```
import pandas as pd
```

```
#Loading the data set
df = pd.read_excel("/content/data.xlsx")
df.head()
```

	patient_id	Age	Sex	ECOG PS	Smoking PY	Smoking Status	Ds Site	Subsite	T	N	...	Cause of Death	Local	Date Local	Regiona
0	RADCURE-0005	62.6	Female	ECOG 0	50	Ex-smoker	Oropharynx	post wall	T4b	N2c	...	Other Cause	NaN	NaT	NaN
1	RADCURE-0006	87.3	Male	ECOG 2	25	Ex-smoker	Larynx	Glottis	T1b	N0	...	Other Cause	NaN	NaT	NaN
2	RADCURE-0007	49.9	Male	ECOG 1	15	Ex-smoker	Oropharynx	Tonsil	T3	N2b	...	NaN	NaN	NaT	NaN
3	RADCURE-0009	72.3	Male	ECOG 1	30	Ex-smoker	Unknown	NaN	T0	N2c	...	NaN	NaN	NaT	NaN
4	RADCURE-0010	59.7	Female	ECOG 0	0	Non-smoker	Oropharynx	Tonsillar Fossa	T4b	N0	...	NaN	NaN	NaT	NaN

5 rows × 34 columns

```
df.columns
```

```
Index(['patient_id', 'Age', 'Sex', 'ECOG PS', 'Smoking PY', 'Smoking Status',
       'Ds Site', 'Subsite', 'T', 'N', 'M ', 'Stage', 'Path', 'HPV',
       'Tx Modality', 'Chemo? ', 'RT Start', 'Dose', 'Fx', 'RT Tech',
       'Last FU', 'Status', 'Length FU', 'Date of Death', 'Cause of Death',
       'Local', 'Date Local', 'Regional', 'Date Regional', 'Distant',
       'Date Distant', '2nd Ca', 'Date 2nd Ca', 'RADCURE-challenge'],
      dtype='object')
```

We can see that all the column names are not meaningful and have spaces in between. Lets name the columns properly.

```
df = df.rename(columns={'T': 'tumor_size',"Smoking PY":"annual_packs_smoked",
                        "Path":"diagnosis_type","Ds Site":"cancer_site",
                        "Subsite":"cancer_subsite","Metastasis Status":"metastasis_status",
                        "Tx Modality":"treatment_type","RT Tech":"radio_therapy_type","RT Start":
                        "Last FU":"last_follow_up","Date of Death":"date_of_death",
                        "Cause of Death":"cause_of_death","Smoking Status":"smoking_status"})
```

```
#Lets check for the null values in the dataset
print(df.shape)
print(df.isnull().sum())
```

```
(3346, 34)
patient_id                 0
Age                        0
Sex                        0
ECOG PS                    1
annual_packs_smoked        5
smoking_status             0
cancer_site                0
cancer_subsite           374
tumor_size                12
N                         13
M                         14
Stage                     27
diagnosis_type             0
HPV                     1629
treatment_type             0
Chemo?                     0
radio_therapy_startDt      0
Dose                       0
Fx                         0
radio_therapy_type         0
last_follow_up             0
Status                     0
Length FU                  0
date_of_death           2288
cause_of_death          2294
Local                   2966
Date Local              2966
Regional                3157
Date Regional           3157
Distant                 2933
Date Distant            2933
2nd Ca                  2905
Date 2nd Ca             2907
RADCURE-challenge          0
dtype: int64
```

```
#Lets fix the null values.
df['smoking_status'].value_counts()
```

```
Ex-smoker      1290
Current        1139
Non-smoker      871
unknown          45
```

```
non-drinker       1
Name: smoking_status, dtype: int64
```

```
# we can see that there data aout drinking in the smoking status , hence dropping it
df = df[df['smoking_status'] != 'non-drinker']
```

```
# Replace NA values in HPV column with Not tested
df['HPV'].fillna('No', inplace=True)
```

```
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['HPV'].fillna('No', inplace=True)
```

```
df['M '].value_counts()
```

```
M0    3327
MX       2
M1       2
Name: M , dtype: int64
```

```
#We can see that we have 99.8 % of the data is about M0 (Benign Tumor) we can drop
#the 4 rows which has MX and M1 and remove the entire column and add in the data
# description that everybody has M0(Benign Tumor).
df.drop('M ', axis=1, inplace=True)
```

```
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df.drop('M ', axis=1, inplace=True)
```

```
# Currently the number of packs smoked is na for people with smoking status unknown
#Replacing them with 0.There are 5 such values
df['annual_packs_smoked'].fillna(0, inplace=True)
```

```
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['annual_packs_smoked'].fillna(0, inplace=True)
```

```
df['cancer_subsite'].fillna('Unknown', inplace=True)
```

```
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['cancer_subsite'].fillna('Unknown', inplace=True)
```

```
# We have 2 columns, "Dead"-> which saves the status if they are alive or not and
#"date_of_death" stores the date when the person died.
#Combining these 2 column to reduce the null values as Alive person's date of death
# will be null
for i in df.index:
  if(df["Status"][i] == "Dead"):
    df["Status"][i] = df["date_of_death"][i]
```

```
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df["Status"][i] = df["date_of_death"][i]
```

```
df['cause_of_death'].fillna('Alive', inplace=True)
```

```
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['cause_of_death'].fillna('Alive', inplace=True)
```

```
df['Local'].fillna('No', inplace=True)
df['Regional'].fillna('No', inplace=True)
df['2nd Ca'].fillna('No', inplace=True)

df = df.rename(columns = {"2nd Ca":"2nd_cancer_site"})
```

```
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Local'].fillna('No', inplace=True)
<ipython-input-25-2c7d12baaeff>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Regional'].fillna('No', inplace=True)
<ipython-input-25-2c7d12baaeff>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['2nd Ca'].fillna('No', inplace=True)
```

```
df.isnull().sum()
```

```
patient_id                 0
Age                        0
Sex                        0
ECOG PS                    1
annual_packs_smoked        0
smoking_status             0
cancer_site                0
cancer_subsite             0
tumor_size                12
N                         13
Stage                     27
diagnosis_type             0
HPV                        0
treatment_type             0
Chemo?                     0
radio_therapy_startDt      0
Dose                       0
Fx                         0
radio_therapy_type         0
last_follow_up             0
Status                     0
Length FU                  0
date_of_death           2287
cause_of_death             0
Local                      0
Date Local              2966
Regional                   0
Date Regional           3156
Distant                 2932
Date Distant            2932
2nd_cancer_site            0
Date 2nd Ca             2906
RADCURE-challenge          0
dtype: int64
```

```
# we have 3346 rows , and in few columns we hav more that 80% of the data empty
df.drop(['Date Local','Date Regional','Distant','Date Distant','Date 2nd Ca','Chemo? ','RADCURE-c
```

```
file_path = 'dtsc_data.xlsx'

df.to_excel(file_path, index=False)
```