

# Basics of Machine Learning

Dmitry Ryabokon, [github.com/dryabokon](https://github.com/dryabokon)





# Lesson 04

## Data Visualization



# Data Visualization

## Summary

- Colormaps
- Squarify
- Pairplot
- Density

# Data Visualization

---

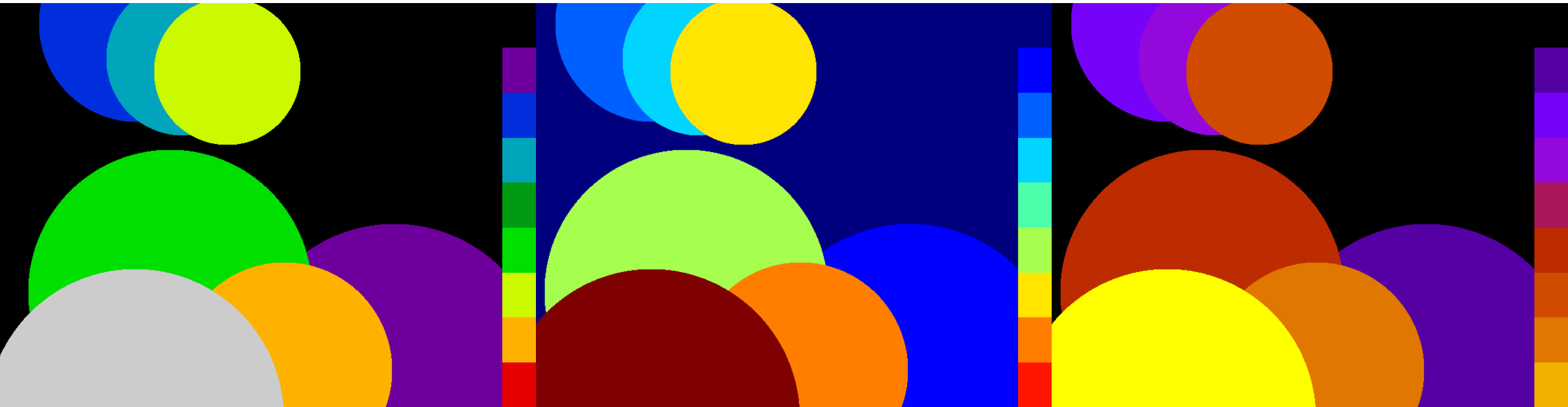
## Tutorials

- `ex_04a_colormaps.py`
- `ex_04b_charts_squarify.py`
- `ex_04c_charts_pairplot.py`
- `ex_04d_charts_density.py`

# Colormaps

# Data charts

## Colormaps



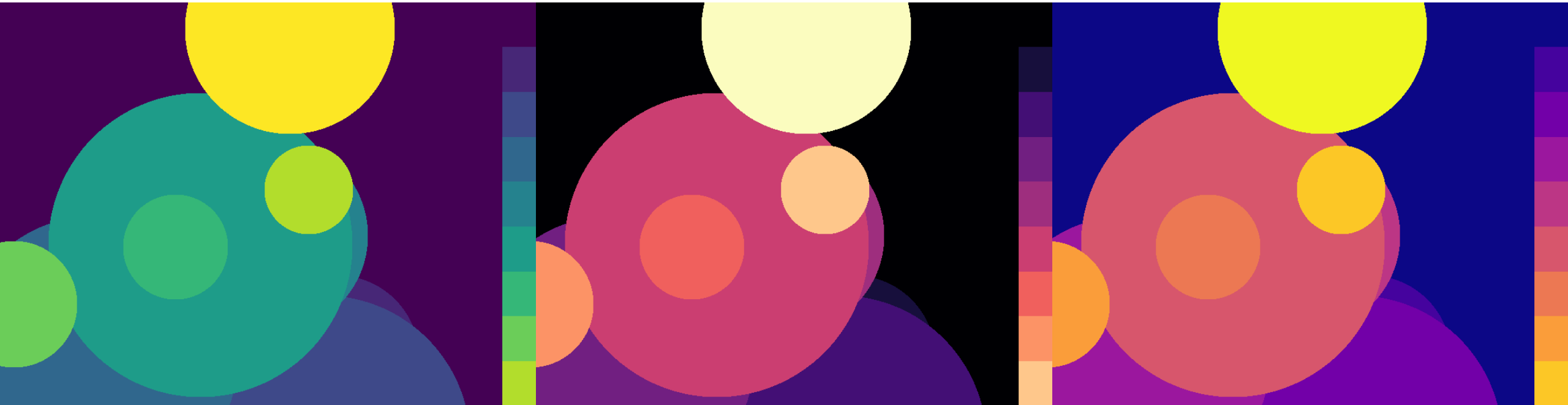
# Data charts

## Colormaps



# Data charts

## Colormaps





# Data charts

## Colormaps



Blues



Greens



PuBu



RdPu



YlOrBr



BuGn



Greys



PuBuGn



Reds



YlOrRd



BuPu



Oranges



PuRd



YlGn



GnBu



OrRd



Purples



YlGnBu

# Data charts

## Colormaps



# Data charts

## Colormaps



afmhot



cool



gist\_yarg



spring



autumn



copper



gray



summer



binary



gist\_gray



hot



winter



bone



gist\_heat



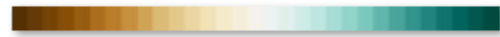
pink



Wistia

# Data charts

## Colormaps



BrBG



PRGn



RdYlBu



bwr



PuOr



RdYlGn



coolwarm



RdBu



seismic



PiYG



RdGy



Spectral

# Data charts

## Colormaps



Accent



Pastel2



tab10



Dark2



Set1



tab20



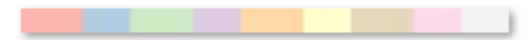
Paired



Set2



tab20b



Pastel1



Set3



tab20c

# Data charts

## Colormaps



brg



gist\_earth



gnuplot



ocean



CMRmap



gist\_ncar



gnuplot2



prism



cubehelix



gist\_rainbow



jet



rainbow



flag



gist\_stern



nipy\_spectral

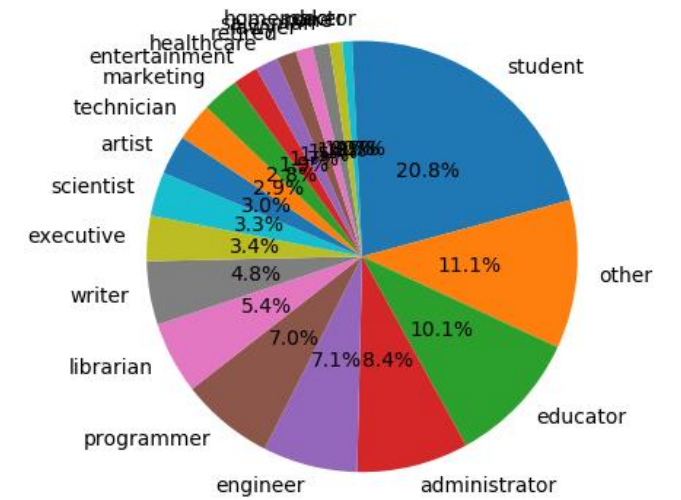
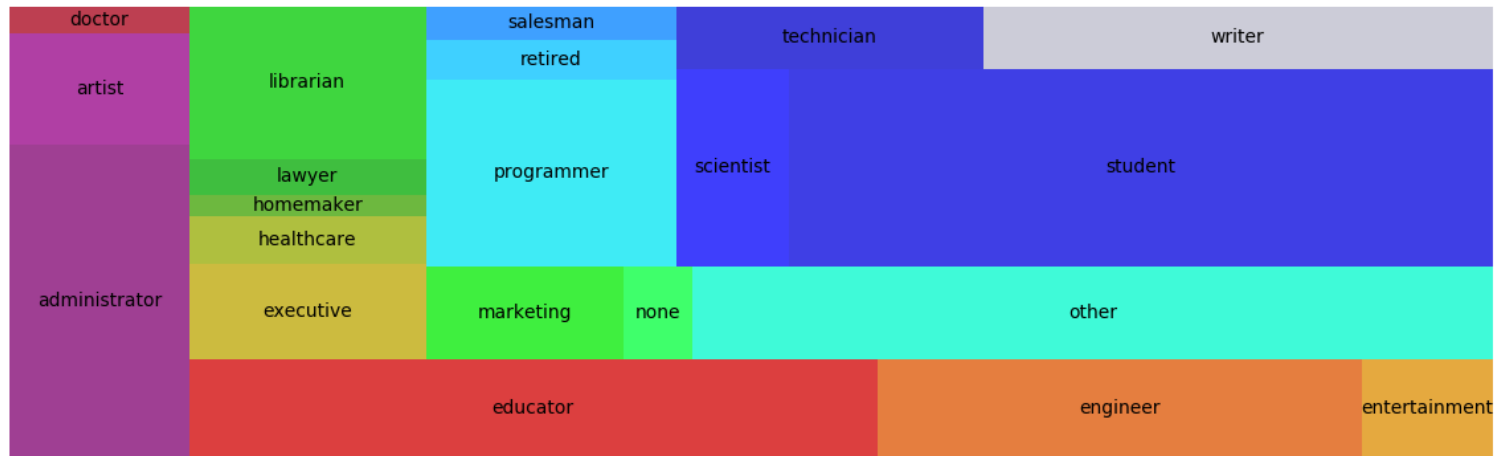


terrain

**Sqarify**

# Data charts

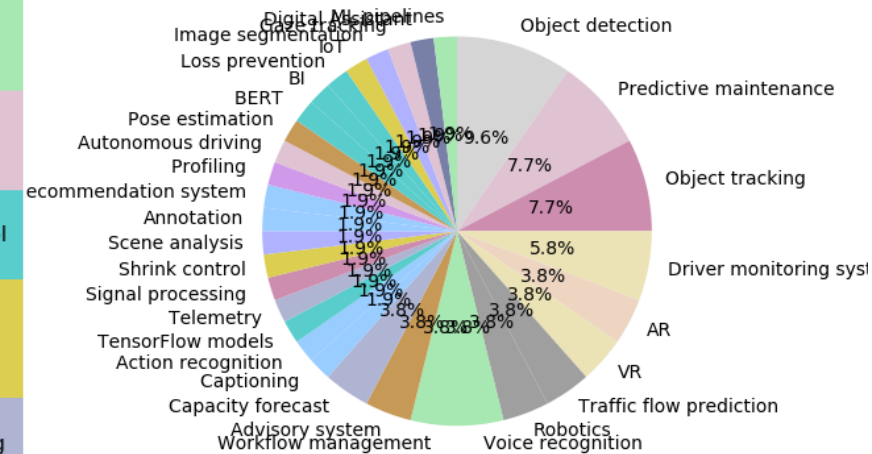
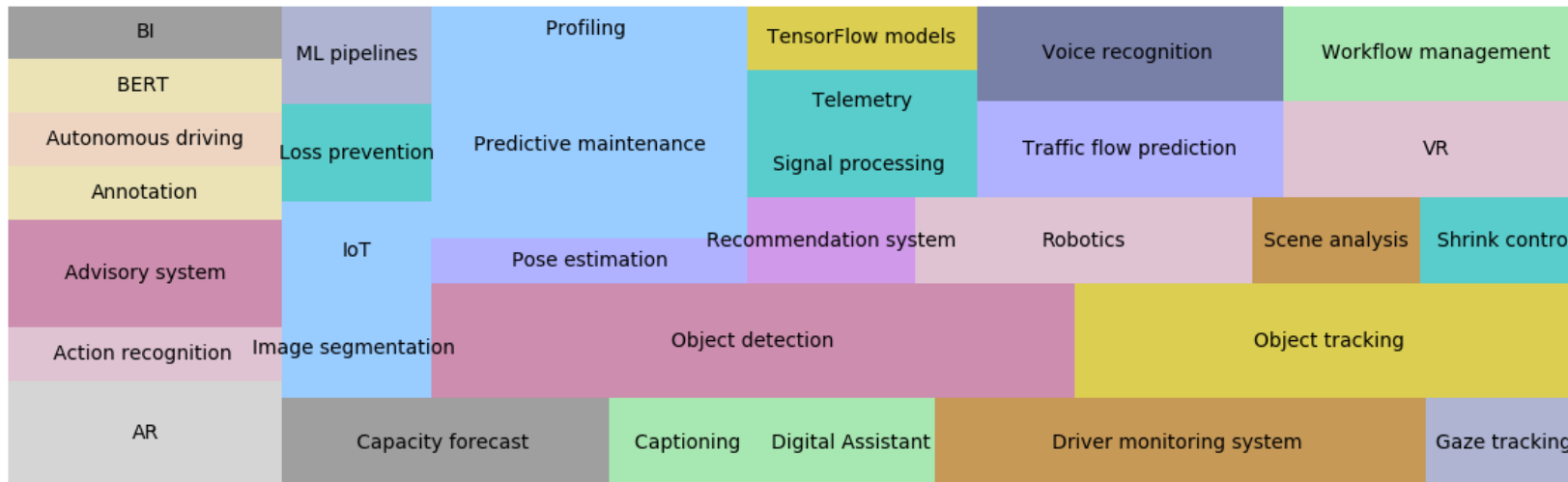
## Squarify: example 01





# Data charts

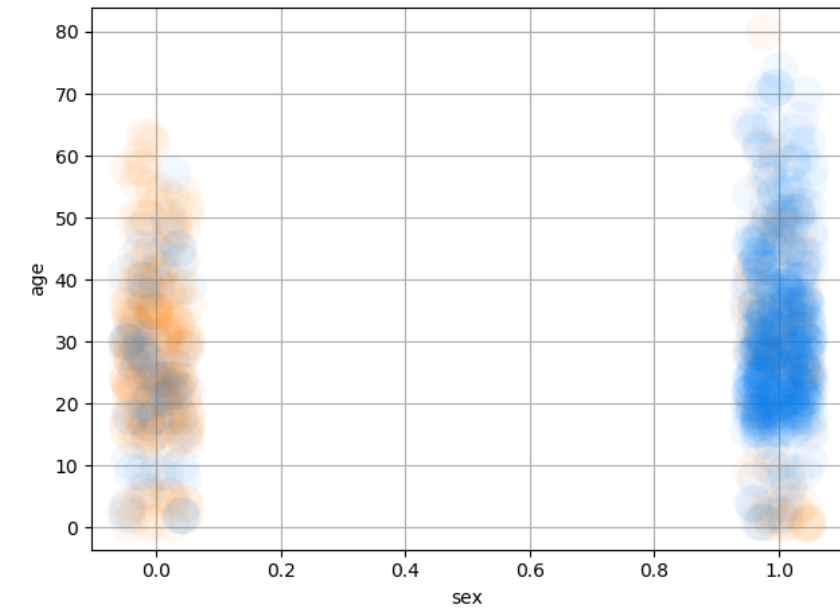
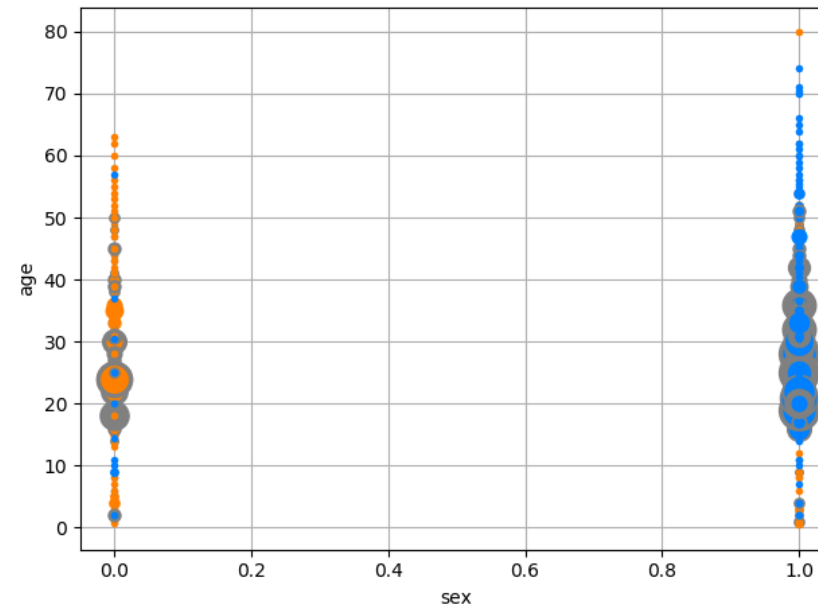
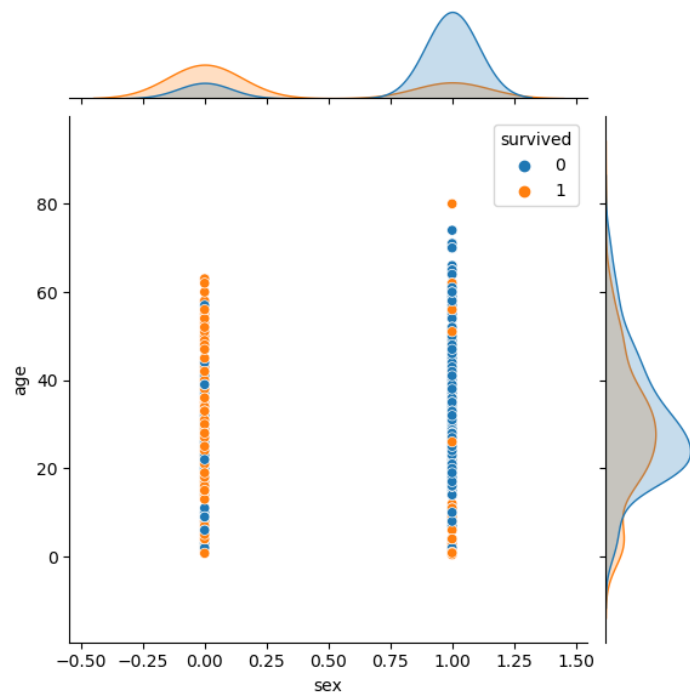
## Squarify: example 02



# Pairplot

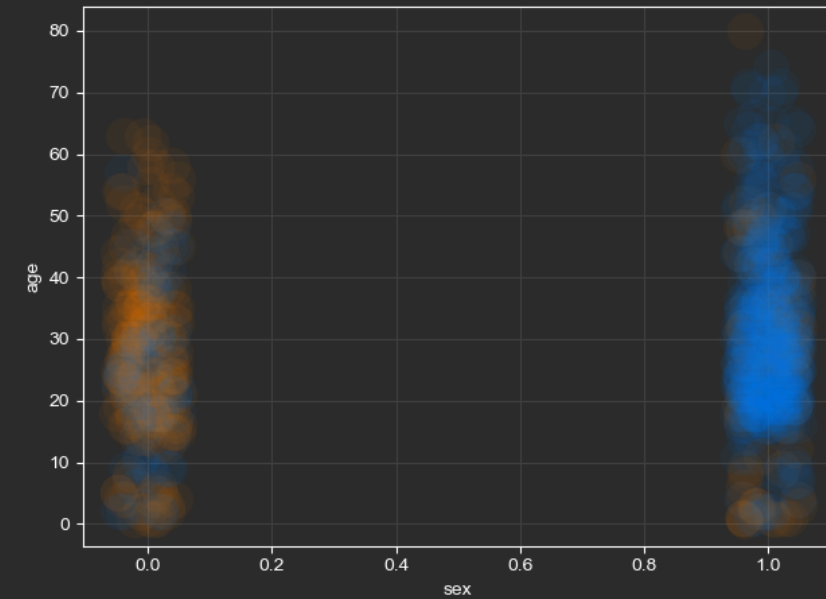
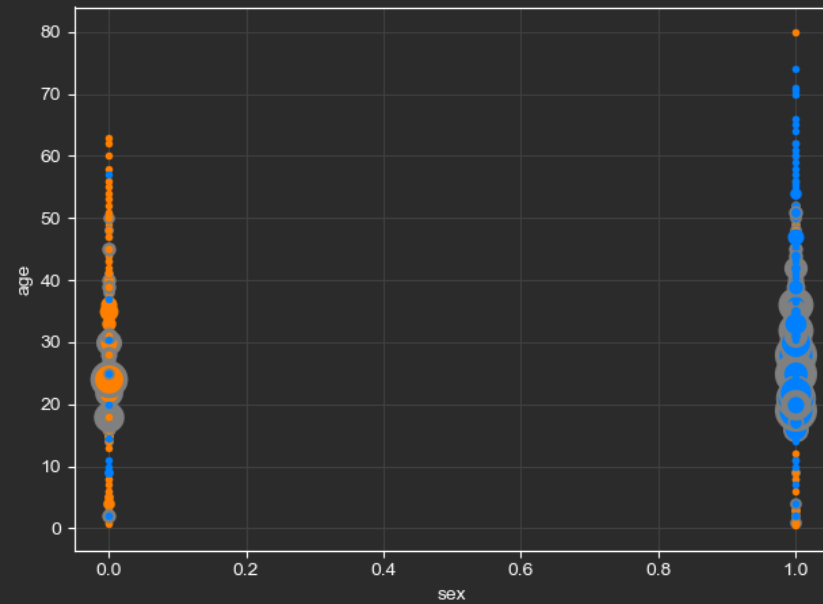
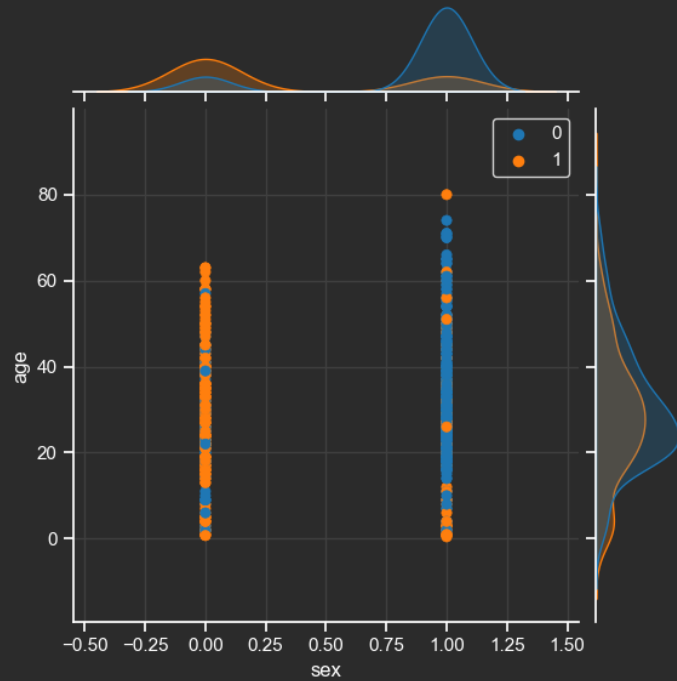
# Data charts

## Pairplot: titanic dataset



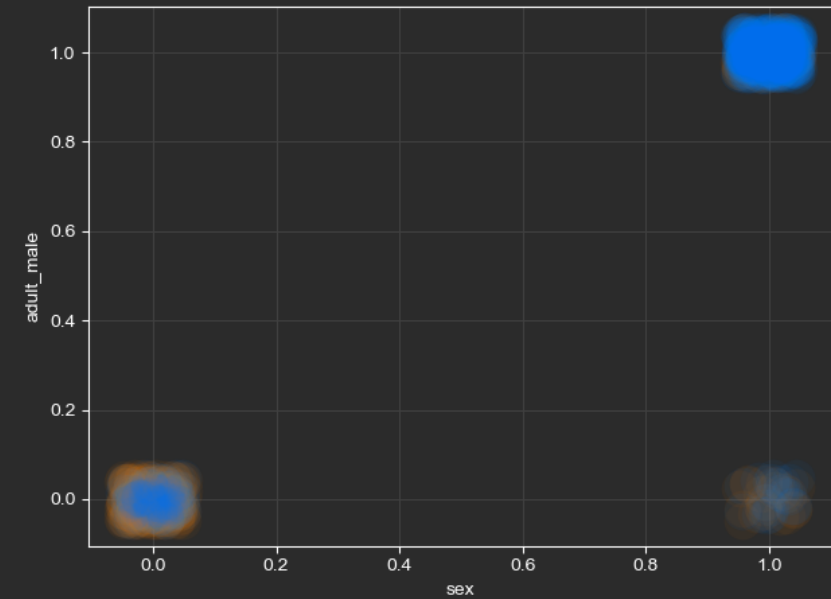
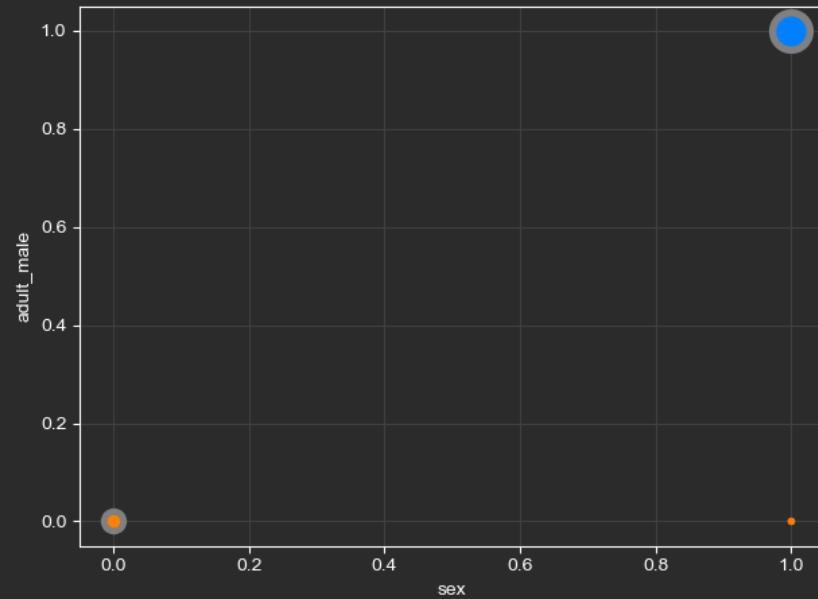
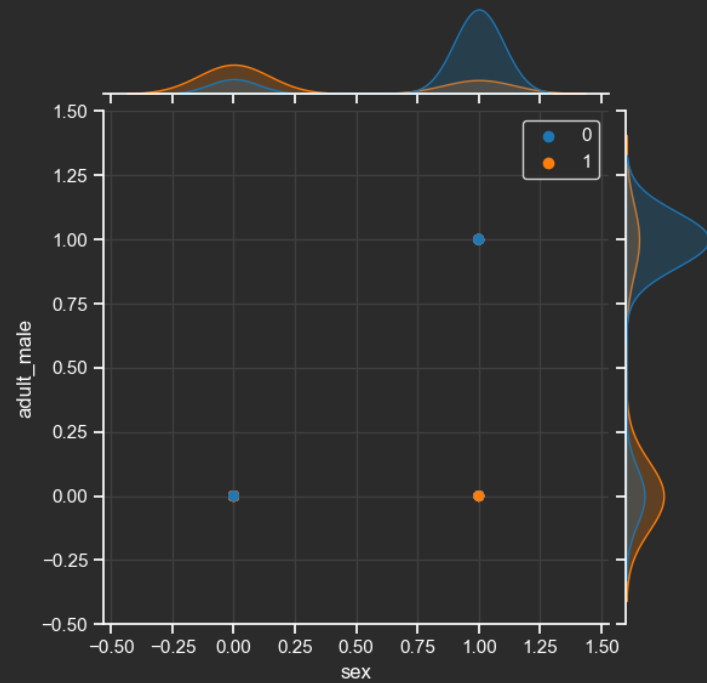
# Data charts

## Pairplot: titanic dataset



# Data charts

## Pairplot: titanic dataset



# Data charts

## Pairplot: titanic dataset

```
columns = df0.columns.to_numpy()
target = columns[idx_target]
idx = numpy.delete(numpy.arange(0, len(columns)), idx_target)
pal = numpy.array(['tab10', 'hus1', 'Set2', 'Paired', 'hls'])[0]

for i in range(len(idx)-1):
    for j in range(i+1, len(idx)):
        c1, c2 = columns[idx[i]], columns[idx[j]]
        df = df0[[target, c1, c2]]
        df = df.dropna()
        df = tools_DF.hash_categoricals(df)

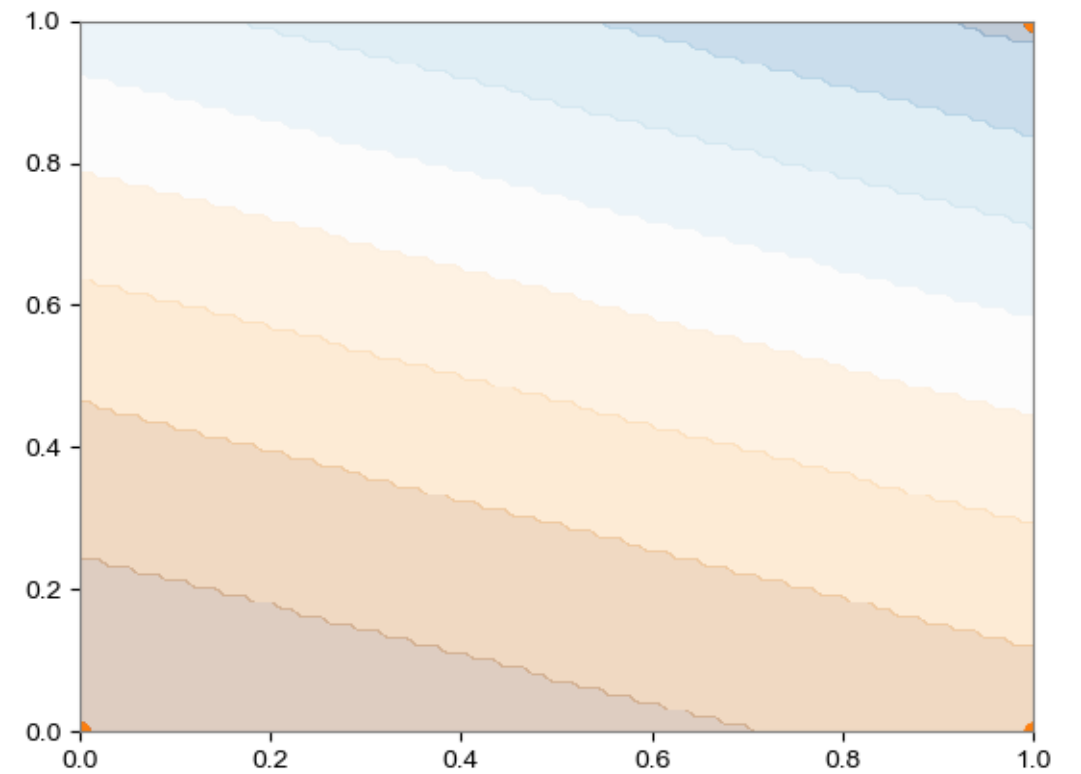
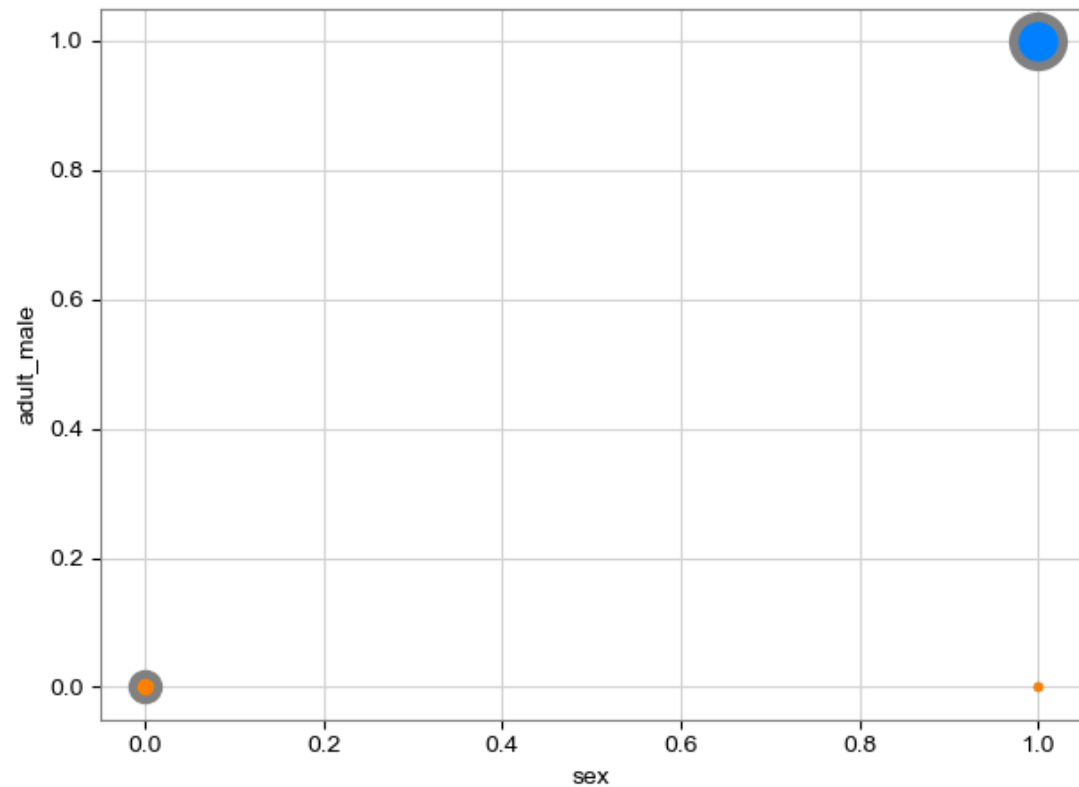
        fig = plt.figure()
        self.turn_light_mode(fig)
        J = seaborn.jointplot(data=df, x=c1, y=c2, hue=target, palette=pal, edgecolor=None)
```

# Density

A horizontal band of glowing particles and dust stretches across the middle of the image. The particles are small, bright white dots of varying sizes, some appearing as short streaks, suggesting motion or a dense field of matter. The background is a deep, dark gray, with the particle band creating a sense of depth and focus.

# Data charts

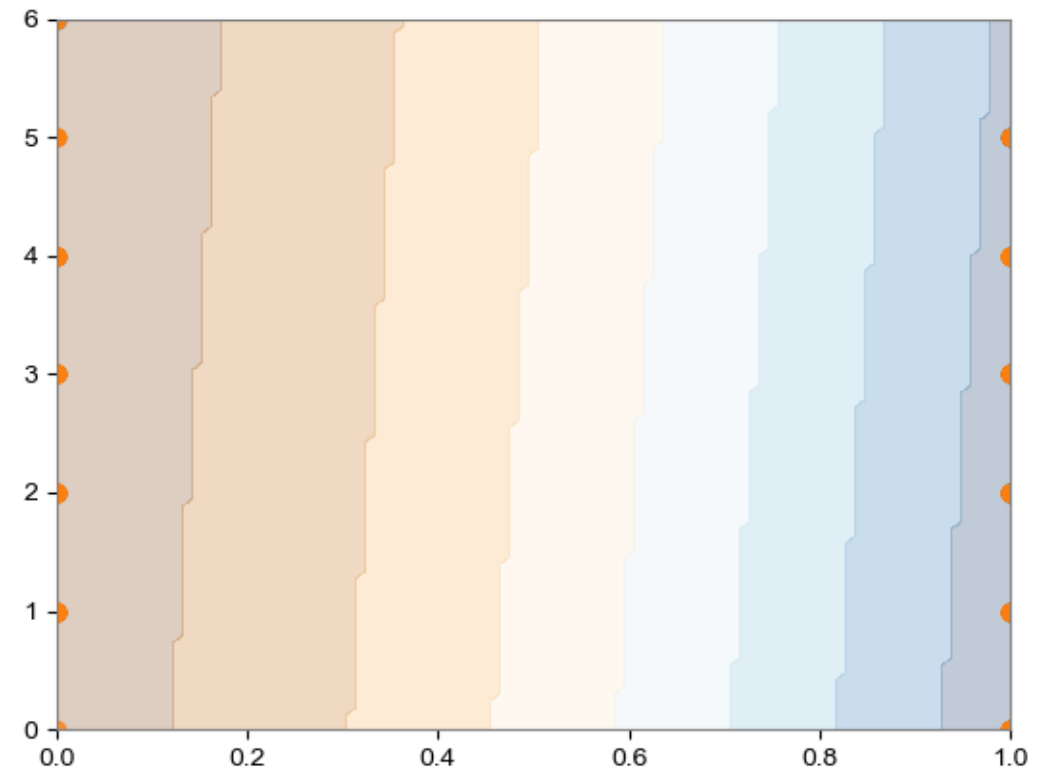
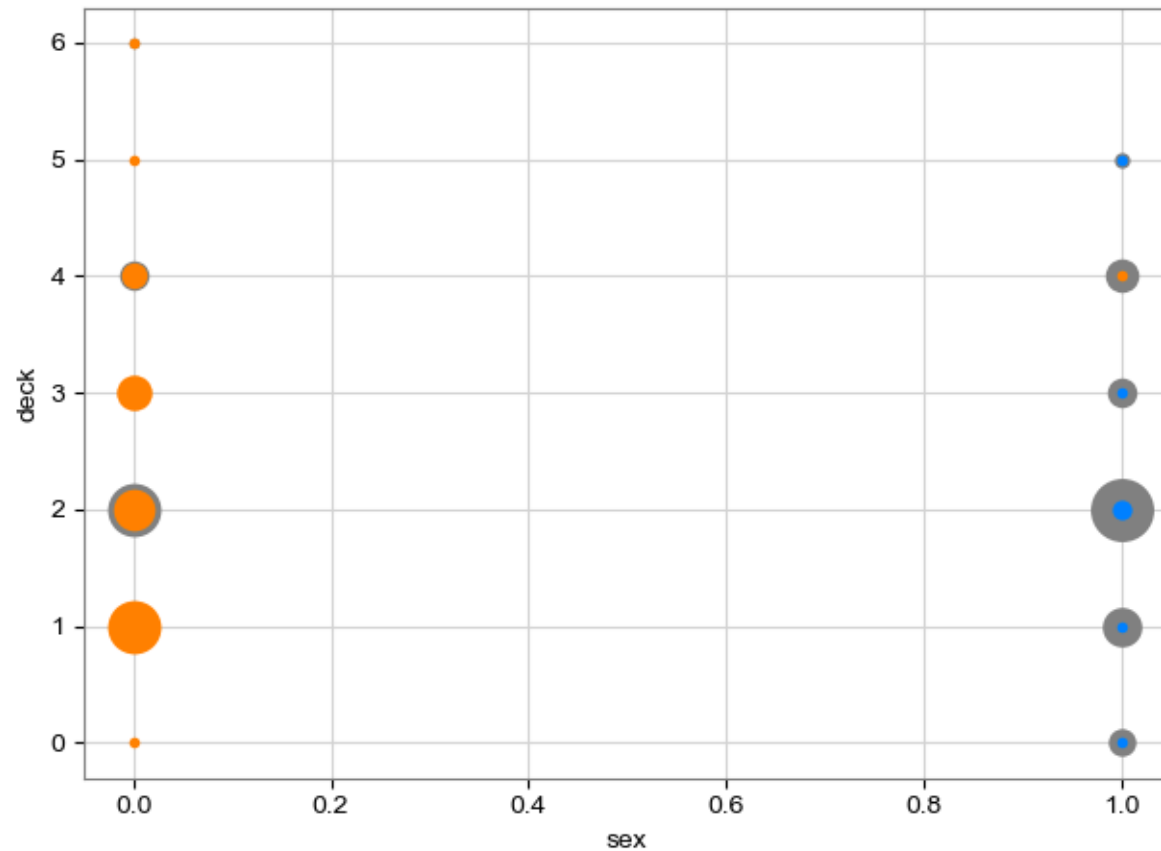
## Density plot: titanic dataset





# Data charts

## Density plot: titanic dataset



# Time Series

# Data charts

## Time Series: pandas

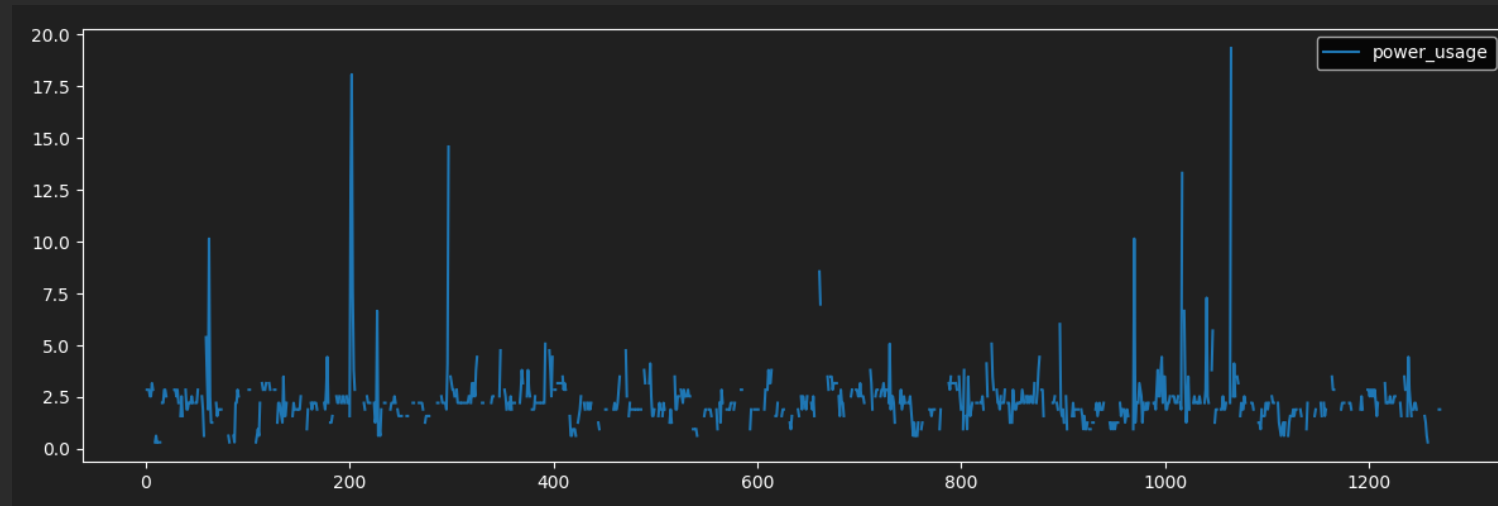
```
fig = plt.figure(figsize=(12, 4))
ax = plt.gca()
```

```
if darkmode:
    plt.style.use('dark_background')
    clr = numpy.array((32, 32, 32)) / 255
    fig.patch.set_facecolor(clr)
    ax.set_facecolor(clr)
    ax.spines['bottom'].set_color('white')
    ax.spines['top'].set_color('white')
    ax.spines['left'].set_color('white')
    ax.spines['right'].set_color('white')
    ax.xaxis.label.set_color('white')
    ax.tick_params(axis='x', colors='white')
    ax.tick_params(axis='y', colors='white')
```

```
else:
    plt.style.use('classic')
    fig.patch.set_facecolor((1, 1, 1))
```

```
X = None if idx_feature is None else df.columns[idx_feature]
df.plot(x=X, y=df.columns[idx_target], ax=ax)
plt.tight_layout()
```

```
if filename_out is not None:
    plt.savefig(filename_out, facecolor=fig.get_facecolor())
    plt.close()
```



# Data charts

## Time Series: seaborn

```
fig = plt.figure(figsize=(12, 4))
```

```
pal = numpy.array(['tab10', 'hus1', 'Set2', 'Paired', 'hls'])[0]  
colors = sns.color_palette(palette=pal, n_colors=1)
```

```
if darkmode:
```

```
    plt.style.use('dark_background')  
    clr = numpy.array((32, 32, 32)) / 255  
    fig.patch.set_facecolor(clr)  
    plt.gca().set_facecolor(clr)
```

```
else:
```

```
    plt.style.use('classic')  
    fig.patch.set_facecolor((1, 1, 1))
```

```
X = df.index if idx_feature is None else df.columns[idx_feature]  
markers = 'o'
```

```
if idx_feature is None:
```

```
    if mode=='pointplot':
```

```
        g = sns.pointplot(data=df, x=X, y=df.columns[idx_target], scale=0.25, markers=markers, color=colors[0])  
        g.set(xticks=[])
```

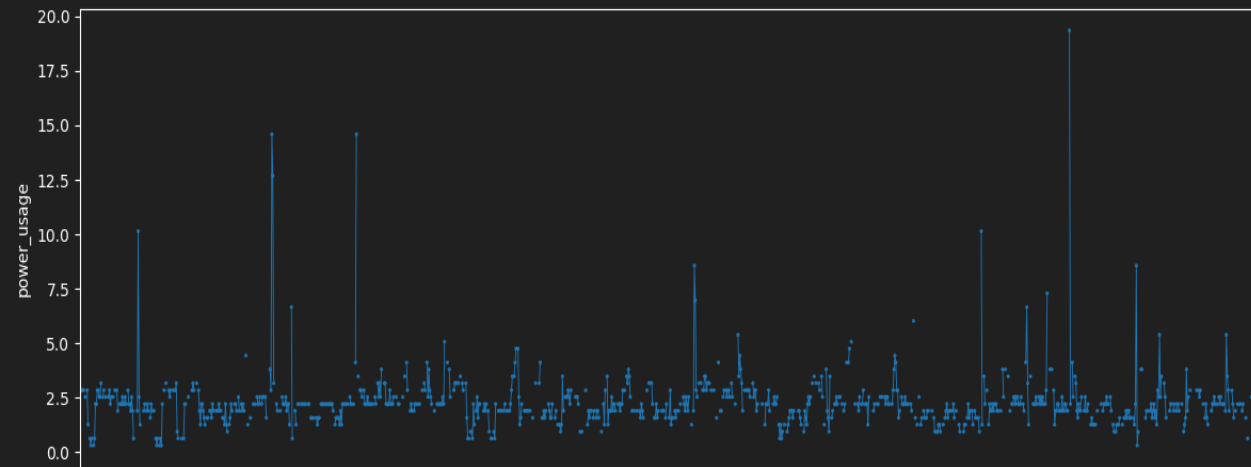
```
    else:
```

```
        g = sns.lineplot(data=df, x=X, y=df.columns[idx_target], color=colors[0])
```

```
plt.tight_layout()
```

```
if filename_out is not None:
```

```
    plt.savefig(filename_out, facecolor=fig.get_facecolor())  
    plt.close()
```



## References

- <https://www.tutorialdocs.com/article/python-matplotlib-tutorial.html>

