Visualization for Data Science

From Theory to Tooling

Points: occupy a single position in the plane

Square
Point
Circle

Lines: extend in one dimension (length)

Bar
Line
Tick

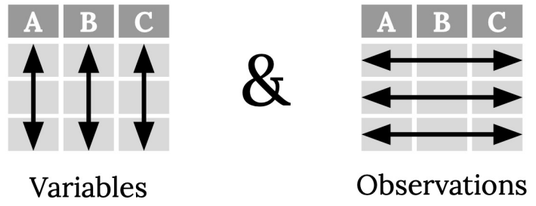
Areas: extend in two dimensions (enclosed region)

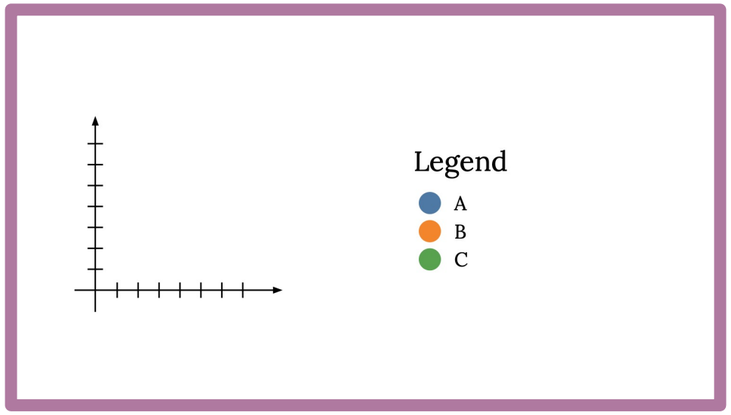
Arc
Rect
Area

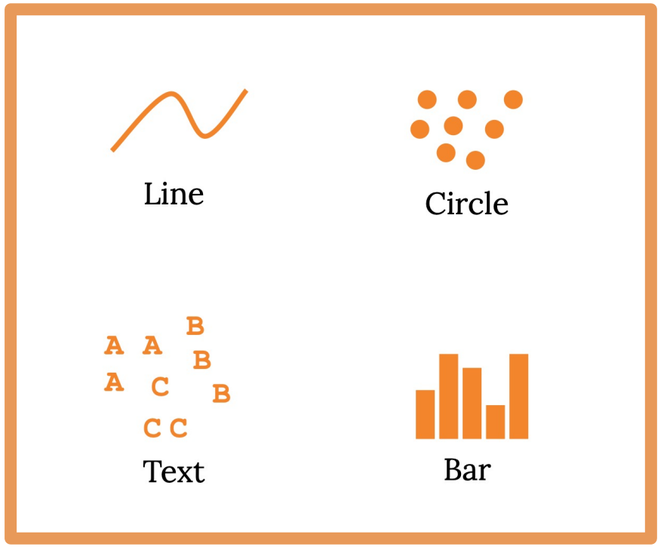
Altair Mark	Munzner Category	Why?
`point`	Point (0D)	Represents a single position; shape/color/size can encode attributes.
`circle`	Point (0D)	Same as `point`, but always circular; still a single position in space.
`square`	Point (0D)	Same as `circle`, but square shape; still marks a single position.
`line`	Line (1D)	Extends in one dimension to connect positions (e.g., trends over time).
`tick`	Line (1D)	A short line segment marking a position; encodes value via placement/length.
`bar`	Line (1D)	Perceived by length, not area; extends in one dimension from a baseline.
`area`	Area (2D)	Encloses a filled region under a line; encodes value by both length and fill.
`rect`	Area (2D)	Encodes two dimensions of position (e.g., heatmaps); perceived as a filled cell.
`arc`	Area (2D)	A filled wedge (sector of a circle); encodes proportion as angle and area.

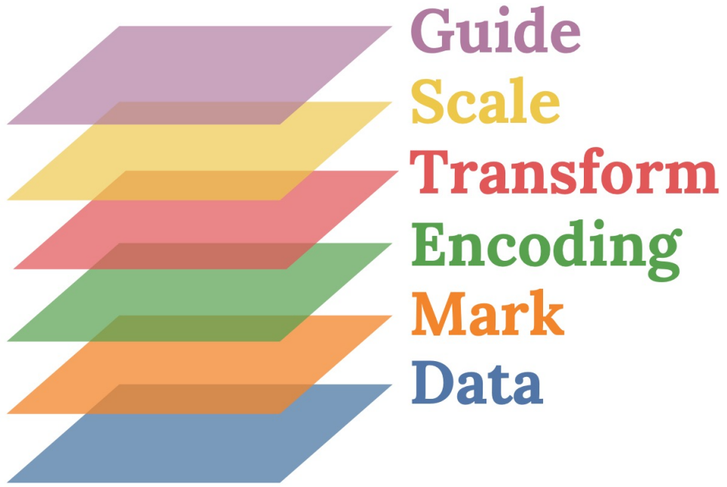
From Theory to Tooling - Channels

Munzner Channel Category	**Altair/Vega-Lite Channels**	**Notes / Examples**
Position	`x`, `y`, `latitude`, `longitude`	Encodes data by spatial location. Geographic uses `lat/lon`.
Size	`size`	Size of points, thickness of lines
Shape	`shape`	Glyph shape for point marks (`circle`, `square`, `triangle`, etc.).
Color	`color`, `fill`, `stroke`	Encodes quantitative (via luminance/saturation) or categorical (via hue) values. `fill` = inside, `stroke` = outline.
Tilt / Angle	`angle`	Rotation of glyphs (e.g., wedges in a pie chart, tilted text).
Transparency	`opacity`	Helps with overplotting, can encode quantitative intensity. Sometimes it merges with color saturation.
	`detail`	Adds grouping distinctions without changing position. Functions like a “non-positional differentiator.”
	`text`, `tooltip`	Direct labeling (`text`) or interactive display (`tooltip`).
	`order`	Drawing order of marks (important for lines and stacking).
	`facet`, `row`, `column`	Breaks visualization into panels, aligning with Munzner’s “small multiples” strategy.

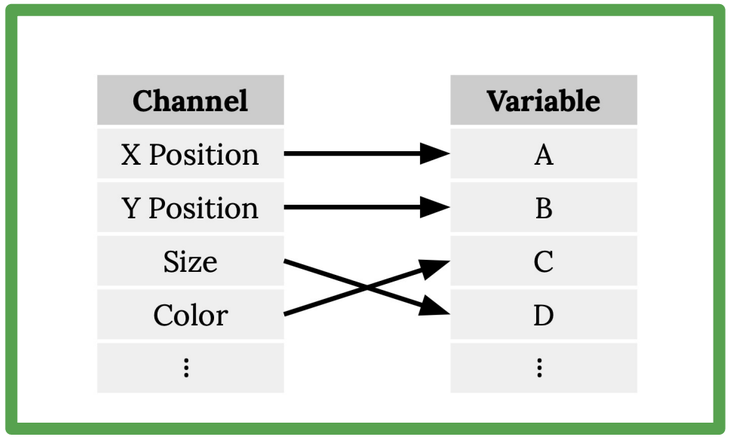


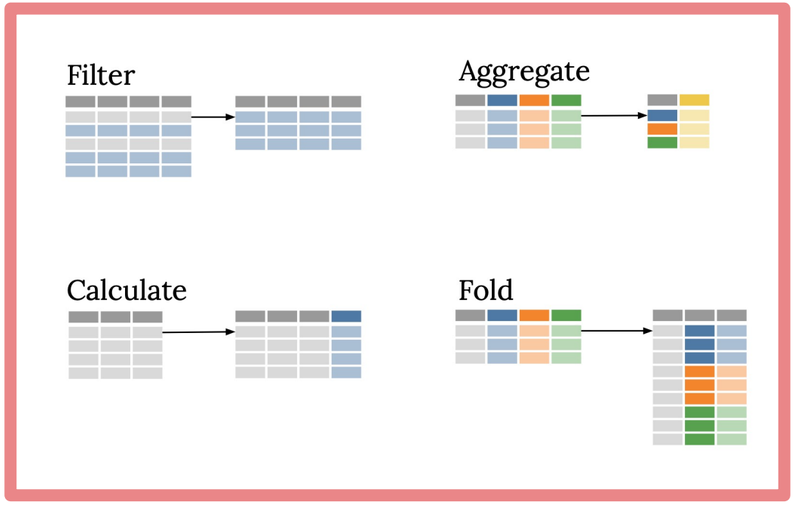






$$f(\text{domain}) \rightarrow \text{range}$$





Altair Basics

Create a Chart Object

Attach data to the Chart Object

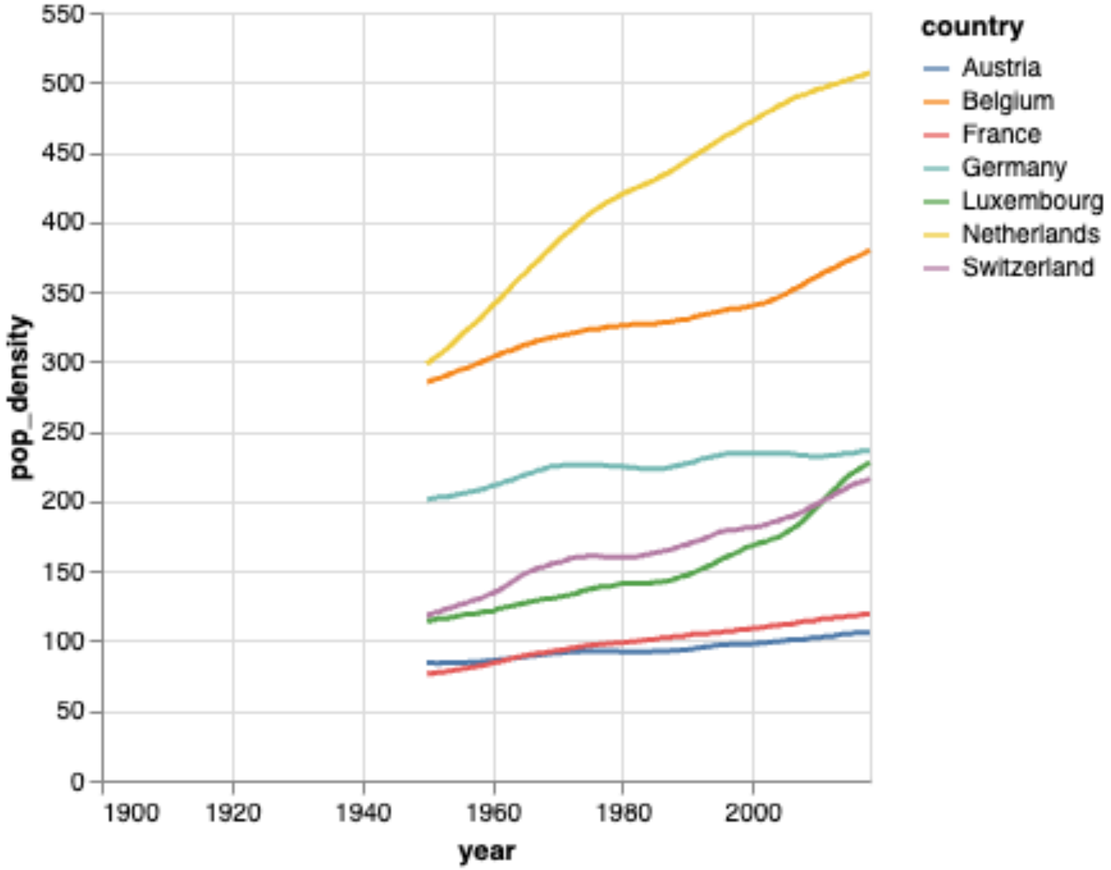
Specify the mark type

```
alt.Chart(data).mark_bar().encode(
        channel_1 = 'column1',
        channel_2 = 'column2',
)
```

Specify each channel and what data attribute it encodes

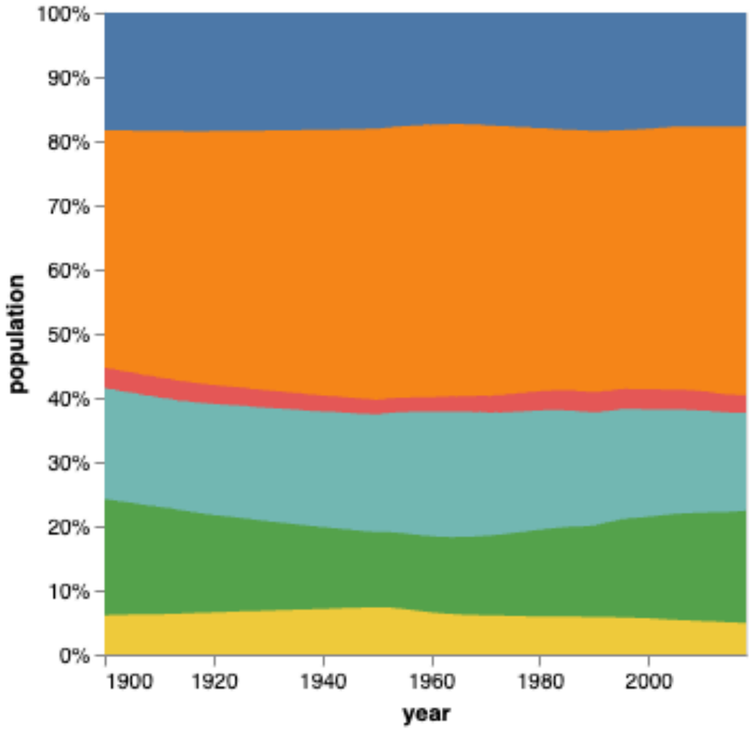
Tutorial 3 Recap Clicker Question

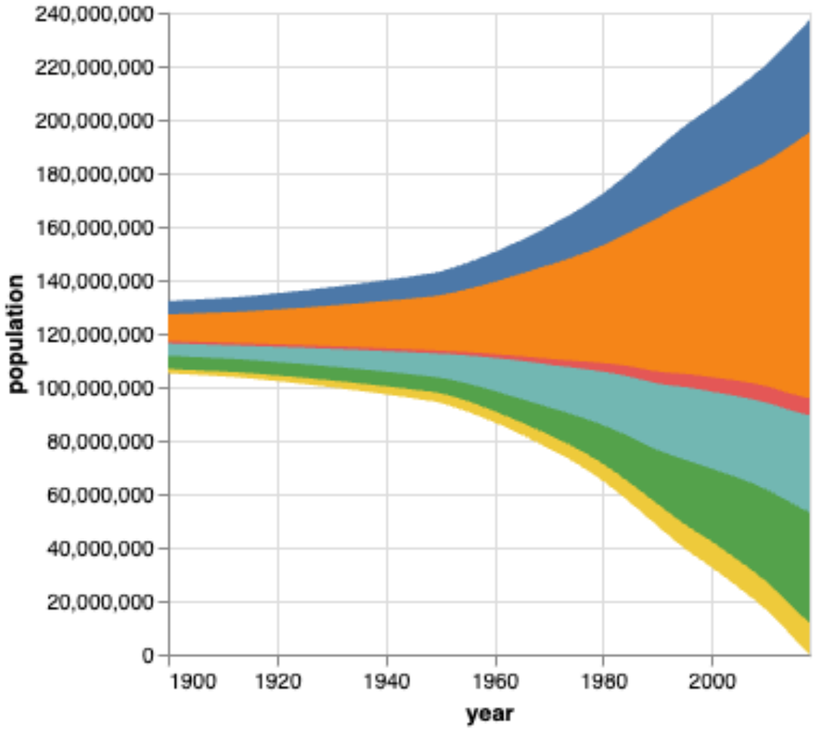
Why does this chart look like this?

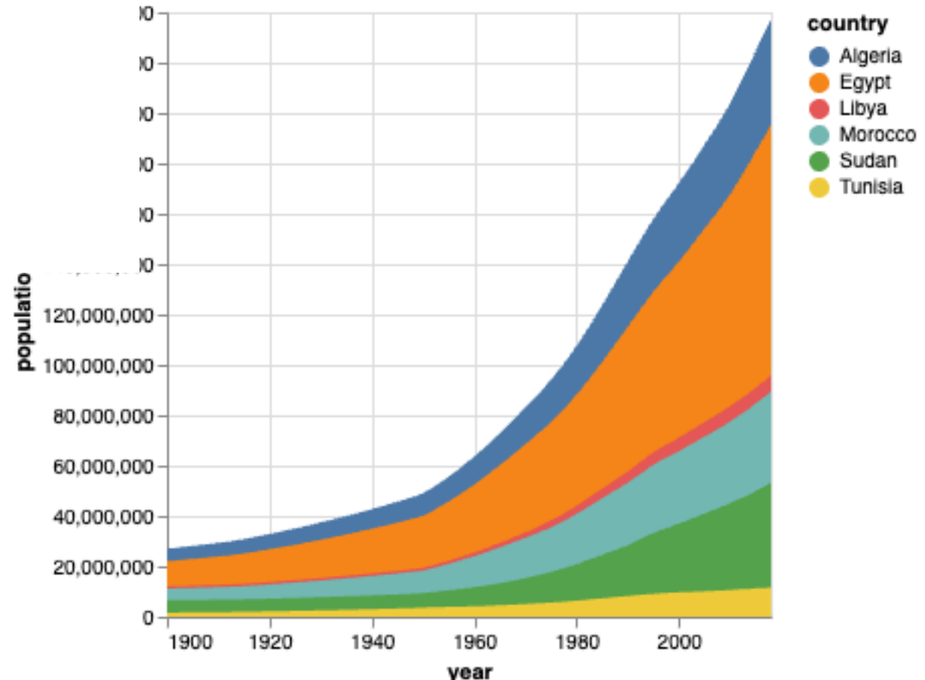


Tutorial 3 Recap Clicker Question

What did I do to make the chart look like this?







How would you describe the pace of the tooling aspects of the course?

A. Fast
B. Slow
C. Okay

How would you describe the pace of the theoretical aspects of course?

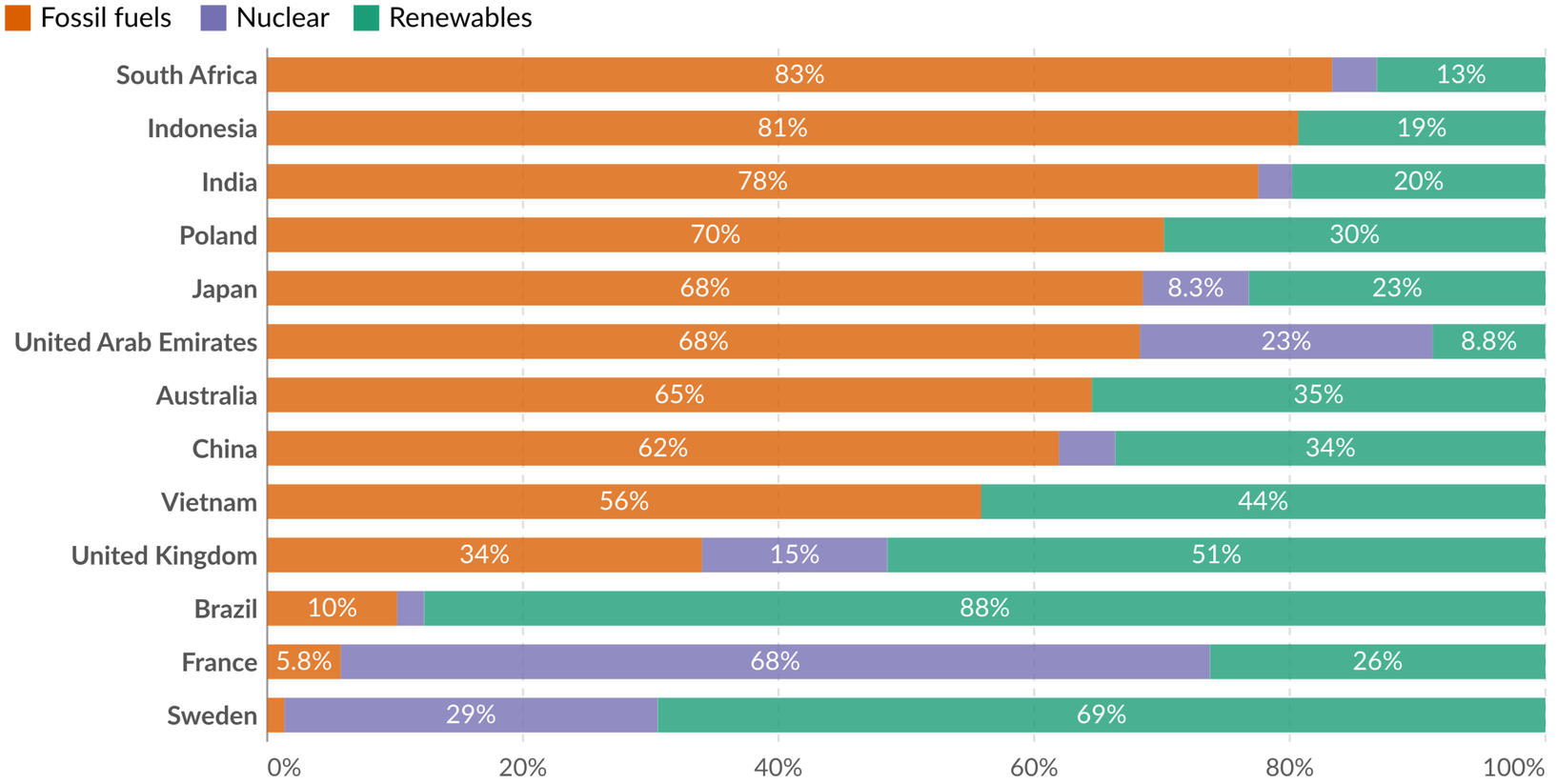
A. Fast
B. Slow
C. Okay

OWID Energy Dataset

Per capita electricity generation from fossil fuels, nuclear and renewables, 2024

Our World in Data

Per capita electricity generation from fossil fuels (coal, gas, and oil), nuclear, and renewables (solar, wind, hydropower, bioenergy, geothermal, wave, and tidal).



Data source: Ember (2025); Energy Institute - Statistical Review of World Energy (2025); Population based on various sources (2024)

OurWorldinData.org/electricity-mix | CC BY

OWID Energy Dataset

Column	Description	Unit
country	Country name	Geographic location
year	Year of observation	Date
nuclear_electricity	Electricity generation from nuclear	TWh
hydro_electricity	Electricity generated from hydropower	TWh
population	Country population	Count
fossil_fuel_consumption	Primary energy consumption from fossil fuels	TWh
coal_production	Coal production volume	TWh
coal_consumption	Coal consumption volume	TWh
nuclear_share_energy	Share of primary energy consumption that comes from nuclear power	%
fossil_share_energy	Share of primary energy consumption that comes from fossil fuels	%
hydro_share_energy	Share of primary energy consumption that comes from hydropower	%
renewables_share_energy	Share of primary energy consumption that comes from other renewables	%
biofuel_consumption	Primary energy consumption from biofuels	TWh
gas_consumption	Primary energy consumption from gas	TWh
oil_consumption	Primary energy consumption from oil	TWh
hydro_consumption	Primary energy consumption from hydropower	TWh
other_renewable_consumption	Primary energy consumption from other renewables	TWh
wind_electricity	Electricity generation from wind power	TWh
solar_electricity	Electricity generation from solar power	TWh
hydro_electricity	Electricity generation from hydropower	TWh

Note: TWh = Terawatt-hours (1 TWh = 1 billion kilowatt-hours)

OWID Energy Dataset

Outline

Exploratory Task – How has the generation of wind energy changed over the last 60 years?

- Data Task: Filter the data to only include records for a select number of countries (feel free to change this list)
- Viz Task: Use a multi-line chart to visualize the wind electricity generation over the given time period

Exploratory Task – How has the composition of low-carbon electricity sources evolved?

- Data Task: create a dataset for hydro, nuclear, solar and wind electricity at the global level
- Viz Task: Use a stacked area chart to visualize the data

Data Task – Create a dataframe for select countries

Which of the following code snippets accomplishes the task

A.

```
owid_data.query('country in @wind_countries').copy()
```

B.

```
owid_data[owid_data['country'].isin(wind_countries)].copy()
```

C.

```
owid_data.loc[owid_data.country.isin(wind_countries)].copy()
```

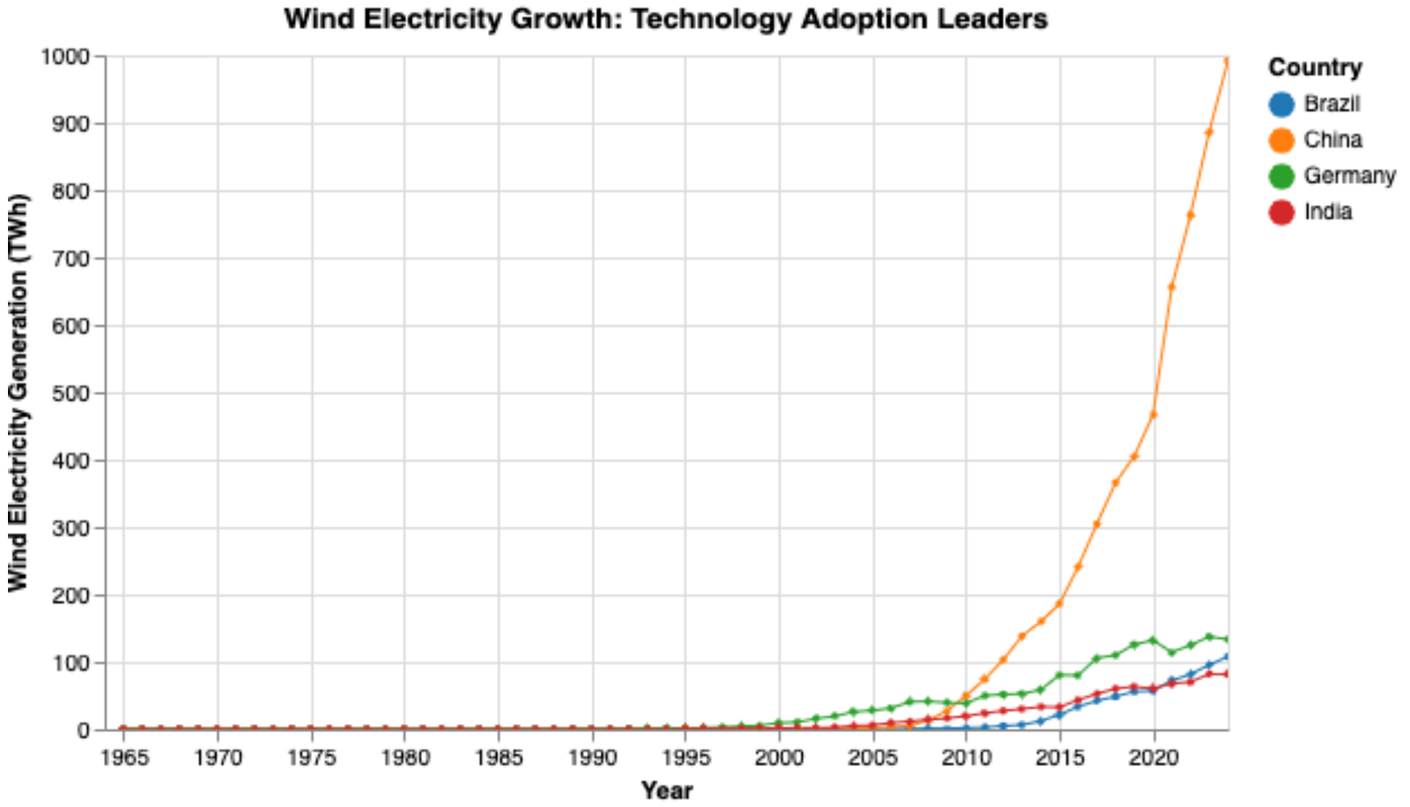
Data Task – Create a dataframe for select countries

```
owid_data.query('country in @wind_countries').copy()

owid_data[owid_data['country'].isin(wind_countries)].copy()

owid_data.loc[owid_data.country.isin(wind_countries)].copy()
```

Viz Task: Create a multi-line chart.

Suggestion – first make the chart, then focus on styling it
(i.e., setting the channel and chart properties)

A. Still working
B. Finished with basic chart
C. Finished with basic chart AND styling
D. Stuck



Option A – Inline encoding

```
wind_comparison = alt.Chart(wind_data).mark_line(
    strokeWidth=1,
    point=alt.OverlayMarkDef(size=10, filled=True)
).encode(
    x=alt.X('year:T',
            title='Year',
            axis=alt.Axis(format='%Y')),
    y=alt.Y('wind_electricity:Q',
            title='Wind Electricity Generation (TWh)',
            axis=alt.Axis(format='.0f')),
    color=alt.Color('country:N',
                    title='Country',
                    scale=alt.Scale(scheme='category10')),
tooltip=[
    alt.Tooltip('country:N', title='Country'),
    alt.Tooltip('year:T', title='Year', format='%Y'),
    alt.Tooltip('wind_electricity:Q', title='Wind Generation (TWh)',
format='.1f')
  ]
```

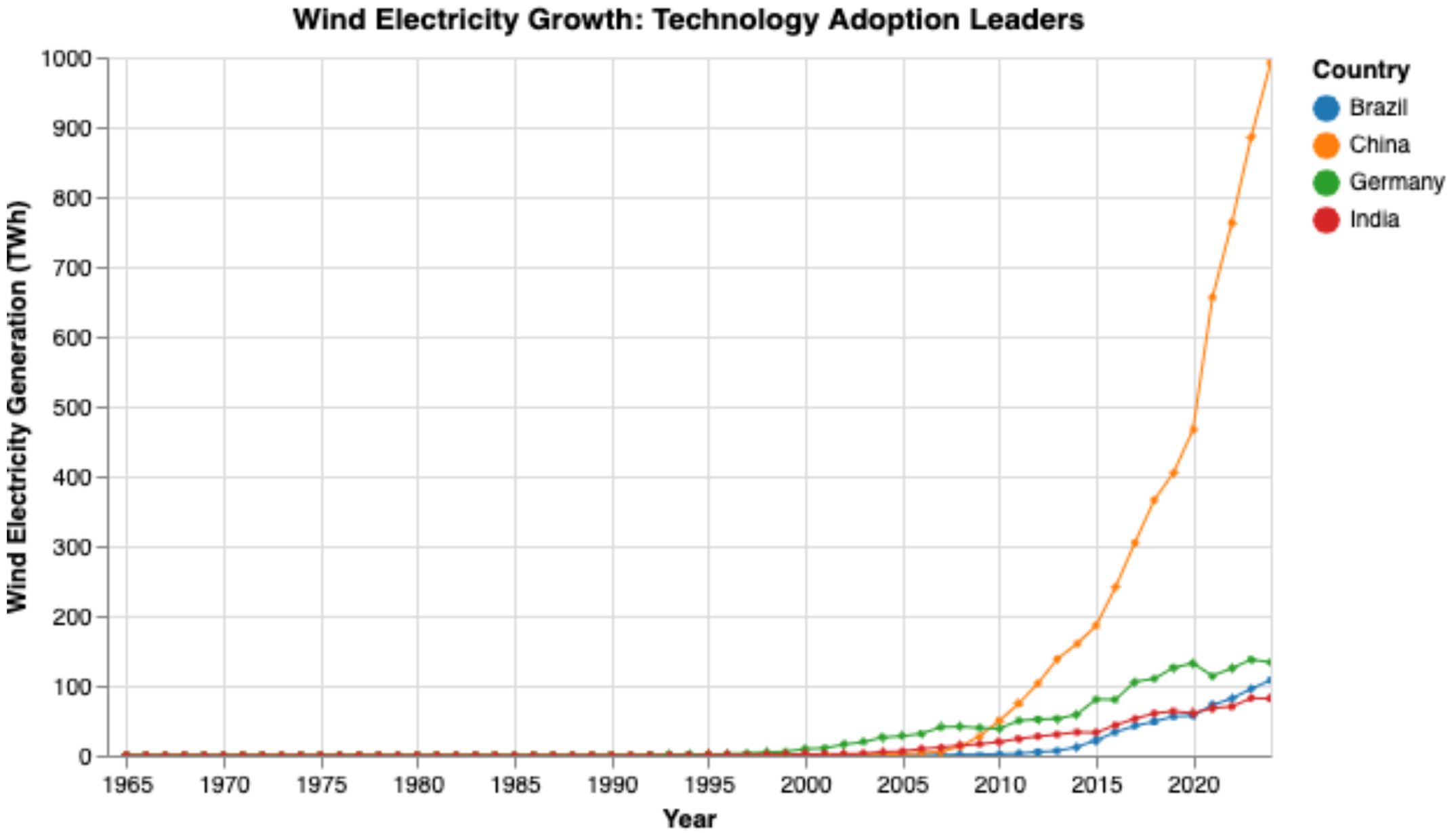
Option B – Method Chaining with reusable channel objects

```
x_prop = alt.X('year:T').title('Year').axis(format='%Y')
y_prop = alt.Y('wind_electricity:Q').title('Wind Electricity Generation
(TWh)').axis(format='.0f')
color_prop = alt.Color('country:N').title('Country').scale(scheme='category10')


wind_comparison_alt = alt.Chart(wind_data).mark_line(
        strokeWidth=1,
        point=alt.OverlayMarkDef(size=10, filled=True)
    ).encode(
        x=x_prop,
        y=y_prop,
        color=color_prop,
        tooltip=tooltip_prop
    ).properties(
        title='Wind Electricity Growth: Technology Adoption Leaders',
        width=500,
        height=300
    ).interactive()
```

Option C – Declarative Dictionary Style

```
wind_comparison_dict = alt.Chart(wind_data).mark_line(
    strokeWidth=1,
    point={"size": 10, "filled": True}
).encode(
    x={"field": "year", "type": "temporal", "title": "Year", "axis":
{"format": "%Y"}},
color={"field": "country", "type": "nominal", "title": "Country",
"scale": {"scheme": "category10"}},
    tooltip=[
        {"field": "country", "type": "nominal", "title": "Country"},
        {"field": "year", "type": "temporal", "title": "Year", "format":
"%Y"},
]
).properties(
    title="Wind Electricity Growth: Technology Adoption Leaders",
    width=500,
    height=300
).interactive()
```

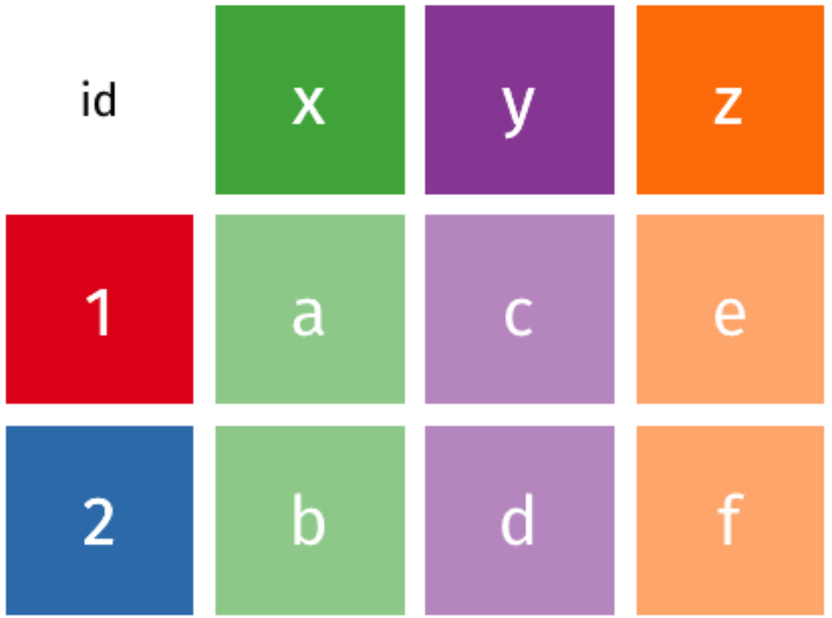
What do you notice?



Data Task: Prepare the global low-carbon dataset for each tech.

5 rows × 132 columns pd.Dataframe

	country	year	iso_code	entity_type	region	population	gdp	biofuel_cons_change_pct	biofuel_cons_change_twh	biofuel_cons_per_capita
1952	World	2020-01-01	NaN	region	NaN	7.887001e+09	1.185900e+14	-5.241	-59.084	135.442
1953	World	2021-01-01	NaN	region	NaN	7.954448e+09	1.260048e+14	5.606	59.890	141.822
1954	World	2022-01-01	NaN	region	NaN	8.021407e+09	1.301126e+14	5.763	65.012	148.743
1955	World	2023-01-01	NaN	region	NaN	8.091735e+09	NaN	10.568	126.085	163.033
1956	World	2024-01-01	NaN	region	NaN	8.161972e+09	NaN	3.613	47.662	167.469



https://github.com/gadenbuie/tidyexplain#spread-and-gather

Data Task: Prepare the global low-carbon dataset

STEP 0: Select the low-carbon sources you will plot.
STEP 1: Filter the dataset to global totals.
STEP 2: Select the columns of interest
STEP 3: Reshape from wide → long.
STEP 4: Clean and replace the technology labels for presentation.

Pandas can help reshape dataframes
- `.melt()`: make wide data long
- `.pivot()`: make long data wide
- `.pivot_table()`: same as `.pivot()` but can handle multiple indexes

Data Task: Prepare the global low-carbon dataset

A - STEP 1: Filter the dataset to global totals.

B - STEP 2: Select the columns of interest

```
selected = world_data[['year'] + low_carbon_sources]
```

C - STEP 3: Reshape from wide → long.

```
world_low_carbon_long = selected.melt(id_vars=['year'],
var_name='technology', value_name='generation')
```

D - STEP 4: Clean and replace the technology labels for presentation.

```
world_low_carbon_long['technology'] =
world_low_carbon_long['technology'].map(tech_names)
```

E – I'm stuck ooooo ☹

Data Wrangling

5 rows × 132 columns pd.Dataframe

	country	year	iso_code	entity_type	region	population	gdp	biofuel_cons_change_pct	biofuel_cons_change_twh	biofuel_cons_per_capita
1952	World	2020-01-01	NaN	region	NaN	7.887001e+09	1.185900e+14	-5.241	-59.084	135.442
1953	World	2021-01-01	NaN	region	NaN	7.954448e+09	1.260048e+14	5.606	59.890	141.822
1954	World	2022-01-01	NaN	region	NaN	8.021407e+09	1.301126e+14	5.763	65.012	148.743
1955	World	2023-01-01	NaN	region	NaN	8.091735e+09	NaN	10.568	126.085	163.033
1956	World	2024-01-01	NaN	region	NaN	8.161972e+09	NaN	3.613	47.662	167.469

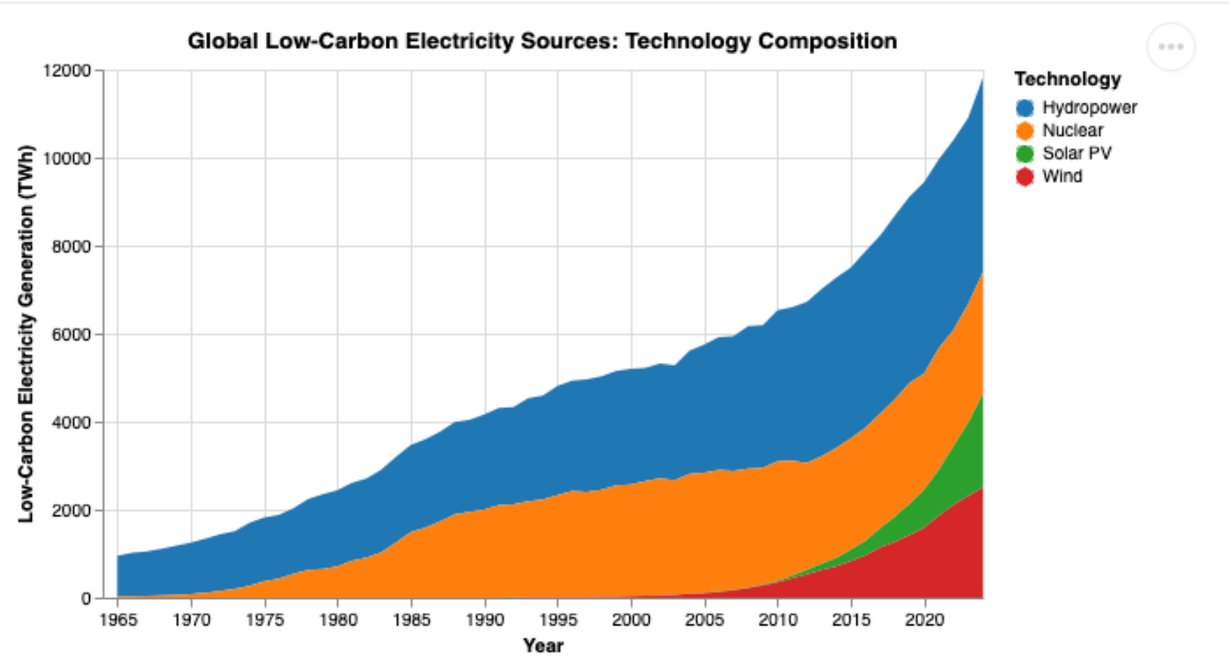
5 rows × 3 columns pd.Dataframe

	year	technology	generation
0	1964-01-01	Hydropower	NaN
1	1965-01-01	Hydropower	923.198
2	1966-01-01	Hydropower	983.817
3	1967-01-01	Hydropower	1005.742
4	1968-01-01	Hydropower	1059.289

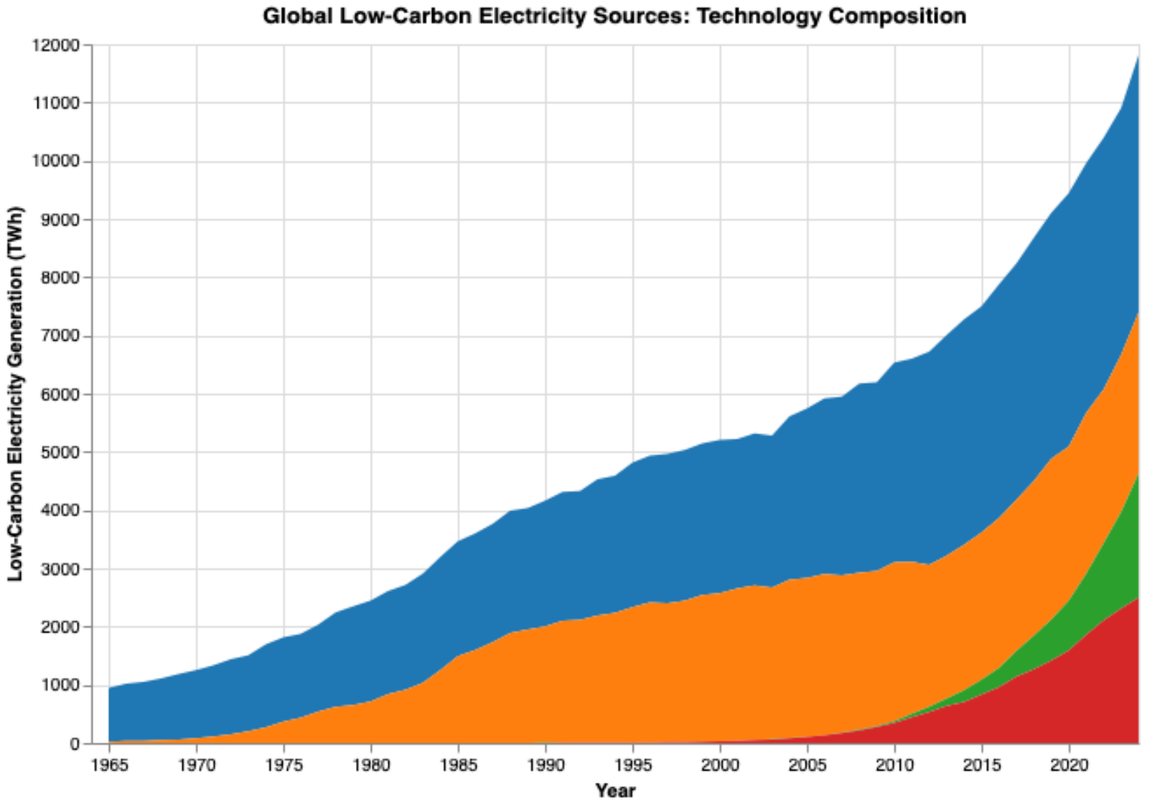
Viz Task: Create a stacked area chart.

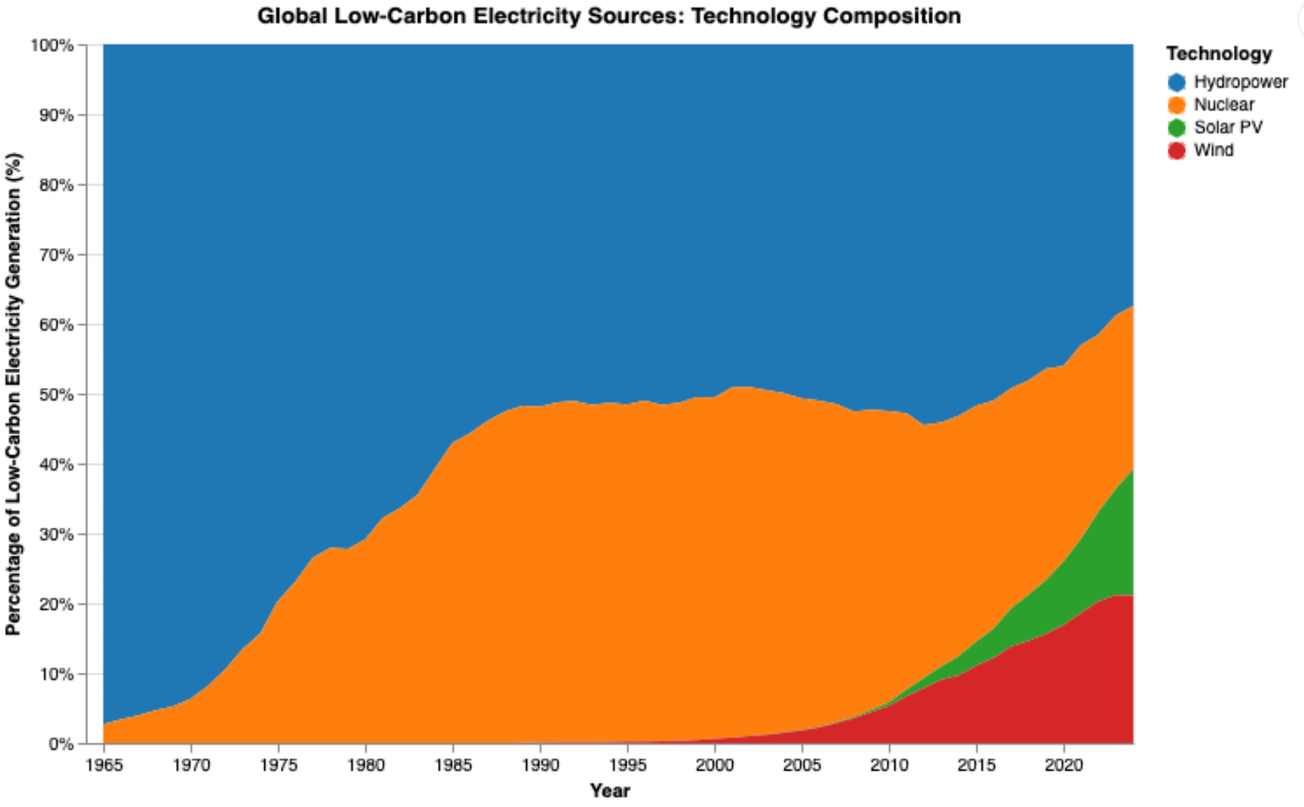
Suggestion – first make the chart, then focus on styling it
(i.e., setting the channel and chart properties)

A. Still working
B. Finished with basic chart
C. Finished with basic chart
   AND styling
D. Stuck




```
color=alt.Color('technology:N',
    scale=alt.Scale(range=['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728']),
    title='Technology'),
```


Global Low-Carbon Electricity Sources: Technology Composition
Low-Carbon Electricity Generation (TWh)
12000
11000
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010
2015
2020
Year


Global Low-Carbon Electricity Sources: Technology Composition
Percentage of Low-Carbon Electricity Generation (%)
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010
2015
2020
Year
Technology
Hydropower
Nuclear
Solar PV
Wind

```
# Stacked area with thoughtful color scheme
low_carbon_stack = alt.Chart(world_low_carbon_long).mark_area().encode(
    x=alt.X('year:T', title='Year'),
    y=alt.Y('generation:Q',
            stack='zero',
            title='Low-Carbon Electricity Generation (TWh)',
            axis=alt.Axis(format='.0f')),
    color=alt.Color('technology:N',
                    scale=alt.Scale(range=['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728']),
                    title='Technology'),
    tooltip=[
        alt.Tooltip('year:T', title='Year', format='%Y'),
        alt.Tooltip('technology:N', title='Technology'),
        alt.Tooltip('generation:Q', title='Generation (TWh)', format='.0f')
    ]
).properties(
    title="Global Low-Carbon Electricity Sources: Technology Composition",
    width=600,
    height=400
)

low_carbon_stack_normalized = low_carbon_stack.encode(y=alt.Y('generation:Q',
                                                              stack='normalize',
                                                              title='Percentage of Low-Carbon Electricity Generation (%)'),
)

low_carbon_stack | low_carbon_stack_normalized
```

Feedback on Lecture style for Coding

A. I like the pace and structure of today's coding lecture
B. I prefer the pace & structure of last week's coding lecture
C. I like to see you make mistakes, so please code.
D. Neither worked for me, I don't like coding, I like singing.
E. Either is fine.

Administrivia – Office Hours

TA Office Hours (online)

Monday 11:00am – 12pm

Wednesday 10:00 – 11am

Instructor Office Hours

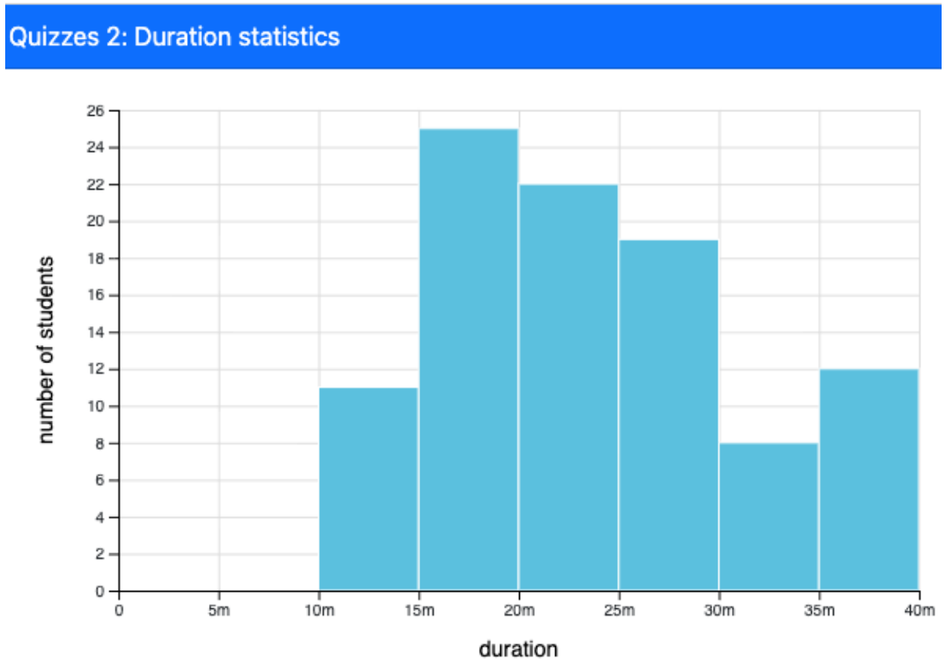
Monday and Wednesday 5 – 6pm (currently in this room)

Tuesdays 2 – 3pm ICCS 227

Participation is Clickers, not PL (unless otherwise stated)

Quiz Updates

- Quiz 1 retake will be around week 7 – 9. It has been re-opened so you can access it
- Quiz 2
- Quiz 3
  - 30 theory – data abstraction, viz description
  - 70% altair coding – Weeks 2 AND 3
  - About 10 minutes longer than Quiz 2



Quiz assesses the minimum that is expected at this point

Outside of working through the lecture in class and the tutorial before class, how much time did you spend studying for Quiz 2

A. 0 hours
B. 1 – 2 hours
C. 3 – 5 hours
D. More than 5 hours

Administrivia

- No in-person class on Monday. That doesn’t mean you shouldn’t show up. Use that time to work in your teams on your Design 1 Artifact submission (due on Sept. 26th)
- In place of an in-person lecture, there will be TWO coding tutorials.