Day 77- 90 days of Analytics : Missing Values in Pandas

In today's video, we looked some ways we can handle missing values with Pandas

The following were mentioned

-One way to deal with empty cells is to remove rows that contain empty cells. This is usually OK, since data sets can be very big, and removing a few rows will not have a big impact on the result. We use the **dropna()** method

```
import pandas as pd
df = pd.read_csv('data.csv')
new_df = df.dropna()
print(new_df)
```

-We should note that by default, the dropna() method returns a new DataFrame, and will not change the original.

-If we want to change the original DataFrame, we use the inplace = True argument. `df.dropna(inplace = True)`

-The **thresh** property of the fillna() method permits us to delete rows with specific a number of missing values per column.

-Another way of dealing with empty cells is to insert a new value instead. This way we do not have to delete entire rows just because of some empty cells. The **fillna()** method allows us to replace empty cells with a value

```
import pandas as pd
df = pd.read_csv('data_store.csv')
new_df = df.fillna(130)
print(new_df)
```

-To only replace empty values for one column, specify the ***column name*** for the DataFrame. Example

```
import pandas as pd
df = pd.read_csv('data_store.csv')
new_df = df["units_sold"].fillna(0)
print(new_df)
```

-We can replace empty values in various columns with different values by passing a dictionary to the fillna() method. Example

```
new_df = df.fillna({
    'visitors':0,
    'units_sold':0,
    'sex':"Not provided"
})
print(new_df)
```

-The replace() method permits us to replace specific values in a data frame. Example

```
import pandas as pd
import numpy as np
df = pd.read_csv('store_data2.csv')
df2 = df.replace(111111,value=np.nan)
print(df2)
```

Link to the YouTube Recording: https://www.youtube.com/watch?v=L1w5QeoOlGc

#90daysofanalytics #community #dataanalysis #dataanalyst #microsoft #msexcel #SQL #powerbi #pythonprogramming #numpy