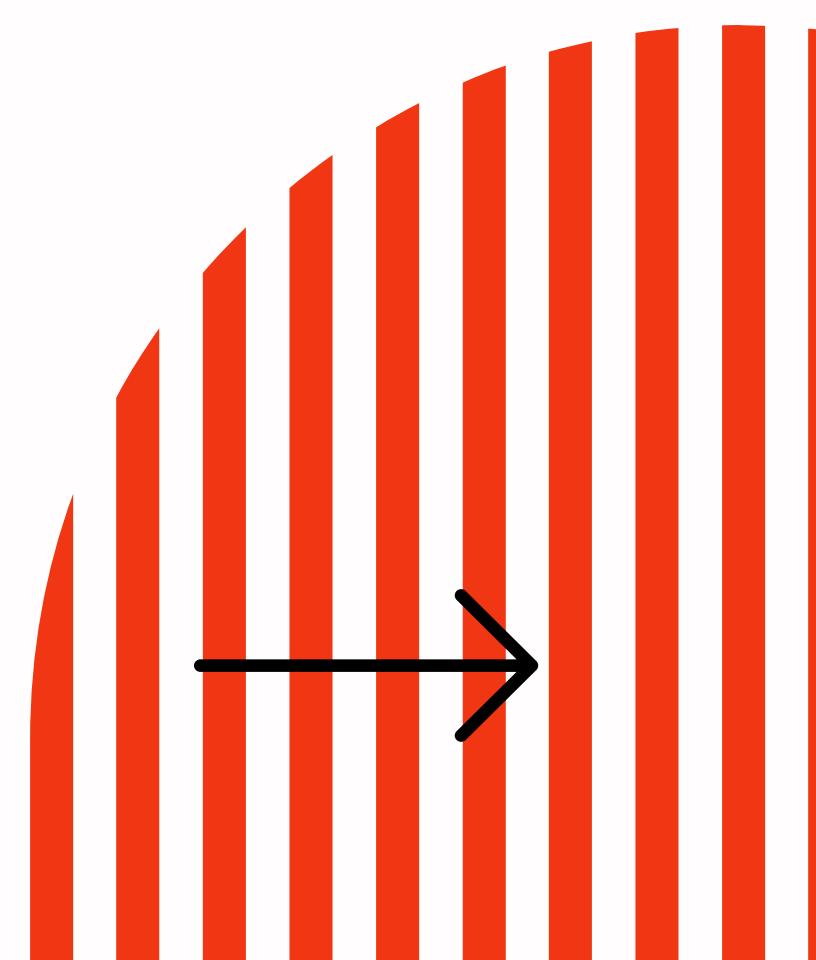
```
df.groupby('col1')['col2'].mean()
```

SQL

```
SELECT col1, AVG(col2)  
FROM table  
GROUP BY col1;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



29

Pivot table

Pandas

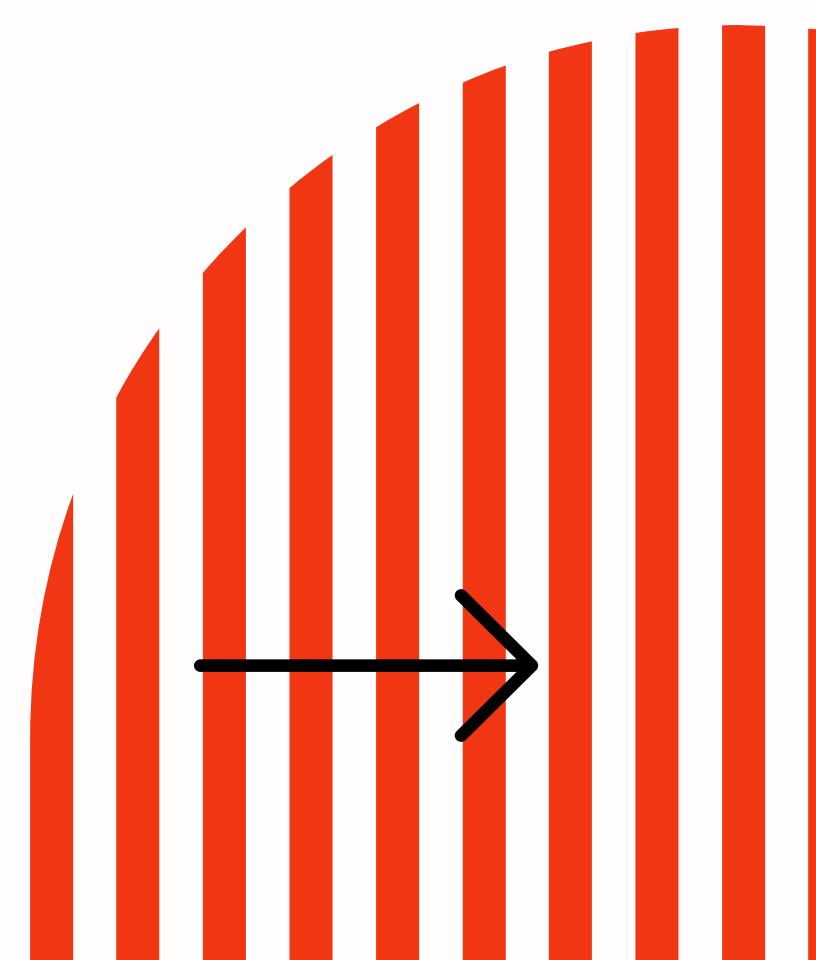
```
df.pivot_table(values='col1',  
index='col2', columns='val1')
```

SQL

```
SELECT * FROM table  
PIVOT (SUM(col1)  
FOR col2 IN (val1, val2));
```



Shwetank Singh
GritSetGrow - GSGLearn.com



30

Unpivot table

Pandas

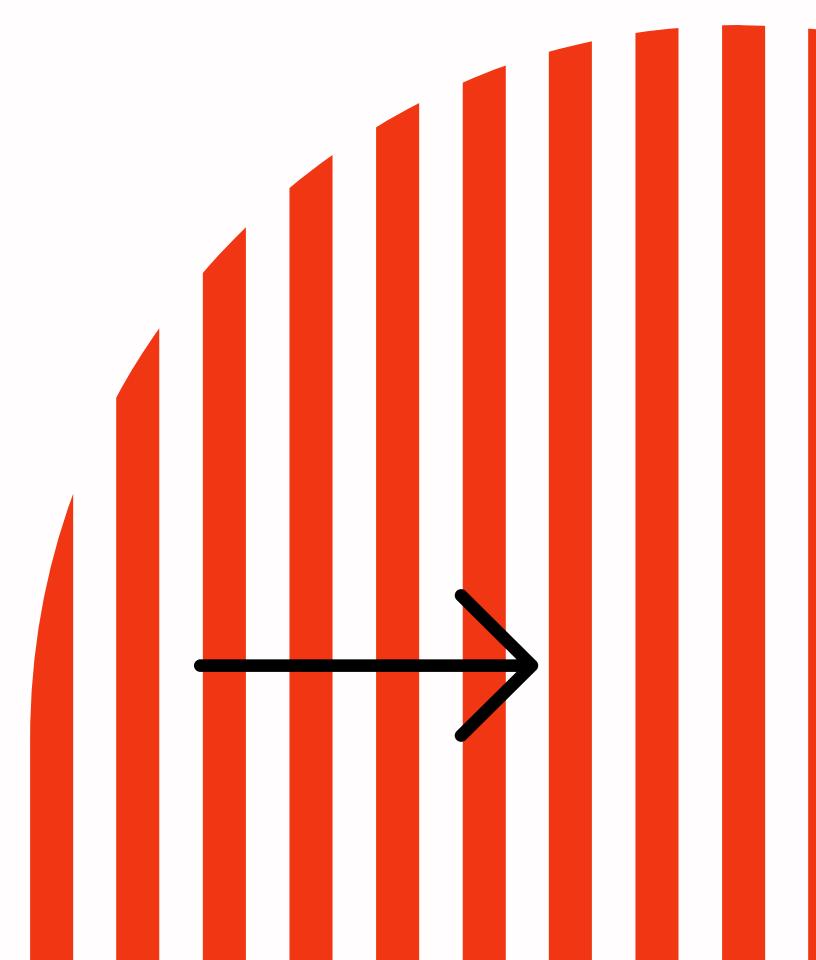
```
df.melt(id_vars=['id'],
         value_vars=['col1', 'col2'])
```

SQL

```
SELECT col1, col2
FROM table
UNPIVOT (col FOR val
IN (col1, col2));
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Calculate cumulative sum

Q1
31

Pandas

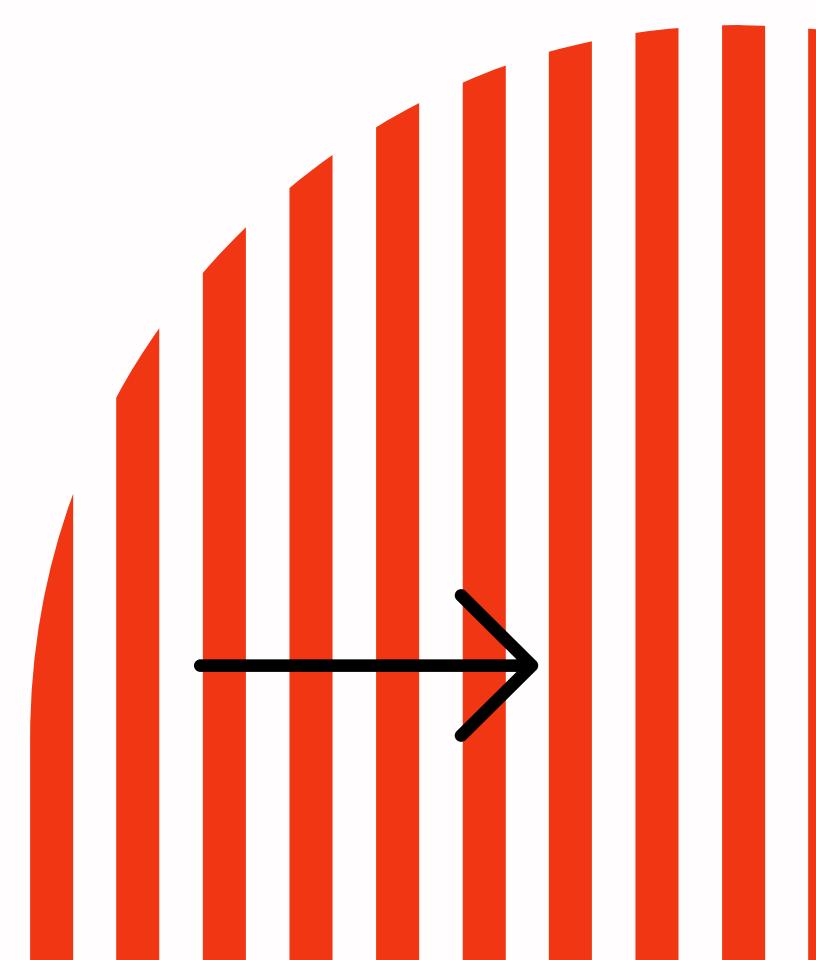
```
df['col2'].cumsum()
```

SQL

```
SELECT col1, SUM(col2)
OVER (ORDER BY col1)
FROM table;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Calculate moving average

SQL

Pandas

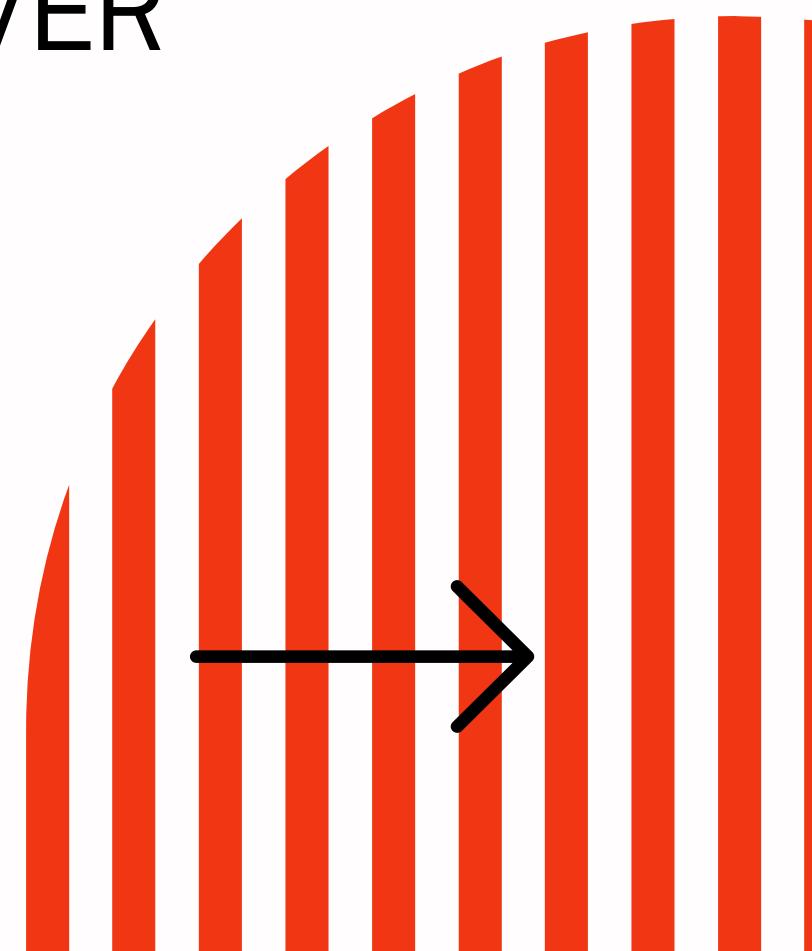
```
df['col2'].rolling(window=3).mean()
```

SQL

```
SELECT col1, AVG(col2) OVER  
(ORDER BY col1 ROWS  
BETWEEN 2 PRECEDING  
AND CURRENT ROW)  
FROM table;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Find the row with maximum value in a column

Pandas

00
99
33
00

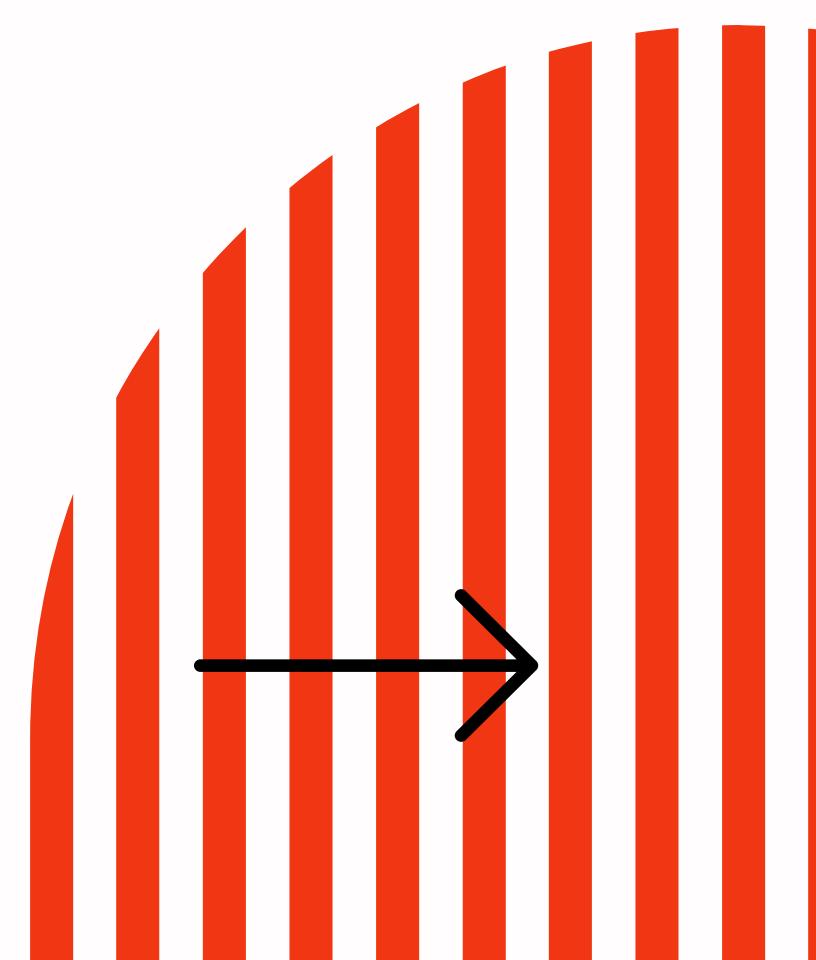
```
df.loc[df['col1'].idxmax()]
```

SQL

```
SELECT * FROM table  
ORDER BY col1 DESC  
LIMIT 1;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Find the row with minimum value in a column

Pandas

SQL

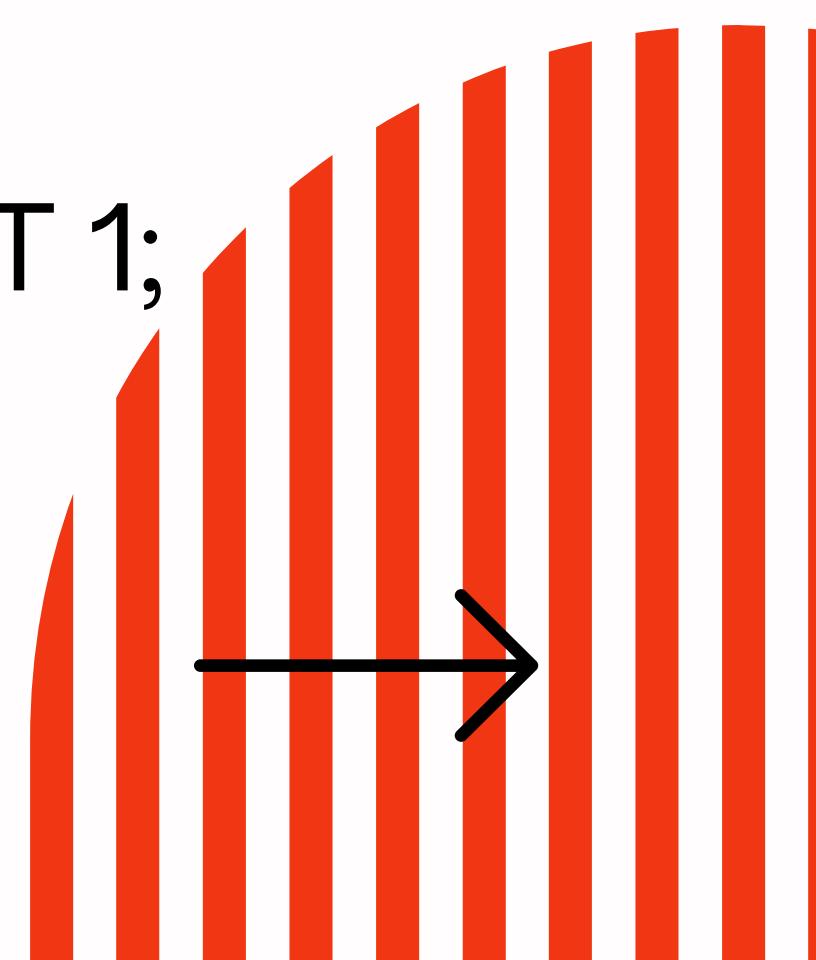
```
df.loc[df['col1'].idxmin()]
```

SQL

```
SELECT * FROM table  
ORDER BY col1 ASC LIMIT 1;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Drop duplicate rows

OF
30

Pandas

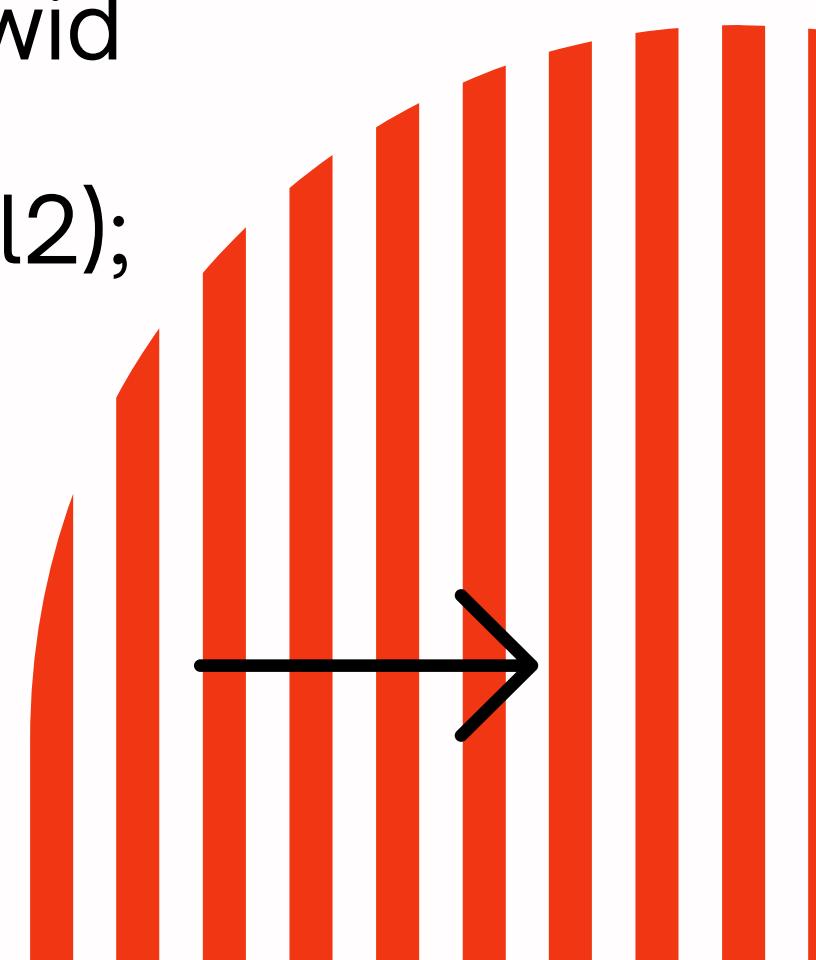
```
df.drop_duplicates()
```

SQL

```
DELETE FROM table WHERE rowid  
NOT IN (SELECT MIN(rowid)  
FROM table GROUP BY col1, col2);
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Calculate the length of a string

36

Pandas

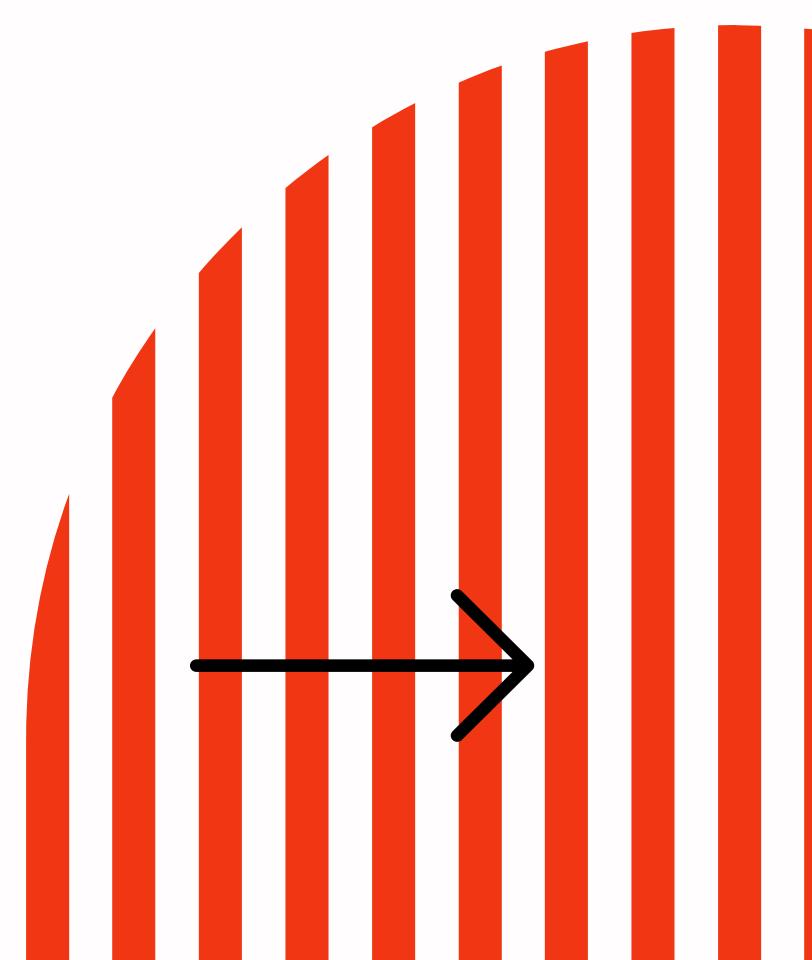
```
df['col1'].str.len()
```

SQL

```
SELECT LENGTH(col1)  
FROM table;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



31

Convert string to uppercase

Pandas

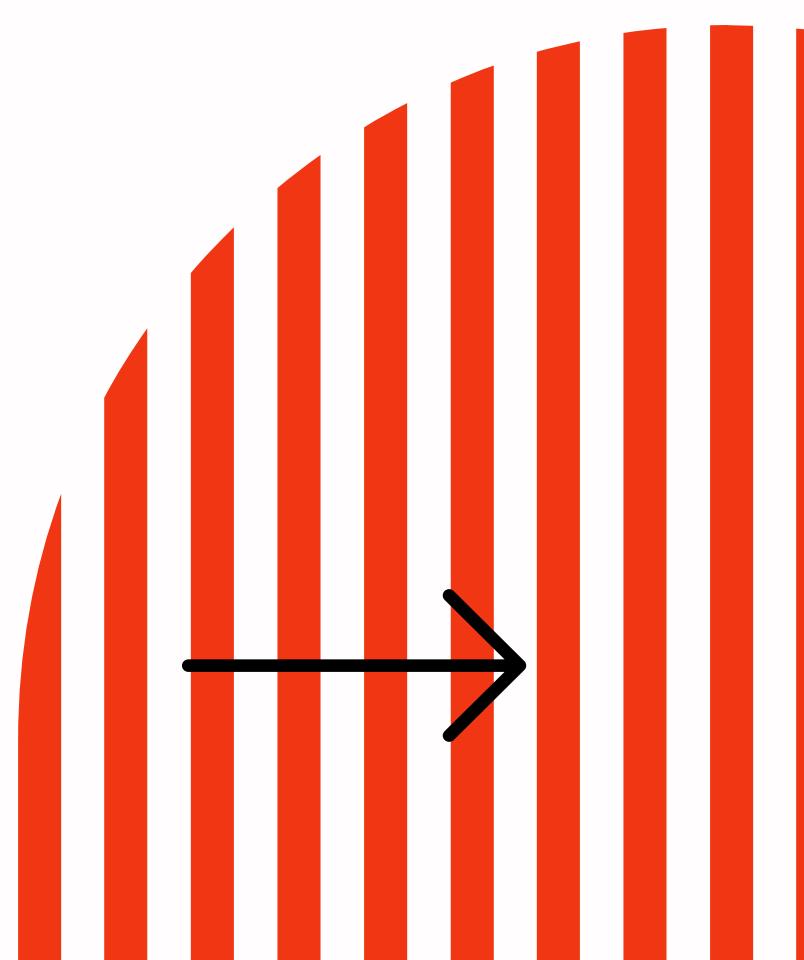
```
df['col1'].str.upper()
```

SQL

```
SELECT UPPER(col1)  
FROM table;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



38

Convert string to lowercase

Pandas

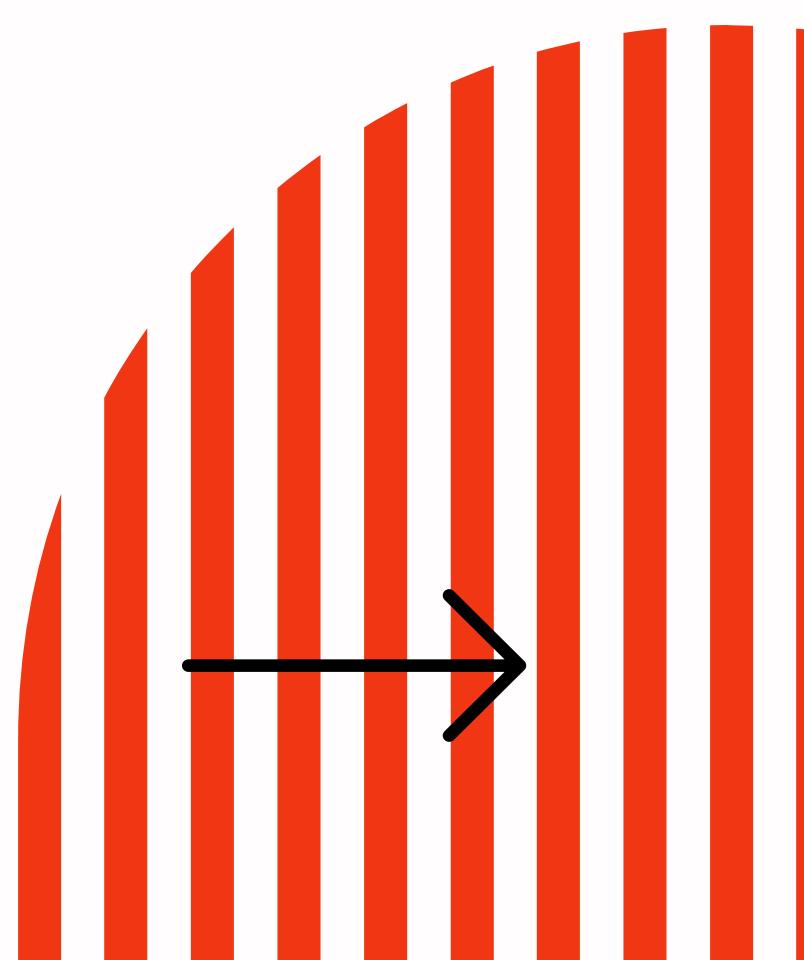
```
df['col1'].str.lower()
```

SQL

```
SELECT LOWER(col1)  
FROM table;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Strip whitespace from string

Pandas

```
df['col1'].str.strip()
```

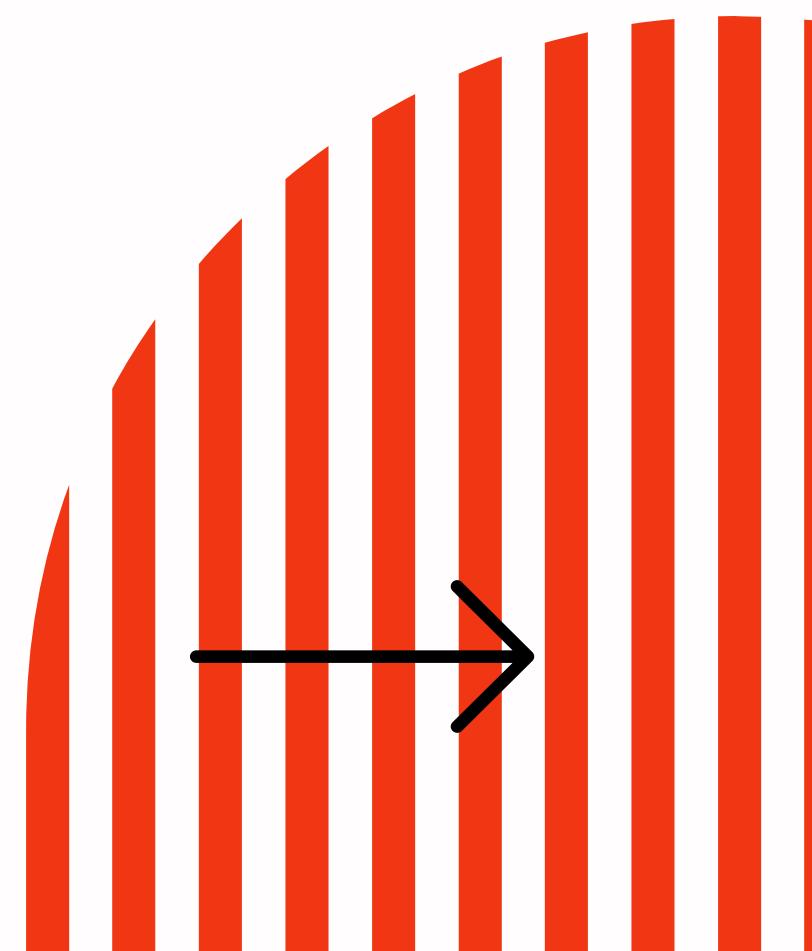
SQL

```
SELECT TRIM(col1)  
FROM table;
```



Shwetank Singh
GritSetGrow - GSGLearn.com

00
39
00
39



Replace a substring

40

Pandas

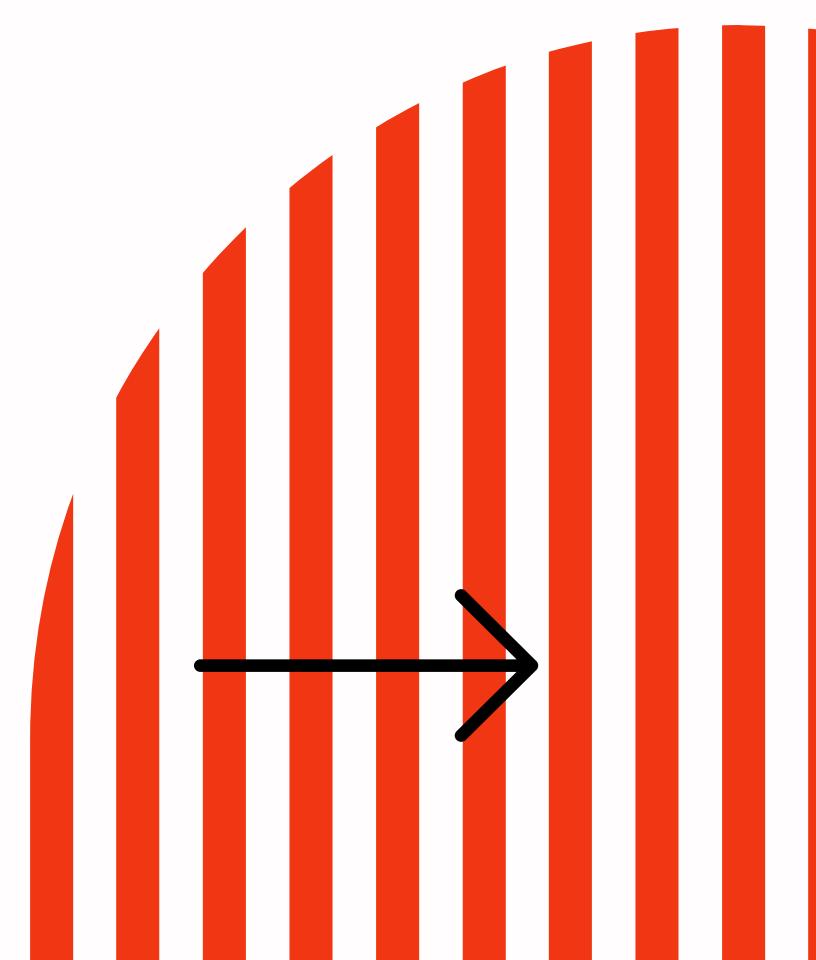
```
df['col1'].str.replace('old',  
'new')
```

SQL

```
SELECT REPLACE(col1,  
'old', 'new')  
FROM table;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Check for null values

Pandas

```
df[df['col1'].isnull()]
```

SQL

```
SELECT * FROM table  
WHERE col1 IS NULL;
```



Shwetank Singh
GritSetGrow - GSGLearn.com

Check for non-null values

Pandas

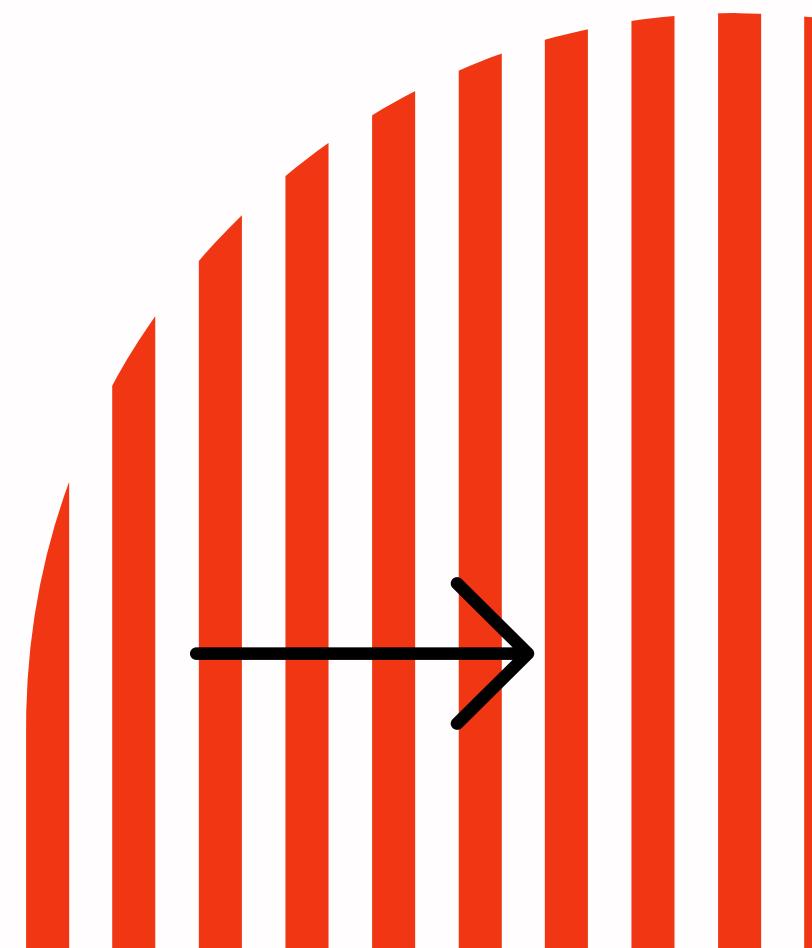
```
df[df['col1'].notnull()]
```

SQL

```
SELECT * FROM table  
WHERE col1 IS NOT  
NULL;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Conditional column based on other columns

10
43

Pandas

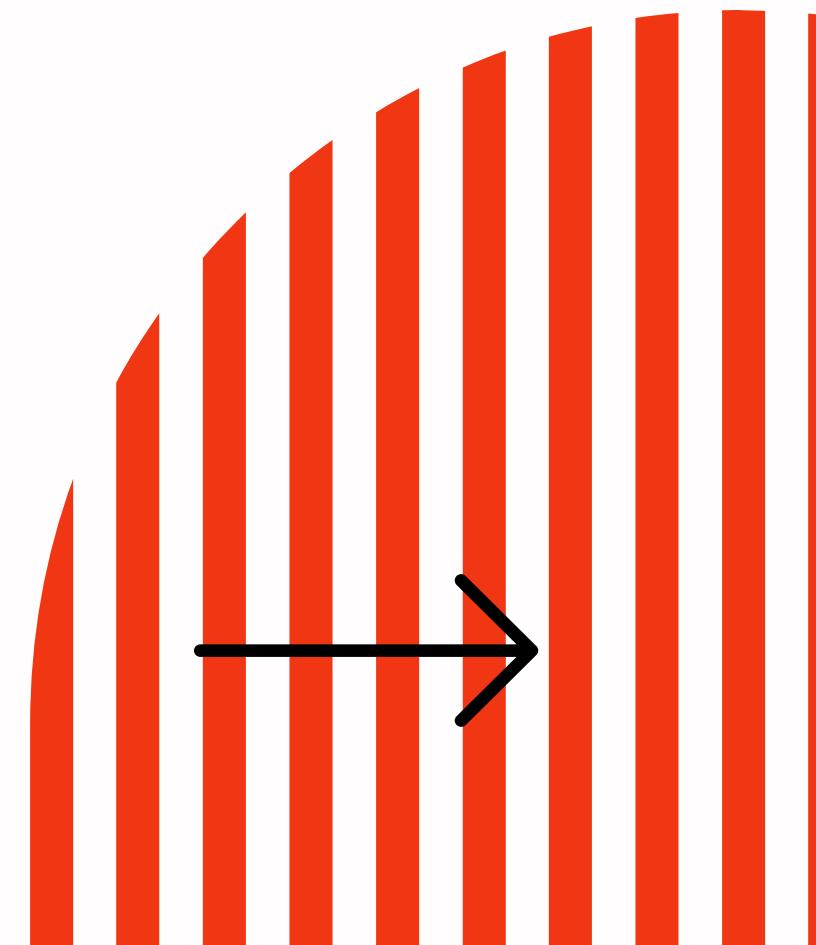
```
df['col3'] =  
df['col2'].apply(lambda x:  
'High' if x > 10 else 'Low')
```

SQL

```
SELECT col1, CASE  
WHEN col2 > 10 THEN  
'High' ELSE 'Low' END AS  
col3 FROM table;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Drop rows with null values

44

Pandas

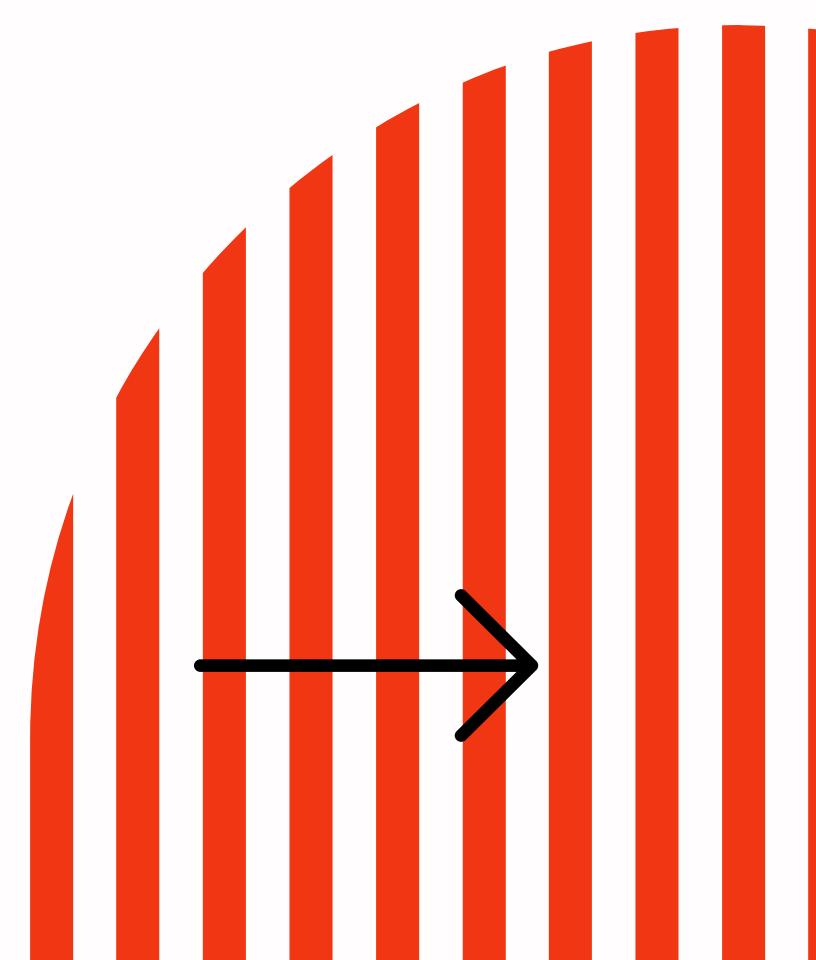
```
df.dropna()
```

SQL

```
DELETE FROM table  
WHERE col1 IS NULL;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Create new column from existing columns

Pandas

```
df['col3'] = df['col1'] +  
df['col2']
```

SQL

```
SELECT col1, col2, (col1  
+ col2) AS col3  
FROM table;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Calculate percentage of a column

40

Pandas

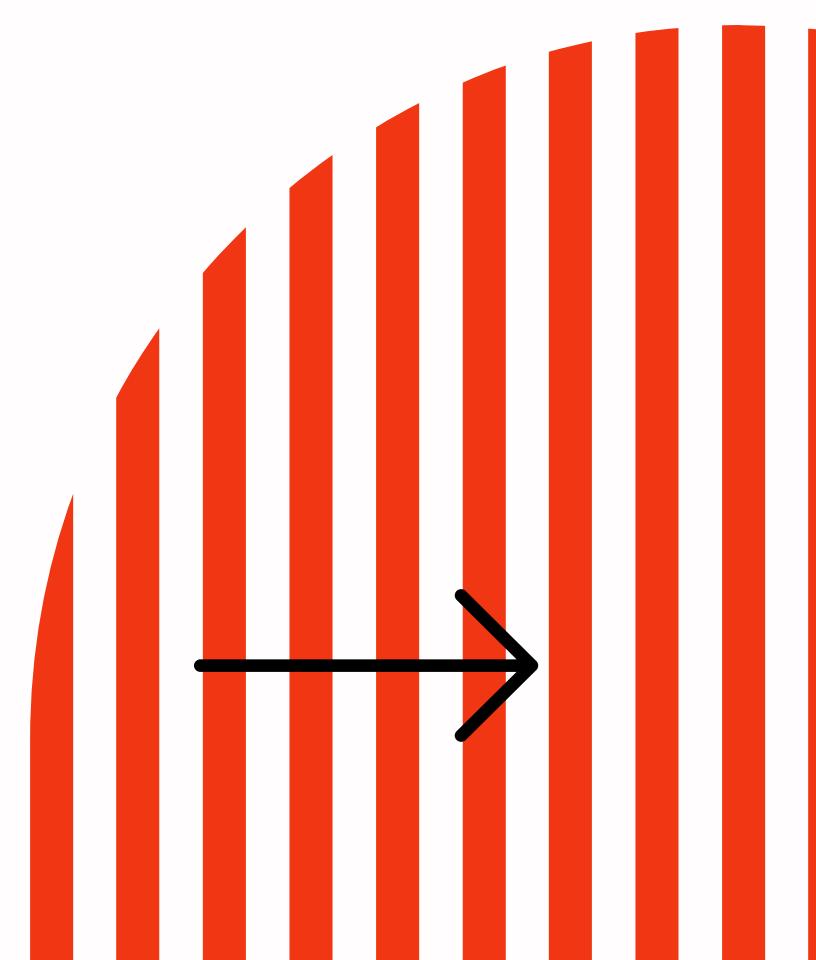
```
df['percentage'] =  
df['col1'] /  
df['col1'].sum() * 100
```

SQL

```
SELECT col1, (col1 /  
SUM(col1) OVER()) * 100  
AS percentage  
FROM table;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Insert a new row



Pandas

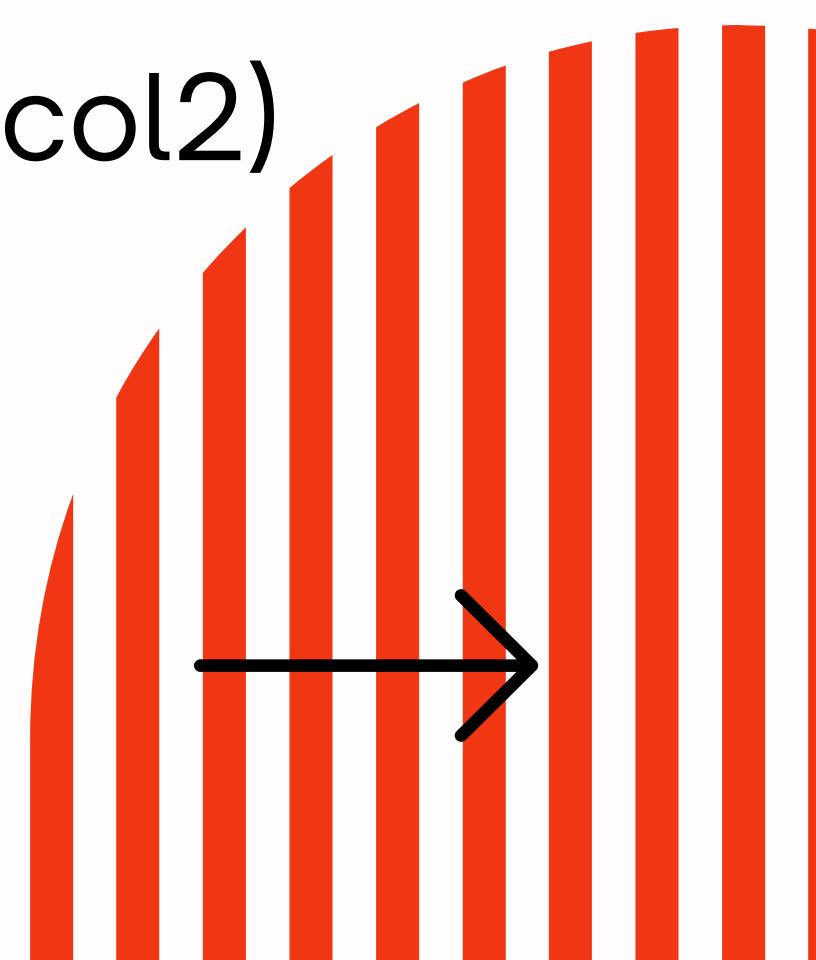
```
df.loc[len(df)] = [val1, val2]
```

SQL

```
INSERT INTO table (col1, col2)  
VALUES (val1, val2);
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Update a row

48

Pandas

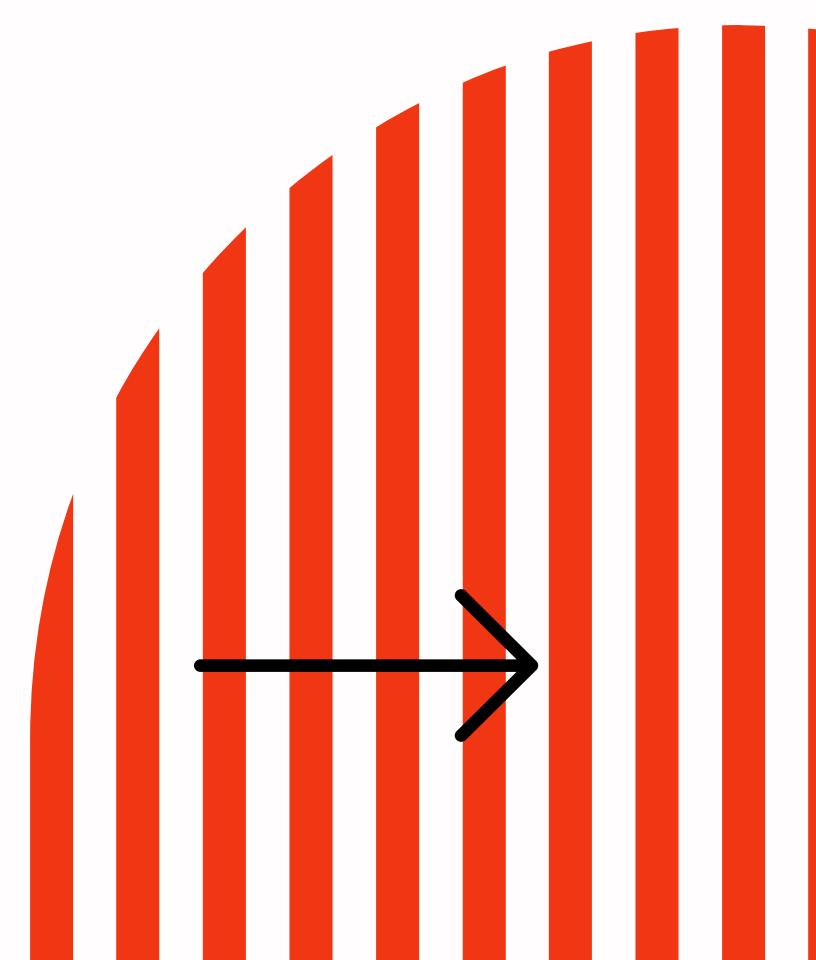
```
df.loc[df['col2'] == val2, 'col1'] = val1
```

SQL

```
UPDATE table  
SET col1 = val1 WHERE  
col2 = val2;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



THANK
you