

Instituto de Computação (IC) Unicamp



Escola de Extensão da Unicamp



UNICAMP

Universidade Estadual de Campinas

INF 331 - Componentização e Reuso de Software
Prof. André Santanché

Aula 01

Orange + notebook

Aluno

Douglas Sermarini - 147730

Repositório : <https://github.com/Douglas019BR/INF331-lab02>

Laboratório 1 – Tarefa 1

Arquivo: [Aula2-lab1-atv1.ipynb](#)

```
[11]: %%plantuml

@startuml

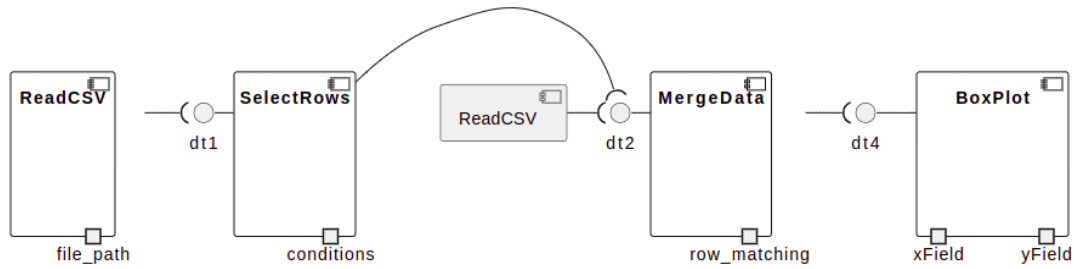
!procedure $property($name)
  portout $name
!endprocedure

component "ReadCSV" as csv1 {
  $property(file_path)
}
component "ReadCSV" as csv2{
  $property(file_path)
}
component SelectRows {
  $property(conditions)
}
component MergeData {
  $property(row_matching)
}
component BoxPlot {
  $property(xField)
  $property(yField)
}

[csv1] --> dt1
dt1 --> [SelectRows]
[SelectRows] --> dt2
[csv2] --> dt2
dt2 --> [MergeData]
[MergeData] --> dt4
dt4 --> [BoxPlot]

@enduml
```

[11]:



Laboratório 2 – Tarefa 1

Arquivo: [aula2-lab2-atv1.ipynb](#)

```
[17]: import matplotlib.pyplot as plt

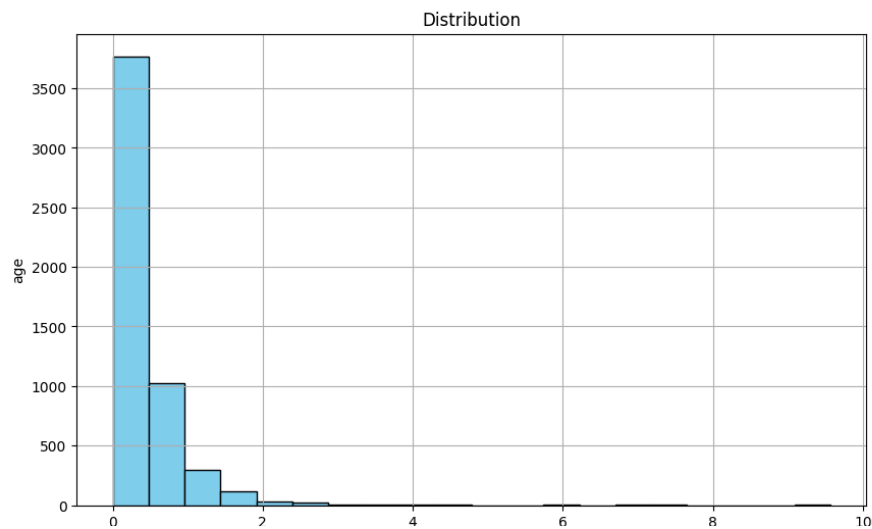
def histogram_plot(data, field, bins, title, y_label):
    plt.figure(figsize=(10, 6))

    # Plotting histogram
    plt.hist(data[field], bins=bins, color='skyblue', edgecolor='black')

    plt.xlabel(field)
    plt.ylabel(y_label)
    plt.title(title)

    plt.grid(True)
    plt.show()
```

```
[19]: histogram_plot(df, field='intake_bw', bins=20, title='Distribution', y_label='age')
```



Laboratório 3 – Tarefa 1

Arquivo: [aula2-lab3-atv1.ipynb](#)

```
[13]: import matplotlib.pyplot as plt
```

```
class HistogramPlot:
    def __init__(self, df, field, bins, title, y_label):
        self.df = df
        self.field = field
        self.bins = bins
        self.title = title
        self.y_label = y_label

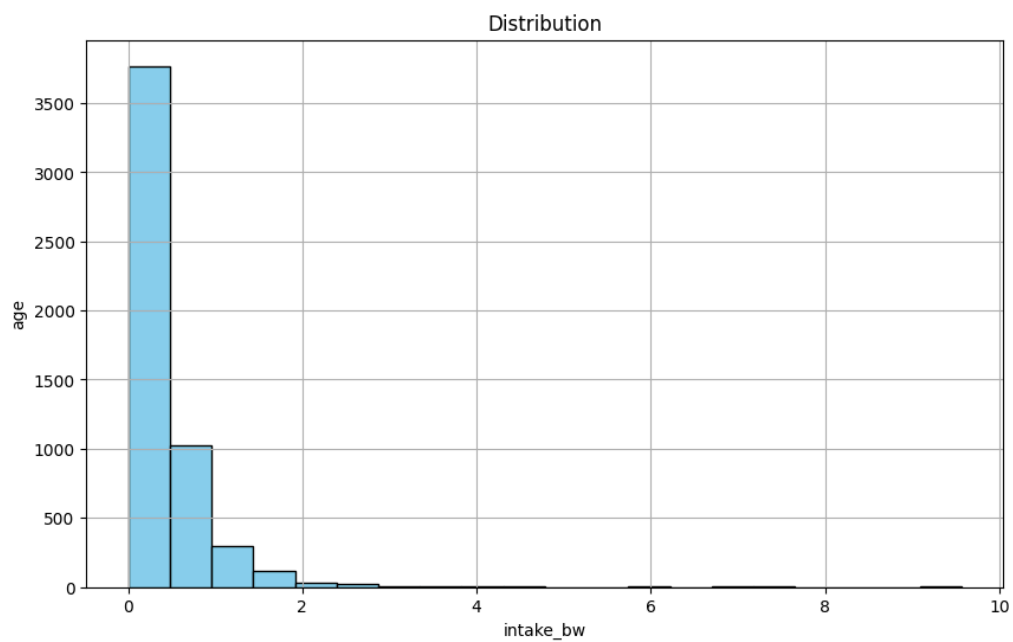
    def plot(self):
        plt.figure(figsize=(10, 6))
        plt.hist(self.df[self.field], bins=self.bins, color='skyblue', edgecolor='black')

        # Adiciona os rótulos e título
        plt.xlabel(self.field)
        plt.ylabel(self.y_label)
        plt.title(self.title)

        # Exibe a grade
        plt.grid(True)

        # Mostra o gráfico
        plt.show()
```

```
[14]: distribution = HistogramPlot(df, 'intake_bw', bins=20, title='Distribution', y_label='age')
distribution.plot()
```

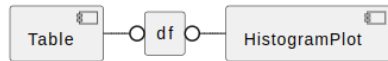


[15]: %%plantuml

```
@startuml
rectangle df
[Table] --> df
df --> [HistogramPlot]
@enduml
```

Writing output for /home/jovyan/notebooks/componentization/49853ae8-78c3-4dbc-b113-f0d88da63040.uml to 49853ae8-78c3-4dbc-b113-f0d88da63040.svg

[15]:



Laboratório 4 – Tarefa 1

Arquivo: [aula2-lab4-atv1.ipynb](#)

[9]: import matplotlib.pyplot as plt

```
class HistogramPlot:
    def __init__(self, field, bins, title, y_label):
        self.field = field
        self.bins = bins
        self.title = title
        self.y_label = y_label
        self.df = None # Inicialmente sem dados

    def plot(self):
        if self.df is None:
            raise ValueError("Data not set. Use the 'update' method to provide data.")

        # Cria o gráfico de histograma
        plt.figure(figsize=(10, 6))
        plt.hist(self.df[self.field], bins=self.bins, color='skyblue', edgecolor='black')

        # Adiciona os rótulos e título
        plt.xlabel(self.field)
        plt.ylabel(self.y_label)
        plt.title(self.title)

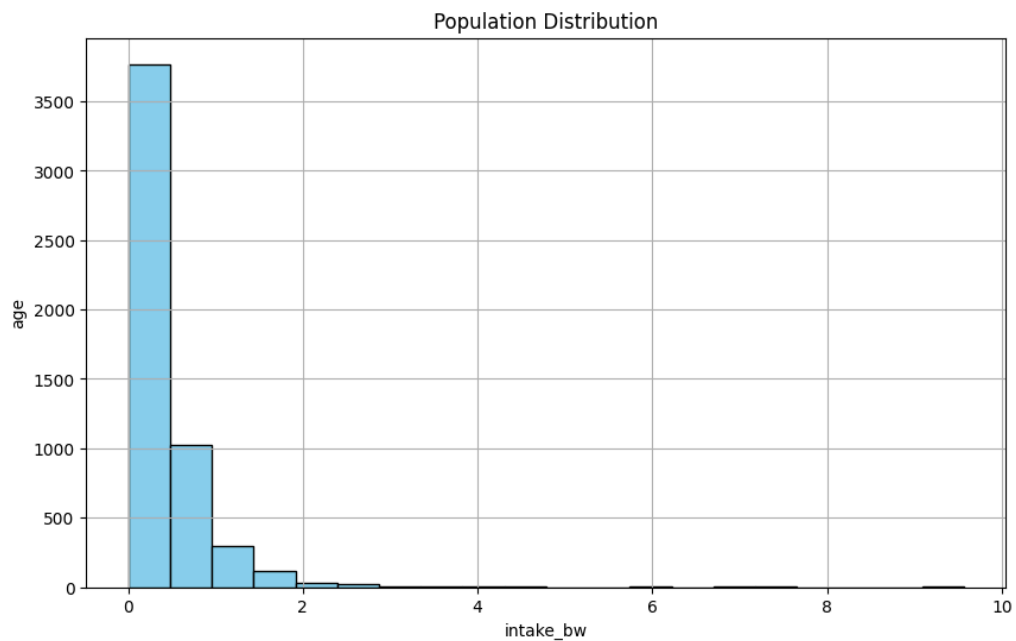
        # Exibe a grade
        plt.grid(True)

        # Mostra o gráfico
        plt.show()

    # Interface para atualizar os dados e replotar
    def update(self, new_data):
        self.df = new_data
        self.plot()
```

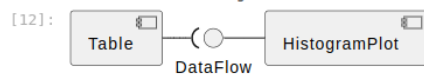
```
[11]: table_intake = Table('intake-person-demo(potato-chips).csv')
hist_plot = HistogramPlot(field='intake_bw', bins=20, title='Population Distribution', y_label='age')

# Connect the ScatterPlot to the DataLoader
table_intake.connect(hist_plot)
```



```
[12]: %%plantuml
@startuml
[Table] -- DataFlow --> [HistogramPlot]
enduml

Writing output for /home/jovyan/notebooks/componentization/c59db57e-336d-44bb-b38d-d82f8101c1bd.uml to c59db57e-336d-44bb-b38d-d82f8101c1bd.svg
```



Laboratório 5 – Tarefa 1

Arquivo: [aula2-lab5-atv1.ipynb](#)

```
[9]: import matplotlib.pyplot as plt
```

```
class HistogramPlot:
    def __init__(self, field=None, bins=None, title=None, y_label=None):
        self._field = field
        self._bins = bins
        self._title = title
        self._y_label = y_label
        self._df = None

    # Property para 'field'
    @property
    def field(self):
        return self._field

    @field.setter
    def field(self, value):
        self._field = value
        self.plot()

    # Property para 'bins'
    @property
    def bins(self):
        return self._bins

    @bins.setter
    def bins(self, value):
        self._bins = value
        self.plot()

    # Property para 'title'
    @property
    def title(self):
        return self._title
```

```
@title.setter
def title(self, value):
    self._title = value
    self.plot()

# Property para 'y_label'
@property
def y_label(self):
    return self._y_label

@y_label.setter
def y_label(self, value):
    self._y_label = value
    self.plot()

def plot(self):
    # Verifica se todas as variáveis foram definidas antes de gerar o gráfico
    if self._df is not None and self._field is not None and self._bins is not None and self._title is not None and self._y_label is not None:
        plt.figure(figsize=(10, 6))
        plt.hist(self._df[self._field], bins=self._bins, color='skyblue', edgecolor='black')

        # Adiciona rótulos e título
        plt.xlabel(self._field)
        plt.ylabel(self._y_label)
        plt.title(self._title)

        # Exibe a grade
        plt.grid(True)

        # Mostra o gráfico
        plt.show()

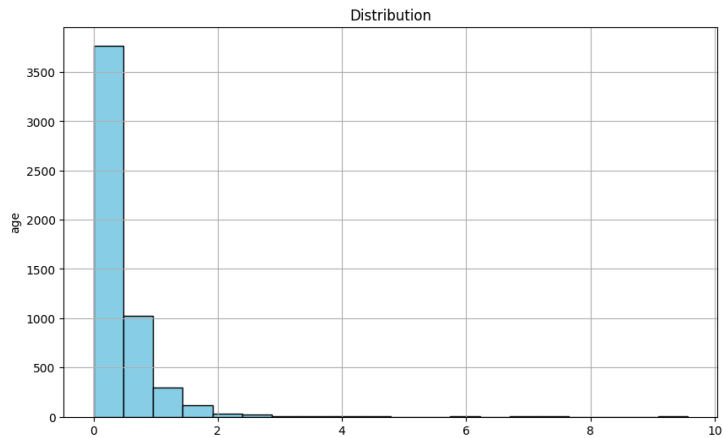
# Interface para atualizar os dados
def update(self, new_data):
    self._df = new_data
    self.plot()
```



```
[11]: # Create instances
table_intake = Table()
table_intake.file_path = 'intake-person-demo(potato-chips).csv'

hist_plot = HistogramPlot()
hist_plot.field='intake_bw'
hist_plot.bins=20
hist_plot.title='Distribution'
hist_plot.y_label='age'

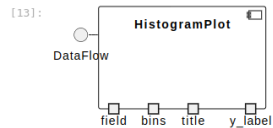
table_intake.connect(hist_plot)
```



```
[13]: %plantuml
!procedure $property($name)
    portout $name
!endprocedure

@startuml
component HistogramPlot {
    $property(field)
    $property(bins)
    $property(title)
    $property(y_label)
}
DataFlow -> [HistogramPlot]
@enduml
```

Writing output for /home/jovyan/notebooks/componentization/ebe0a03d-1fd4-4fd0-98f3-749ee489a140.uml to ebe0a03d-1fd4-4fd0-98f3-749ee489a140.svg



```
[14]: %plantuml
@startuml
[Table] -- DataFlow --> HistogramPlot
@enduml
```

Writing output for /home/jovyan/notebooks/componentization/b09d1894-c689-4781-8960-5f95f39b4105.uml to b09d1894-c689-4781-8960-5f95f39b4105.svg

