

“LA LUNA DEJA UN HISTOGRAMA ABANDONADO EN EL AIRE”

Una introducción a la visualización
de textos en Python con Vega-lite y
Altair de la mano de F. García Lorca

Alejandro Benito-Santos

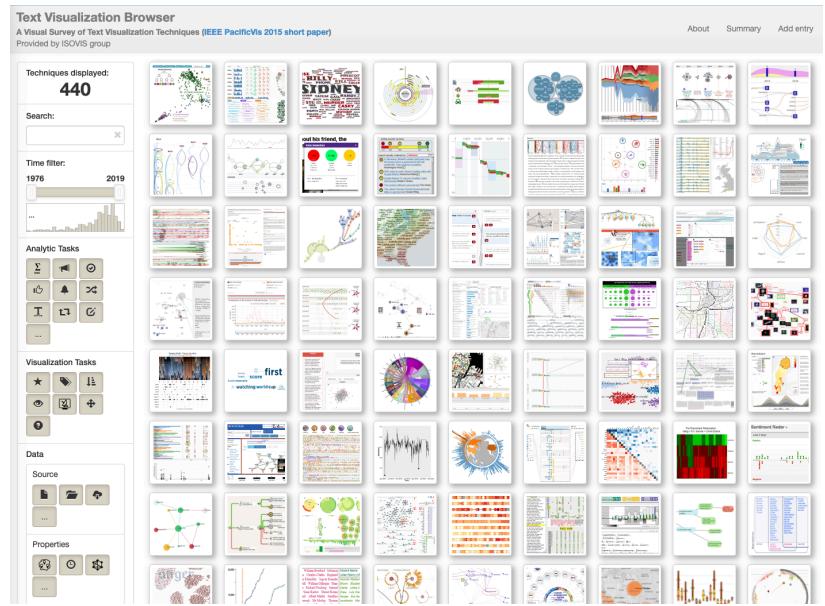
Universidad de Salamanca



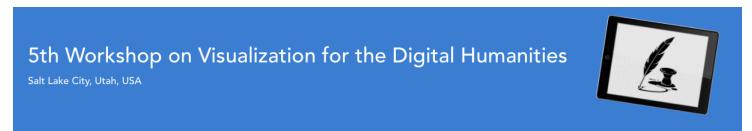
INTRODUCCIÓN

Visualización y Analítica Visual de Textos

- **Combinación de técnicas de minería de textos y visualización para aumentar las capacidades cognitivas de las usuarias.**
- Explotando al máximo el ancho de banda que proporciona el **canal visual**.
- **Aplicaciones:**
 - **Minado de opiniones/sentimientos.**
 - **Resumen automático de textos.**
 - **Lectura Distante (p. ej. en humanidades)**
 - ...



<https://textvis.lnu.se/>



Announcement: This year exceptionally, the VIS4DH workshop will be held entirely virtually, as a consequence of the current COVID-19 restrictions, and following the shift of the IEEE VIS conference to an online format.

More information about the format and access of the virtual platform will be published here soon.

Note on COVID-19: For the latest updates on the COVID-19 impact on the timeline, schedule, and format of the VIS conference, please refer to [the IEEE VIS website](#).

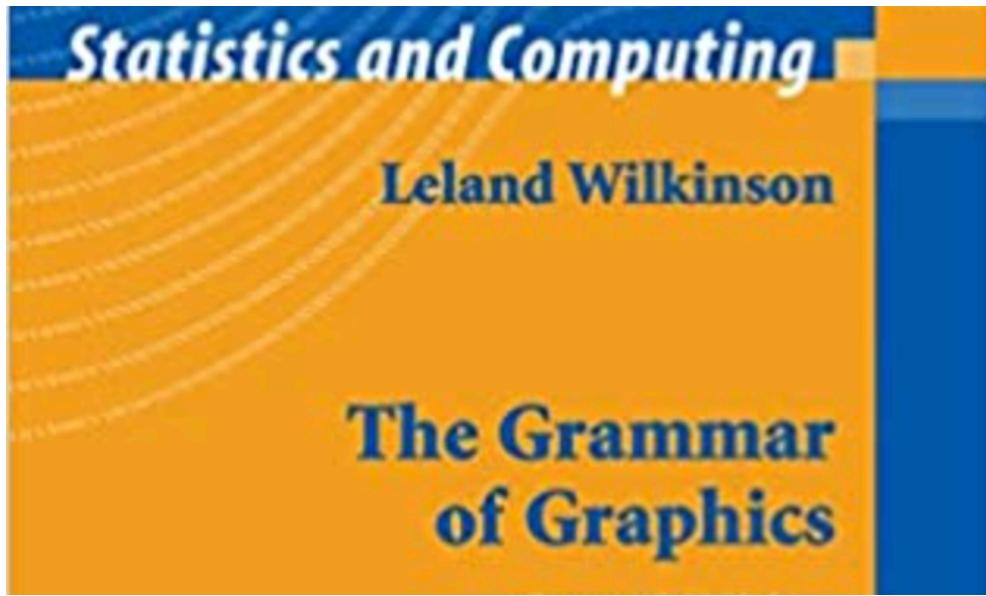
We are happy to announce that the 5th edition of VIS4DH — the Workshop on Visualization for the Digital Humanities — will take place on **Sunday, October 25, 2020** in Salt Lake City, Utah, as part of the [IEEE VIS2020 conference](#).

For 4 years now, the VIS4DH workshop has brought together researchers and practitioners from the fields of visualization and the humanities to discuss new research directions at the intersection of visualization and (digital) humanities research. The general aim is to foster productive collaborations that mutually advances all fields involved. Papers and invited talks from previous years can be found [here](#).

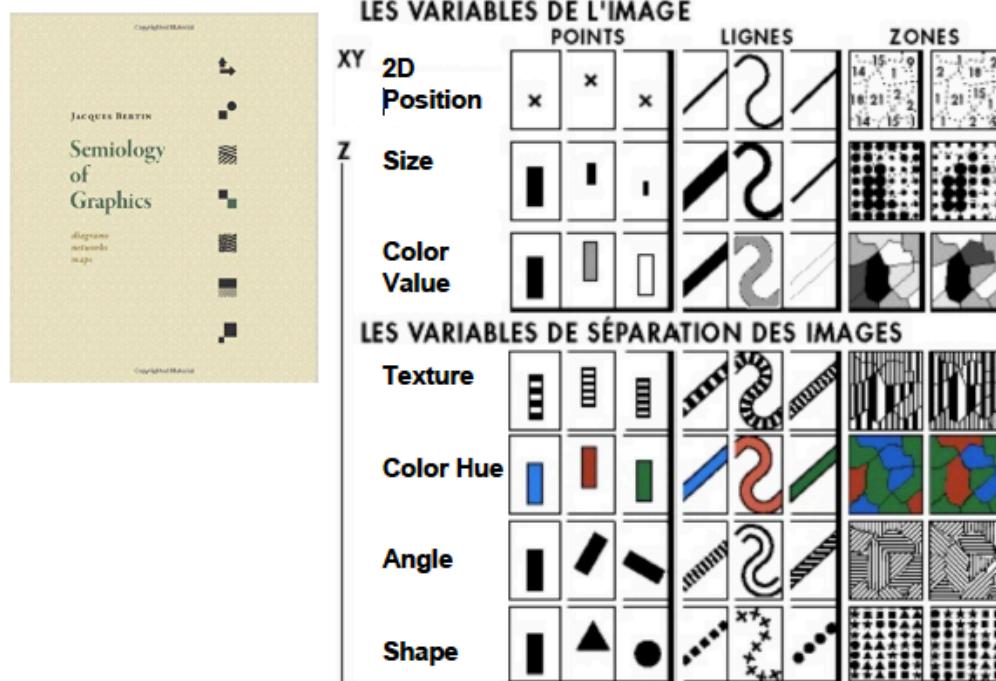
<https://vis4dh.dbvis.de/>

El lenguaje de los gráficos

1. Datos
2. Transformaciones
3. Símbolos gráficos
4. Codificación - mapear campos en los datos a propiedades de cada gráfico
5. Escalas - funciones que mapean datos a escalas visuales
6. Guías – visualización de escalas y anotación (ejes, leyendas, etc.)



Bertin's Semiology of Graphics (1967)

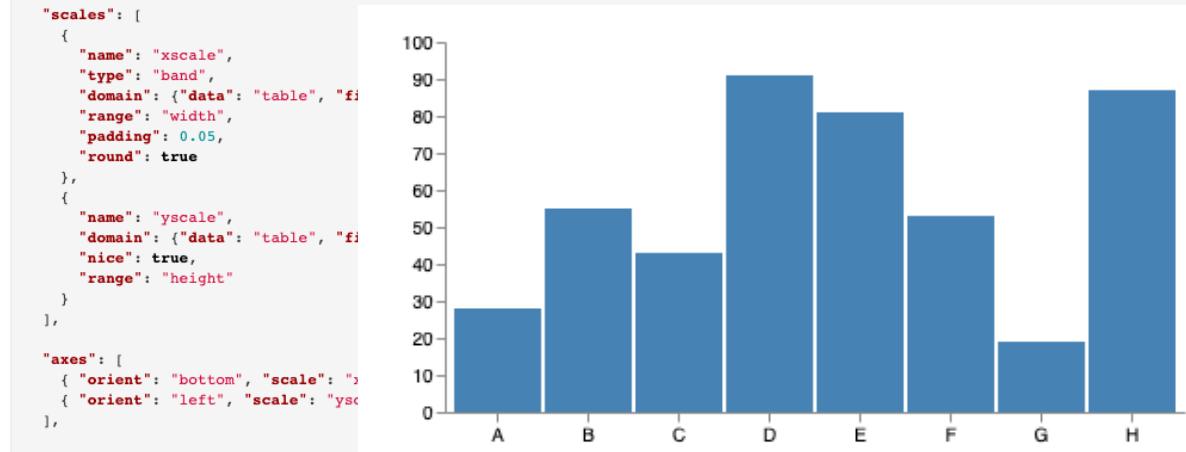


¿Qué es Vega? ¿Qué es Altair?

- Vega es una gramática para gráficos especificacada en JSON
- Creada por investigadores de la Universidad de Washington (Jeffrey Heer & Co.)
- Idónea para la web y cuadernos de programación interactivos:
 - Jupyter
 - Colab
 - Observable
- Pros: Muy expresiva y completa
- Cons: Compleja, alta curva de aprendizaje.

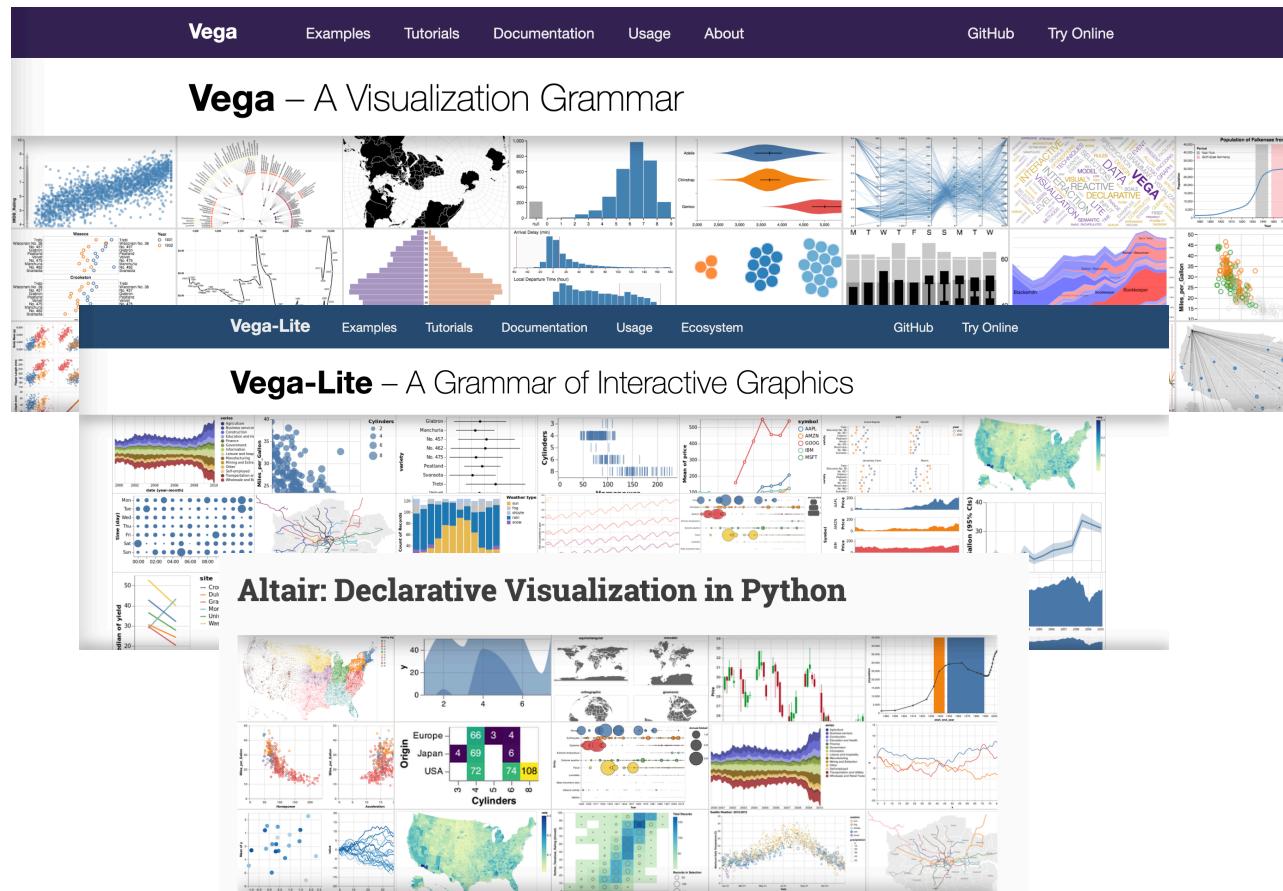
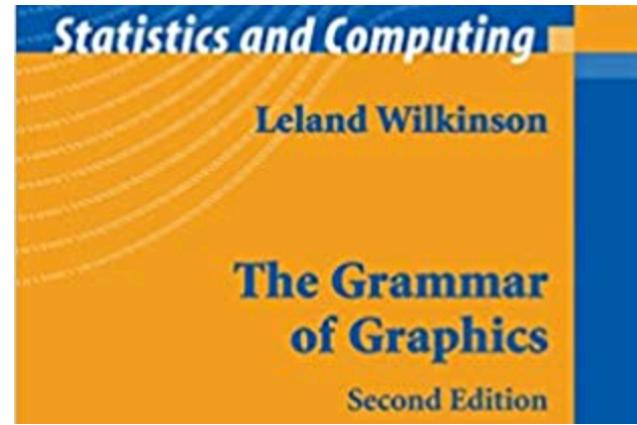
Vega JSON Specification [🔗](#)

```
{  
  "$schema": "https://vega.github.io/schema/vega/v5.json",  
  "description": "A basic bar chart example, with value labels shown upon mouse hover.",  
  "width": 400,  
  "height": 200,  
  "padding": 5,  
  
  "data": [  
    {  
      "name": "table",  
      "values": [  
        {"category": "A", "amount": 28},  
        {"category": "B", "amount": 55},  
        {"category": "C", "amount": 43},  
        {"category": "D", "amount": 91},  
        {"category": "E", "amount": 81},  
        {"category": "F", "amount": 53},  
        {"category": "G", "amount": 19},  
        {"category": "H", "amount": 87}  
      ]  
    }  
  ],  
  
  "signals": [  
    {  
      "name": "tooltip",  
      "value": {},  
      "on": [  
        {"events": "rect:mouseover", "update": "datum"},  
        {"events": "rect:mouseout", "update": "{}"}  
      ]  
    }  
  ],  
  
  "scales": [  
    {  
      "name": "xscale",  
      "type": "band",  
      "domain": {"data": "table", "field": "category"},  
      "range": "width",  
      "padding": 0.05,  
      "round": true  
    },  
    {  
      "name": "yscale",  
      "domain": {"data": "table", "field": "amount"},  
      "nice": true,  
      "range": "height"  
    }  
  ],  
  
  "axes": [  
    { "orient": "bottom", "scale": "xscale"},  
    { "orient": "left", "scale": "yscale"}  
  ],  
  "marks": [  
    {  
      "type": "rect",  
      "x": {"signal": "tooltip"},  
      "y": 0, "y2": {"signal": "tooltip"},  
      "color": "#3182bd",  
      "stroke": "#3182bd",  
      "strokeWidth": 1  
    }  
  ]  
}
```



¿Qué es Vega-Lite? ¿Qué es Altair?

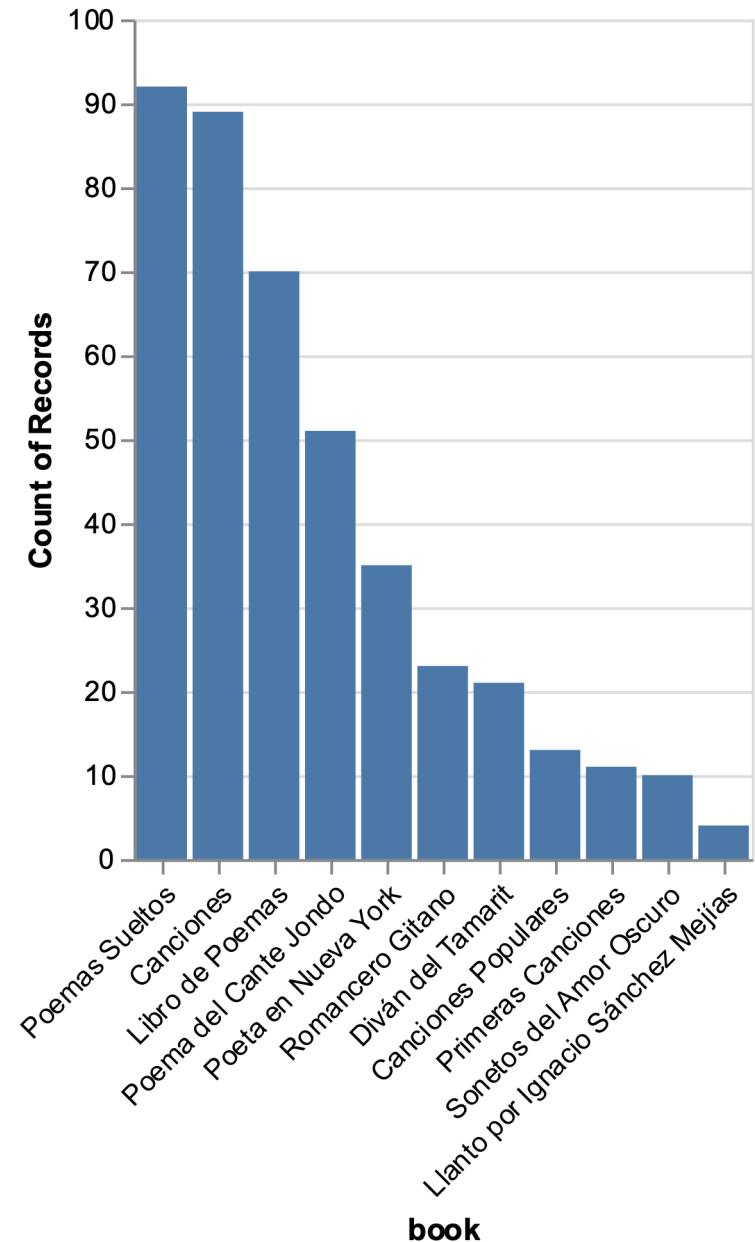
- Creado por expertos en vis
- Vega-Lite: versión reducida de Vega.
- Declarativo
- Trabaja con pd.DataFrame
- Perfecto para:
 - Análisis exploratorio
 - Prototipado
 - Etc.
- Interacción fácil
- Interfaces en:
 - Python: <https://altair-viz.github.io/>
 - R!: <https://github.com/vegawidget/altair>

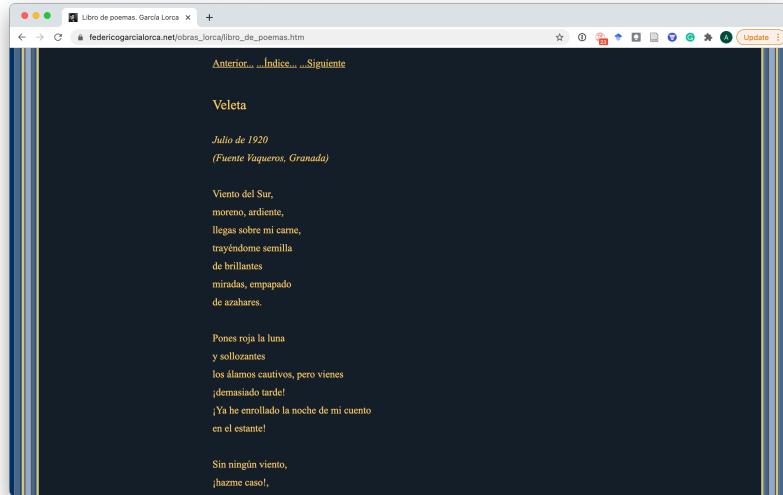


EL CORPUS

(El corpus fue vilmente escraperado desde <https://federicogarcialorca.net/>)

- **Obra poética de Federico García Lorca**
 - **1921 - 1936**
 - **419 poemas**
 - **88952 versos**
 - **11 libros**
 - Libro de poemas (1921)
 - Poema del cante jondo (1921)
 - Primeras canciones (1922)
 - Canciones (1921-1924)
 - Romancero gitano (1924-1927)
 - Poeta en Nueva York (1929-1930)
 - Llanto por Ignacio Sánchez Mejías (1935)
 - Seis poemas gallegos (1935)
 - Diván del Tamarit (1936)
 - Sonetos del amor oscuro (1936)
 - Poemas sueltos
 - Cantares Populares





```
[
  {
    "book": "Canciones",
    "body": "Cantan las siete\ndoncellas.\n\n(Sobre el cielo un arco\\nde ejemplos de ocaso.)\n\nAlma con siete voces\\las siete doncellas.\n\n(En el aire blanco \\nsiete ]",
    "subtitle": "Canción de las siete doncellas (Teoría del arco iris)",
    "title": "Teorías"
  },
  {
    "book": "Canciones",
    "body": "Hinojo, serpiente y junco.\nAroma, rastro y penumbra.\nAire, tierra y soledad.\n(La escala llega a la luna.)",
    "subtitle": "Nocturno esquemático.",
    "title": "Teorías"
  },
  {
    "book": "Canciones",
    "body": "Sábado.\nPuerta de jardín.\nDomingo.\nDía gris.\nSábado.\nArcos azules.\nBrisa.\nDomingo.\nMar con orillas,\nMetas.\nSábado.\nSemilla\\nestreme",
    "subtitle": "Canción del colegial.",
    "title": "Teorías"
  },
  {
    "book": "Canciones",
    "body": "El canto quiere ser luz.\nEn lo oscuro el canto tiene\\nhilos de fósforo y luna.\nLa luz no sabe qué quiere.\nEn sus límites de ópalo,\nse encuentra ella misma",
    "subtitle": "El canto quiere ser luz",
    "title": "Teorías"
  },
  {
    "book": "Canciones",
    "body": "Los días de fiesta\\nván sobre ruedas.\nEl tío-vivo los trae,\ny los lleva.\nCorpus azul.\nBlanca Nochebuena.\nLos días abandonan\\nsu piel, como las culetas",
    "subtitle": "Tío vivo.",
    "title": "Teorías"
  },
  {
    "book": "Canciones",
    ...
  }
]
```

	book	body	subtitle	title
0	Canciones	Cantan las siete\ndoncellas.\n\n(Sobre el cielo un arco\\nde ejemplos de ocaso.)\n\nAlma con siete voces\\las siete doncellas.\n\n(En el aire blanco \\nsiete]	Canción de las siete doncellas (Teoría del arco iris)	Teorías
1	Canciones	Hinojo, serpiente y junco.\nAroma, rastro y penumbra.	Nocturno esquemático.	Teorías
2	Canciones	Sábado.\nPuerta de jardín.\nDomingo.\nDía gris.	Canción del colegial.	Teorías
3	Canciones	El canto quiere ser luz.\nEn lo oscuro el canto tiene\\nhilos de fósforo y luna.	El canto quiere ser luz	Teorías
4	Canciones	Los días de fiesta\\nván sobre ruedas.\nEl tío-vivo los trae,\n	Tío vivo.	Teorías
...
414	Sonetos del Amor Oscuro	¡Ay voz secreta del amor oscuro!\n¡ay balido s...	NaN	Ay voz secreta del amor oscuro
415	Sonetos del Amor Oscuro	Tengo miedo a perder la maravilla\\nde tus ojos...	NaN	Soneto de la dulce queja
416	Sonetos del Amor Oscuro	Noche arriba los dos con luna llena,\nno me pu...	NaN	Noche del amor insomne
417	Sonetos del Amor Oscuro	¿Te gustó la ciudad que gota a gota\\nlabró el ...	NaN	El poeta pregunta a su amor por la ciudad enca...
418	Sonetos del Amor Oscuro	Tú nunca entenderás lo que te quiero\\nporque d...	NaN	El amor duerme en el pecho del poeta

419 rows x 4 columns

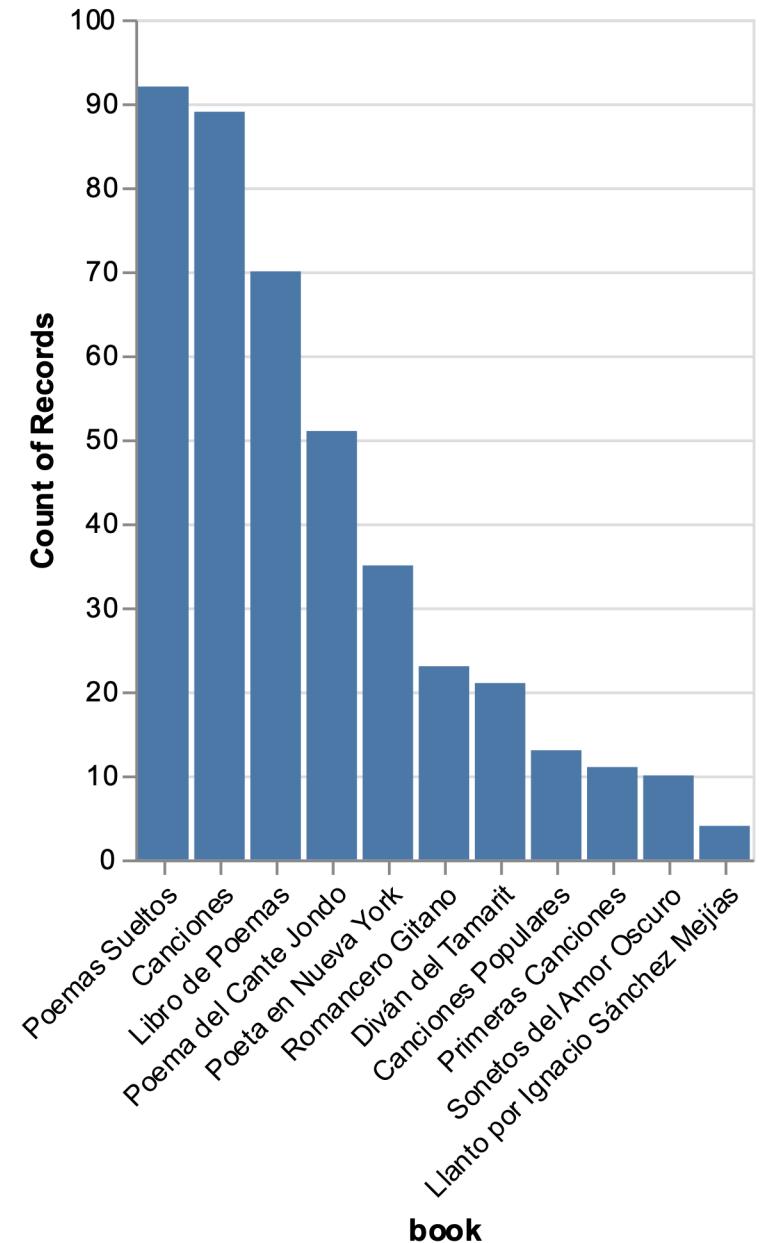


```
poems_df = pd.read_json("poems.json")
```



	book	body	subtitle	title
0	Canciones	Cantan las siete\ndoncellas.\n\n(Sobre el ciel...	Canción de las siete doncellas (Teoría del arc...	Teorías
1	Canciones	Hinojo, serpiente y junco.\nAroma, rastro y pe...	Nocturno esquemático.	Teorías
2	Canciones	Sábado.\nPuerta de jardín.\n\nDomingo.\nDía gr...	Canción del colegial.	Teorías
3	Canciones	El canto quiere ser luz.\nEn lo oscuro el cant...	El canto quiere ser luz	Teorías
4	Canciones	Los días de fiesta\nvan sobre ruedas.\nEl tío-...	Tío vivo.	Teorías
...
414	Sonetos del Amor Oscuro	¡Ay voz secreta del amor oscuro!\njay balido s...	NaN	Ay voz secreta del amor oscuro
415	Sonetos del Amor Oscuro	Tengo miedo a perder la maravilla\nde tus ojos...	NaN	Soneto de la dulce queja
416	Sonetos del Amor Oscuro	Noche arriba los dos con luna llena,\nyo me pu...	NaN	Noche del amor insomne
417	Sonetos del Amor Oscuro	¿Te gustó la ciudad que gota a gota\nlabró el ...	NaN	El poeta pregunta a su amor por la ciudad enca...
418	Sonetos del Amor Oscuro	Tú nunca entenderás lo que te quiero\nporque d...	NaN	El amor duerme en el pecho del poeta

419 rows × 4 columns



IMPERATIVO VS DECLARATIVO

D3 (o matplotlib)

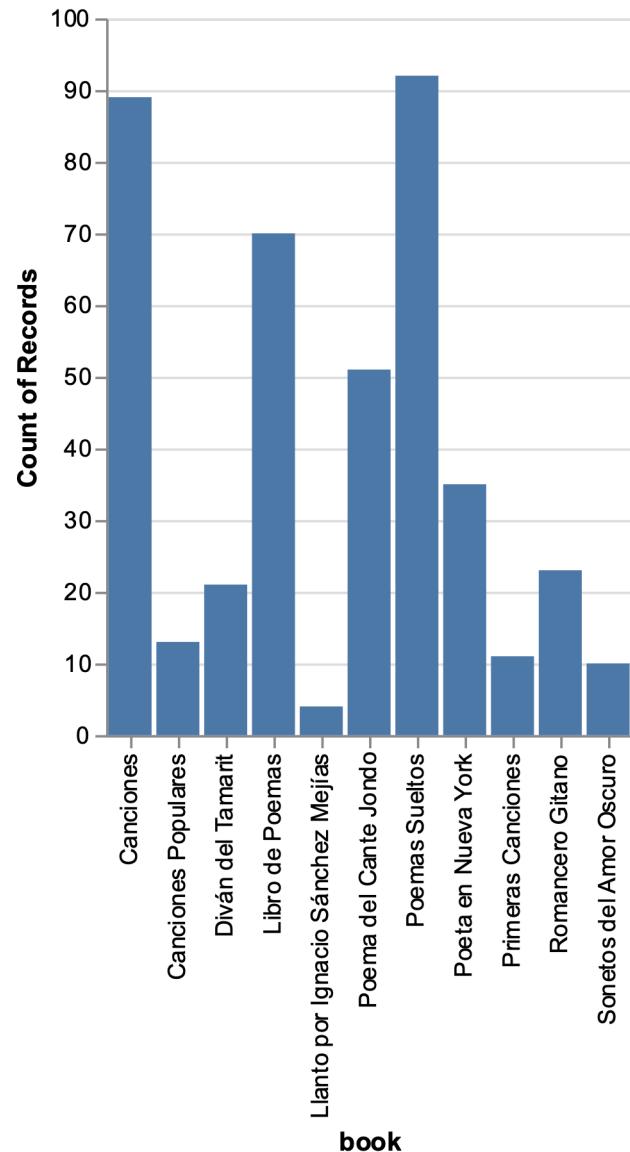
1. Añadir el SVG.
2. Agrupar los datos (d3.group/d3.nest)
3. Calcula la cuenta total de entradas por grupo (d3.rollup)
4. Con las cuentas y el número de grupos, crear las escala para los ejes X e Y., d3.scaleBand, d3.scaleLinear...
5. Crear los ejes X e Y, d3.axisBottom, d3.axisLeft.
6. Pintar un rectángulo de ancho scaleX.bandwidth, alto xScale(d.count), x = scaleX.bandwidth*group_number, y = 0.
7. @!#! No funciona!
8. Suspiro largo.

Vega-Lite (o ggplot2)

1. Quiero hacer un rectángulo por grupo.
2. El alto del rectángulo ha de ser directamente proporcional al número de elementos en el grupo.
3. Ya está! Admirar el resultado.
4. Suspiro largo de amor.

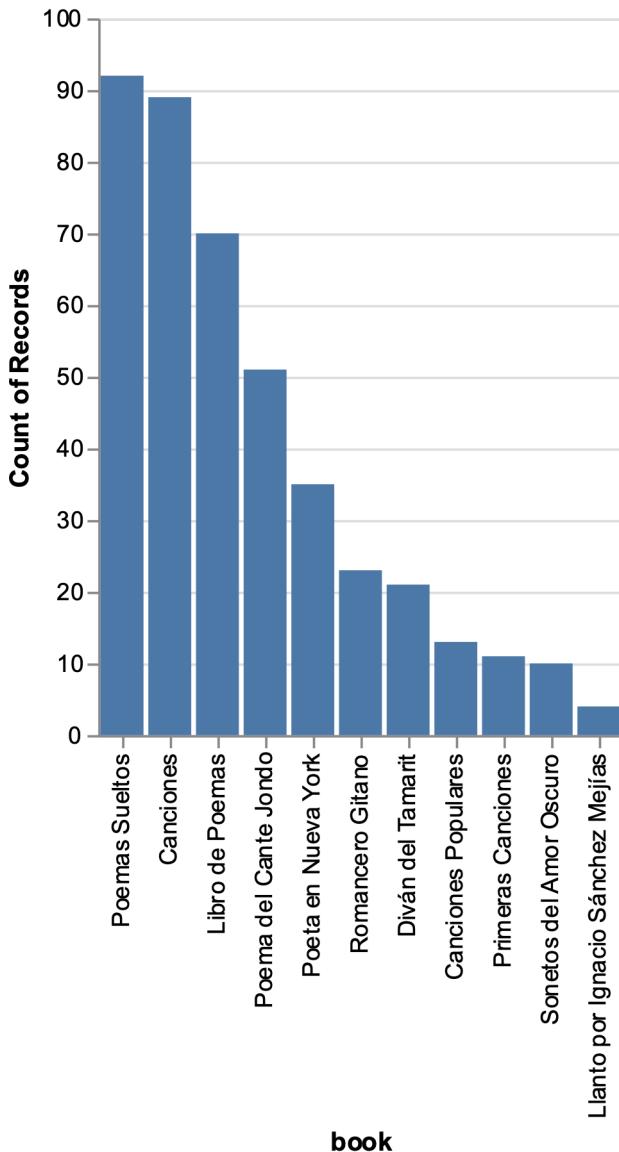
Codificando

```
alt.Chart(poems_df).mark_bar().encode(  
    x='book:N',  
    y='count()'  
)
```



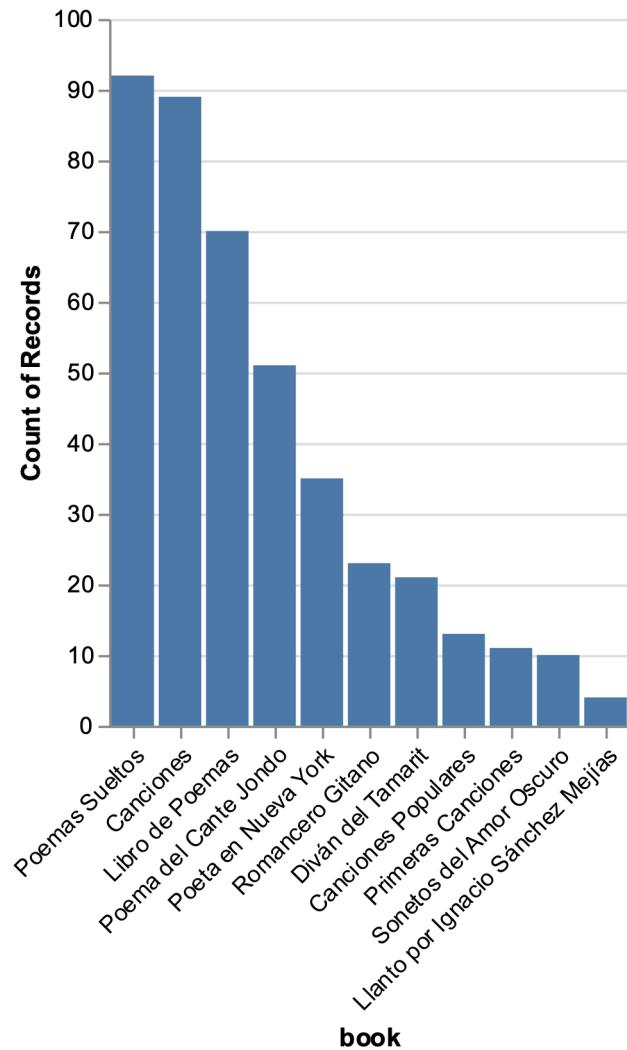
Ordenando el eje

```
alt.Chart(poems_df).mark_bar().encode(  
    x=alt.X('book:N',  
           sort='-' + y),  
    y='count()'  
)
```



Previendo la tortícolis

```
alt.Chart(poems_df).mark_bar().encode(  
    x=alt.X('book:N',  
            sort=-y',  
    axis=alt.Axis(labelAngle=-45)),  
    y='count()'  
)
```



EXPLORANDO PATRONES RÍTMICOS

```

!pip install rantanplan spacy-stanza
import rantanplan
!python -m spacy download es_core_news_md
!python -m spacy_affixes download es
from rantanplan.core import get_scansion

```

- (1) *Cuando el alba me despierta*
Cuan-doel-al-ba-me-des-pier-ta
 -- + --- +- 8
 (Miguel de Unamuno)

```

[{'phonological_groups': [{'is_stressed': True, 'syllable': 'Can'},
    {'is_stressed': False, 'is_word_end': True, 'syllable': 'tan'},
    {'is_stressed': False, 'is_word_end': True, 'syllable': 'las'},
    {'is_stressed': True, 'syllable': 'sie'},
    {'is_stressed': False, 'is_word_end': True, 'syllable': 'te'}],
 'rhythm': {'length': 5, 'stress': '+---', 'type': 'pattern'},
 'tokens': [{}{'pos': 'VERB',
    'stress_position': -2,
    'word': [{}{'is_stressed': True, 'syllable': 'Can'},
        {'is_stressed': False, 'is_word_end': True, 'syllable': 'tan'}]},
    {}{'pos': 'DET',
    'stress_position': 0,
    'word': [{}{'is_stressed': False, 'is_word_end': True, 'syllable': 'las'}]},
    {}{'pos': 'NUM',
    'stress_position': -2,
    'word': [{}{'is_stressed': True, 'syllable': 'sie'},
        {'is_stressed': False, 'is_word_end': True, 'syllable': 'te'}]}],
 'phonological_groups': [{}{'is_stressed': False, 'syllable': 'don'},
    {}{'is_stressed': True, 'syllable': 'ce'},
    {}{'is_stressed': False, 'is_word_end': True, 'syllable': 'llas'}],
 'rhythm': {'length': 3, 'stress': '-+-', 'type': 'pattern'},
 'tokens': [{}{'pos': 'NOUN'}]
}
```

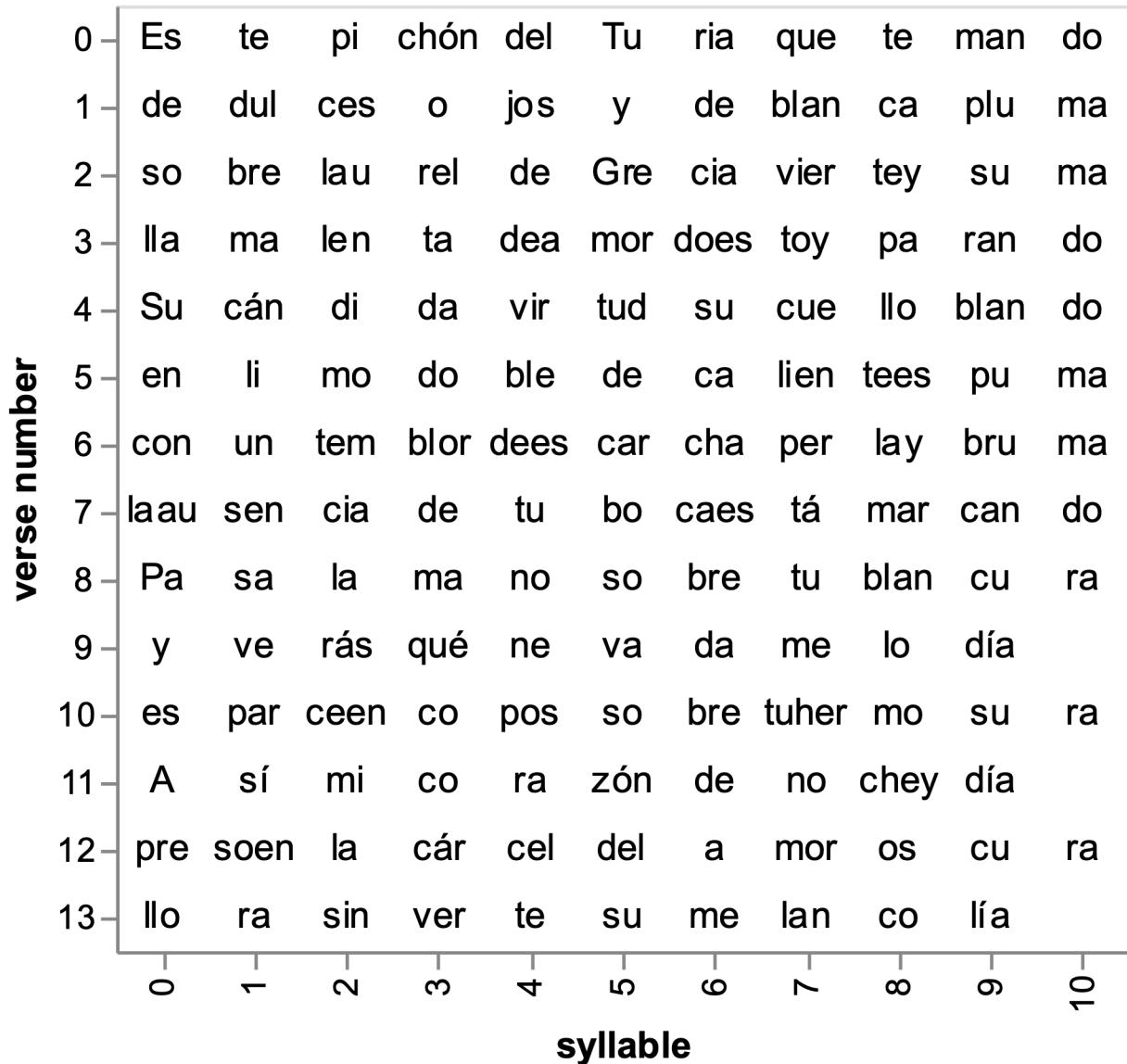
- J. de la Rosa, S. Ros, y E. González-Blanco, «Predicting metrical patterns in Spanish poetry with language models», arXiv:2011.09567 [cs], nov. 2020, Accedido: ene. 29, 2021. [En línea]. Disponible en: <http://arxiv.org/abs/2011.09567>.
- J. de la Rosa, Á. Pérez, L. Hernández, S. Ros, y E. González-Blanco, «Rantanplan, Fast and Accurate Syllabification and Scansion of Spanish Poetry», Procesamiento del Lenguaje Natural, vol. 65, n.º 0, Art. n.º 0, sep. 2020.
- <https://github.com/linhd-postdata/rantanplan>

poem	x	y	syllable	is_stressed		book	n_verses	verse_length
0	0	0	Can	1	Canciones	14	5	
1	0	1	tan	0	Canciones	14	5	
2	0	2	las	0	Canciones	14	5	
3	0	3	sie	1	Canciones	14	5	
4	0	4	te	0	Canciones	14	5	
...
88948	418	5	ce	1	Sonetos del Amor Oscuro	14	11	
88949	418	6	chan	0	Sonetos del Amor Oscuro	14	11	
88950	418	7	to	0	Sonetos del Amor Oscuro	14	11	
88951	418	8	da	0	Sonetos del Amor Oscuro	14	11	
88952	418	9	vía	1	Sonetos del Amor Oscuro	14	11	

88953 rows × 8 columns

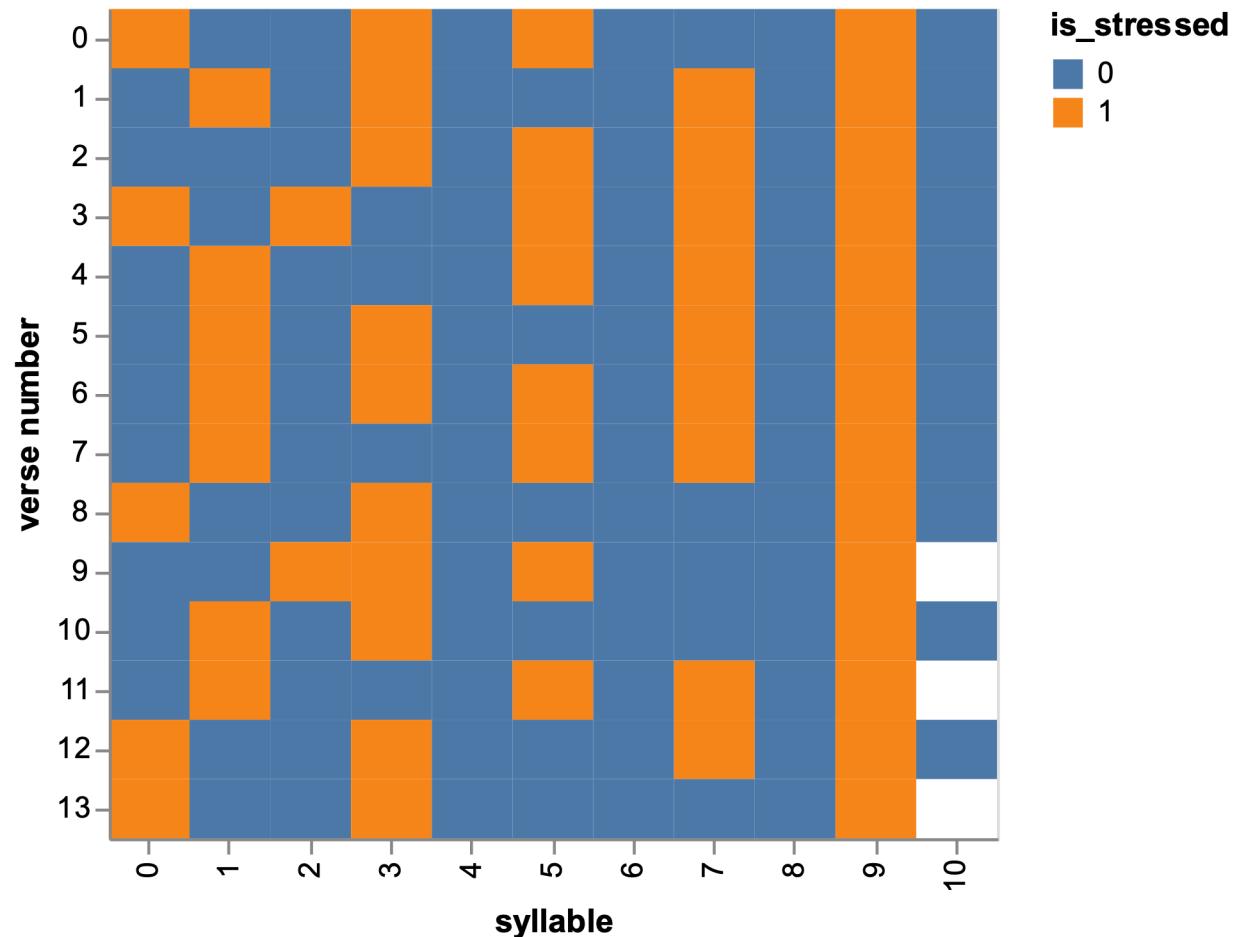
Solo Texto

```
alt.Chart(  
    poem  
) .encode(  
    x=alt.X('x:O',  
            title='syllable') ,  
    y=alt.Y('y:O',  
            title='verse number')  
) .mark_text() .encode(  
    text='syllable:N',  
    color=alt.value('black')  
)
```

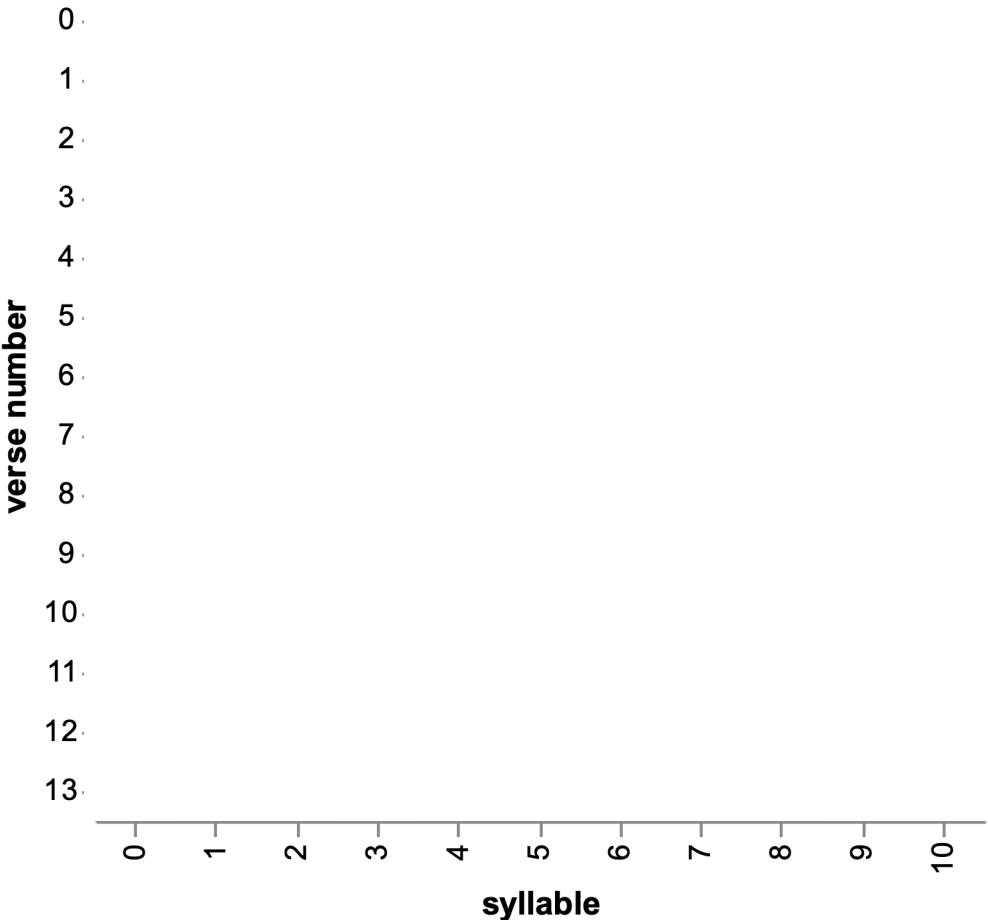


Solo un Heatmap

```
alt.Chart(  
    poem  
) .encode(  
    x=alt.X('x:O' ,  
            title='syllable') ,  
    y=alt.Y('y:O' ,  
            title='verse number')  
) .mark_rect() .encode(  
    color='is_stressed:N'  
)
```



```
base = alt.Chart(  
    verses_df  
) .encode(  
    x=alt.X('x:O',  
            title='syllable'),  
    y=alt.Y('y:O',  
            title='verse number')  
)
```



```

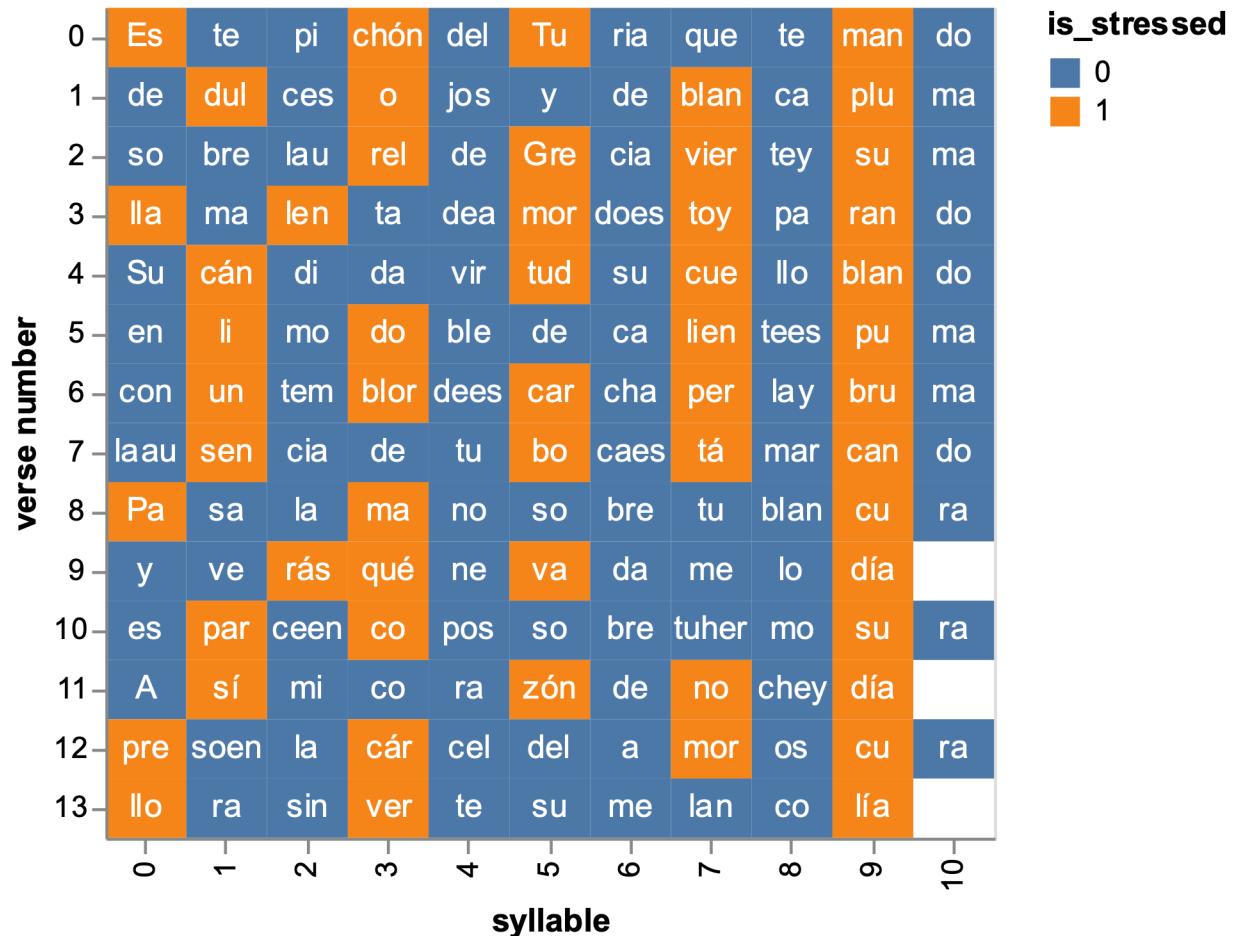
base = alt.Chart(
    verses_df
) .encode(
    x=alt.X('x:O',
            title='syllable'),
    y=alt.Y('y:O',
            title='verse number')
)

heatmap = base.mark_rect().encode(
    color='is_stressed:N'
)

text = base.mark_text().encode(
    text='syllable:N',
    color=alt.value('white')
)

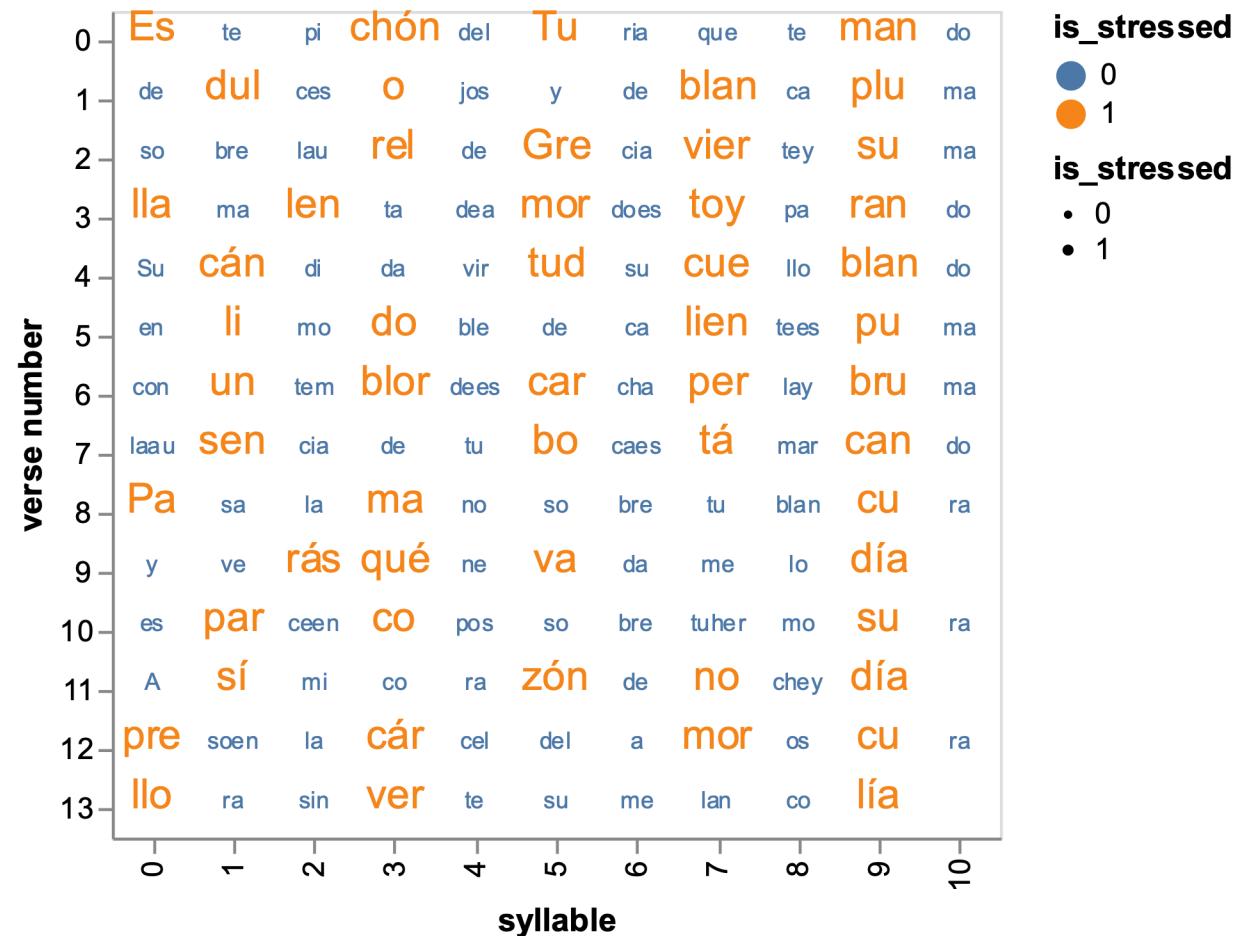
(heatmap + text)

```



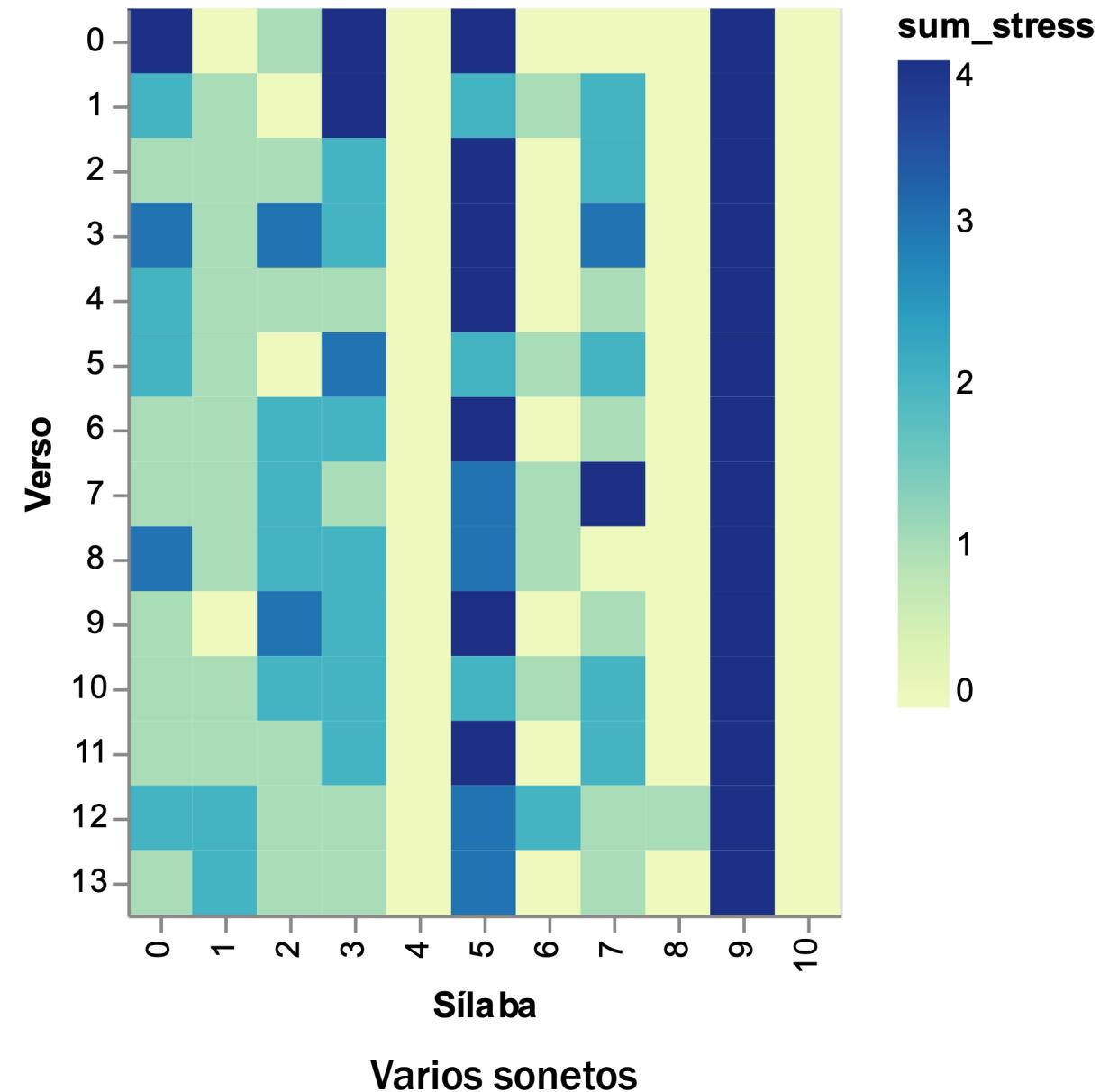
Visualizando (con) Texto

```
alt.Chart(  
    poem  
) .encode(  
    x=alt.X('x:O',  
            title='syllable'),  
    y=alt.Y('y:O',  
            title='verse number')  
) .mark_text(baseline='alphabetic') .encode(  
    text='syllable:N',  
    size=alt.Size('is_stressed:N',  
    scale=alt.Scale(  
        domain=[0, 1],  
        range=[8,14])),  
    color='is_stressed:N'  
)
```



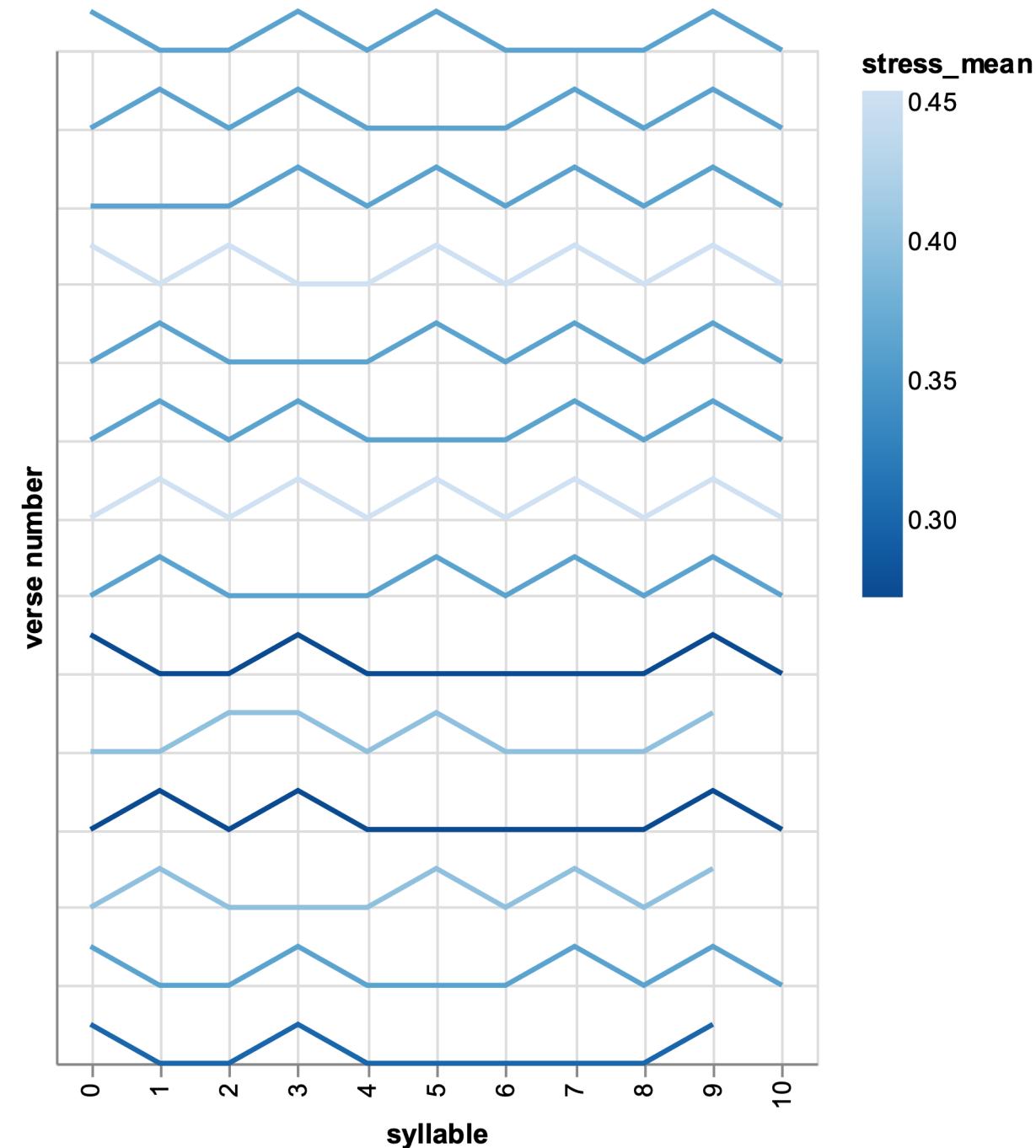
Agregando: Heatmap

```
alt.Chart(poems_verses_df).mark_rect()  
).encode(  
    x=xAxis,  
    y=yAxis,  
    color='sum_stress:Q',  
    tooltip=['sum_stress:Q']  
) .transform_aggregate(  
    sum_stress='sum(is_stressed)',  
    groupby=[ 'x' , 'y' ]  
)
```



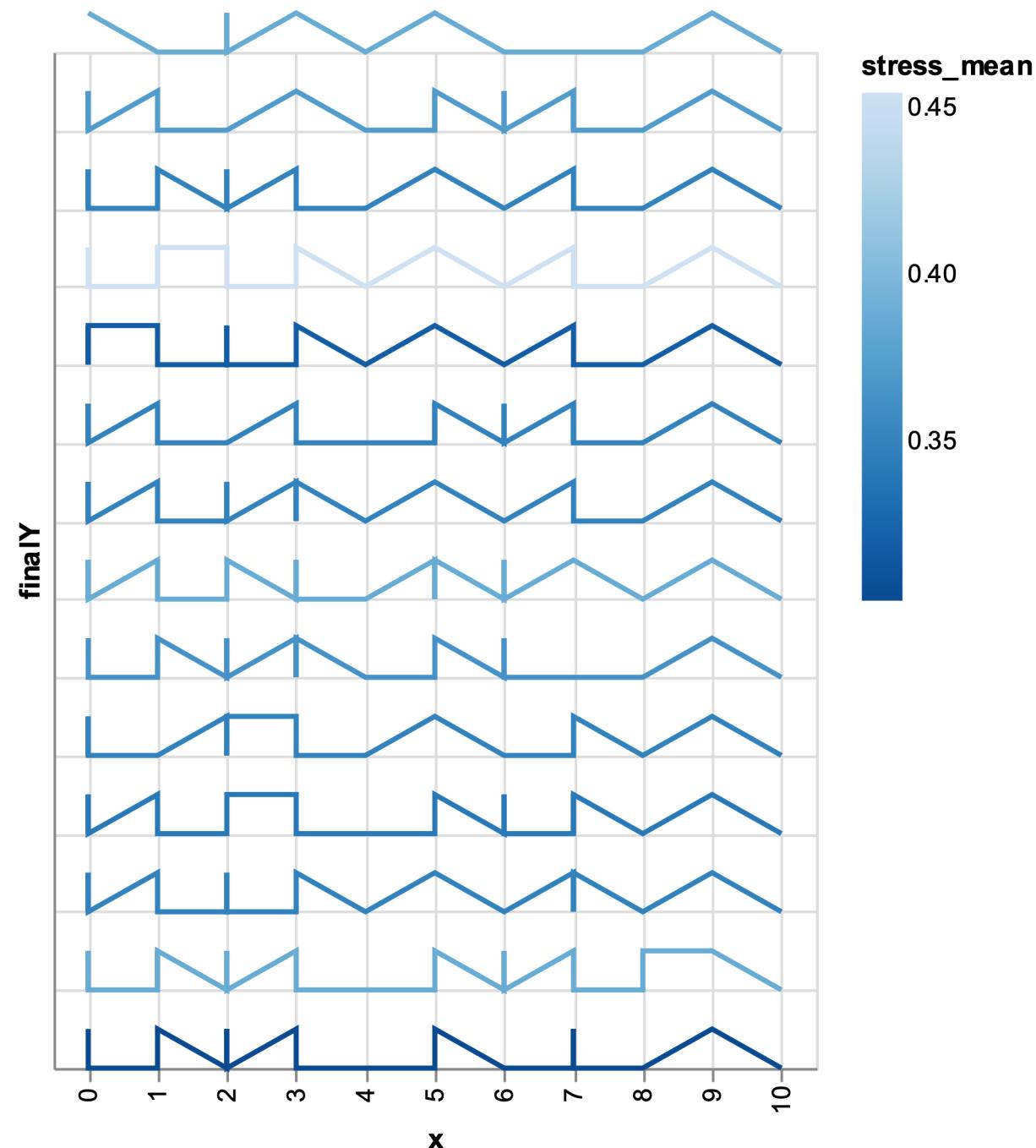
Sparklines: un poema

```
chart.transform_joinaggregate(
    stress_mean='mean(is_stressed)' ,
    groupby=['y']
).transform_calculate(
    finalY = 'datum.y - datum.is_stressed*0.5'
).encode(
    x=alt.X('x:O', axis=alt.Axis(grid=True) ,
            title='syllable') ,
    y=alt.Y('finalY:Q' ,
            ...
    color=alt.Color('stress_mean:Q' ,
                    sort='descending') ,
    ...
)
```



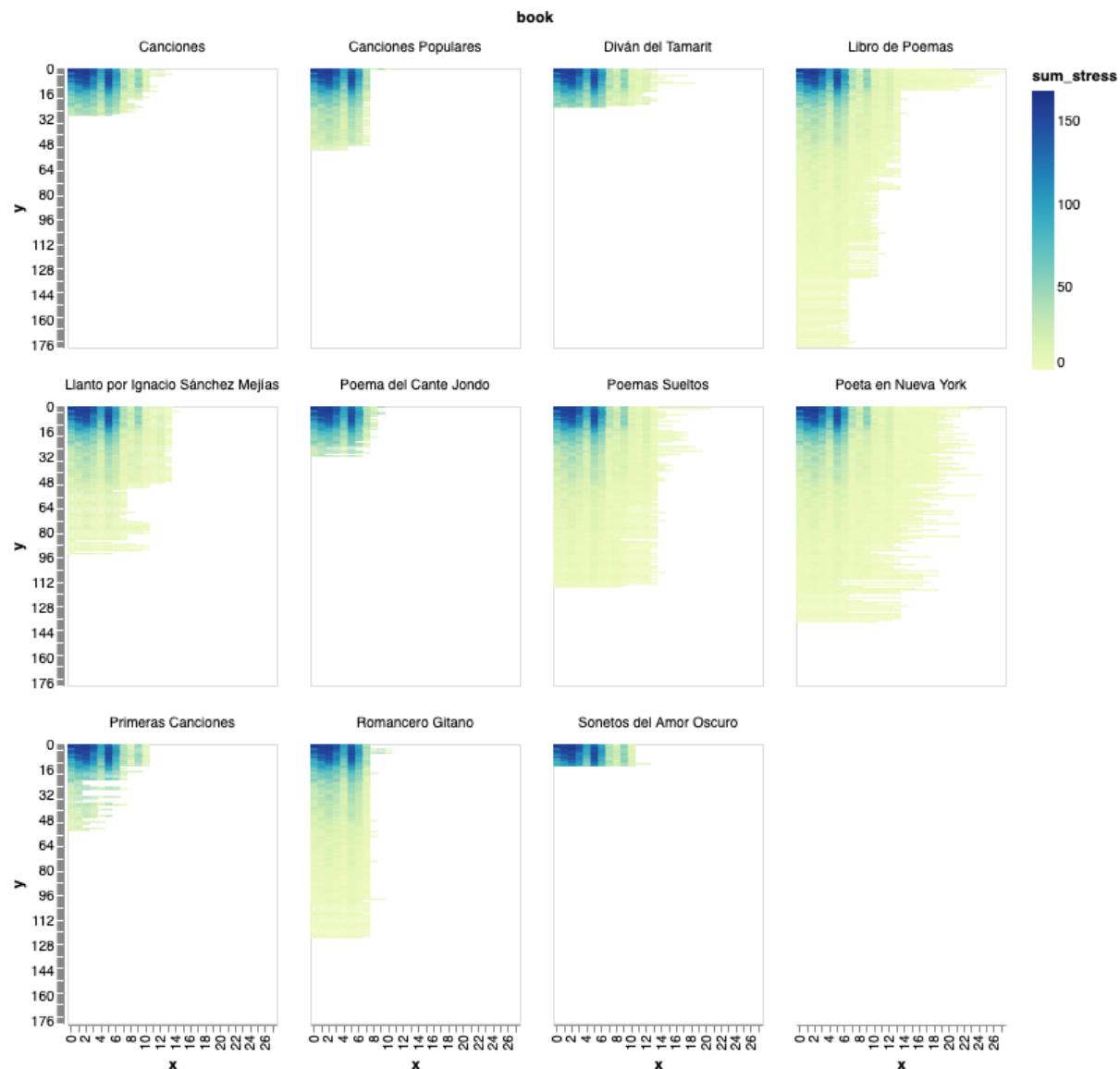
Sparklines: varios poemas

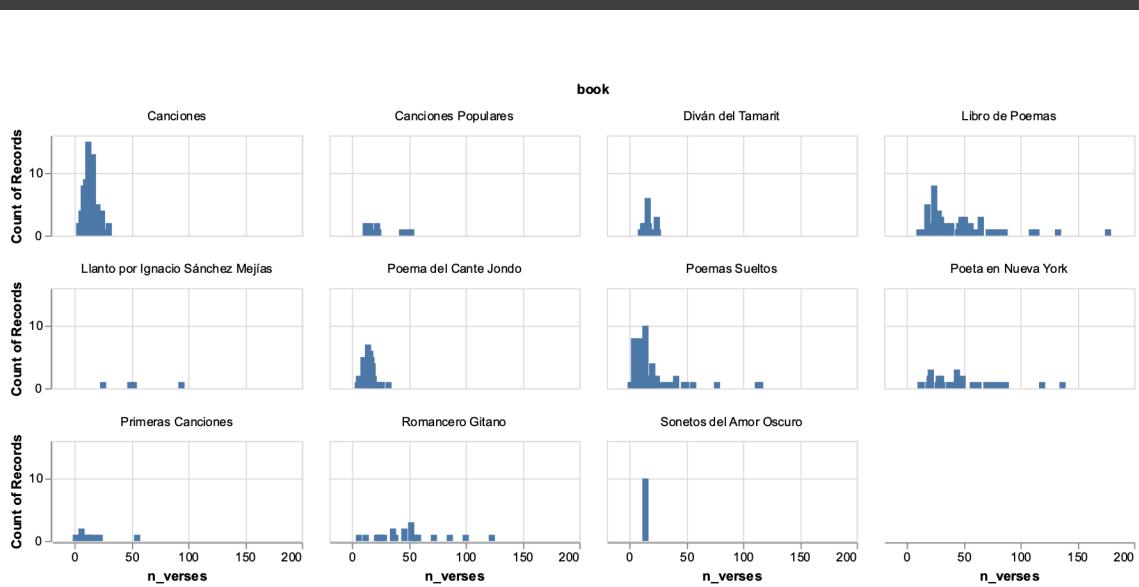
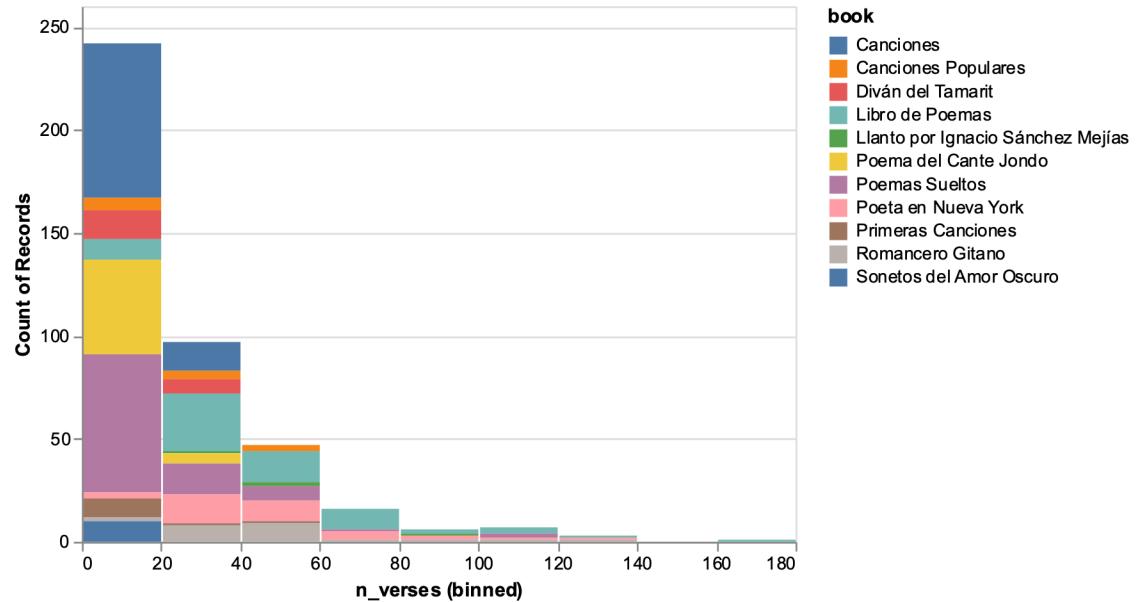
```
chart.transform_joinaggregate(  
    stress_mean='mean(is_stressed)' ,  
    groupby=['y'])  
.transform_calculate(  
    finalY = 'datum.y - datum.is_stressed*0.5'  
).encode(  
    x=alt.X('x:O', axis=alt.Axis(grid=True) ,  
           title='syllable') ,  
    y=alt.Y('finalY:Q' ,  
            ...  
    color=alt.Color('stress_mean:Q' ,  
                  sort='descending') ,  
    ...
```



Desagregando

```
alt.Chart(verses_df).mark_rect().encode(  
    x='x:O',  
    y='y:O',  
    color='sum_stress:Q',  
    facet=alt.Facet('book:N', columns=4)  
)  
.transform_joinaggregate(  
    sum_stress='sum(is_stressed)',  
    groupby=['x', 'y'])  
)
```

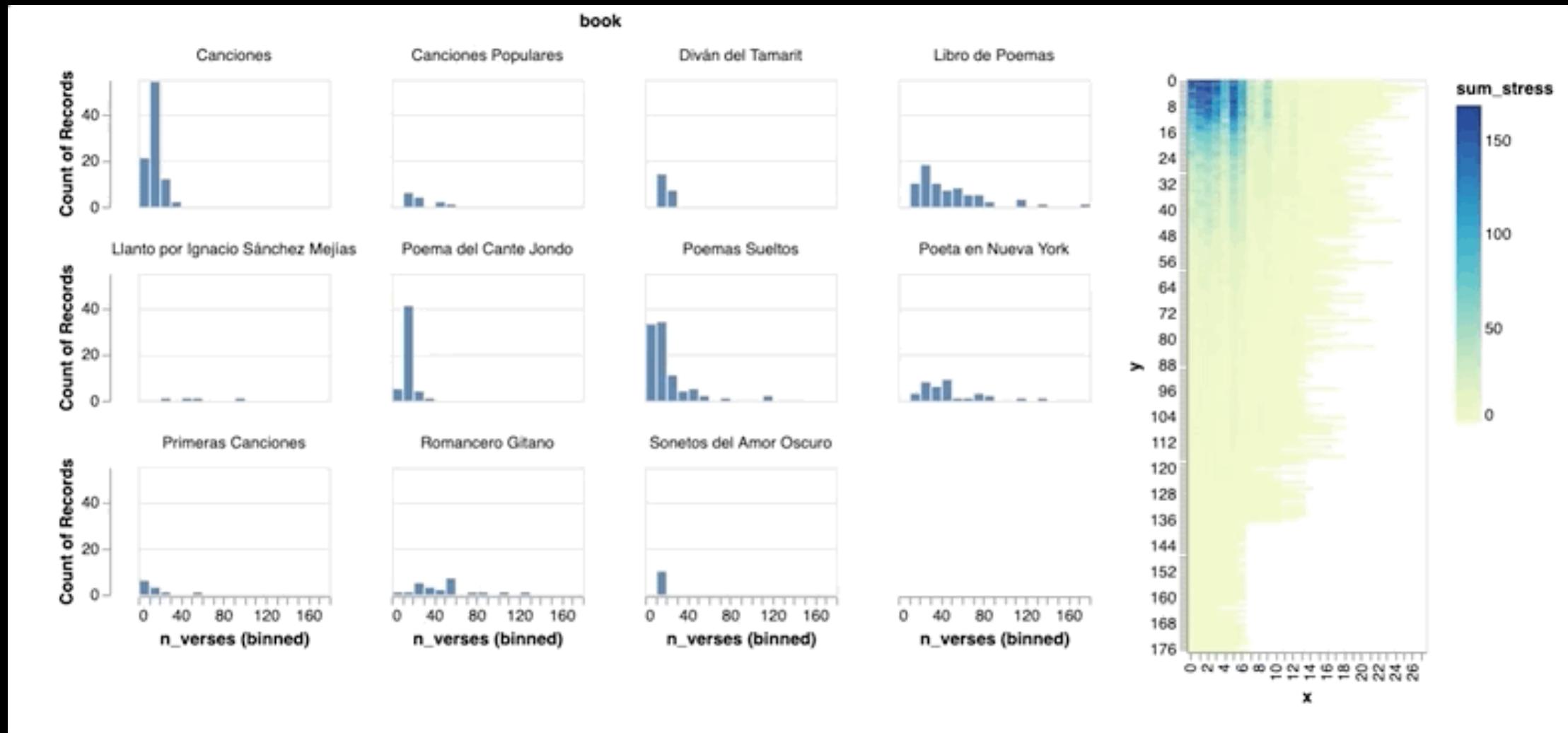




Interacción: Brushing+Linking

```
brush = alt.selection_interval(encodings=[ 'x' ])  
...  
background = base.encode(  
    color=alt.value('#ddd')  
) .add_selection(brush)  
  
highlight = base.transform_filter(brush)  
  
bars_multi = alt.layer(  
    background,  
    highlight,  
    data=poems_df,  
) .facet(  
    'book:N' ,  
    columns=4  
)  
...  
(bars_multi | heatmap)
```

Interacción: Brushing+Linking



VISUALIZANDO TEMÁTICAS

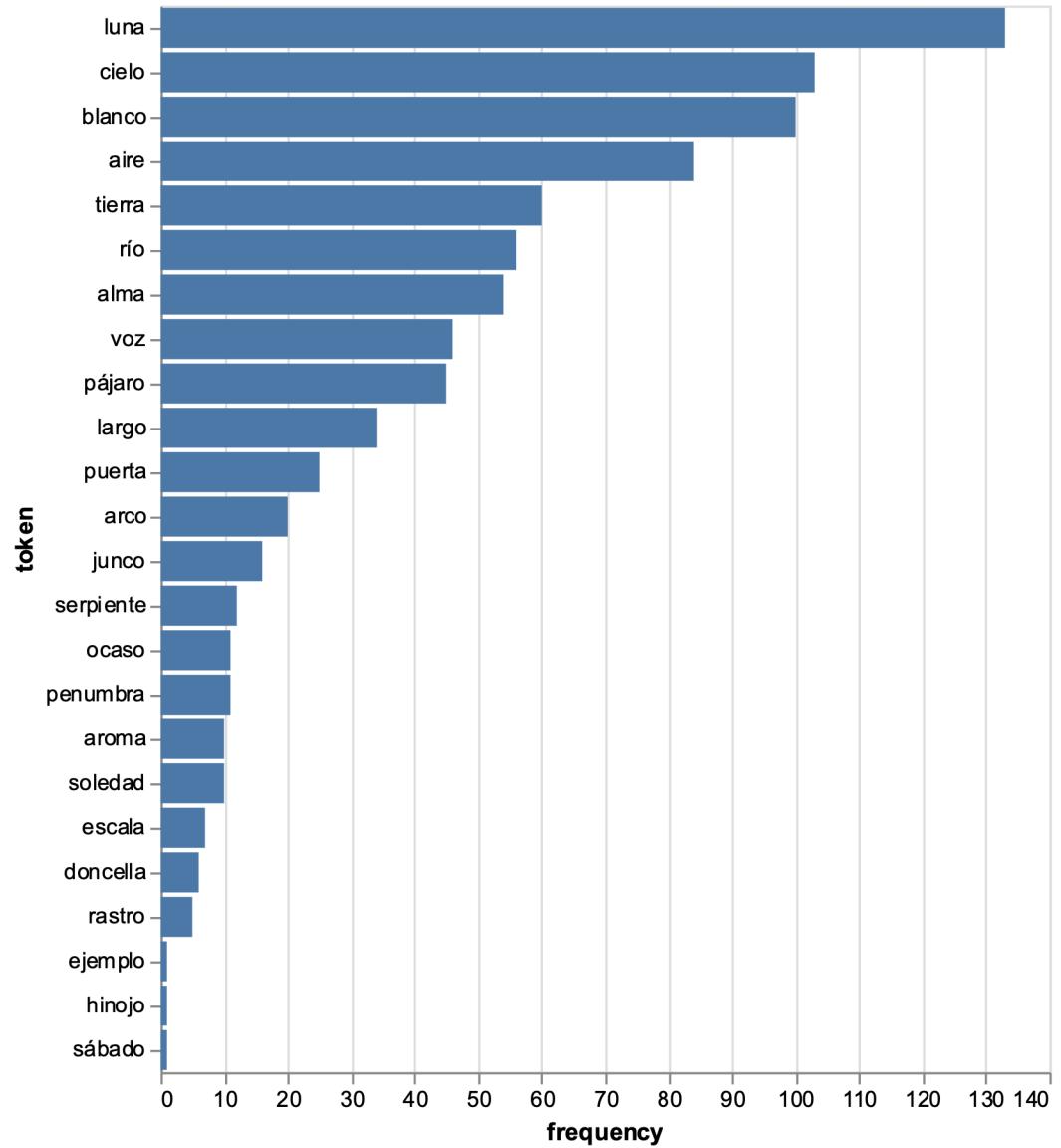
Lematizado y selección (nombres y adjetivos)

```
lemmatized_poems = []
for i, chunked_poem in enumerate(chunked_poems_full):
    if i % 10 == 0:
        print('Processed (%d/%d)' % (i, len(chunked_poems_full)))
    tokens = []
    for token in nlp(" ".join(list(flatten(chunked_poem)))):
        if token.pos_ in ['NOUN', 'ADJ']:
            tokens.append(token.lemma_)
    lemmatized_poems.append(tokens)
print('Done')

dictionary = corpora.Dictionary(lemmatized_poems)
doc_term_matrix = [dictionary.doc2bow(poem) for poem in
                   lemmatized_poems]
```

Tokens populares

```
alt.Chart(freqs_df).transform_window  
(  
    rank='rank(frequency)',  
).transform_filter(  
    alt.datum.rank < 25  
) .mark_bar().encode(  
    y=alt.Y('token:N', sort=-x'),  
    x='frequency:Q'  
)
```

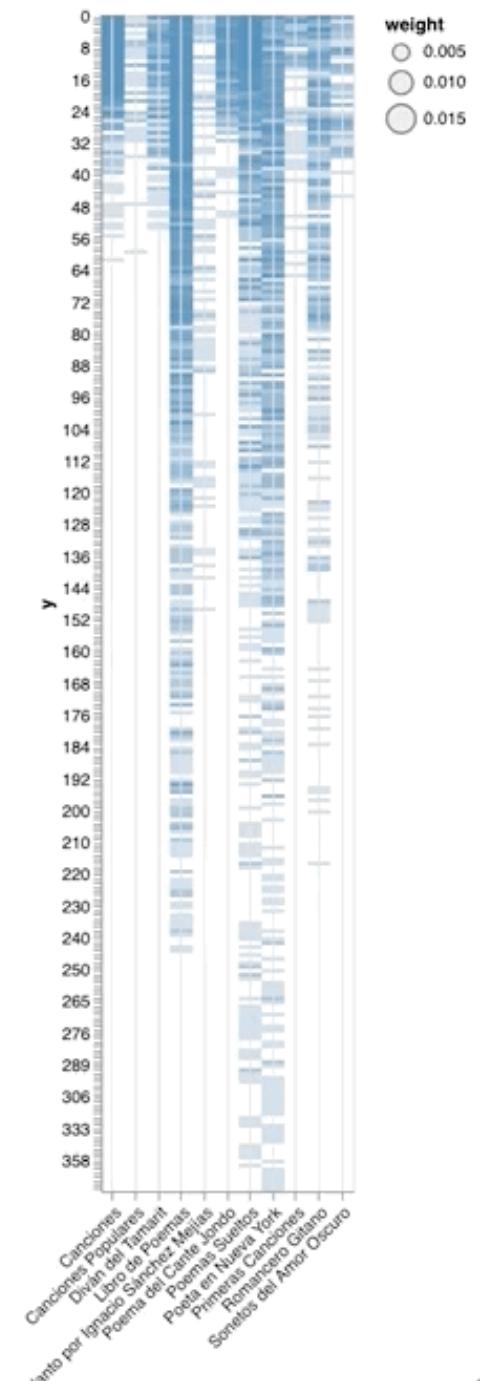
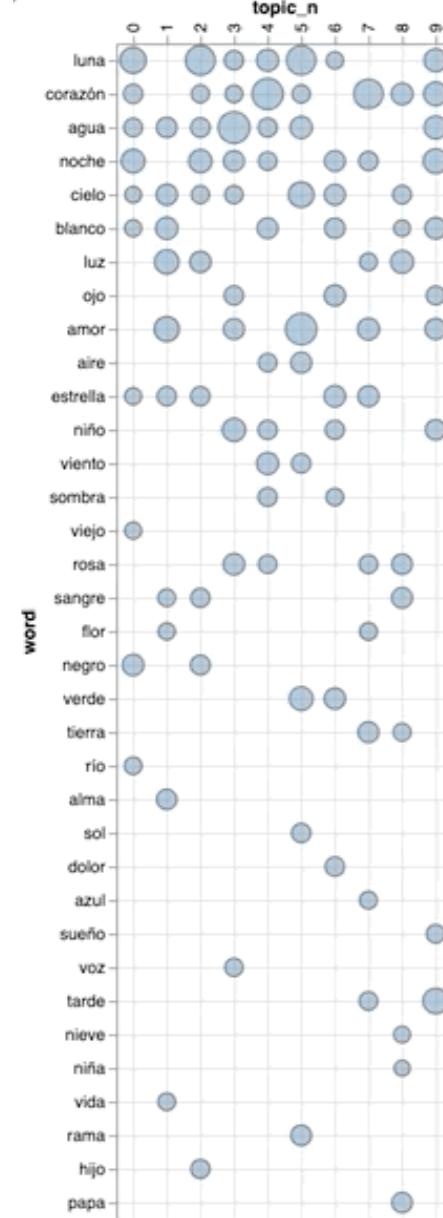
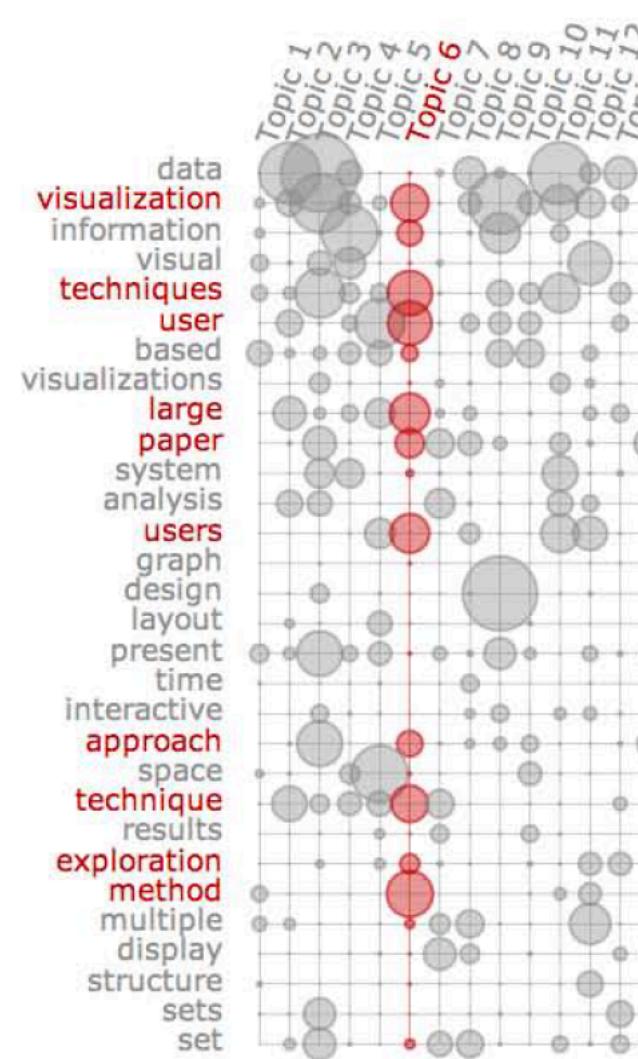


Topic Modeling (LDA)

```
ldamodel = LdaModel(doc_term_matrix,
                     num_topics=number_of_topics,
                     alpha='auto',
                     eval_every=5,
                     id2word = dictionary)
```

```
ldamodel.print_topics(num_topics=number_of_topics, num_words=number_of_words)
```

```
[0,
 '0.012*luna' + 0.010*noche' + 0.009*negro' + 0.007*corazón' + 0.006*agua' + 0.006*rio' + 0.005*blanco' + 0.005*viejo' + 0.005*estrella' + 0.005*cielo''),
(1,
 '0.011*amor' + 0.011*luz' + 0.009*blanco' + 0.008*cielo' + 0.008*agua' + 0.008*alma' + 0.007*estrella' + 0.006*sangre' + 0.006*vida' + 0.005*flor''),
(2,
 '0.016*luna' + 0.010*noche' + 0.008*luz' + 0.007*negro' + 0.007*agua' + 0.007*estrella' + 0.007*sangre' + 0.007*hijo' + 0.006*corazón' + 0.006*cielo''),
(3,
 '0.018*agua' + 0.010*niño' + 0.009*rosa' + 0.008*amor' + 0.008*noche' + 0.007*ojos' + 0.007*luna' + 0.006*voz' + 0.006*cielo' + 0.006*corazón''),
(4,
 '0.018*corazón' + 0.009*viento' + 0.009*luna' + 0.008*blanco' + 0.007*niño' + 0.007*agua' + 0.007*sombra' + 0.006*aire' + 0.006*noche' + 0.006*rosa''),
(5,
 '0.018*amor' + 0.016*luna' + 0.012*cielo' + 0.010*verde' + 0.009*agua' + 0.008*aire' + 0.008*rama' + 0.007*viento' + 0.007*sol' + 0.006*corazón''),
(6,
 '0.009*estrella' + 0.008*ojos' + 0.008*verde' + 0.008*cielo' + 0.008*noche' + 0.008*blanco' + 0.007*dolor' + 0.007*niño' + 0.006*sombra' + 0.006*luna''),
(7,
 '0.016*corazón' + 0.009*amor' + 0.008*estrella' + 0.008*tierra' + 0.007*noche' + 0.006*rosa' + 0.006*tarde' + 0.006*azul' + 0.006*luz' + 0.006*flor''),
(8,
 '0.010*luz' + 0.009*corazón' + 0.008*sangre' + 0.008*rosa' + 0.008*papa' + 0.007*cielo' + 0.006*tierra' + 0.005*nieve' + 0.005*blanco' + 0.005*niña''),
(9,
 '0.012*tarde' + 0.011*noche' + 0.010*corazón' + 0.010*luna' + 0.010*agua' + 0.009*blanco' + 0.008*amor' + 0.008*niño' + 0.007*ojos' + 0.006*sueño')]
```



[J. Chuang, C. D. Manning, y J. Heer, «Termite: Visualization Techniques for Assessing Textual Topic Models», en *Proceedings of the International Working Conference on Advanced Visual Interfaces*, New York, NY, USA, 2012, pp. 74–77, doi: [10.1145/2254556.2254572](https://doi.org/10.1145/2254556.2254572).

Resumen

- Vega-Lite tiene una curva de aprendizaje no muy empinada y permite crear gráficos correctos de manera sencilla y declarativa.
 - Muchas decisiones de diseño han sido ya tomadas y son perceptualmente óptimas.
- Algunas cosas aún no implementadas
 - Animaciones
 - Layouts complejos: Treemaps, grafos, Sankeys, etc.
 - ...

GRACIAS

Notebook y slides: <https://github.com/ale0xb/visualizing-garcialorca>

¿Preguntas?

RECURSOS

- A. Satyanarayan, D. Moritz, K. Wongsuphasawat, y J. Heer, «Vega-Lite: A Grammar of Interactive Graphics», *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, n.º 1, pp. 341-350, ene. 2017, doi: [10.1109/TVCG.2016.2599030](https://doi.org/10.1109/TVCG.2016.2599030).
- J. VanderPlas et al., «Altair: Interactive Statistical Visualizations for Python», *The Journal of Open Source Software*, vol. 3, p. 1057, dic. 2018, doi: [10.21105/joss.01057](https://doi.org/10.21105/joss.01057).
- J. Chuang, C. D. Manning, y J. Heer, «Termite: Visualization Techniques for Assessing Textual Topic Models», en *Proceedings of the International Working Conference on Advanced Visual Interfaces*, New York, NY, USA, 2012, pp. 74–77, doi: [10.1145/2254556.2254572](https://doi.org/10.1145/2254556.2254572).
- J. de la Rosa, S. Ros, y E. González-Blanco, «Predicting metrical patterns in Spanish poetry with language models», *arXiv:2011.09567 [cs]*, nov. 2020, Accedido: ene. 29, 2021. [En línea]. Disponible en: [http://arxiv.org/abs/2011.09567](https://arxiv.org/abs/2011.09567).
- J. de la Rosa, Á. Pérez, L. Hernández, S. Ros, y E. González-Blanco, «Rantanplan, Fast and Accurate Syllabification and Scansion of Spanish Poetry», *Procesamiento del Lenguaje Natural*, vol. 65, n.º 0, Art. n.º 0, sep. 2020.
- Jake Vanderplas, *Exploratory Data Visualization with Vega, Vega-Lite, and Altair - PyCon 2018*. 2018.
<https://youtu.be/ms29ZPUKxbU>
- Wongsuphasawat, Moritz, and Satyanarayan, *Vega Lite: A Grammar of Interactive Graphics - OpenVisConf 2017*.
<https://youtu.be/9uaHRWj04D4>
- <https://altair-viz.github.io/>
- https://vega.github.io/vega-lite/tutorials/getting_started.html
- <https://github.com/uwdata/visualization-curriculum>
- <https://courses.cs.washington.edu/courses/cse512/19sp/>
- <https://vis4dh.dbvis.de/>