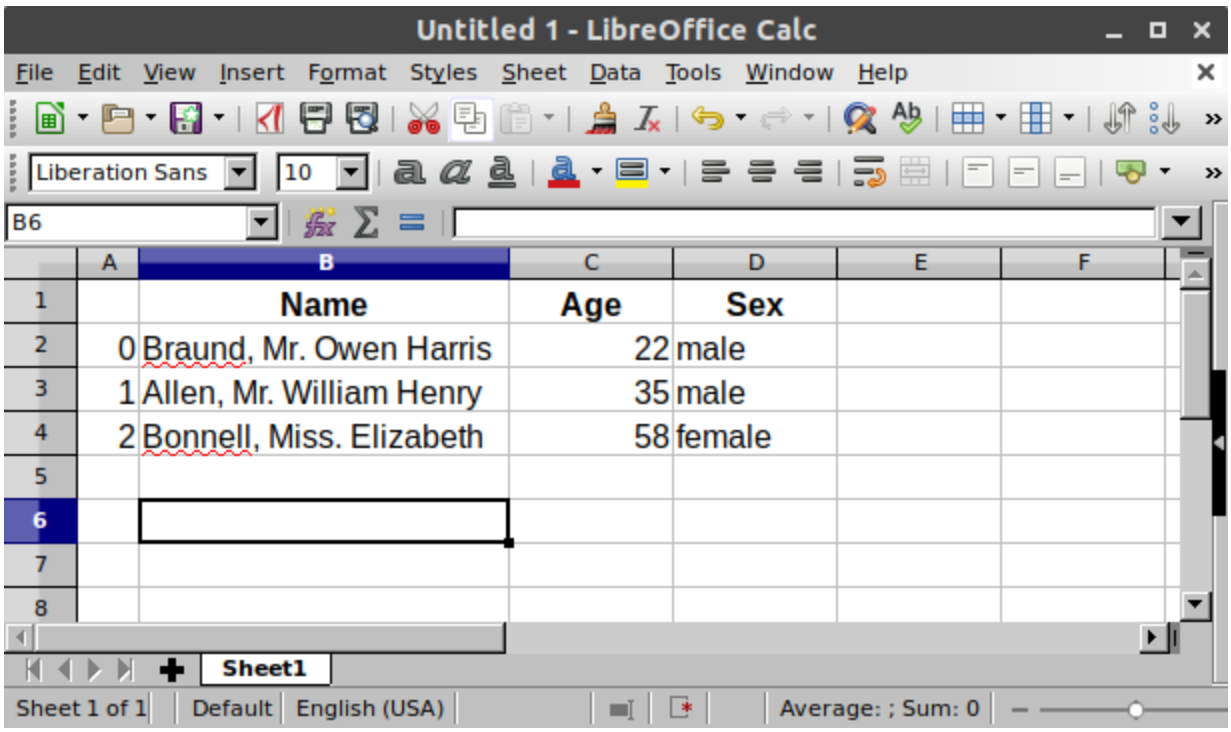
https://github.com/jakevdp/data-USstates/

```
import pandas as pd
df = pd.DataFrame(
    ...:     {
    ...:         "Name": [
    ...:             "Braund, Mr. Owen Harris",
    ...:             "Allen, Mr. William Henry",
    ...:             "Bonnell, Miss. Elizabeth",
    ...:         ],
    ...:         "Age": [22, 35, 58],
    ...:         "Sex": ["male", "male", "female"],
    ...:     }
    ...: )
    ...:
```

```
df
```




```
df["Age"]
ages = pd.Series([22, 35, 58], name="Age")

df["Age"].max()
ages.max()
df.describe()
```

****

Titanic Dataset

```
titanic = pd.read_csv("data/titanic.csv")

titanic.to_excel("titanic.xlsx", sheet_name="passengers", index=False)

titanic = pd.read_excel("titanic.xlsx", sheet_name="passengers")
```

```
titanic.dtypes

titanic.info()

titanic.head()


titanic["Name"].str.lower()


titanic["Name"].str.split(",")


titanic["Surname"] = titanic["Name"].str.split(",").str.get(0)


titanic["Surname"]


titanic["Name"].str.contains("Countess")


titanic[titanic["Name"].str.contains("Countess")]


titanic["Name"].str.len()


titanic["Name"].str.len().idxmax()


ages = titanic["Age"]


ages.head()


above_35 = titanic[titanic["Age"] > 35]


above_35.shape
class_23 = titanic[(titanic["Pclass"] == 2) | (titanic["Pclass"] == 3)]


class_23.head()


adult_names = titanic.loc[titanic["Age"] > 35, "Name"]

adult_names.head()


titanic.iloc[9:25, 2:5]
```

****

How to calculate summary statistics

```
titanic["Age"].mean()
titanic[["Age", "Fare"]].median()
```

```
titanic[["Age", "Fare"]].describe()

titanic.agg(
       {
           "Age": ["min", "max", "median", "skew"],
           "Fare": ["min", "max", "median", "mean"],
       }
   )

titanic[["Sex", "Age"]].groupby("Sex").mean()
titanic.groupby("Sex").mean(numeric_only=True)
titanic.groupby("Sex")["Age"].mean()
titanic.groupby(["Sex", "Pclass"])["Fare"].mean()
```

No. of records by category

```
titanic["Pclass"].value_counts()

titanic.groupby("Pclass")["Pclass"].count()
```

Reshaping layout of tabes

```
titanic.sort_values(by="Age").head()
titanic.sort_values(by=['Pclass', 'Age'], ascending=False).head()

************************************************************

no2 = air_quality[air_quality["parameter"] == "no2"]
no2_subset = no2.sort_index().groupby(["location"]).head(2)
no2_subset
no2_subset.pivot(columns="location", values="value")
no2.head()
no2.pivot(columns="location", values="value").plot()

*********************************************************

air_quality = pd.read_csv("data/air_quality_no2.csv", index_col=0, parse_dates=True)
air_quality.head()

air_quality.plot()

air_quality["station_paris"].plot()



air_quality.plot.scatter(x="station_london", y="station_paris", alpha=0.5)
```

```
plt.show()


air_quality.plot.box()

axs = air_quality.plot.area(figsize=(12, 4), subplots=True)

plt.show()
fig, axs = plt.subplots(figsize=(12, 4))
air_quality.plot.area(ax=axs)


axs.set_ylabel("NO$_2$ concentration")
fig.savefig("no2_concentrations.png")
plt.show()
```

```
fig, axs = plt.subplots(figsize=(12, 4))        # Create an empty Matplotlib Figure and
Axes
air_quality.plot.area(ax=axs)                   # Use pandas to put the area plot on the
prepared Figure/Axes
axs.set_ylabel("NO$_2$ concentration")          # Do any Matplotlib customization you
like
fig.savefig("no2_concentrations.png")           # Save the Figure/Axes using the existing
Matplotlib method.
plt.show()                                      # Display the plot
```

****

```
air_quality = pd.read_csv("data/air_quality_no2.csv", index_col=0, parse_dates=True)
air_quality.head()


air_quality["london_mg_per_cubic"] = air_quality["station_london"] * 1.882
air_quality.head()
air_quality["ratio_paris_antwerp"] = (air_quality["station_paris"] /
air_quality["station_antwerp"]
)
air_quality.head()
air_quality_renamed = air_quality.rename(
   ...:     columns={
   ...:         "station_antwerp": "BETR801",
   ...:         "station_paris": "FR04014",
   ...:         "station_london": "London Westminster",
   ...:     }
   ...: )


air_quality_renamed = air_quality_renamed.rename(columns=str.lower)
```