```
'''
Load meteorite_landing csv
update the year column to only contain the year
  - convert it to a numeric data type
  - create a new column indicaiting whether the meteorite was observed
falling before 1970
set the index to the id column
extract all the rows in IDs between 10 036 and 10 040

bonus : There's a data entry error in the year column, Can you find it
'''

from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import pandas as pd

df = pd.read_csv('/content/drive/MyDrive/Meteorite_Landings.csv')

df['year'] = pd.to_datetime(df['year'], errors='coerce').dt.year

df['fell_before_1970'] = (df['fall'] == 'Fell') & (df['year'] < 1970)

df = df.set_index('id')
df.index = pd.to_numeric(df.index, errors='coerce')
df = df.sort_index()

print("Rows with IDs between 10036 and 10040:")
display(df.loc[10036:10040])

print("Minimum year:", df['year'].min())
print("Maximum year:", df['year'].max())
print("\nSuspicious year entries:")
display(df.loc[(df['year'] < 1800) | (df['year'] > 2026), ['name',
'year']])
```

```
Rows with IDs between 10036 and 10040:

/tmp/ipython-input-2732244590.py:5: UserWarning: Could not infer
format, so each element will be parsed individually, falling back to
`dateutil`. To ensure parsing is consistent and as-expected, please
specify a format.
  df['year'] = pd.to_datetime(df['year'], errors='coerce').dt.year
```


```
{"summary":"{\n  \"name\": \"display(df\",\n  \"rows\": 4,\n
\"fields\": [\n    {\n      \"column\": \"id\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 1,\n
\"min\": 10036,\n        \"max\": 10039,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          10037,\n
```

```
10039,\n          10036\n        ],\n        \"semantic_type\": \"\",\
n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"name\",\n      \"properties\": {\n        \"dtype\":
\"string\",\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          \"Enon\",\n          \"Ensisheim\",\n
\"Enigma\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"nametype\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 1,\n        \"samples\":
[\n          \"Valid\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"recclass\",\n      \"properties\": {\n        \"dtype\":
\"string\",\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          \"Iron, ungrouped\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"mass (g)\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
62127.04696225308,\n        \"min\": 94.0,\n        \"max\":
127000.0,\n        \"num_unique_values\": 4,\n        \"samples\": [\n
763.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"fall\",\n      \"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 2,\n        \"samples\": [\n          \"Fell\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"year\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 50.63924696649164,\n        \"min\": 1883.0,\n
\"max\": 1974.0,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          1967.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"reclat\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
8.222662098219265,\n        \"min\": 30.3,\n        \"max\":
47.86667,\n        \"num_unique_values\": 4,\n        \"samples\": [\n
39.86667\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"reclong\",\n      \"properties\": {\n        \"dtype\": \"number\",\
n        \"std\": 91.75584875348396,\n        \"min\": -83.95,\n
\"max\": 109.5,\n        \"num_unique_values\": 4,\n
\"samples\": [\n          -83.95\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"GeoLocation\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 4,\n        \"samples\": [\n
\"(39.86667, -83.95)\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"fell_before_1970\",\n      \"properties\": {\n        \"dtype\":
\"boolean\",\n        \"num_unique_values\": 1,\n        \"samples\":
```

[\n   false\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}

```
Minimum year: 1688.0
Maximum year: 2101.0
```

```
Suspicious year entries:
```

{"summary":"{\n \"name\": \"display(df\",\n \"rows\": 40,\n
\"fields\": [\n   {\n   \"column\": \"id\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\": 10213,\n
\"min\": 425,\n   \"max\": 57150,\n   \"num_unique_values\":
40,\n   \"samples\": [\n   17994,\n   15446,\n
14757\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"name\",\n   \"properties\": {\n   \"dtype\": \"string\",\n
\"num_unique_values\": 40,\n   \"samples\": [\n
\"Ogi\",\n   \"Mauerkirchen\",\n   \"Luponnas\"\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"year\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\":
60.30919796328217,\n   \"min\": 1688.0,\n   \"max\":
2101.0,\n   \"num_unique_values\": 33,\n   \"samples\": [\n
1792.0,\n   1753.0,\n   1773.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   }\n]\n}","type":"dataframe"}