PARIS LODRON UNIVERSITÄT SALZBURG

Plotly including ggplotly (extension on ggplot2)

Ankitha Kakade & Roland Weiszmüller

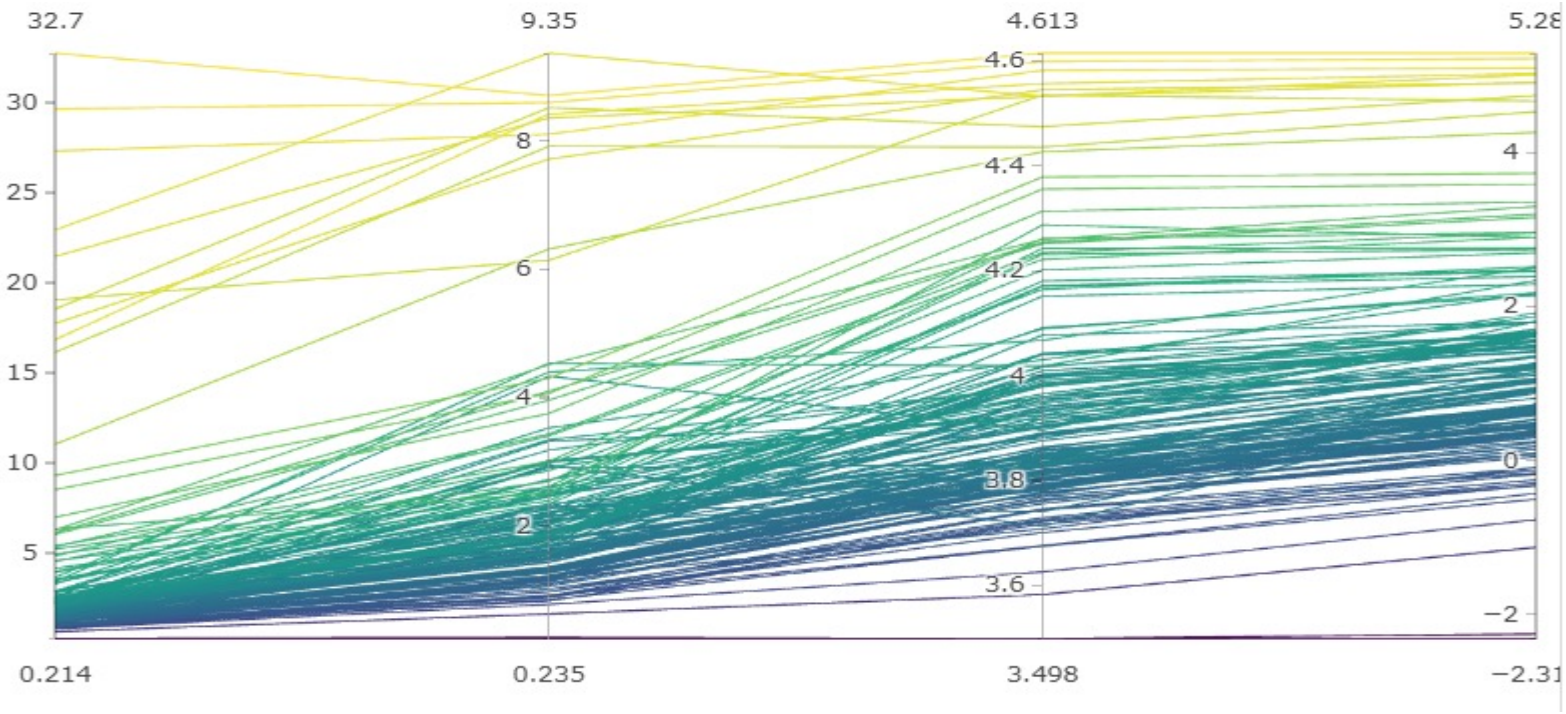
SE Statistics Visualization and More Using "R“

April 30th, 2024

Outline

- Plotly Introduction
- What, Where and Why “plotly”?
- Exercise on plotly
- Introduction to ggplot2
- Basics and exercises.

Interpret the plot (Discussion)



What is plotly ?

- plotly is an R package for creating interactive web-based graphs via the open-source JavaScript graphing library plotly.js.

- By default, the plotly R package runs locally in your web browser or in the RStudio viewer.

- This function maps R objects to [plotly.js](https://plotly.com/javascript/), an (MIT licensed) web-based interactive charting library.

- plot_ly() tries to create a sensible plot based on the information you give it. If you don't provide a trace type, plot_ly() will infer one.

Why plotly ?

- plotly has a functional interface: every plotly function takes a plotly object as it's first input argument and returns a modified plotly object.

- plotly integrates seamlessly with popular programming languages like R, Python, and JavaScript, as well as with web frameworks like Shiny and Dash, enabling users to incorporate interactive plots into their data analysis pipelines and web applications.

Where do we us plotly ?

1. Data Analysis and Exploration

2. Reporting and Presentation

3. Dashboard & Web Developments

4. Education and Research

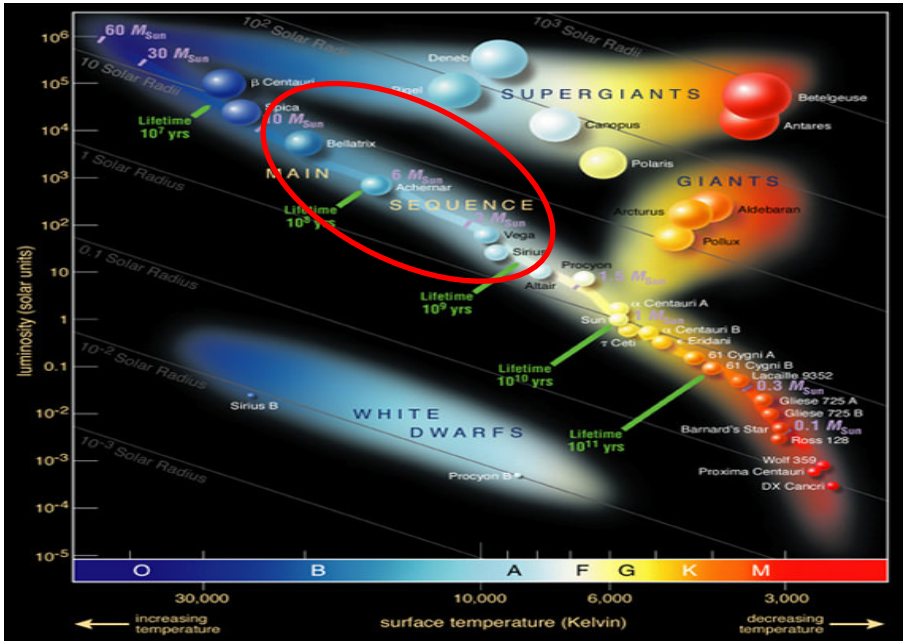
Motivation For interactivity

- Understanding Complex Relationships
- Exploratory Data Analysis
- Enhanced Communication and Engagement
- Facilitating Decision-Making

Ploty

Short introduction to the dataset – I.

- Source: https://academic.oup.com/mnras/article/382/3/1073/1008562
- Class A,B
  - surface temperature 7000-30.000 K



Short introduction to the dataset – I.

PARIS LODRON UNIVERSITÄT SALZBURG

- Source: https://academic.oup.com/mnras/article/382/3/1073/1008562

- Class A,B :
  - surface temperature 7000-30.000 K

- Changing luminosity stars :
  - the total radiation emitted by a celestial body over a specified period of time.
  -

Short introduction to the dataset – I.

PARIS LODRON UNIVERSITÄT SALZBURG

- Source: https://academic.oup.com/mnras/article/382/3/1073/1008562

- Class A,B :
  - surface temperature 7000-30.000 K

- Changing luminosity stars :
  - the total radiation emitted by a celestial body over a specified period of time.
  -

- Magnitude :
  - apparent brightness – what we can see with our eyes

$$m = -2,5 * \lg \left(\frac{F}{F_{Vega}} \right)$$

Short introduction to the dataset – II.

PARIS LODRON UNIVERSITÄT SALZBURG

- Source: https://academic.oup.com/mnras/article/382/3/1073/1008562

GCVS	m_Name	SpType	Mass	e_Mass	Rad	e_Rad	logTeff	e_logTeff	logL	e_logL	VMAG	e_VMAG
RT And	A	F8V	1.240	0.030	1.260	0.015	3.785	0.015	0.290	0.060	4.00	0.15
RT And	B	K3	0.910	0.020	0.900	0.013	3.675	0.010	-0.435	0.040	6.20	0.20
BW Aqr	B	F7V	1.390	0.020	1.790	0.040	3.810	0.007	0.700	0.030	2.98	0.09

Short introduction to the dataset – II.

PARIS LODRON UNIVERSITÄT SALZBURG

- Source: https://academic.oup.com/mnras/article/382/3/1073/1008562

Geral Class of Variable Stars
(Subclass of Main-sequence)

Examples :
- Pulsars class
- Rotacion class
- Double stars class

GCVS	m_Name	SpType	Mass	e_Mass	Rad	e_Rad	logTeff	e_logTeff	logL	e_logL	VMAG	e_VMAG
RT And	A	F8V	1.240	0.030	1.260	0.015	3.785	0.015	0.290	0.060	4.00	0.15
RT And	B	K3	0.910	0.020	0.900	0.013	3.675	0.010	-0.435	0.040	6.20	0.20
BW Aqr	B	F7V	1.390	0.020	1.790	0.040	3.810	0.007	0.700	0.030	2.98	0.09

Short introduction to the dataset – II.

PARIS LODRON UNIVERSITÄT SALZBURG

- Source: https://academic.oup.com/mnras/article/382/3/1073/1008562

GCVS	m_Name	SpType	Mass	e_Mass	Rad	e_Rad	logTeff	e_logTeff	logL	e_logL	VMAG	e_VMAG
RT And	A	F8V	1.240	0.030	1.260	0.015	3.785	0.015	0.290	0.060	4.00	0.15
RT And	B	K3	0.910	0.020	0.900	0.013	3.675	0.010	-0.435	0.040	6.20	0.20
BW Aqr	B	F7V	1.390	0.020	1.790	0.040	3.810	0.007	0.700	0.030	2.98	0.09

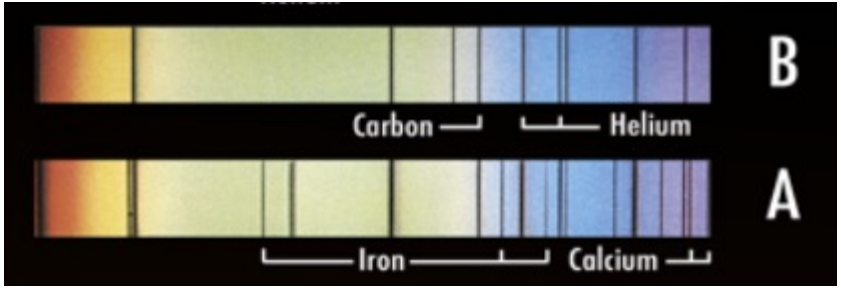
Short introduction to the dataset – II.

PARIS LODRON UNIVERSITÄT SALZBURG

- Source: https://academic.oup.com/mnras/article/382/3/1073/1008562

Main Seq.

Type of spectal lines



GCVS	m_Name	SpType	Mass	e_Mass	Rad	e_Rad	logTeff	e_logTeff	logL	e_logL	VMAG	e_VMAG
RT And	A	F8V	1.240	0.030	1.260	0.015	3.785	0.015	0.290	0.060	4.00	0.15
RT And	B	K3	0.910	0.020	0.900	0.013	3.675	0.010	-0.435	0.040	6.20	0.20
BW Aqr	B	F7V	1.390	0.020	1.790	0.040	3.810	0.007	0.700	0.030	2.98	0.09

Short introduction to the dataset – II.

PARIS LODRON UNIVERSITÄT SALZBURG

- Source: https://academic.oup.com/mnras/article/382/3/1073/1008562

All units of measurement are dimensionless!

Mass Radius Temperature Luminosity Magnitude

GCVS	m_Name	SpType	Mass	e_Mass	Rad	e_Rad	logTeff	e_logTeff	logL	e_logL	VMAG	e_VMAG
RT And	A	F8V	1.240	0.030	1.260	0.015	3.785	0.015	0.290	0.060	4.00	0.15
RT And	B	K3	0.910	0.020	0.900	0.013	3.675	0.010	-0.435	0.040	6.20	0.20
BW Aqr	B	F7V	1.390	0.020	1.790	0.040	3.810	0.007	0.700	0.030	2.98	0.09

(and their measurement errors)

First steps

PARIS LODRON UNIVERSITÄT SALZBURG

```r
library(dplyr)     # Data manipulation
library(lubridate)     # Date-time manipulation
library(plotly)     # Interactive plots
library(orca)     # Static image export
library(magrittr)     # Pipe operator
library(reticulate)     # Interface with Python
library(kaleido)     # Static image export
```

0.) Included libraries

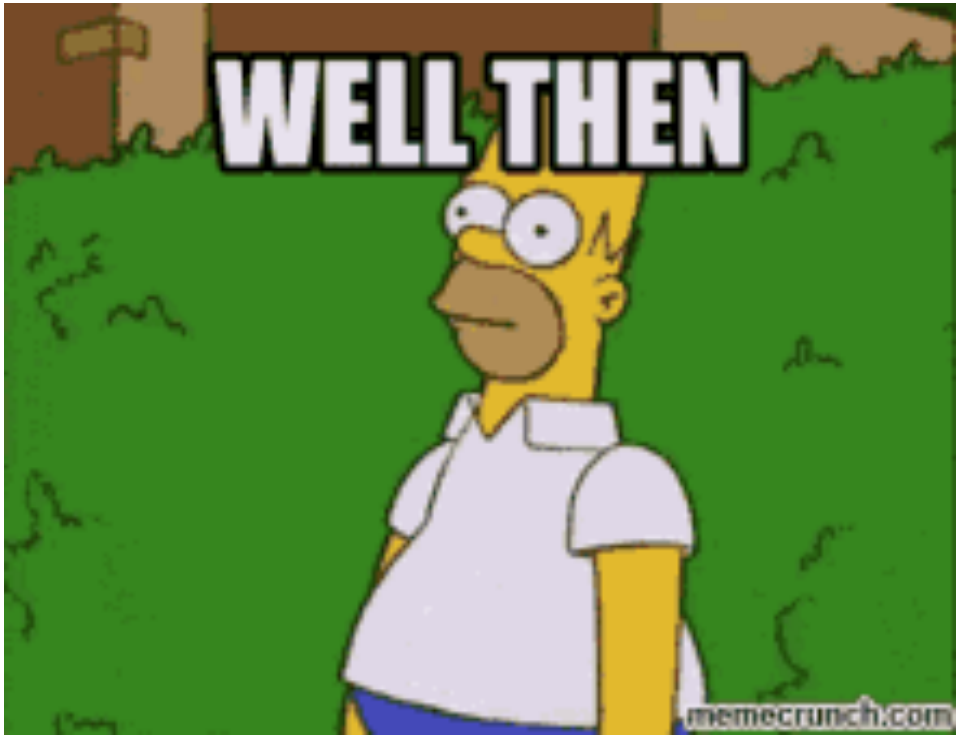
1.) Read
2) Drop empty values
3.) Stored in correct format

```r
# Read the CSV file
solar <- read.csv("C:/Users/weisz/Downloads/solar.csv")

# Remove rows with missing values
solar <- na.omit(solar)

# Step 2: Data Conversion
# Convert columns to numeric
solar$Mass <- as.numeric(solar$Mass)
solar$e_Mass <- as.numeric(solar$e_Mass)
solar$logL <- as.numeric(solar$logL)
solar$e_logL <- as.numeric(solar$e_logL)
solar$logTeff <- as.numeric(solar$logTeff)
solar$e_logTeff <- as.numeric(solar$e_logTeff)
solar$Rad <- as.numeric(solar$Rad)
solar$e_Rad <- as.numeric(solar$e_Rad)
solar$VMAG <- as.numeric(solar$VMAG)
solar$e_VMAG <- as.numeric(solar$e_VMAG)
solar$Teff <- as.numeric(solar$Teff)
```

Let's see the code!



Introduction to ggplot2:

PARIS LODRON UNIVERSITÄT SALZBURG

What is ggplot2?

- ggplot2 is a system for declaratively creating graphics, based on *The Grammar of Graphics*.

- You provide the data, tell ggplot2 how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.

The grammar of graphics consists of several key components

- **Data**: Start with your dataset, which contains the variables you want to visualize.

- **Aesthetic Mapping**: Map variables in your dataset to visual properties of the plot, such as x and y positions, colors, shapes, and sizes.

- **Geometric Objects (Geoms)**: Choose geometric objects, or geoms, to represent your data. Geoms define the type of plot you want to create, such as points, lines, bars, or polygons.

- **Statistical Transformations (Stats)**: Apply statistical transformations to your data, such as aggregations or transformations, before plotting.

- **Scales**: Customize the scales of your plot, including axis scales and legends.

- **Facets**: Divide your data into subsets and create multiple plots (facets) based on these subsets.

- **Themes**: Apply visual themes to customize the appearance of your plot, including background colors, gridlines, and fonts.

Relation between ggplot and plotly.

ggplot2 was developed by Hadley Wickham(2007)

ggplot2 + plotly => ggplotly !!!

ggplotly() returns a plotly object.

Example of ggplot2:

library(ggplot2) # Create g ggplot2

aes() - aesthetics

Geom_point is a function used add points to a plot

static_plot <- ggplot(data, aes(x = x_var, y = y_var)) + geom_point() +

labs(title = "Static Plot Title",

labs() - labels

x = "X-axis Label", y = "Y-axis Label")

Conversion of static to interactive:

Converts static into interactive!

```
interactive_plot <- ggplotly(static_plot)
print(interactive_plot )
```

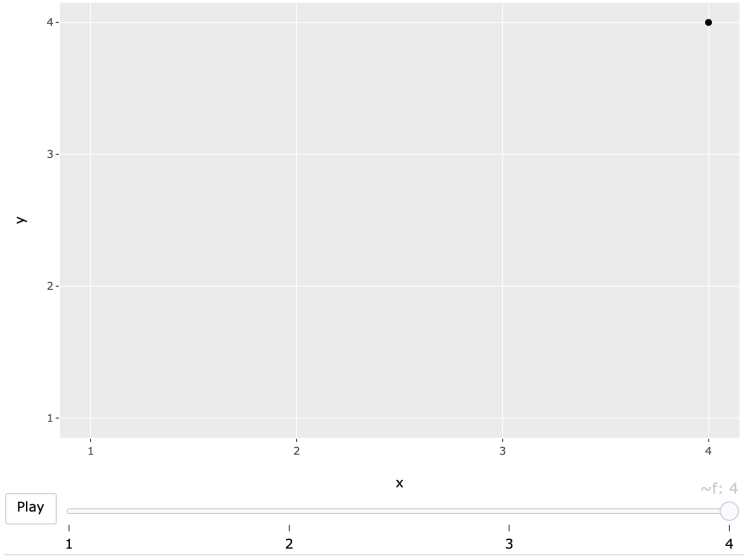
Quick Peek into ggplotly:

```r
library(plotly)
library(ggplot2)

df <- data.frame(
  x = c(1,2,4,4),
  y = c(1,2,3,4),
  f = c(1,2,3,4)
)

p <- ggplot(df, aes(x, y)) +
  geom_point(aes(frame = f))

ggplotly(p)
```



Introduction to GAPMINDER dataset

PARIS LODRON UNIVERSITÄT SALZBURG

	country	continent	year	lifeExp	pop	gdpPercap
1	Afghanistan	Asia	1952	28.801	8425333	779.4453
2	Afghanistan	Asia	1957	30.332	9240934	820.8530
3	Afghanistan	Asia	1962	31.997	10267083	853.1007
4	Afghanistan	Asia	1967	34.020	11537966	836.1971
5	Afghanistan	Asia	1972	36.088	13079460	739.9811
6	Afghanistan	Asia	1977	38.438	14880372	786.1134
7	Afghanistan	Asia	1982	39.854	12881816	978.0114
8	Afghanistan	Asia	1987	40.822	13867957	852.3959
9	Afghanistan	Asia	1992	41.674	16317921	649.3414
10	Afghanistan	Asia	1997	41.763	22227415	635.3414
11	Afghanistan	Asia	2002	42.129	25268405	726.7341
12	Afghanistan	Asia	2007	43.828	31889923	974.5803

It provides data about the population, life expectancy and GDP in different countries of the world from 1952 to 2007

- Now let's move to exercises!!!!
  - Please copy the codes available on the website.

Mulitple Trace Animations

PARIS LODRON UNIVERSITÄT SALZBURG

X

Y

Specify color by

Size = point size
frame = animation frame
Ids = identification

```
p <- ggplot(gapminder, aes(gdpPercap, lifeExp, color = continent)) +
  geom_point(aes(size = pop, frame = year, ids = country)) +
  scale_x_log10()

ggplotly(p)
```

This line scales the x-axis using a logarithmic scale. It's used to adjust the scale of the x-axis to better visualize data that spans a wide range of values.

NOTE: ‘scale_x_log10()’ there are different scaling functions

Add Animation Options

```
p <- ggplot(gapminder, aes(gdpPercap, lifeExp, color = continent)) +
  geom_point(aes(size = pop, frame = year, ids = country)) +
  scale_x_log10()

fig <- ggplotly(p)

fig <- fig %>%
  animation_opts(
    1000, easing = "elastic", redraw = FALSE
  )

fig
```

To add options to the plot, first convert ggplot2 plot to plotly variable with ggplotly() and then, add options to that variable.

Add Button Options

```
p <- ggplot(gapminder, aes(gdpPercap, lifeExp, color = continent)) +
  geom_point(aes(size = pop, frame = year, ids = country)) +
  scale_x_log10()

fig <- ggplotly(p)

fig <- fig %>%
  animation_opts(
    1000, easing = "elastic", redraw = FALSE
  )

fig <- fig %>%
  animation_button(
    x = 1, xanchor = "right", y = 0, yanchor = "bottom"
  )

fig
```

Let’s move to EXERCISE!!!

References:

- https://plotly-r.com/improving-ggplotly
- https://plotly.com/ggplot2/getting-started/
- https://plotly.com/r/
- https://plotly.com/ggplot2/animations/#basic-example

Questions

Thank You for Your Attention !

Appendix (plotly codes)

Plotly

Matplotlib-lookout
2020-02-25

First steps

```
pip3 install plotly
pip3 install https://dash.plotly.com/dash-core-components.zip
plotly_orca
npm install -g orca
```

```
import numpy as np
import plotly.express as px
df = px.data.iris()
px.scatter(df, x="petal_length", y="petal_width")
px.scatter(df, x="petal_length", y="petal_width", color="species")
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length")
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species")
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species")
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species", facet_col="species")
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species", facet_col="species", trendline="ols")
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species", facet_col="species", trendline="lowess")
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species", facet_col="species", trendline="lowess", marginal_x="rug")
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species", facet_col="species", trendline="lowess", marginal_x="rug", marginal_y="rug")
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species", facet_col="species", trendline="lowess", marginal_x="rug", marginal_y="rug", log_x=True)
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species", facet_col="species", trendline="lowess", marginal_x="rug", marginal_y="rug", log_x=True, log_y=True)
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species", facet_col="species", trendline="lowess", marginal_x="rug", marginal_y="rug", log_x=True, log_y=True, log_z=True)
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species", facet_col="species", trendline="lowess", marginal_x="rug", marginal_y="rug", log_x=True, log_y=True, log_z=True, log_color=True)
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species", facet_col="species", trendline="lowess", marginal_x="rug", marginal_y="rug", log_x=True, log_y=True, log_z=True, log_color=True, log_size=True)
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species", facet_col="species", trendline="lowess", marginal_x="rug", marginal_y="rug", log_x=True, log_y=True, log_z=True, log_color=True, log_size=True, log_hover=True)
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species", facet_col="species", trendline="lowess", marginal_x="rug", marginal_y="rug", log_x=True, log_y=True, log_z=True, log_color=True, log_size=True, log_hover=True, log_animation=True)
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species", facet_col="species", trendline="lowess", marginal_x="rug", marginal_y="rug", log_x=True, log_y=True, log_z=True, log_color=True, log_size=True, log_hover=True, log_animation=True, log_frame=True)
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species", facet_col="species", trendline="lowess", marginal_x="rug", marginal_y="rug", log_x=True, log_y=True, log_z=True, log_color=True, log_size=True, log_hover=True, log_animation=True, log_frame=True, log_facet_row=True)
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species", facet_col="species", trendline="lowess", marginal_x="rug", marginal_y="rug", log_x=True, log_y=True, log_z=True, log_color=True, log_size=True, log_hover=True, log_animation=True, log_frame=True, log_facet_row=True, log_facet_col=True)
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species", facet_col="species", trendline="lowess", marginal_x="rug", marginal_y="rug", log_x=True, log_y=True, log_z=True, log_color=True, log_size=True, log_hover=True, log_animation=True, log_frame=True, log_facet_row=True, log_facet_col=True, log_trendline=True)
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species", facet_col="species", trendline="lowess", marginal_x="rug", marginal_y="rug", log_x=True, log_y=True, log_z=True, log_color=True, log_size=True, log_hover=True, log_animation=True, log_frame=True, log_facet_row=True, log_facet_col=True, log_trendline=True, log_marginal_x=True)
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species", facet_col="species", trendline="lowess", marginal_x="rug", marginal_y="rug", log_x=True, log_y=True, log_z=True, log_color=True, log_size=True, log_hover=True, log_animation=True, log_frame=True, log_facet_row=True, log_facet_col=True, log_trendline=True, log_marginal_x=True, log_marginal_y=True)
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species", facet_col="species", trendline="lowess", marginal_x="rug", marginal_y="rug", log_x=True, log_y=True, log_z=True, log_color=True, log_size=True, log_hover=True, log_animation=True, log_frame=True, log_facet_row=True, log_facet_col=True, log_trendline=True, log_marginal_x=True, log_marginal_y=True, log_facet_row=True)
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species", facet_col="species", trendline="lowess", marginal_x="rug", marginal_y="rug", log_x=True, log_y=True, log_z=True, log_color=True, log_size=True, log_hover=True, log_animation=True, log_frame=True, log_facet_row=True, log_facet_col=True, log_trendline=True, log_marginal_x=True, log_marginal_y=True, log_facet_row=True, log_facet_col=True)
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species", facet_col="species", trendline="lowess", marginal_x="rug", marginal_y="rug", log_x=True, log_y=True, log_z=True, log_color=True, log_size=True, log_hover=True, log_animation=True, log_frame=True, log_facet_row=True, log_facet_col=True, log_trendline=True, log_marginal_x=True, log_marginal_y=True, log_facet_row=True, log_facet_col=True, log_marginal_x=True)
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species", facet_col="species", trendline="lowess", marginal_x="rug", marginal_y="rug", log_x=True, log_y=True, log_z=True, log_color=True, log_size=True, log_hover=True, log_animation=True, log_frame=True, log_facet_row=True, log_facet_col=True, log_trendline=True, log_marginal_x=True, log_marginal_y=True, log_facet_row=True, log_facet_col=True, log_marginal_x=True, log_marginal_y=True)
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species", facet_col="species", trendline="lowess", marginal_x="rug", marginal_y="rug", log_x=True, log_y=True, log_z=True, log_color=True, log_size=True, log_hover=True, log_animation=True, log_frame=True, log_facet_row=True, log_facet_col=True, log_trendline=True, log_marginal_x=True, log_marginal_y=True, log_facet_row=True, log_facet_col=True, log_marginal_x=True, log_marginal_y=True, log_facet_row=True)
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species", facet_col="species", trendline="lowess", marginal_x="rug", marginal_y="rug", log_x=True, log_y=True, log_z=True, log_color=True, log_size=True, log_hover=True, log_animation=True, log_frame=True, log_facet_row=True, log_facet_col=True, log_trendline=True, log_marginal_x=True, log_marginal_y=True, log_facet_row=True, log_facet_col=True, log_marginal_x=True, log_marginal_y=True, log_facet_row=True, log_facet_col=True)
px.scatter(df, x="petal_length", y="petal_width", color="species", size="sepal_length", hover_name="species", facet_row="species", facet_col="species", trendline="lowess", marginal_x="rug", marginal_y="rug", log_x=True, log_y=True, log_z=True, log_color=True, log_size=True, log_hover=True, log_animation=True, log_frame=True, log_facet_row=True, log_facet_col=True, log_trendline=True, log_marginal_x=True, log_marginal_y=True, log_facet_row=True, log_facet_col
```