

Generalized Template Files for Dropping or Deleting any columns or rows

```
class dropColumns(object):  
    def removeColByName(self,source_df,colName):  
        source_df.drop(columns=colName,axis=1,inplace =True)  
        return source_df  
  
    def removeMultipleColsByName(self,source_df,colName):  
        source_df.drop(columns=colName,axis=1,inplace =True)  
        return source_df  
  
    def removeSingleColByIndex(self,source_df,indexNum):  
        source_df.drop(source_df.columns[[indexNum]],inplace =True,axis = 1)  
        return source_df  
  
    def removeMultiColByIndex(self,source_df,indexNum):  
        source_df.drop(source_df.columns[[indexNum]],inplace =True,axis = 1)  
        return source_df  
  
    def removeColumnsHavingNulls(self,source_df,thresholdVal=0.95):  
        null_percentage = source_df.isnull().sum()/source_df.shape[0]  
        col_to_drop = null_percentage[null_percentage>=thresholdVal].keys()  
        output_df = source_df.drop(col_to_drop, axis=1)  
        return output_df
```

1. Dropping Columns By Given Column Name:

```
obj = dropColumns() // Creating Object  
obj.removeColByName(source_df, 'Name')
```

Adding pandas data frame in first parameter and column name in single inverted commas.

2. Dropping Columns By Given Column Name:

```
obj = dropColumns() // Creating Object  
obj.removeMultipleColsByName (source_df, ['ID','Role'])
```

Adding pandas data frame in first parameter and different column names in a list with single inverted commas.

3. Dropping Columns By Given Column Index:

```
obj = dropColumns() // Creating Object  
obj.removeSingleColByIndex (source_df,0)
```

Adding pandas data frame in first parameter and column index in second

4. Dropping Columns By Given Column Index:

```
obj = dropColumns() // Creating Object  
obj.removeMultiColByIndex (source_df , [0,1])
```

Adding pandas data frame in first parameter and multiple column index in a list in second column.

5. Dropping Columns By Given Condition (For example Nulls):

```
obj = dropColumns() // Creating Object  
obj.removeColumnsHavingNulls (source_df , thresholdVal = 0.4)
```

Adding pandas data frame in first parameter and threshold value in second, this threshold value is the value through which we will remove our entire column. So for example if there are columns having equal to or more than 40 percent null values than this function will remove those columns entirely.

Default threshold value is 95 percent.

Dropping Rows by Index and by any other condition

```
class dropRows(object):

    def removeSingleRowByIndex(self,source_df,indexNum):
        source_df_RowRemoved = source_df.drop(index=indexNum,axis = 0)
        return source_df_RowRemoved

    def removeMultiRowsByIndex(self,source_df,indexNum):
        source_df_RowRemoved = source_df.drop(index=indexNum,axis = 0)
        return source_df_RowRemoved

    def removeRowsWithAtleastOneNull(self,source_df):
        source_df_RowRemoved = source_df.dropna()
        return source_df_RowRemoved

    def removeRowsWithAllNulls(self,source_df):
        source_df_RowRemoved = source_df.dropna(how='all')
        return source_df_RowRemoved

    def removeSubsetOrRowsWithNulls(self,source_df,cols):
        source_df_RowRemoved = source_df.dropna(subset=cols)
        return source_df_RowRemoved

    def removeSubsetAndRowsWithNulls(self,source_df,cols):
        source_df_RowRemoved = source_df.dropna(subset=cols, how='all')
        return source_df_RowRemoved

    def removeRowsWithThresh(self,source_df,threshold=0.4):
        source_df_RowRemoved = source_df.dropna(thresh=threshold)
        return source_df_RowRemoved
```

1. Dropping Row By Given Index:

```
df_ColRemoved = obj.removeSingleRowByIndex(df_titanic,2)
```

Provide pandas df in first param and index number in second param

2. Dropping Multiple Rows By Given Index:

```
df_ColRemoved = obj.removeSingleRowByIndex(df_titanic,[1,3])
```

Provide pandas df in first param and multiple index number as list in second param

3. Dropping Rows according to the condition:

```
## Lets remove rows having atleast one null value
```

```
df_ColRemoved = obj.removeRowsWithAtleastOneNull(df_titanic)
```

```
## drop all the rows whose columns' values are all null
```

```
df_ColRemoved = obj.removeRowsWithAllNulls(df_titanic)
```

```
## Lets remove rows having subsets, like if col A OR colB has nulls then  
remove
```

```
df_ColRemoved =
```

```
obj.removeSubsetOrRowsWithNulls(df_titanic,['Age','Embarked'])
```

```
## Lets remove rows having subsets, like if col A AND colB has nulls then  
remove
```

```
df_ColRemoved =
```

```
obj.removeSubsetAndRowsWithNulls(df_titanic,['Cabin','Age'])
```

```
## Lets remove rows with some kind of threshold to remove
## For instance, if you want to drop all the columns that have more than
one null values, then you need to specify thresh to be len(df.columns) — 1
```

```
lengthOfDataset = len(df_titanic.columns)
df_ColRemoved =
obj.removeRowsWithThresh(df_titanic,threshold=(lengthOfDataset-1))
```

```
*****
*****
*****THE*END*****
*****
*****
```