

PEC2

Estudio de técnicas de visualización de datos

Enrique Callejas Castro

Visualización de datos

Máster Universitario de Ciencia de Datos

Primera parte: Pyramid Chart



Datos

Descripción técnica

- **Contenido:** frecuencia de uso de los lenguajes de programación en GitHub.
- **Procesamiento:** se filtraron los datos de 2021 y se agruparon los *issues* por lenguaje.
- **Tipo:** frecuencias.
- **Fuente:** GitHub Programming Languages Data (Wen, 2022)



```

match: function(e, t, n) {
  var r, i = 0;
  n = n.length;
  n = Math.min(n, 100);
  if (n) {
    if (n > 1; i++) {
      if (r = t.call(e[i], n), r === !1) break;
    } else {
      if (r = t.call(e[i], n), r === !1) break;
    }
  }
  if (i > 1; i++) {
    if (r = t.call(e[i], i, e[i]), r === !1) break;
  }
  if (i > 1; i++) {
    if (r = t.call(e[i], i, e[i]), r === !1) break;
  }
  return e;
},
trim: function(e) {
  return null == e ? "" : e.trim();
},
function(e) {
  return null == e ? "" : (e + "").replace(/ /g, "");
},
},
merge: function(e, t) {
  return null != e && (t instanceof Object) ? t.merge(e, "string" == typeof e ? [e] : e) : t.call(e, t), e;
},
indexOf: function(e, t, n) {
  var r;
  if (n) {
    if (n) return e.call(t, n, t);
    for (r = 0; r < e.length; r++) {
      if (t === e[r]) return r;
    }
  }
  return -1;
}

```

Elaboración

Carga de los datos

```
df = pd.read_csv("issues.csv")
df = df[df['year']==2021]

df = pd.DataFrame(df.groupby('name')['count'].sum().nlargest(10))
df.reset_index(inplace=True)
df["count"] = df["count"]//1000

display(df)
```

Fuente: elaboración propia

	name	count
0	Python	328
1	JavaScript	275
2	Java	228
3	TypeScript	190
4	Go	172
5	C++	168
6	PHP	117
7	C#	102
8	C	90
9	Ruby	48

Elaboración

Creación de la figura

```
# Creación de la paleta de colores
cmap = cm.get_cmap('Pastell', 10)
color_list = [matplotlib.colors.rgb2hex(cmap(i)[:3]) for i in range(cmap.N)]

# Creación de la figura
fig = go.Figure(go.Funnel(
    y = df["name"].to_list(),
    x = df["count"].to_list(),
    textposition = "inside",
    textinfo = "label+value",
    opacity = 0.85, marker = {"color": color_list}))

fig.update_yaxes(visible=False, showticklabels=False)
fig.show()
```

Fuente: elaboración propia

Visualización

Frecuencia de uso (en miles de *issues*) de lenguajes de programación en GitHub: Top 10 en 2021

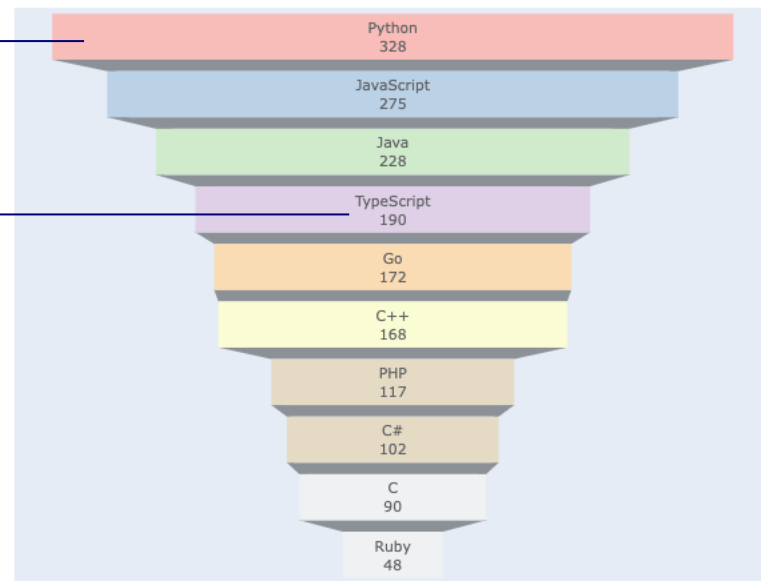
barras de frecuencia

etiqueta:

[1] lenguaje de programación

valor:

[2] miles de *issues* en dicho lenguaje

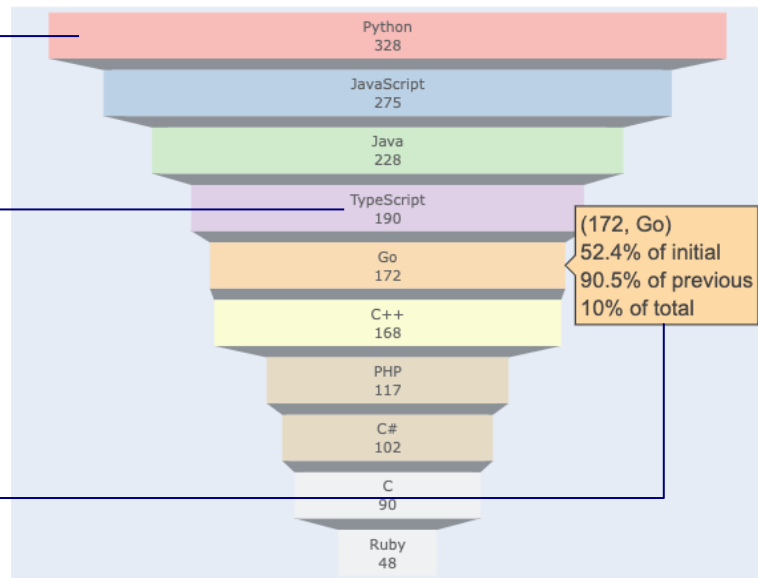


Herramienta para realizar la visualización: Librería *plotly* de Python

Visualización

Frecuencia de uso (en miles de *issues*) de lenguajes de programación en GitHub: Top 10 en 2021

- barras de frecuencia**
- etiqueta:**
[1] lenguaje de programación
valor:
[2] miles de *issues* en dicho lenguaje
- información adicional:**
porcentajes acumulados



Herramienta para realizar la visualización: Librería *plotly* de Python

Segunda parte: Gauge diagram



Datos

Descripción técnica

- **Contenido:** resultados de las elecciones municipales de San Cristóbal de La Laguna en las elecciones de 2015 y 2019.
- **Procesamiento:** se cargaron los datos del archivo .xml a un fichero .xlsx.
- **Tipo:** frecuencias.
- **Fuente:** Resultados de las elecciones municipales (El País, 2019)



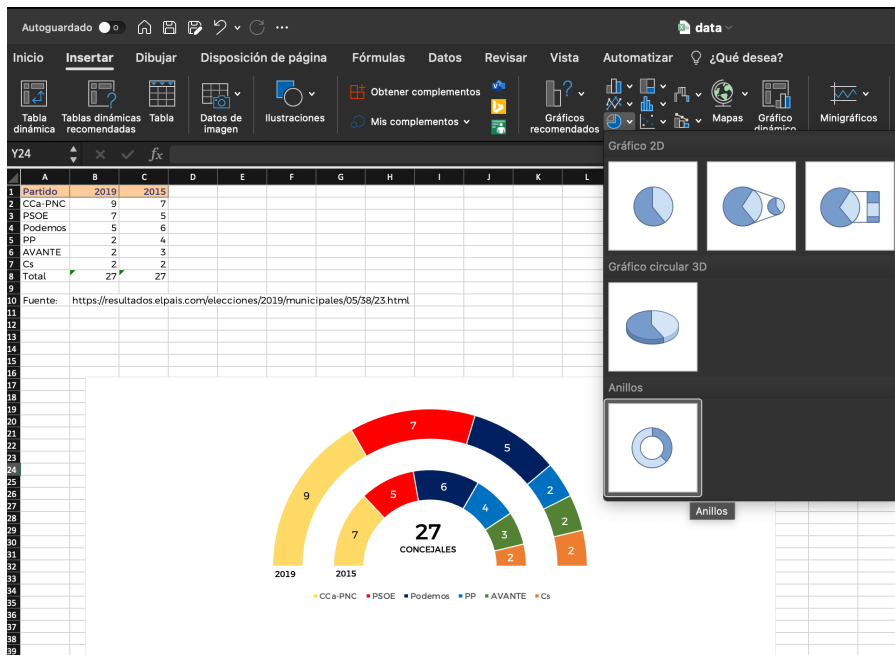
Elaboración

Carga de los datos

	A	B	C	D
1	Partido	2019	2015	
2	CCa-PNC	9	7	
3	PSOE	7	5	
4	Podemos	5	6	
5	PP	2	4	
6	AVANTE	2	3	
7	Cs	2	2	
8	Total	27	27	
9				

Elaboración

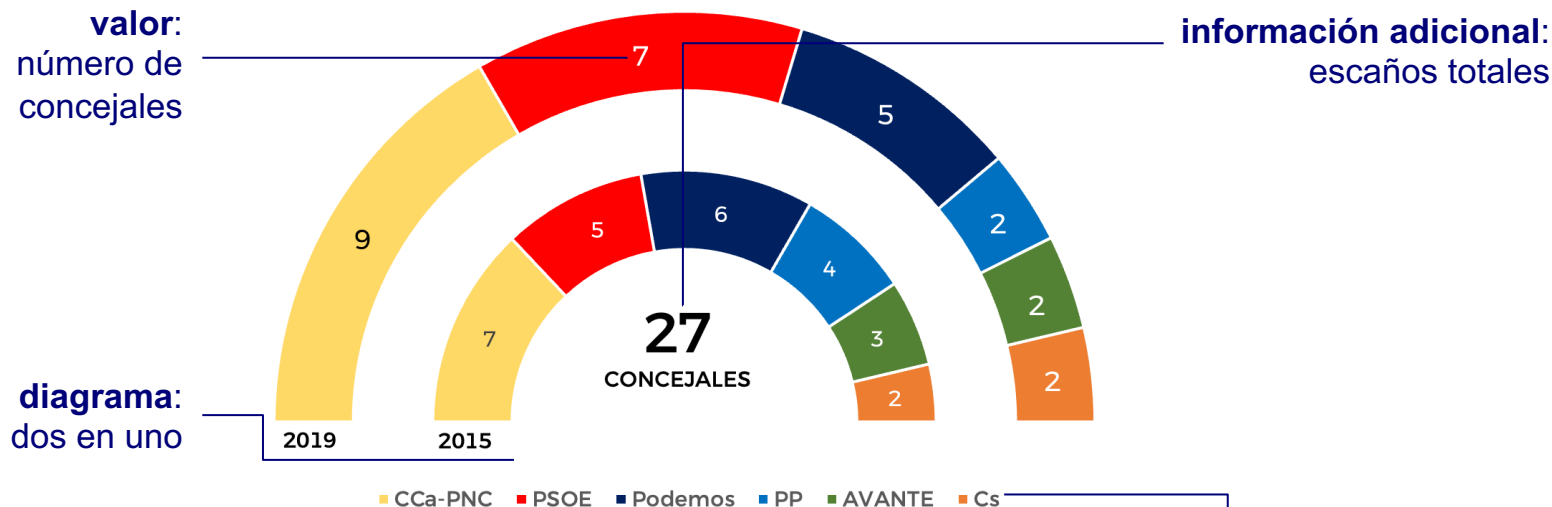
Creación de la figura



Fuente: elaboración propia

Visualización

Resultados de las elecciones municipales de San Cristóbal de La Laguna

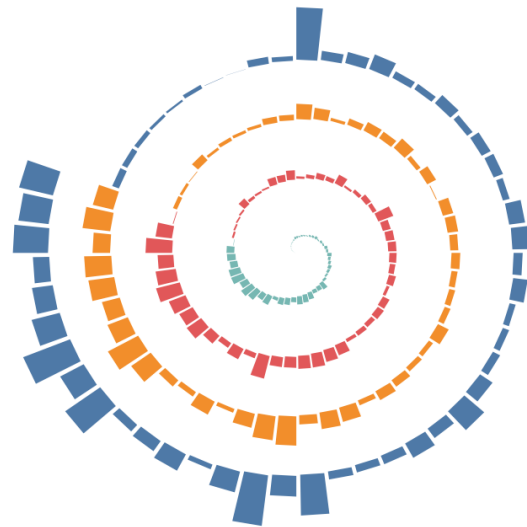


leyenda:

los partidos se presentan
mediante colores

Herramienta para realizar la visualización: Microsoft Excel

Tercera parte: Spiral Plot



Datos

Descripción técnica

- **Contenido:** número de casos de COVID-19 entre 2020 y 2022 en España.
- **Procesamiento:** se cargaron los datos del archivo .csv y se filtraron por país.
- **Tipo:** frecuencias.
- **Fuente:** Our World in Data (OWID, 2022).



Elaboración

Carga de los datos

```

{r}
covid <- read.csv("owid-covid-data.csv", stringsAsFactors = FALSE)
covid

```

Description: df [233,048 × 67]

iso_code <chr>	continent <chr>	location <chr>	date <chr>	total_cases <dbl>	new_cases <dbl>	new_cases_smoothed <dbl>	total_deaths <dbl>	new_deaths <dbl>	new_deaths_smoothed <dbl>
AFG	Asia	Afghanistan	2020-02-24	5	5	NA	NA	NA	NA
AFG	Asia	Afghanistan	2020-02-25	5	0	NA	NA	NA	NA
AFG	Asia	Afghanistan	2020-02-26	5	0	NA	NA	NA	NA
AFG	Asia	Afghanistan	2020-02-27	5	0	NA	NA	NA	NA
AFG	Asia	Afghanistan	2020-02-28	5	0	NA	NA	NA	NA
AFG	Asia	Afghanistan	2020-02-29	5	0	0.714	NA	NA	NA
AFG	Asia	Afghanistan	2020-03-01	5	0	0.714	NA	NA	NA
AFG	Asia	Afghanistan	2020-03-02	5	0	0.000	NA	NA	NA
AFG	Asia	Afghanistan	2020-03-03	5	0	0.000	NA	NA	NA
AFG	Asia	Afghanistan	2020-03-04	5	0	0.000	NA	NA	NA

1-10 of 233,048 rows | 1-10 of 67 columns

Previous **1** 2 3 4 5 6 ... 100 Next

Elaboración

Creación de la figura

```

{r}
# Initialize.
spiral_initialize_by_time(xlim=c("2020-01-01 00:00:00", "2022-10-01 00:00:00"),
  unit_on_axis = "days", period="years",
  start=90, end=(709/365)*360+(28/365)*360+90,
  flip="horizontal")

# Create the track.
spiral_track(ylim=c(0, ymax*.7),
  background=FALSE, background_gp = gpar(col = NA, fill = NA))

# Use a polygon.
spiral_polygon(x=c(spain$dt, rev(spain$dt)),
  y=c(spain$new_cases_smoothed/2, -rev(spain$new_cases_smoothed/2)),
  gp = gpar(col="#d32e2b", fill="#d32e2b50"))

# Middle baseline.
spiral_lines(x=spain$dt, y=0)

# Text.
spiral_text(x="2020-01-01", y=50000, text="2020",
  just = "right",
  gp=gpar(cex=1, fontfamily="Courier"))
spiral_text(x="2021-01-01", y=50000, text="2021",
  just = "right",
  gp=gpar(cex=1, fontfamily="Courier"))
spiral_text(x="2022-01-01", y=50000, text="2022",
  just = "right",
  gp=gpar(cex=1, fontfamily="Courier"))

```

Fuente: Flowing Data

Visualización

Casos de COVID-19 en España

ejes la espiral:

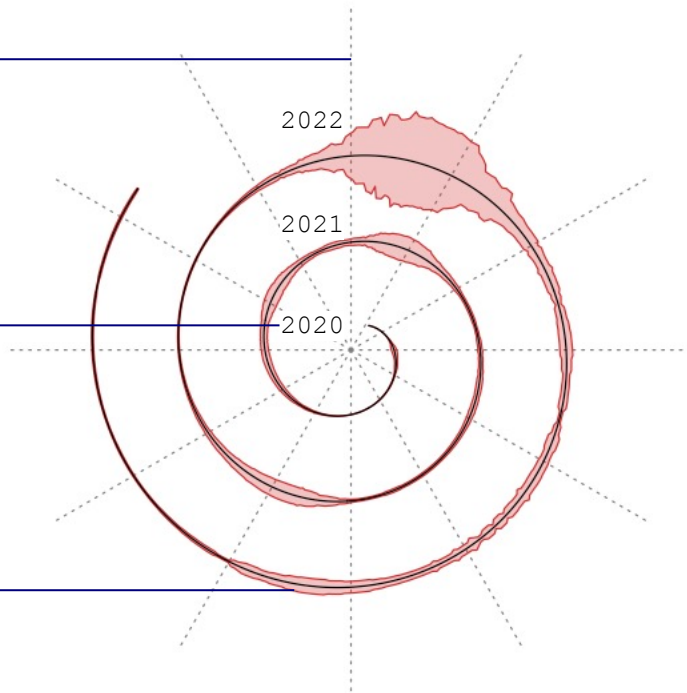
cada eje
representa un mes

texto:

etiqueta que marca
cada año, a su paso
por el eje de enero

espiral:

forma que refleja la
frecuencia de casos



Herramienta para realizar la visualización: Librería *spiralize* de R

Visualización

Casos de COVID-19 en China

ejes la espiral:

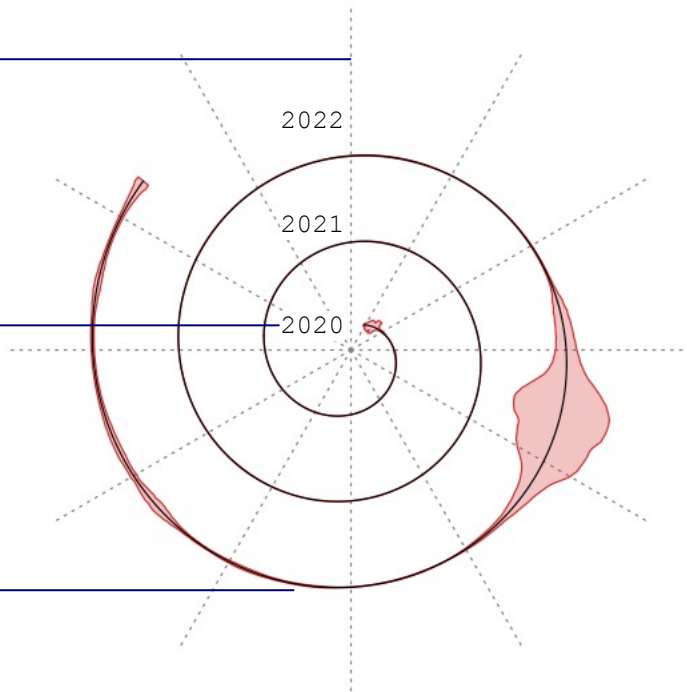
cada eje
representa un mes

texto:

etiqueta que marca
cada año, a su paso
por el eje de enero

espiral:

forma que refleja la
frecuencia de casos



Herramienta para realizar la visualización: Librería *spiralize* de R

Referencias

- El País. (2019). *Resultados de las elecciones municipales*. Recuperado de <https://resultados.elpais.com/elecciones/2019/municipales/05/38/23.html>
- Flowing Data. (2022). *A Quick and Easy Way to Make Spiral Charts in R*. Recuperado de <https://flowingdata.com/2022/01/10/a-quick-and-easy-way-to-make-spiral-charts-in-r/>
- OWID. (2022). *COVID-19 data*. Recuperado de <https://github.com/owid/covid-19-data/blob/master/public/data/owid-covid-data.csv>
- Wen, I. (2022). *GitHub Programming Languages Data*. Recuperado de <https://www.kaggle.com/datasets/isaacwen/github-programming-languages-data>

Visualizaciones disponibles en: https://github.com/enriquecallejascastro/visualizacion_de_datos

PEC2

Estudio de técnicas de visualización de datos

Enrique Callejas Castro

Visualización de datos

Máster Universitario de Ciencia de Datos
