matplotlib vs. ggplot2

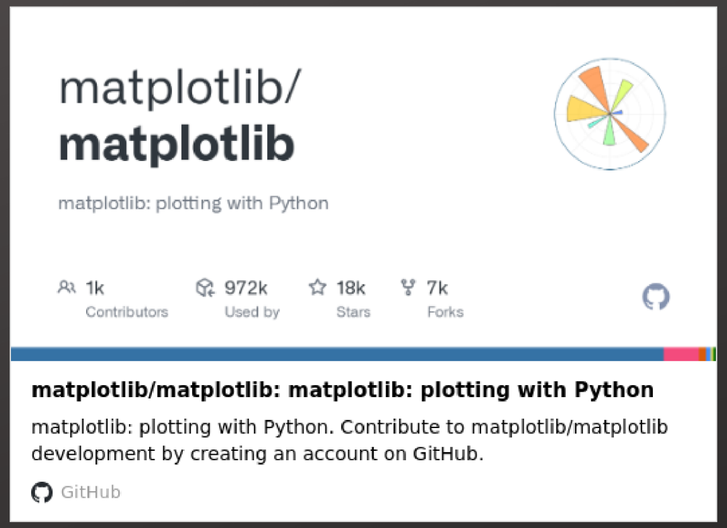
Rocío Romero Zaliz
rocio@ugr.es

16 / 11 / 23

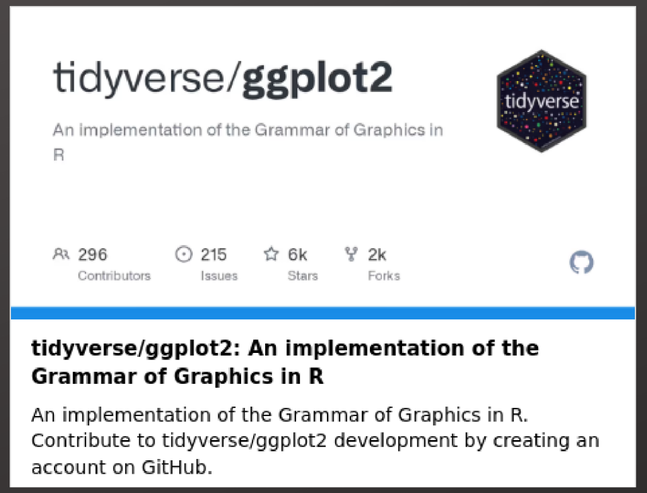
PyData Granada

matplotlib

- Matplotlib es una herramienta de visualización de datos multiplataforma basada en el framework Numpy y Scipy concebida por John Hunter en 2002.



ggplot2

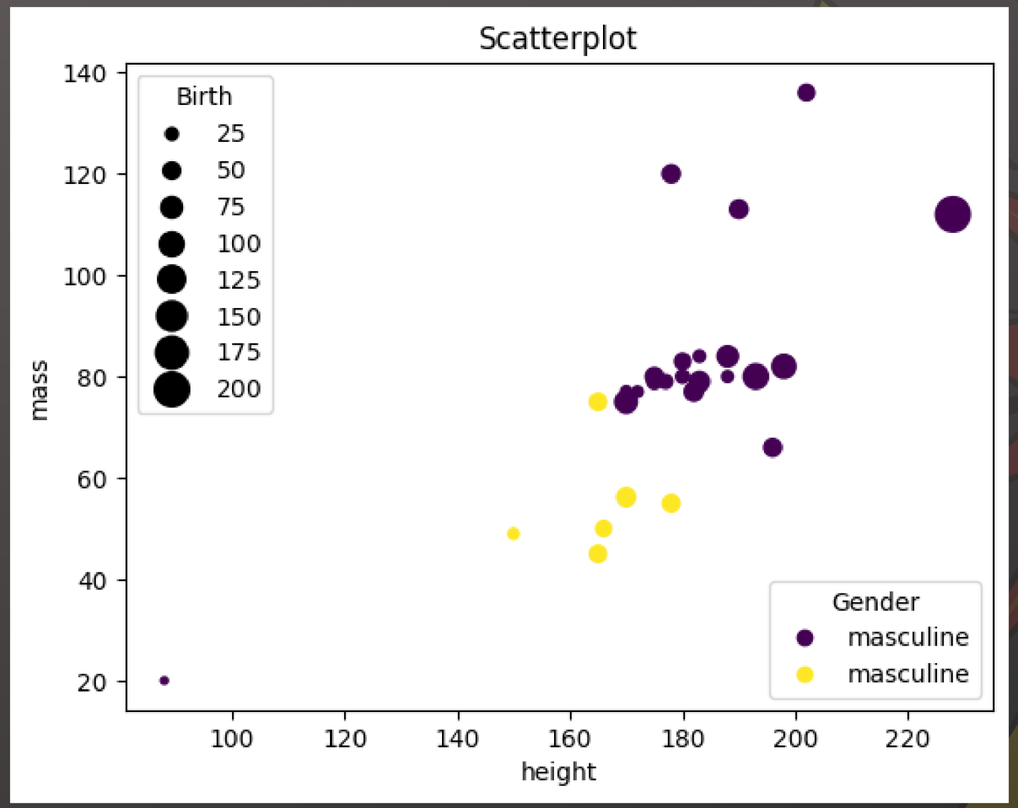


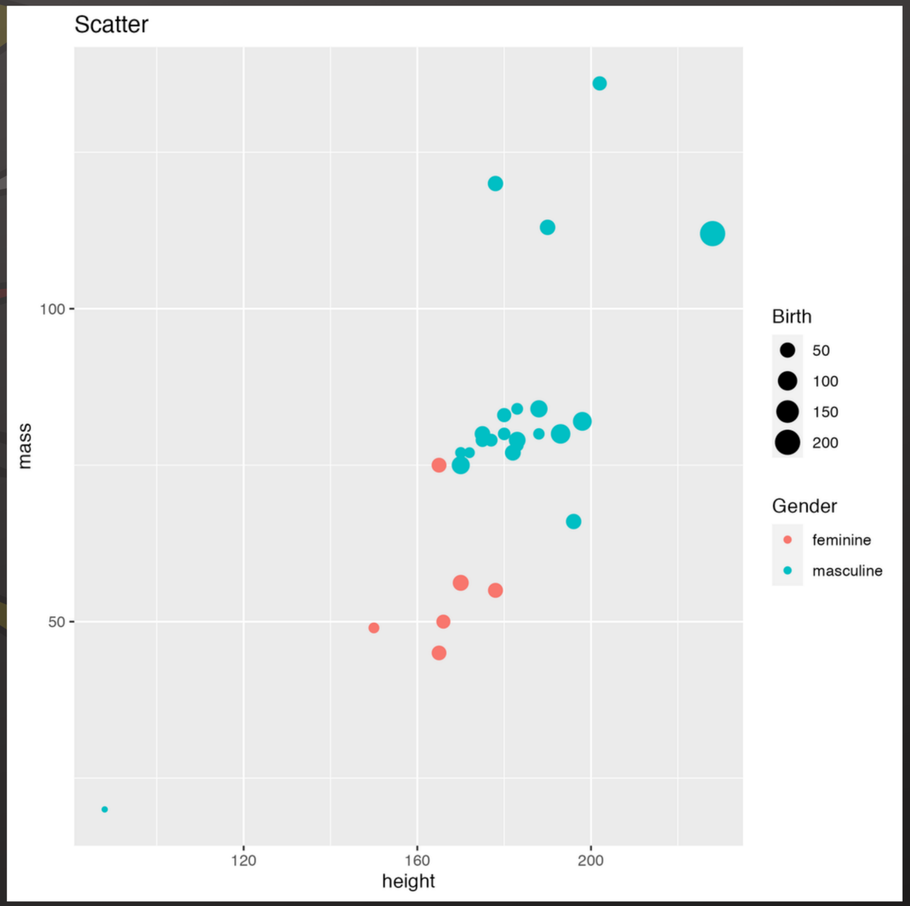
- ggplot2 tiene una gramática subyacente, basada en la Gramática de los Gráficos (Wilkinson 2005), que permite componer gráficos combinando componentes independientes.

round one

matplotlib

ggplotz





matplotlib

```
scatter = plt.scatter(x=df["height"],
y=df["mass"], c=pd.factorize(df["gender"])[0],
s=df['birth_year'])

# Adding legend to the plot
handles = scatter.legend_elements("colors")[0]
first_legend = plt.legend(handles=handles,
labels=list(df["gender"]), title='Gender',
loc='lower right')
handles = scatter.legend_elements("sizes")[0]
plt.gca().add_artist(first_legend)
plt.legend(handles=handles,
labels=scatter.legend_elements("sizes")[1],
title='Birth', loc='upper left')
plt.xlabel("height")
plt.ylabel("mass")

# Title to the plot
plt.title("Scatterplot")
plt.show()
```

VS

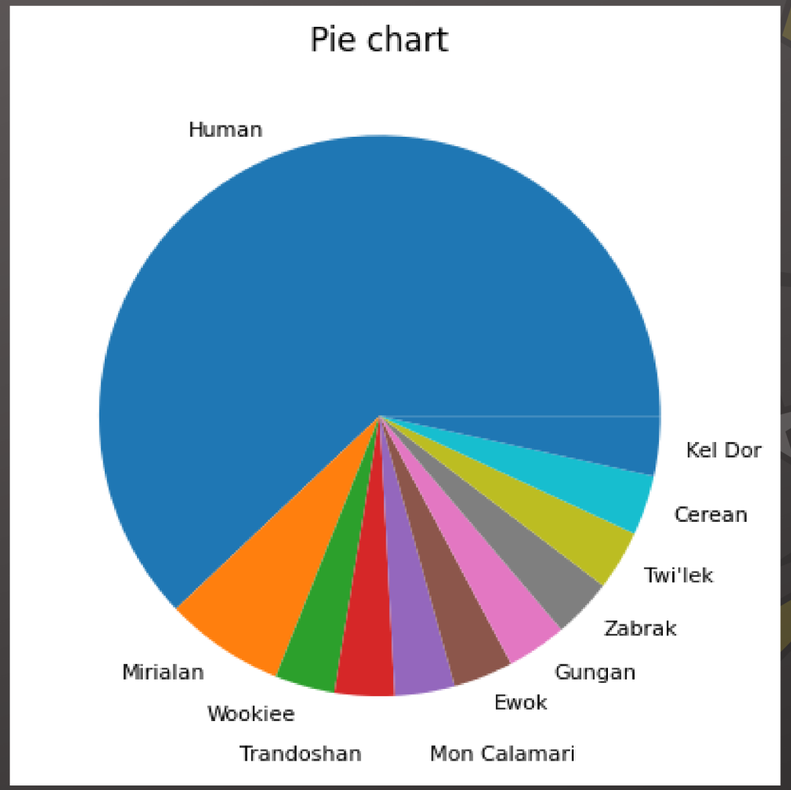
ggplot2

```
ggplot(starwars %>% drop_na(), aes(x=height,
y=mass, color=gender, size=birth_year)) +
geom_point() + labs(color="Gender",
size="Birth", title="Scatter")
```

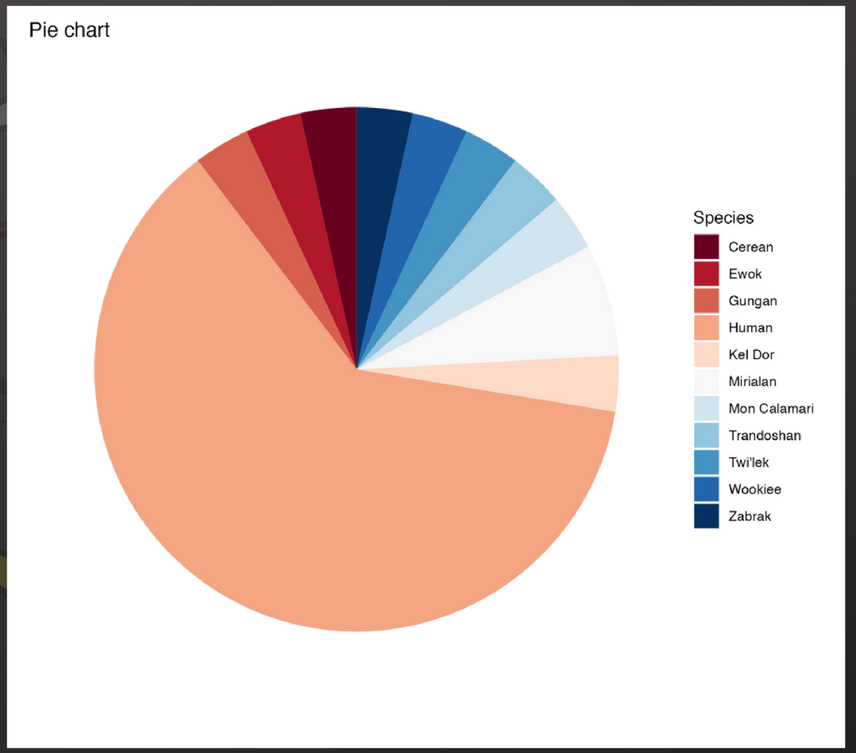
round two

matplotlib

ggplotz



VS



matplotlib

```
 count = df['species'].value_counts()

# This will plot a simple pie chart
patches, texts, autotexts = plt.pie(count,
labels=count.keys(), autopct='', textprops=
{'fontsize': 8})

texts[3]._y =-1.2
texts[4]._y =-1.2

# Title to the plot
plt.title("Pie chart")
plt.show()
```

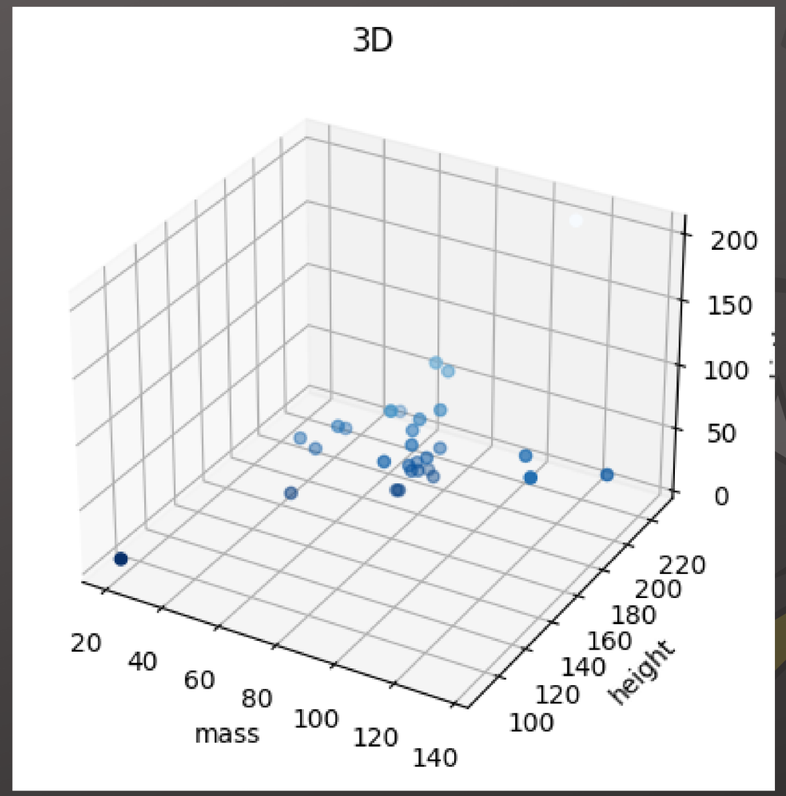
ggplot2

```
ggplot(starwars %>% drop_na() %>% group_by(species)
%>% count(), aes(x="", y=n, fill=species)) +
geom_col() + coord_polar(theta = "y") +
scale_fill_brewer(palette = "RdBu") + theme_void()
+ labs(title = "Pie chart", fill = "Species")
```

round three

matplotlib

ggplot2



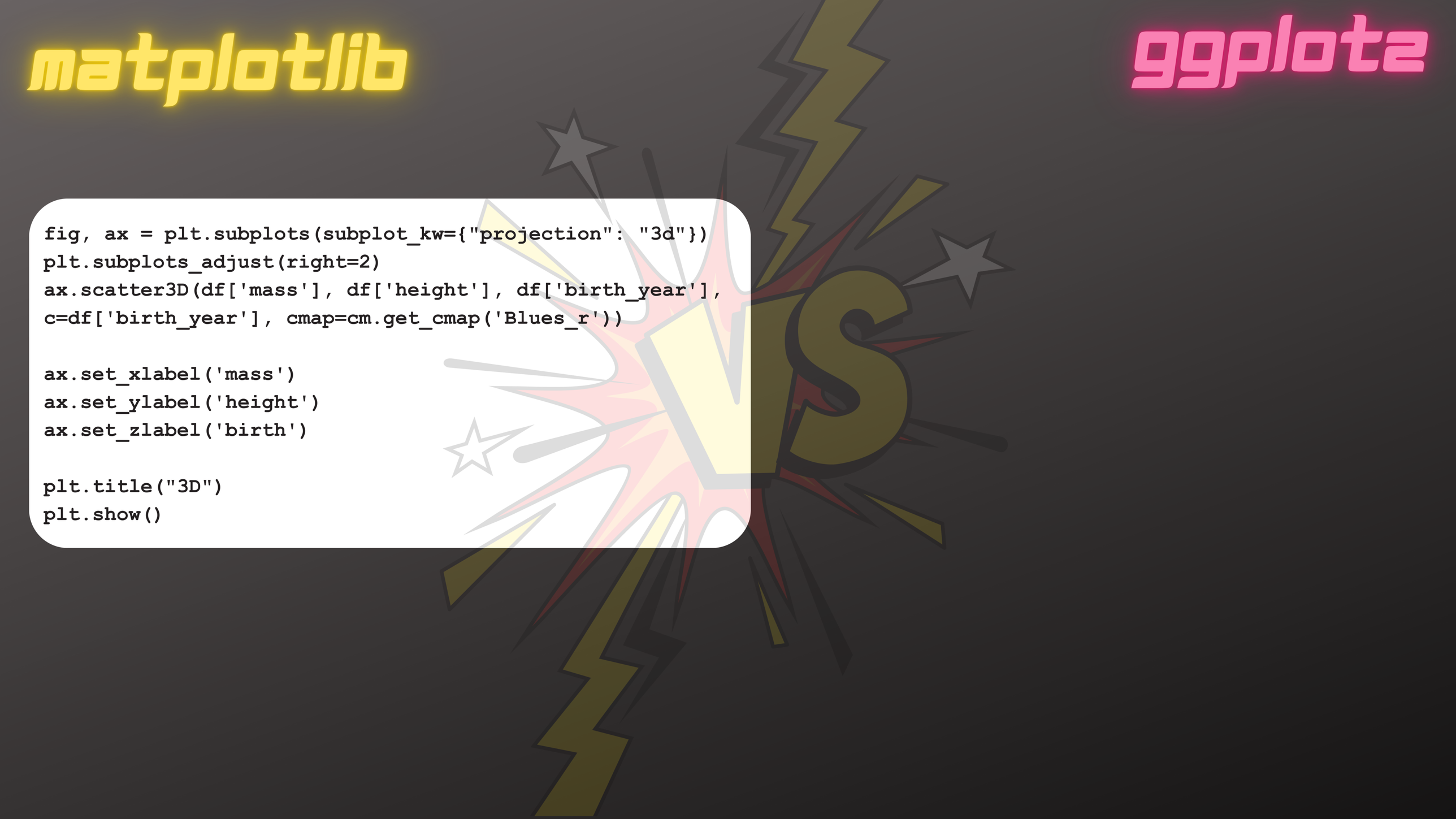
VS

404

matplotlib

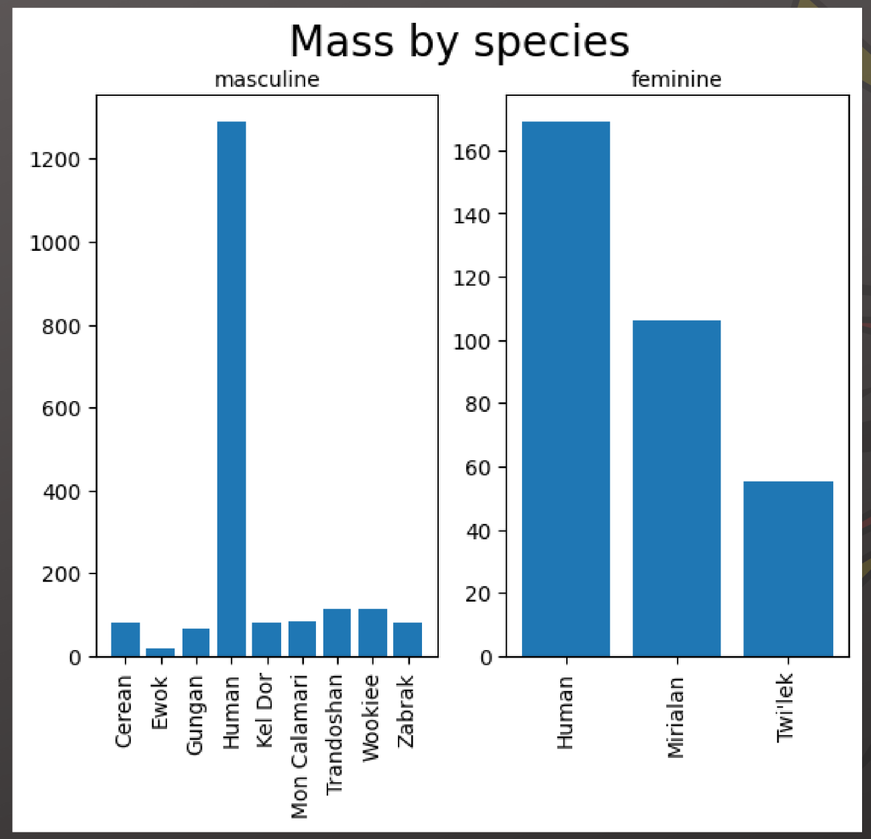
ggplot2

```
fig, ax = plt.subplots(subplot_kw={"projection": "3d"})
plt.subplots_adjust(right=2)
ax.scatter3D(df['mass'], df['height'], df['birth_year'],
c=df['birth_year'], cmap=cm.get_cmap('Blues_r'))

ax.set_xlabel('mass')
ax.set_ylabel('height')
ax.set_zlabel('birth')

plt.title("3D")
plt.show()
```



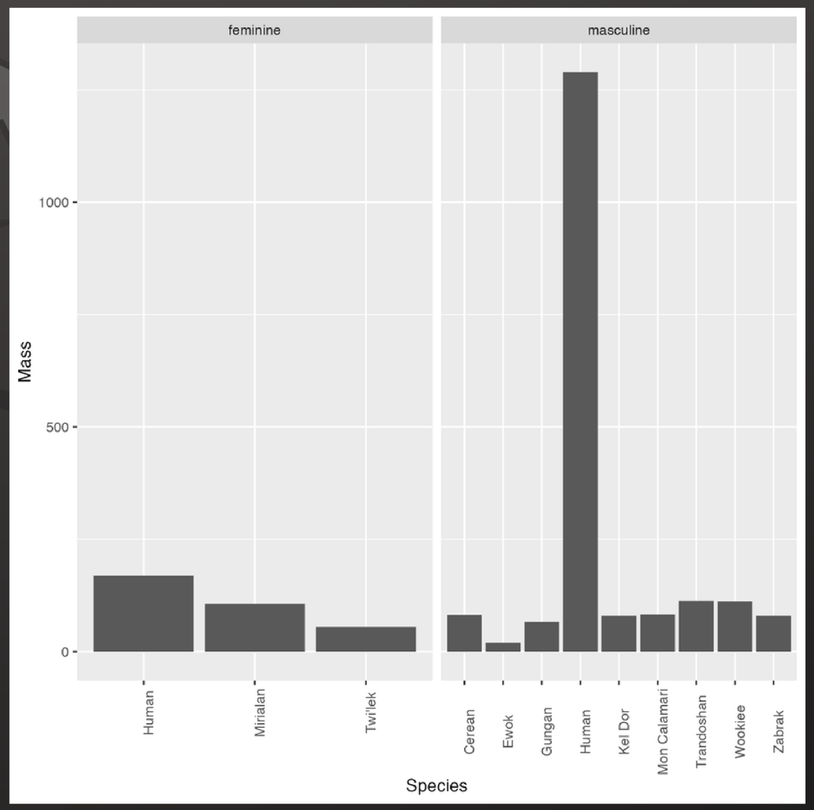
round four

matplotlib



VS

ggplot2



matplotlib

```
fig, axes = plt.subplots(1, len(set(df['gender'])))

for x, gender in enumerate(set(df['gender'])):
    data = df[(df['gender'] == gender)]
    data = data.groupby(['species']).mass.sum()

    left = [k[0] for k in enumerate(data)]
    right = [k[1] for k in enumerate(data)]

    axes[x].bar(left, right, label="%s" % (gender))
    axes[x].set_xticks(left, minor=False)
    axes[x].set_xticklabels(data.index.values,
rotation=90)
    axes[x].set_xlabel(gender, loc='center')
    axes[x].xaxis.set_label_position('top')

    fig.suptitle('Mass by species', fontsize=20)
```

ggplot2

```
ggplot(starwars %>% drop_na() %>%
group_by(species, gender) %>%
summarize(suma=sum(mass)), aes(x=species,
y=suma)) + geom_col() + facet_wrap(~gender,
scales="free_x") + labs(x="Species", y="Mass") +
theme(axis.text.x = element_text(angle = 90))
```

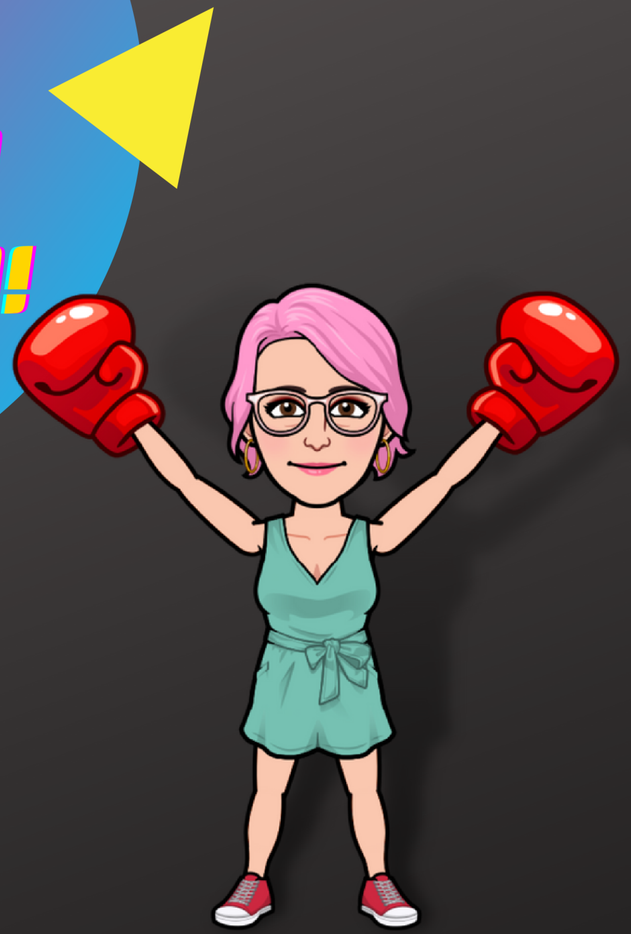
curiosdades

- Se puede usar ggplot2 en python con el paquete **plotnine**.
- Se puede usar matplotlib en R con el comando **Pyrun**.

conclusiones

- **matplotlib:**
  - Código largo
  - Hay que especificarle todo
  - Se necesitan ciclos explícitos
- **ggplot2:**
  - Código elegante y reducido
  - Asume las cosas más obvias
  - Funciones que ahorran trabajo

¡gracias por vuestra atención!



rocio@ugr.es

http://decsai.ugr.es/~rocio

@RCRZ_UGR

SintonIA: la IA en las ondas

@sintonia_dasci

SCAN ME

DaSCI

CITIC-UGR