



---

# Visualización de Información y Analítica Visual

— Hernán Valdivieso López ([hfvaldivieso@uc.cl](mailto:hfvaldivieso@uc.cl)) —

---

# Resumen clase 6

# Validación de visualización

Aplicando el *framework* de Tamara Munzner... *what, why y how*



## Domain situation

Situación de dominio



## Data/task abstraction

Abstracción de datos y tareas



## Visual encoding/interaction idiom

Codificación visual e interacciones



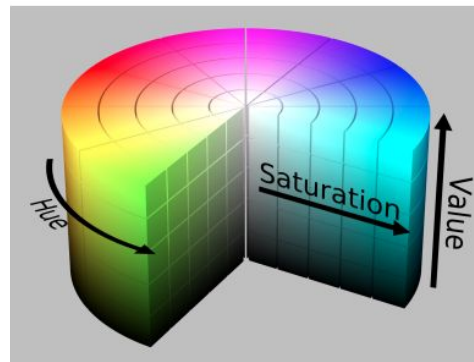
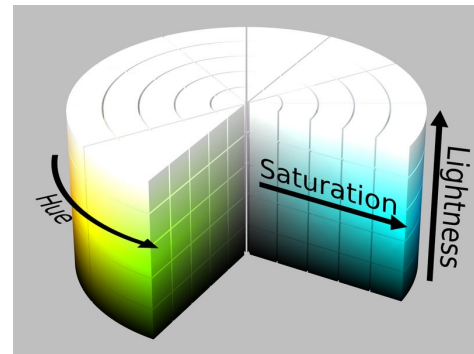
## Algorithm

Algoritmo

# Modelos de color - HSL y HSV

Es más intuitivo para nuestra percepción y es usado fuertemente por artistas y diseñadores.

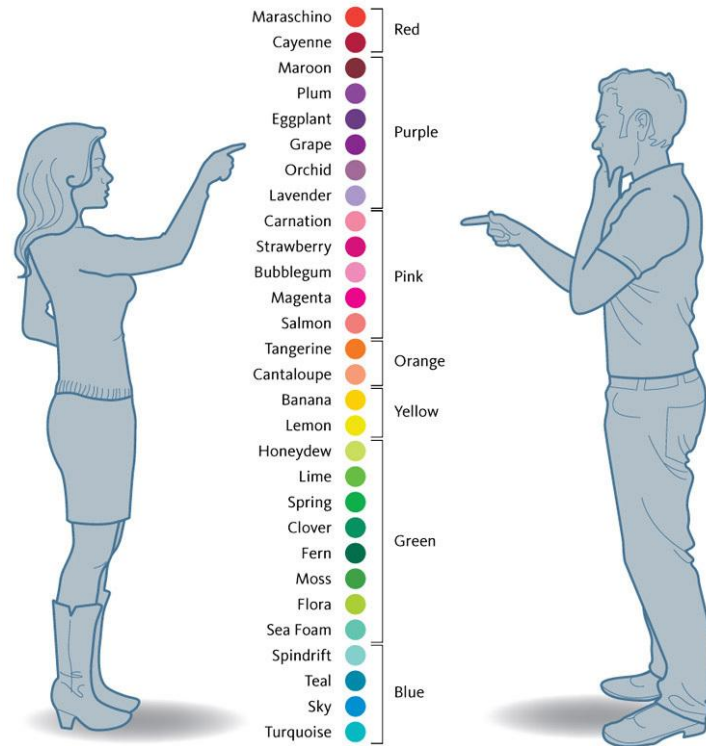
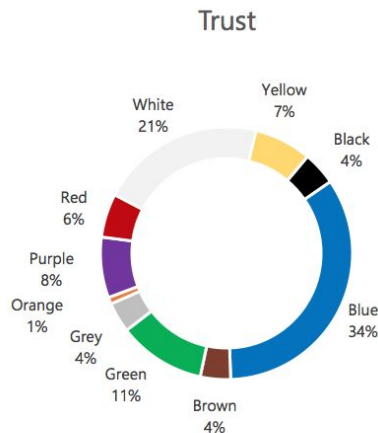
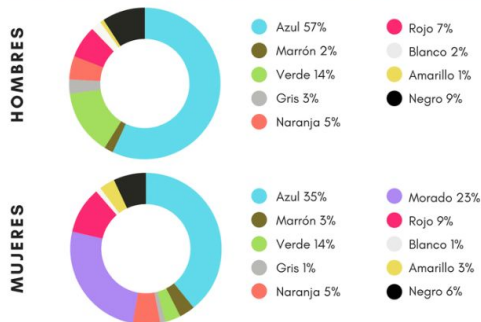
- **Matiz (*hue*)** captura la tonalidad del color. Normalmente conocemos como colores puros, dejando de lado la mezcla del blanco y del negro.
- **Saturación (*saturation*)** especifica la intensidad del color. Qué tan “vivo” o “puro” está.
- **Luminosidad (*lightness*)** especifica la cantidad de luz que recibe el color. Análogo a una linterna.
- **Valor (*value*)** especifica que tan “brillante” es el color. 0% es que no brilla nada (será negro) y 100% implica que brilla totalmente (no hay nada de negro en el color).



# Psicología y percepción del color

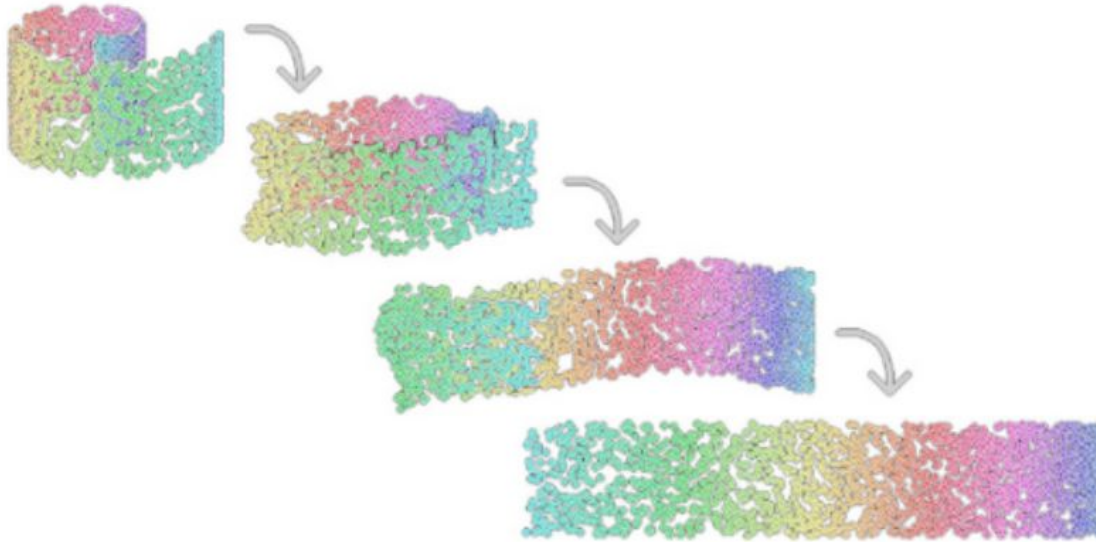


## COLORES FAVORITOS ENTRE HOMBRES Y MUJERES



# Reducción de dimensionalidad

- Transformación de datos multidimensionales a datos de pocas dimensiones sin perder las caracterizaciones originales.
- Son un campo de estudio importante en *data science* y estadística.



# Clase 7: Streamlit

# Temas de la clase - Streamlit

1. ¿Qué es Streamlit?
2. Funciones básicas/Demo.
3. Ejemplos aplicados





# ¿Qué es Streamlit?

---

# ¿Qué es Streamlit?

- Streamlit es una librería open-source de Python, que nos permite crear aplicaciones web.
- Utilizada popularmente para *machine learning* y *data science*.
  - Puedes entrenar un modelo y dejarlo disponible para su uso.
- Su ventaja es que no es necesario aprender tecnologías web (JavaScript) para hacer visualizaciones interactivas.
- Tiene compatibilidad con Altair, Matplotlib, plotly, folium\*, entre otros.
- Documentación: [Streamlit documentation](#)
- Excelente tutorial: [30 days of Streamlit](#).

# ¿Cómo disponibilizo mi *dashboard*?

- Se puede desplegar aplicaciones de forma gratuita en: [Streamlit Share](#).
- Basta con conectar un repositorio de GitHub.
-  **Importante** : si se usa librerías adicionales a Streamlit, se debe incluir un archivo **requirements.txt**.

[← Back](#)

## Deploy an app

Repository ⓘ

[Paste GitHub URL](#)

hernan4444/repo

Branch

master

Main file path

streamlit\_app.py

App URL (optional)

.streamlit.app

[Advanced settings...](#)

Deploy!

# Funciones básicas

---

# Dashboard de ejemplo: Airbnb

- Aprendemos diferentes componentes que aplicaremos para construir el siguiente dashboard <https://demo-uc.streamlit.app/>.
- Empezaremos desde la siguiente [plantilla](#).
- La solución final se puede encontrar [aquí](#).

## Airbnb

Este dashboard muestra información sobre diferentes Airbnb en 10 ciudades.

Ver datos

Cantidad mínima de camas



	anfitrión/a	tiempo_respuesta	es_superhost	ciudad	país	latitud	longitud
284	360,021,647	dentro de una hora	No	Istanbul	Turkey	41.017	28.955
460	14,875,223	dentro de una hora	No	Istanbul	Turkey	41.0372	28.981
544	118,021,719	dentro de una hora	No	Sydney	Australia	-33.8667	150.932
561	153,721	dentro de una hora	Si	Rio de Janeiro	Brazil	-22.9643	-43.175
1,599	6,114,680	dentro de algunas horas	No	Rome	Italy	41.8975	12.466
1,691	225,319,445	dentro de una hora	No	Sydney	Australia	-33.8191	150.988
2,238	15,488,907	dentro de una hora	No	Rio de Janeiro	Brazil	-22.9979	-43.414
2,283	85,893,416	dentro de una hora	No	Istanbul	Turkey	41.0362	28.986
2,365	57,213,517	dentro de algunos días o más	No	Hong Kong	China	22.3147	114.172
2,399	146,317,254	dentro de algunos días o más	No	Rome	Italy	41.8978	12.496

## Filtrar por país

Selecciona un país

Todos

## Mapa de todos los Airbnb

# Funciones básicas: Texto

Elementos para incluir textos en diferentes formatos → [Text elements - Streamlit Docs](#)

- Título de nuestra app

```
st.title('This is a title')  
st.title('_Streamlit_ is :blue[cool] :sunglasses:')
```

**This is a title**

***Streamlit* is cool 🕶️**

# Funciones básicas: Texto

Elementos para incluir textos en diferentes formatos → [Text elements - Streamlit Docs](#)

- Markdown

```
st.markdown("*Streamlit* is **really** ***cool***.")
```

```
st.markdown('''  
:red[Streamlit] :orange[can] :green[write] :blue[text] :violet[in]  
:gray[pretty] :rainbow[colors] and :blue-background[highlight] text.  
''')
```

*Streamlit* is **really** *cool*.

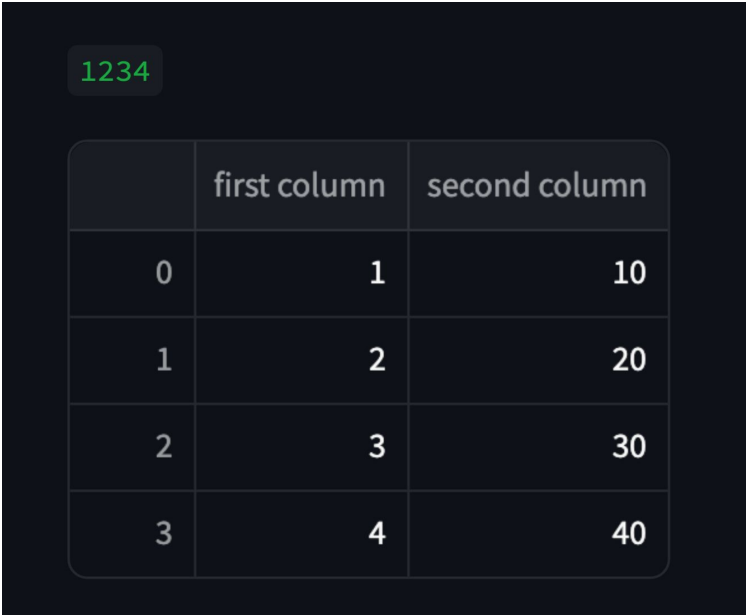
Streamlit can write text in pretty colors and highlight text.

# Funciones básicas: Texto

Elementos para incluir textos en diferentes formatos → [Text elements - Streamlit Docs](#)

- Texto y más (método "mágico")

```
data = pd.DataFrame({  
    'first column': [1, 2, 3, 4],  
    'second column': [10, 20, 30, 40],  
})  
st.write(1234)  
st.write(data)
```



The image shows a Streamlit interface with a dark background. At the top left, there is a green text element displaying the number '1234'. Below it, a table is displayed with two columns: 'first column' and 'second column'. The table has four rows of data, indexed from 0 to 3.

	first column	second column
0	1	10
1	2	20
2	3	30
3	4	40



# Funciones básicas: Texto

Elementos para incluir textos en diferentes formatos → [Text elements - Streamlit Docs](#)

- Otros

```
st.text('This is some text.')
```

```
st.latex(r'''  
a + ar + a r^2 + a r^3 + \cdots + a r^{n-1} =  
\sum_{k=0}^{n-1} ar^k =  
a \left(\frac{1-r^n}{1-r}\right)  
''')
```

This is some text.

$$a + ar + ar^2 + ar^3 + \dots + ar^{n-1} = \sum_{k=0}^{n-1} ar^k = a \left( \frac{1-r^n}{1-r} \right)$$

# Código en Python

Vamos al código  

Pongamos título y veamos el dataset

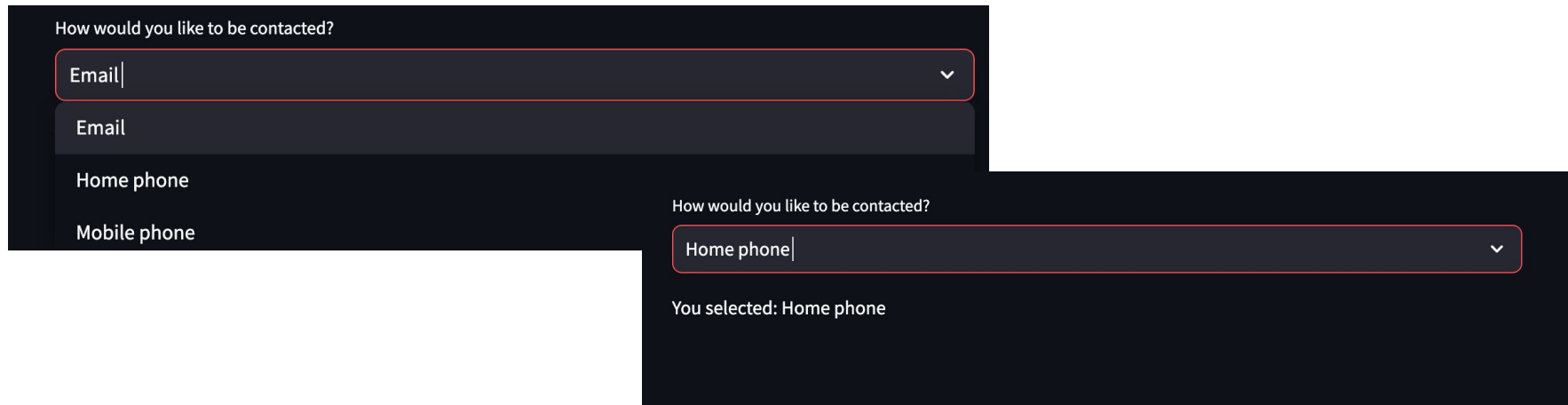
# Funciones básicas: Input

Elementos para incluir interactividad → [Input widgets - Streamlit Docs](#)

- Selectores

```
option = st.selectbox(
    "How would you like to be contacted?",
    ["Email", "Home phone", "Mobile phone"])
```

```
st.write("You selected:", option)
```



How would you like to be contacted?

Email

Email

Home phone

Mobile phone

How would you like to be contacted?

Home phone

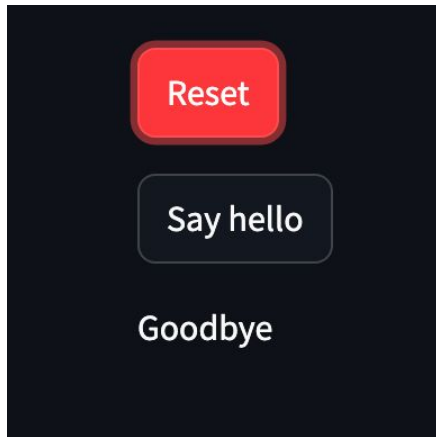
You selected: Home phone

# Funciones básicas: Input

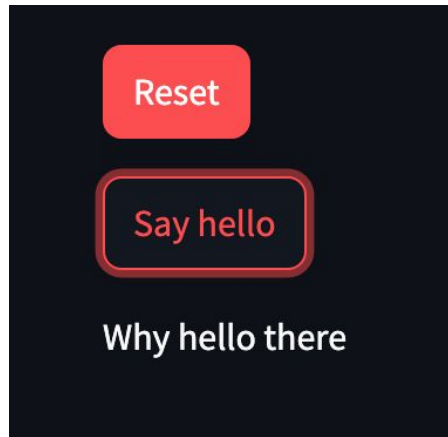
Elementos para incluir interactividad → [Input widgets - Streamlit Docs](#)

- Botón

```
st.button("Reset", type="primary")
if st.button("Say hello"):
    st.write("Why hello there")
else:
    st.write("Goodbye")
```



Presionar Reset



Presionar Say hello

# Funciones básicas: Input

Elementos para incluir interactividad → [Input widgets - Streamlit Docs](#)

- Slider

```
age = st.slider("How old are you?", 0, 130, 25)  
st.write("I'm ", age, "years old")
```

## Slider

How old are you?



I'm 71 years old.

# Código en Python

Vamos al código  

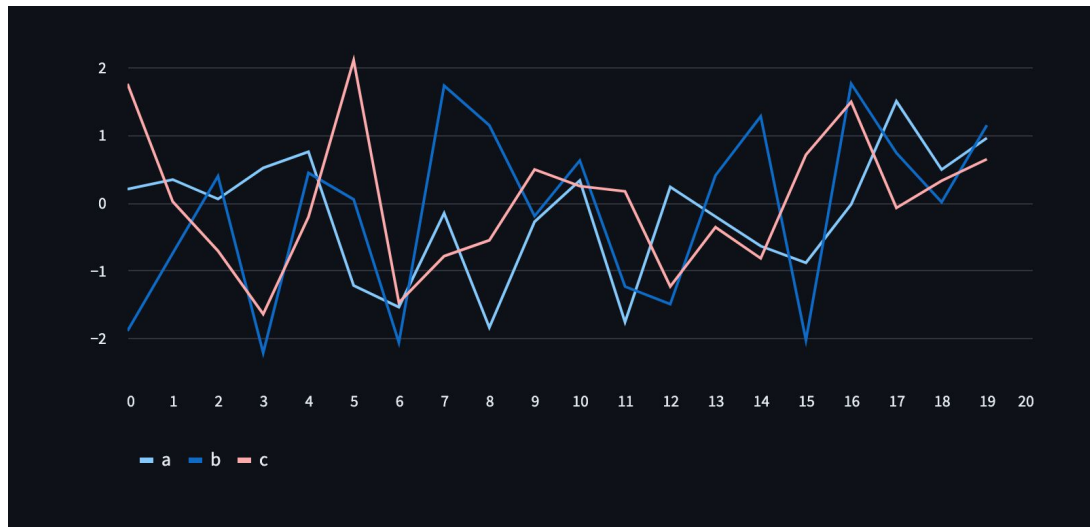
Incluyamos un filtrador por camas y conectemos al  
*dataset*

# Funciones básicas: Gráficos

Elementos para agregar visualizaciones → [Chart elements - Streamlit Docs](#)

- Gráficos directos de streamlit

```
chart_data = pd.DataFrame(np.random.randn(20, 3), columns=["a", "b", "c"])
st.line_chart(chart_data)
```



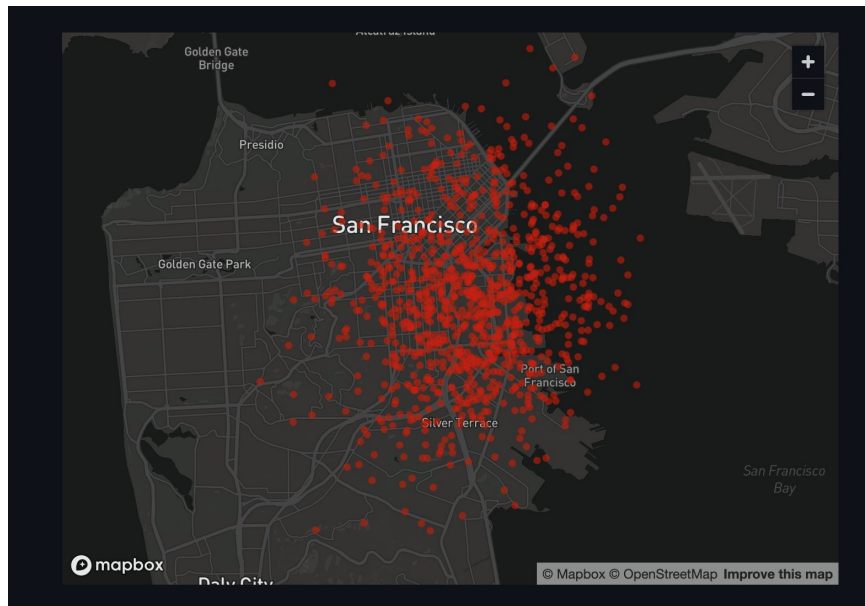
# Funciones básicas: Gráficos

Elementos para agregar visualizaciones → [Chart elements - Streamlit Docs](#)

- Mapa de puntos

```
df = pd.DataFrame(np.random.randn(1000, 2) / [50, 50] + [37.76, -122.4],  
                  columns=['lat', 'lon'])
```

```
st.map(df)
```





# Funciones básicas: Gráficos

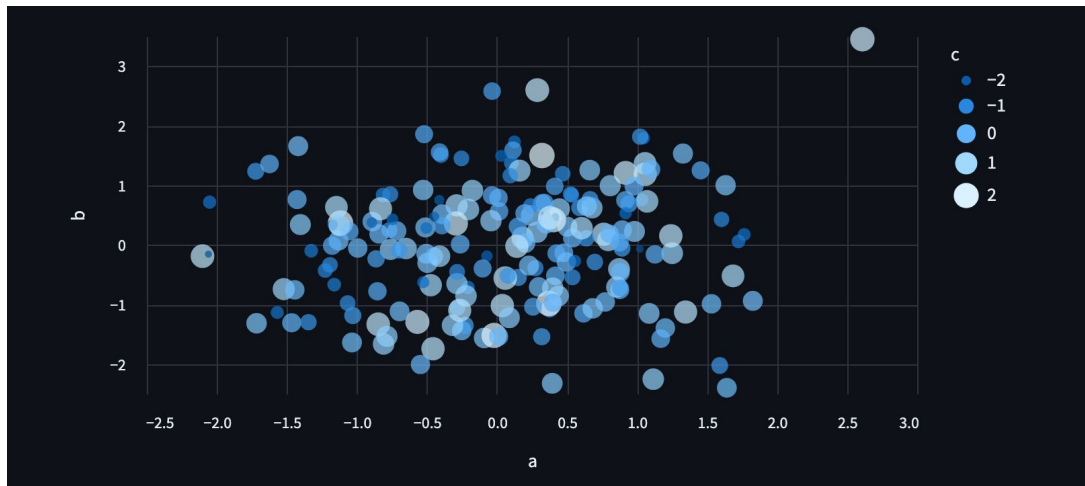
Elementos para agregar visualizaciones → [Chart elements - Streamlit Docs](#)

- Gráficos de Altair

```
chart_data = pd.DataFrame(np.random.randn(20, 3), columns=["a", "b", "c"])
```

```
c = alt.Chart(chart_data).mark_circle()  
    .encode(x="a", y="b", size="c", color="c", tooltip=["a", "b", "c"])
```

```
st.altair_chart(c)
```





# Funciones básicas: Layout

Elementos para definir el layout → [Layouts and Containers - Streamlit Docs](#)

- Columnas

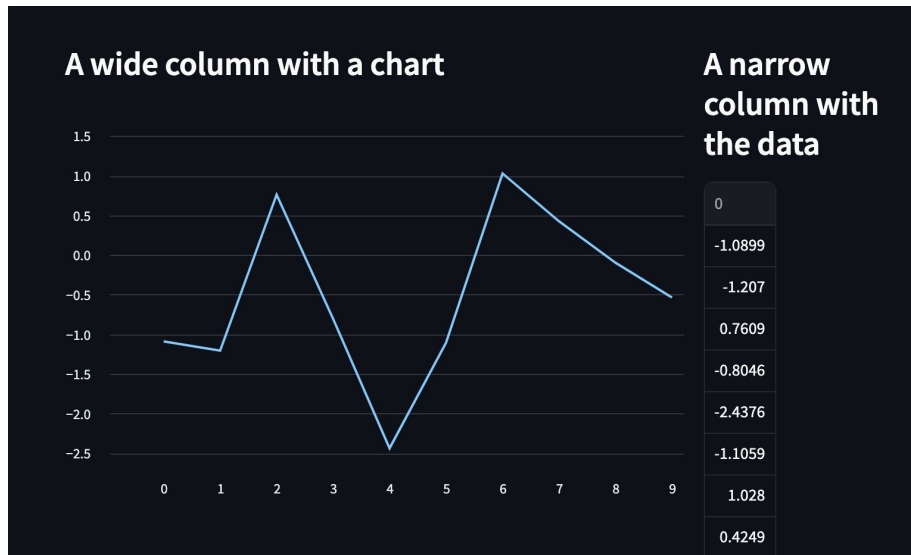
```
col1, col2 = st.columns([3, 1]) # Columna 1 es 3 veces más grande que columna 2
data = [-1.08, -1.20, 0.76, -0.80, -2.43, -1.10, 1.02, 0.42, -0.09, -0.53]
```

```
col1.subheader("A wide column  
with a chart")
```

```
col1.line_chart(data)
```

```
col2.subheader("A narrow  
column with  
the data")
```

```
col2.write(data)
```



# Código en Python

Vamos al código  

Agreguemos un mapa y gráficos al *dashboard*

# Ejemplos Aplicados

---

# Ejemplos Aplicados

- [Sophisticated Palette](#)
- [Streamlit theme for Plotly charts!](#)
- [JULO improves financial inclusion in Indonesia with Streamlit](#)
- [Wissam Siblini uses Streamlit for pathology detection in chest radiographs](#)
- [How Delta Dental uses Streamlit to make lightning-fast decisions](#)

# Taller evaluado

(Reducción de  
Dimensionalidad)

Desarrollo en clases

---



---

# Visualización de Información y Analítica Visual

— Hernán Valdivieso López ([hfvaldivieso@uc.cl](mailto:hfvaldivieso@uc.cl)) —

---