## Kiran Gadhave, Zach Cutler, Alexander Lex http://vdl.sci.utah.edu

# **Persist:** Persistent and Reusable Interactions in Computational Notebooks







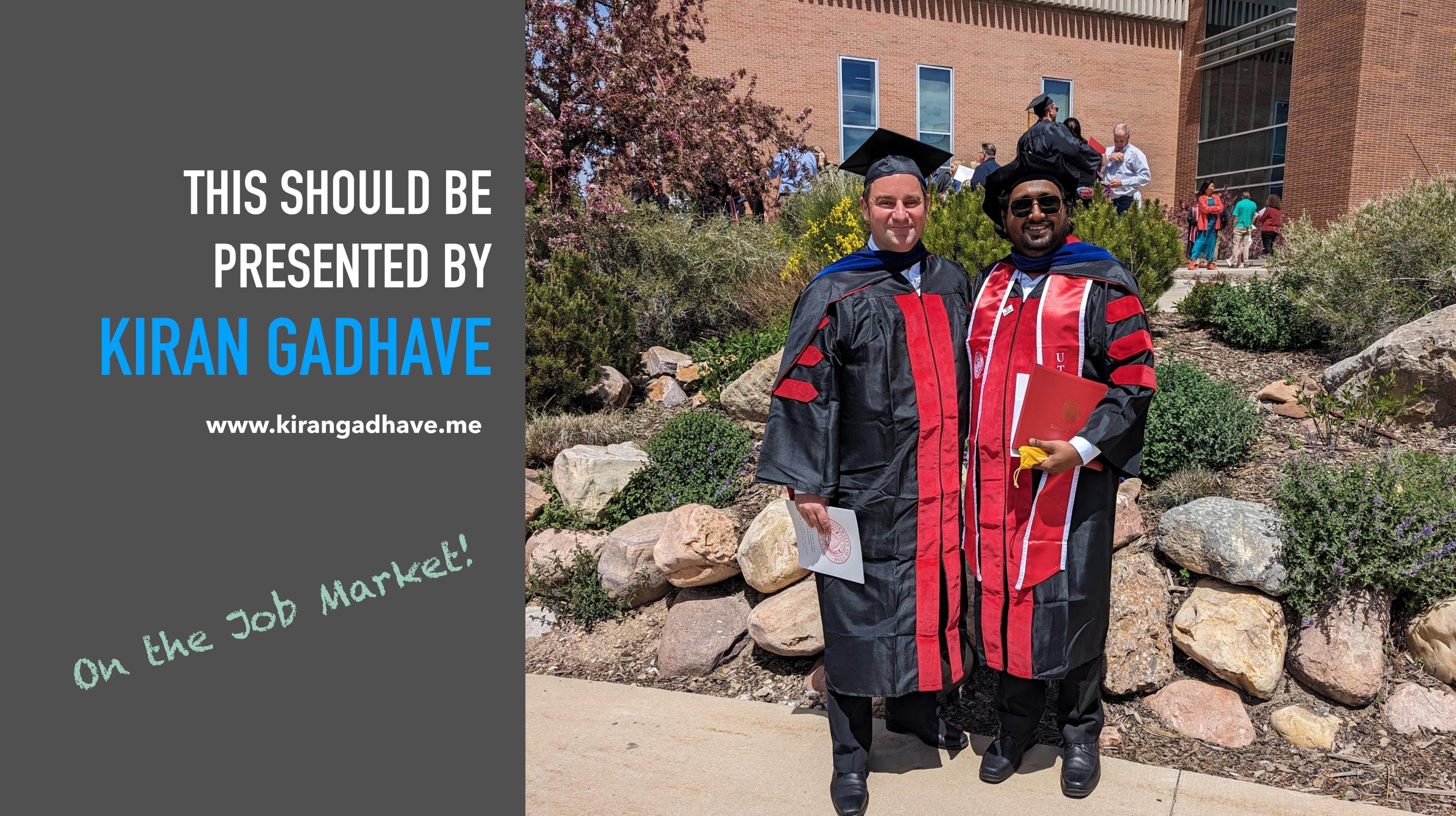


VISUAIZATION design lab

## THE UNIVERSITY OF UTAH







## PERSISTENT AND REUSABLE INTERACTIONS IN **COMPUTATIONAL NOTEBOOKS**

€	௴ 0× 8₽ 88	B 72 74 🗉	Reset Trrack Delete da	itasets				
Q Search							≣≣ Trrack B≣ Summary	
- #	Date ↑↓ ፡፡፡	: Place ↑↓ ፡፡፡ :	Trigger		s ↑↓ ፡፡፡ :   ;Aspect ↑∖	↓ ፡፡፡ :   Elevation_feet ↑↓ :	Root	
2338	3-23-2023	Dry Creek	Natural	Change column 'Trigger' data type ►	West	8000	<ul> <li>Rename column ;Weak Lay</li> <li>Rename column ;Trigger to</li> </ul>	
955	1-19-2014	Whitney Basin	Snowmobiler	団 Drop column 'Trigger'	East	10500	Drop column Comment	
1028	2-21-2014	Chalk Creek	Natural	🖉 Rename column 'Trigger'	Northeast	10600	<ul> <li>O Updated column 'Depth_i</li> <li>O Updated column 'Depth_i</li> </ul>	
1024	2-17-2014	Upper Weber Canyon	Natural	<b>Ξ↑</b> Sort by Trigger ascending	Northeast	10400	Updated column 'Depth_i Finished removing null values from       O     O     Changed column 'Dept	
998	2-12-2014	Upper Weber Canyon	Natural	<b>Ξ↓</b> Sort by Trigger descending	Northeast	10400	• Sort (descending) by 'Dep	
938	1-14-2014	Upper Weber Canyon	Explosive	-🛱 Pin to left	East	10300	O Drop column Coordinates	(1
✓ 1299	1-26-2016	Currant Creek Peak	Snowmobiler	➡ Pin to right	Southwest	9500	<pre>selector = alt.selection_interval("selector", encodings=[</pre>	
1044	2-28-2014	Chalk Creek	Natural	🕅 Unpin	Northeast	10600	<pre>scatterplot = alt.Chart().mark_point().encode(x="Elevatio chart = alt.Chart().mark_bar().encode(</pre>	n_feet:Q", y="Depth_inches:Q")
2348	3-30-2023	Bunnels	Natural	Reset column size	Northeast	10800	<pre>x=alt.X("Month:0").sort([10]), y="count()",</pre>	
1977	4-6-2021	Blue Ice	Natural	X Hide Trigger column	Northeast	10400	<pre>opacity=alt.condition(selector, alt.value(1), alt.val</pre>	ue(0.3))
3 of 2392 rov	N(s) selected			≣≣ Show all columns	ws per page 10	) of 2,392 K × >	).add_params(selector).properties(width=400)	
Dataframe nam		s_df (¯) ≛ [↗ 前) (no_coordir	nates_df <u>ि</u> 古 🗹 前	final_df 「同 亡 I」 前			PR.PersistChart(chart   scatterplot, data=df_outlier_remo	Reset Trrack Delete o
							600 Select a category ⑦	160
							Avalanche Season ×	140-
_	_						500	120-
Ju	oyter I	Plugin					9400 - Start	100
							Niddle	80
							End	a <sub>60</sub>
ttps	://vdl.s	sci.utah	.edu/	persist/			No Assignment	
in	install mersist ext						Month	0 2,000 4,000 6,000 8,000 Elevation_feet

pip install persist ext

Demo Notebook: <u>https://tinyurl.com/5db7nynn</u>

G 📥 Dataframe name. 🖈 df\_with\_season 🛛 🖨

E Reset Trrack Delete datasets Avalanche Season ⊱ Trrack 🛛 🗄 Summary Middle œ 😑 No Assignment 140 Start 120 00 **Q** Root Add new category 'Avalan... 000000 Add new option 'Start' to ... 000000 80 -Add new option 'Middle' t... 00 Add new option 'End' to c... 000 0000 0 Selected Month (10 to 12) Assign Avalanche Season... Selected Month (1 to 3) Assign Avalanche Season... Selected Month (4 to 6) 2,000 4.000 6,000 8,000 10,000 12,000 Elevation feet Dataframes: 🗭 df\_with\_season 🛱 📥



## DATASET EXAMPLE: HISTORICAL AVALANCHES IN UTAH

## Avalanches are a major hazard in Utah Utah Avalanche Center collects data about avalanches, including where it occurred (location, elevation), how it occurred, how big it was, etc.



## WHAT IS THIS TALK ABOUT?

## Supposed you're doing data analysis in Python

# What's the pandas code... • ...to drop a column?

• ...to change the order of columns? • ...to change the label of a column?

Nothing here is hard, but it's annoying.







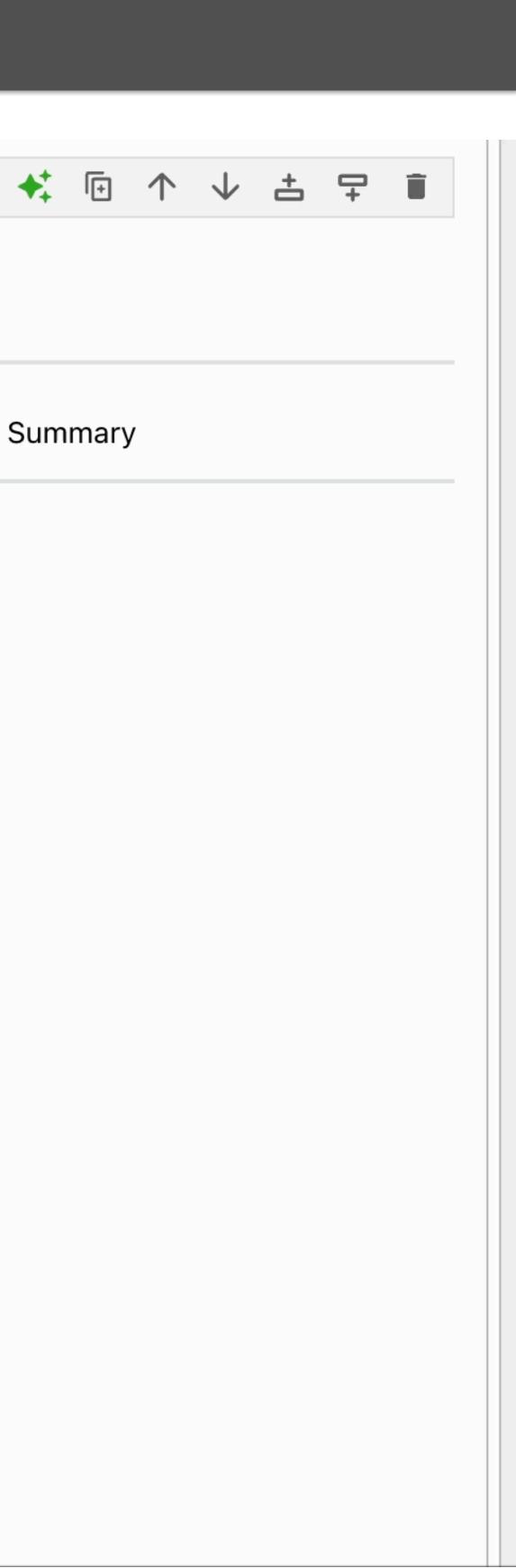


## PERSIST MAKES THIS EASY

[4]: PR.PersistTable(avalanches, df\_name="avalanches")

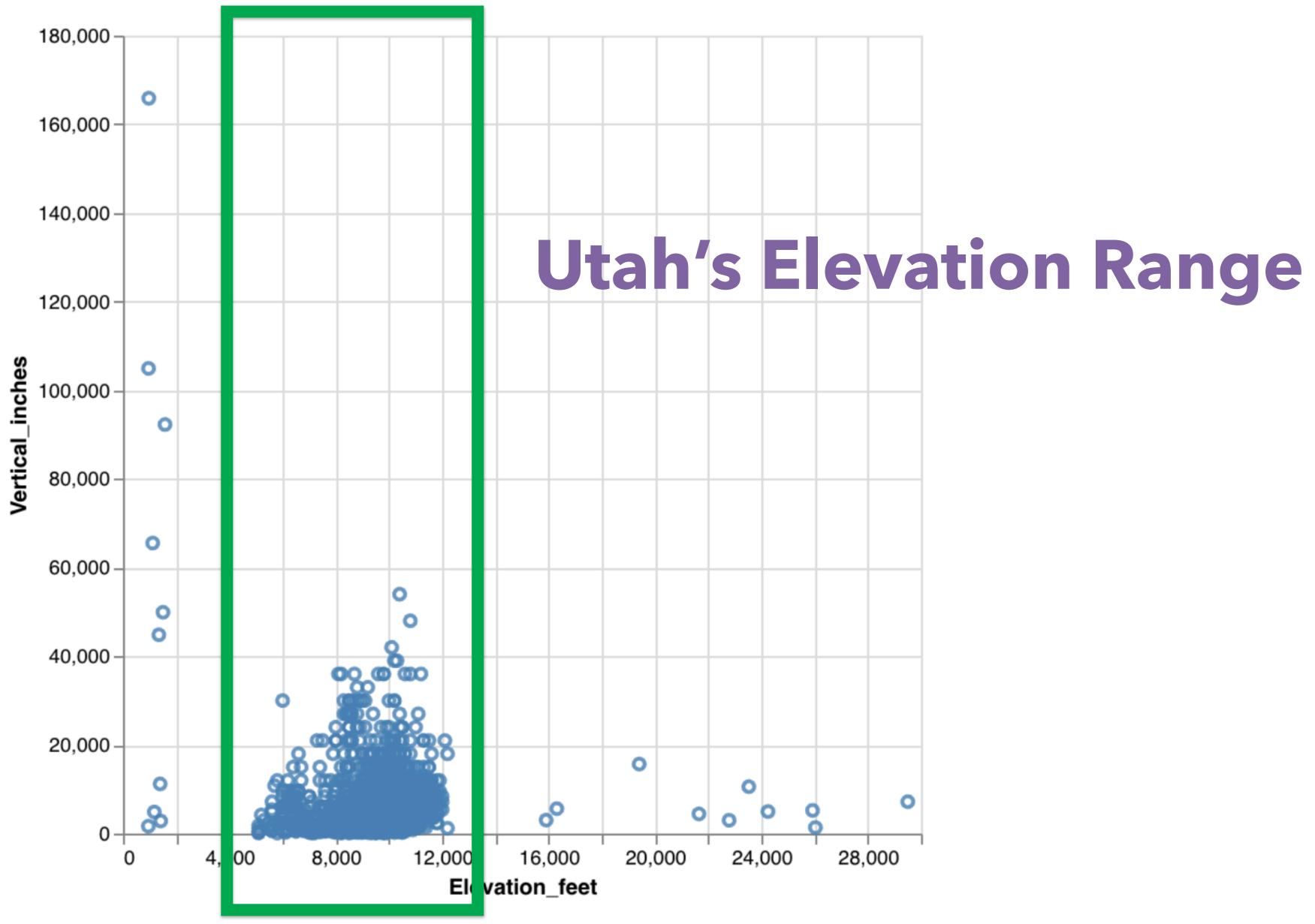
Search		•					ୃ\$⊷ Trrack 📲 Summary
#	;Region ↑↓ ፡፡፡	: Month	↑↓ ። : <b>Day</b>	↑↓ ፡፡፡ : Year	↑↓ ፡፡፡ : ;Trigger ↑↓ ፡፡፡	: ;Weak Layer ↑↓ ::	Root
) 1	Salt Lake	11	9	2012	Snowboarder	New Snow/Old Snow	
2	Salt Lake	11	11	2012	Skier	New Snow/Old Snow	
3	Salt Lake	11	11	2012	Skier	Facets	
4	Salt Lake	11	11	2012	Skier	New Snow	
5	Salt Lake	11	11	2012	Skier	Facets	
6	Salt Lake	11	10	2012	Skier	New Snow/Old Snow	
7	Salt Lake	11	12	2012	Skier	Facets	
8	Salt Lake	12	8	2012	Skier	Facets	
9	Salt Lake	12	9	2012	Skier	Facets	
10	Salt Lake	12	10	2012	Skier	Facets	
				Rows per pag	ge 10 0 1-10 of 2,392	2  < < > >	

Dataframe name	
🖈 avalanches   🗖 📩	



## WHAT IS THIS TALK **ABOUT?**

## Have you ever plotted something and wished you could just "fix" things as you spot them?

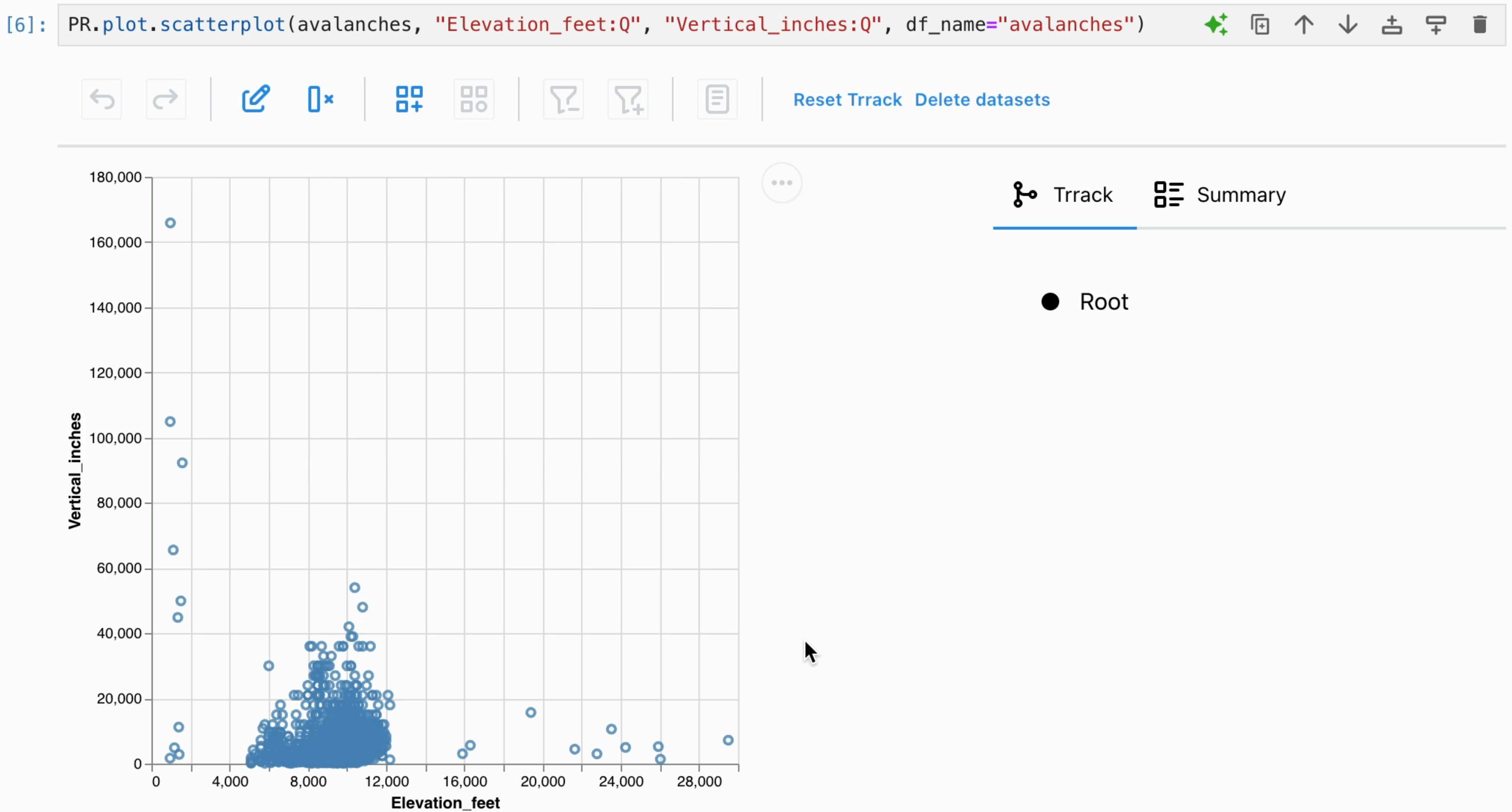


vas valanche th (big) dee

## **Elevation where the avalanche occurred**



## PERSIST MAKES THIS EASY





## SO WHAT'S SPECIAL HERE?

## Lots of vis tools support these operations Most data wrangling happens in code: it's just more powerful **Opportunity: bring interactive operations to code!**

## Persist works **INSIDE your Jupyter Notebook**

# BRIDGING BETWEEN DATA ANALYSIS MODALITIES

# BRIDGING BETWEEN MODALTES

# What are Modalities?

## DATA ANALYSIS 1. Interactive Vis 2. Code

## INTERACTIVE VISUALIZATION: BENEFITS

## Intuitive Easy to use Uses human perceptual capabilities

Line	Up Soc	cer Sta	ts 🔒					
Agg	Rank	Sele	${\mathbb T}$ Player		Current	T Current		Po
•••	•••	↓ <u>1</u> •••		Ļ	· · · · · B	↓ <sup>A</sup> Q <b>T</b> ···	Ļ	2 B
				Sup				
		_		_			_	CF
	1		Dmitri Poloz			Zenit St. Peters		W
	2	_	Emiliano Rigoni			Zenit St. Peters		W
	3	_	Sebastián Driussi			Zenit St. Peters		CF
	4	_	Aleksandr Kokorin			Zenit St. Peters		CF
	5	_	Anton Zabolotnyi			Zenit St. Peters		CF
	6	_	Quincy Promes			Spartak Mosco		W
	7	_	Pedro Rocha			Spartak Mosco		W
	8	_	Lorenzo Melgarejo			Spartak Mosco		W
	9	_	Zelimkhan Bakaev			Spartak 2 Mosc		W
	10		Luiz Adriano		RUS - Pre	Spartak Mosco		CF
	11		Zé Luís		RUS - Pre	Spartak Mosco		CF
	12		Ahmed Musa		RUS - Pre	CSKA Moscow		CF
	13		Fedor Chalov		RUS - Pre	CSKA Moscow		CF
	14		Timur Zhamaletdinov		RUS - Pre	CSKA Moscow		CF
	15		Wanderson		RUS - Pre	FK Krasnodar		W
	16		Joãozinho		RUS - Pre	FK Krasnodar		W
	17		Andrei Ivan		RUS - Pre	FK Krasnodar		W
	18		Ricardo Laborde		RUS - Pre	FK Krasnodar		W
	19		Magomed-Shapi Suleyr		RUS - Pre	FK Krasnodar		W
	20		Fedor Smolov		RUS - Pre	FK Krasnodar		CF
	21		lvan Ignatjev		RUS - Pre	FK Krasnodar		CF
	22		Alan Kasaev		RUS - Pre	Lokomotiv Mos		W
	23		Jefferson Farfán		RUS - Pre	Lokomotiv Mos		W
	24		Arshak Koryan		RUS - Pre	Lokomotiv Mos		W
	25		Éder		RUS - Pre	Lokomotiv Mos		CF
	26		Ari		RUS - Pre	Lokomotiv Mos		CF
	27		Gökdeniz Karadeniz		RUS - Pre	Rubin Kazan		W
	28		Rifat Zhemaletdinov		RUS - Pre	Rubin Kazan		W
	29		Sardar Azmoun		RUS - Pre	Rubin Kazan		CF
	30		Léo Jabá		RUS - Pre	Akhmat Grozny		W
	31		Bernard Berisha		RUS - Pre	Akhmat Grozny		W
	32		Magomed Mitrishev		RUS - Pre	Akhmat Grozny		W
	33		Odise Roshi		RUS - Pre	Akhmat Grozny		W
	34		Khalid Kadyrov		RUS - Pre	Akhmat Grozny		W
	35		Bekim Balaj		RUS - Pre	Akhmat Grozny		CF
	36		Ablaye Mbengue		RUS - Pre	Akhmat Grozny		CF
	37		Zaur Sadaev		RUS - Pre	Akhmat Groznv	I	CF



## INTERACTIVE VISUALIZATION: DOWNSIDES

## Limited Expressivity Some operations are difficult

e.g., conditional queries. Not reusable

need to redo analysis when data changes Not reproducible

Agg	Rank	Sele	op Player	Current	op Current	•	Po
	•••	$\downarrow \frac{1}{9}  \bullet \bullet \bullet$	↓ <sup>A</sup> <sub>z</sub> Q T ····		↓ <sup>A</sup> Q <b>T</b>	↓ª	Β
						0	CF
	1		Dmitri Poloz	RUS - Pre	Zenit St. Peters		W
	2	_	Emiliano Rigoni		Zenit St. Peters		W
	3	_	Sebastián Driussi		Zenit St. Peters		CF
	4	_	Aleksandr Kokorin		Zenit St. Peters		CF
	5	_	Anton Zabolotnyi		Zenit St. Peters		CF
	6	_	Quincy Promes		Spartak Moscov		W
	7	_	Pedro Rocha	=	Spartak Moscov		W
	8		Lorenzo Melgarejo	_	Spartak Moscov		W
	9		Zelimkhan Bakaev		Spartak 2 Mosc		W
	10		Luiz Adriano	_	Spartak Moscov		CF
	11		Zé Luís	_	Spartak Moscov		CF
	12	_	Ahmed Musa	_	CSKA Moscow		CF
	13	_	Fedor Chalov		CSKA Moscow		CF
	14	_	Timur Zhamaletdinov		CSKA Moscow		CF
	15		Wanderson	RUS - Pre	FK Krasnodar		W
	16		Joãozinho	RUS - Pre	FK Krasnodar		W
	17		Andrei Ivan	RUS - Pre	FK Krasnodar		W
	18		Ricardo Laborde	RUS - Pre	FK Krasnodar		W
	19		Magomed-Shapi Suleyr	RUS - Pre	FK Krasnodar		W
	20		Fedor Smolov	RUS - Pre	FK Krasnodar		CF
	21		Ivan Ignatjev	RUS - Pre	FK Krasnodar		CF
	22		Alan Kasaev	RUS - Pre	Lokomotiv Mos		W
	23		Jefferson Farfán	RUS - Pre	Lokomotiv Mos		W
	24		Arshak Koryan	RUS - Pre	Lokomotiv Mos		W
	25		Éder	RUS - Pre	Lokomotiv Mos		CF
	26		Ari	RUS - Pre	Lokomotiv Mos		CF
	27		Gökdeniz Karadeniz	RUS - Pre	Rubin Kazan		W
	28		Rifat Zhemaletdinov	RUS - Pre	Rubin Kazan		W
	29		Sardar Azmoun	RUS - Pre	Rubin Kazan		CF
	30		Léo Jabá	RUS - Pre	Akhmat Grozny		W
	31		Bernard Berisha	RUS - Pre	Akhmat Grozny		W
	32		Magomed Mitrishev	RUS - Pre	Akhmat Grozny		W
	33		Odise Roshi	RUS - Pre	Akhmat Grozny		W
	34		Khalid Kadyrov	RUS - Pre	Akhmat Grozny		W
	35		Bekim Balaj	RUS - Pre	Akhmat Grozny		CF
	36		Ablaye Mbengue	RUS - Pre	Akhmat Grozny		CF
	37		Zaur Sadaev	RUS - Pre	Akhmat Groznv		CF

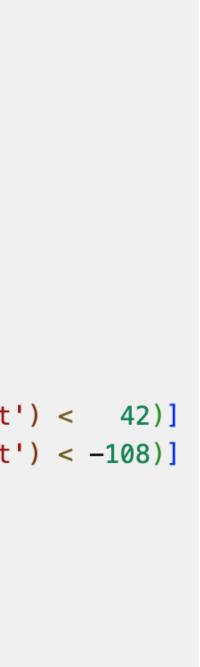


## CODE BENEFIS

## Flexible and powerful you basically can do anything Reusable if your data changes, re-run Reproducible everything is documented

1	# Keep
2	avy_df
3	
4	# Remov
5	avy_df
6	
7	<pre># Split</pre>
8	avy_df[
9	
10	# Remov
11	avy_df
12	avy_df
13	
14	# Keep
15	avy_df
16	avy_df.

```
this cell
= pd.read_csv('./avalanches.csv')
e NaN coordinates
= avy_df[avy_df['Coordinates']==avy_df['Coordinates']]
into latitude & longitude
['lat', 'lon']] = avy_df['Coordinates'].str.split(',', expand=True)
ve values outside of Utah bounds
= avy_df[ (36 < avy_df['lat'].astype('float')) & (avy_df['lat'].astype('float') < 42)]</pre>
= avy_df[(-114 < avy_df['lon'].astype('float')) & (avy_df['lon'].astype('float') < -108)]</pre>
columns we need
= avy_df[['Date', 'Region', 'Trigger', 'lat', 'lon']]
head (
```

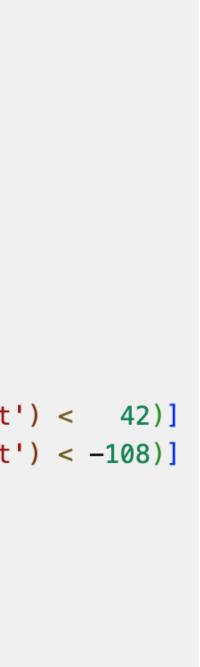


## CODE / SCRIPTING: DOWNSIDES

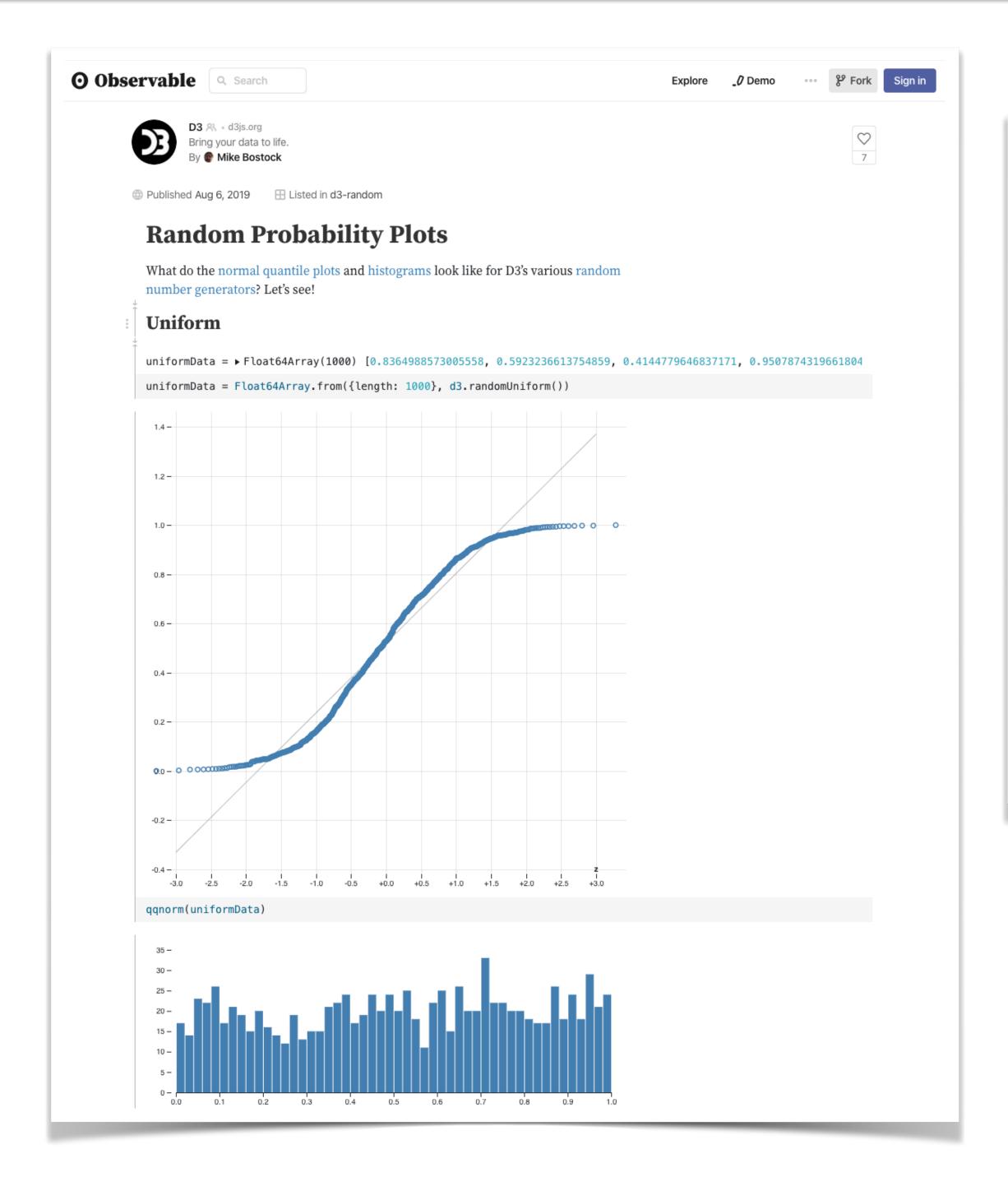
## It's hard requires extensive training reading documentation not discoverable It's time consuming even simple things require effort Some operations are difficult e.g., labeling data points

1	# Кеер
2	avy_df
3	
4	# Remov
5	avy_df
6	
7	# Split
8	avy_df[
9	
10	# Remov
11	avy_df
12	avy_df
13	
14	# Keep
15	avy_df
16	avy_df.

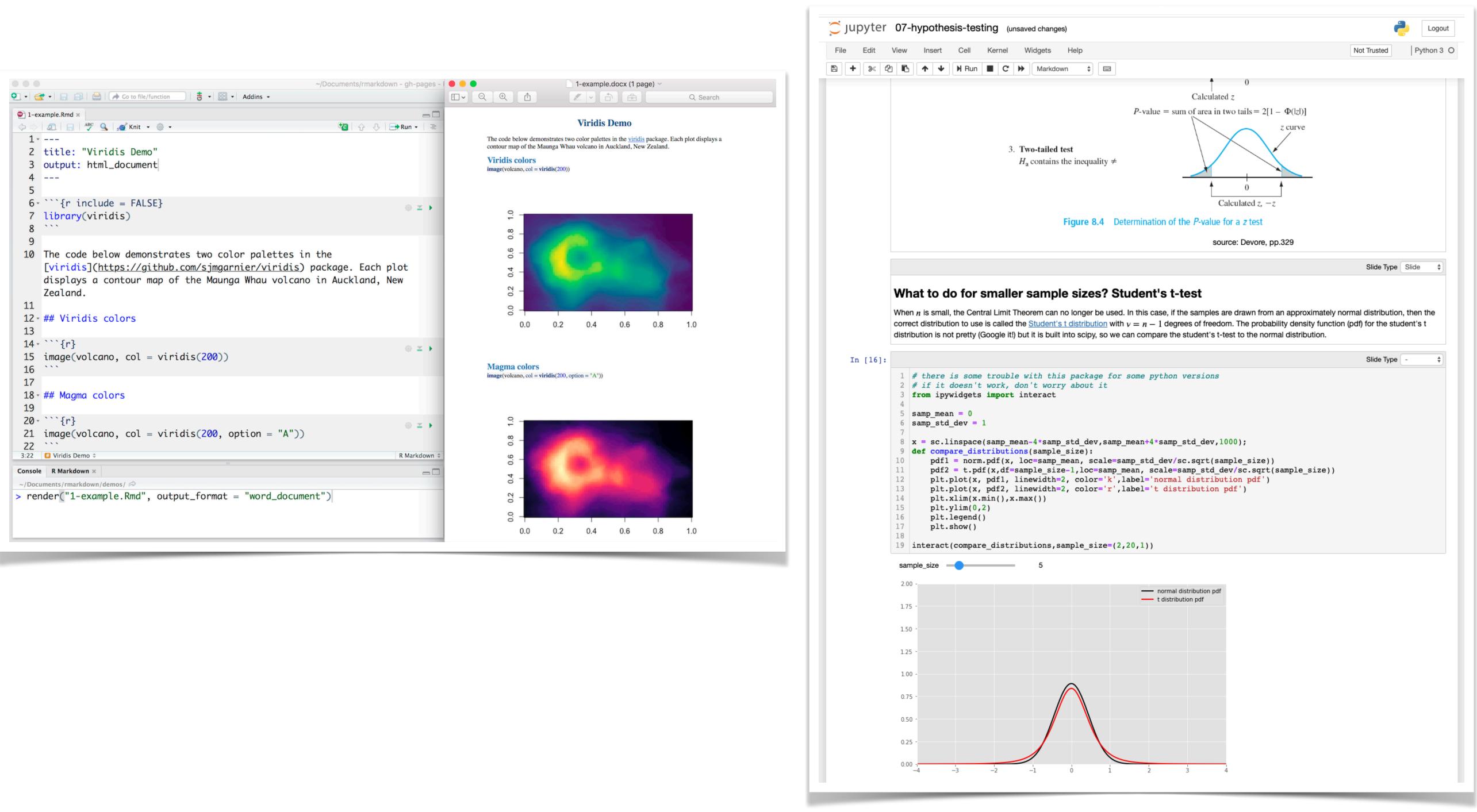
```
this cell
= pd.read_csv('./avalanches.csv')
ve NaN coordinates
= avy_df[avy_df['Coordinates']==avy_df['Coordinates']]
into latitude & longitude
['lat', 'lon']] = avy_df['Coordinates'].str.split(',', expand=True)
ve values outside of Utah bounds
= avy_df[ (36 < avy_df['lat'].astype('float')) & (avy_df['lat'].astype('float') < 42)]</pre>
= avy_df[(-114 < avy_df['lon'].astype('float')) & (avy_df['lon'].astype('float') < -108)]</pre>
columns we need
= avy_df[['Date', 'Region', 'Trigger', 'lat', 'lon']]
head ()
```



## **COMPUTATIONAL NOTEBOOKS: A MIDDLE GROUND?**



## Observable



## R Markdown

## Jupyter Notebooks

## **COMPUTATIONAL NOTEBOOKS: A MIDDLE GROUND?**

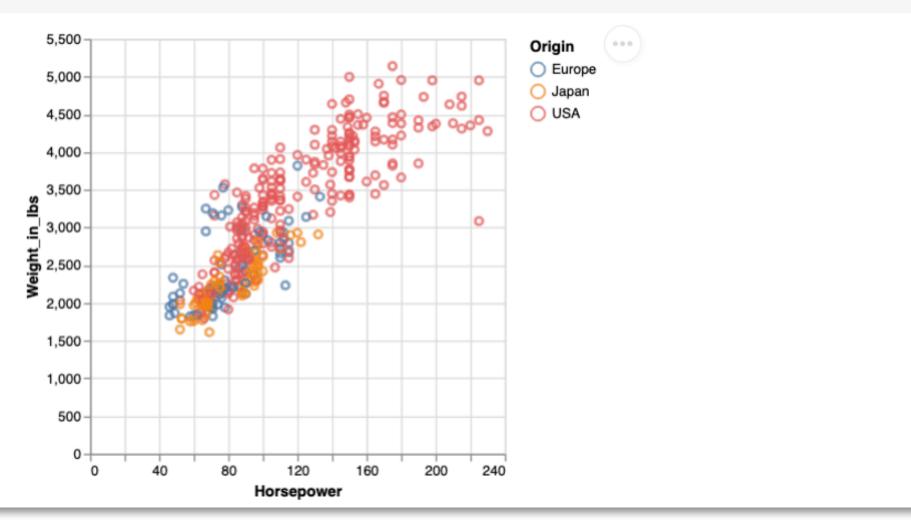
## Yes Afford both scripting and interactive visualization But visualizations are a dead end can't "use" interaction in code e.g., changing a label, or filtering a value

## Cars Dataset Analysis

Here we will see the relationship between weight of the car and horsepower

```
[6] selection = alt.selection_interval()
    chart = alt.Chart(cars_df).mark_point().encode(
          x="Horsepower:Q",
          y="Weight_in_lbs:Q",
           color=alt.condition(
              selection, "Origin:N", alt.value("gray")
         ).properties
           width=300
           height=300
          .add_params
             selection
```

chart

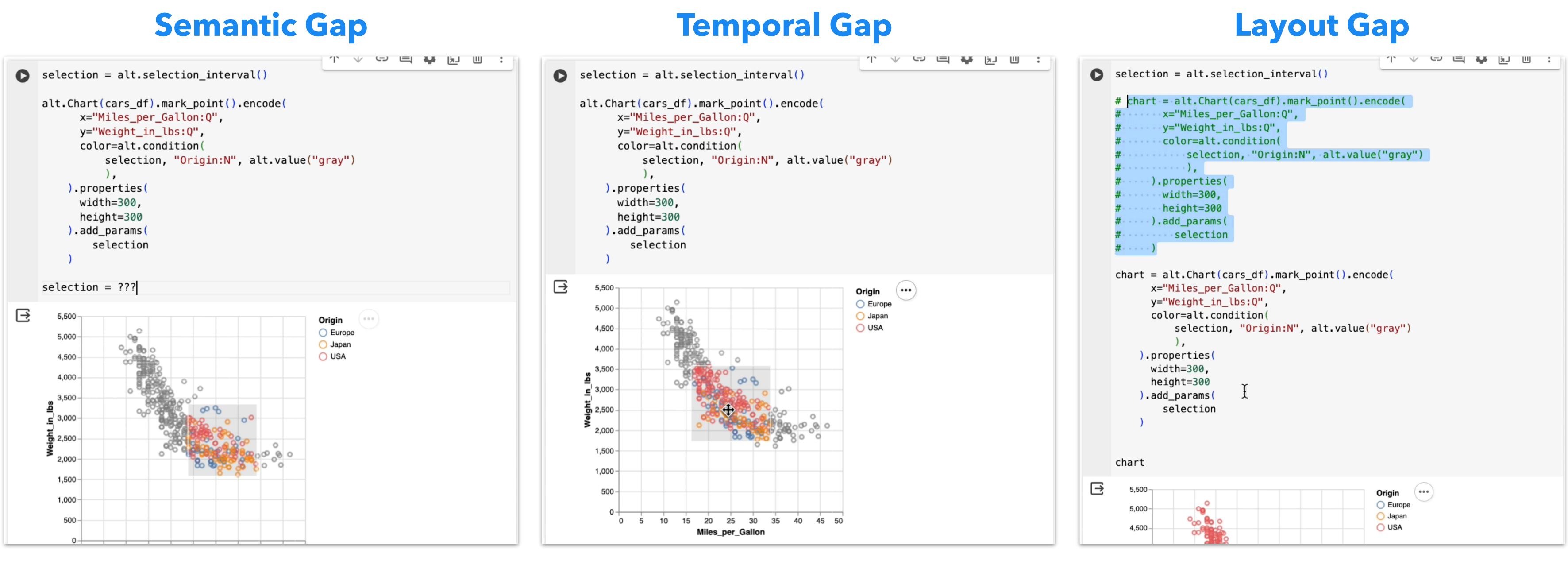


## Code

## Visualizations



## **GAPS BETWEEN CODE AND INTERACTIVE OUTPUTS\***



Information only flows from code to visualization

Changes made to code are preserved Changes made to vis are lost

Changes in code are messy

\*[Wu, Hellerstein, Satyanarayan, UIST 2020]



# THESIS: BRIDGING BETWEEN CODEAND INTERACTIVE VIS IS USEFULEasy handoffs are important!

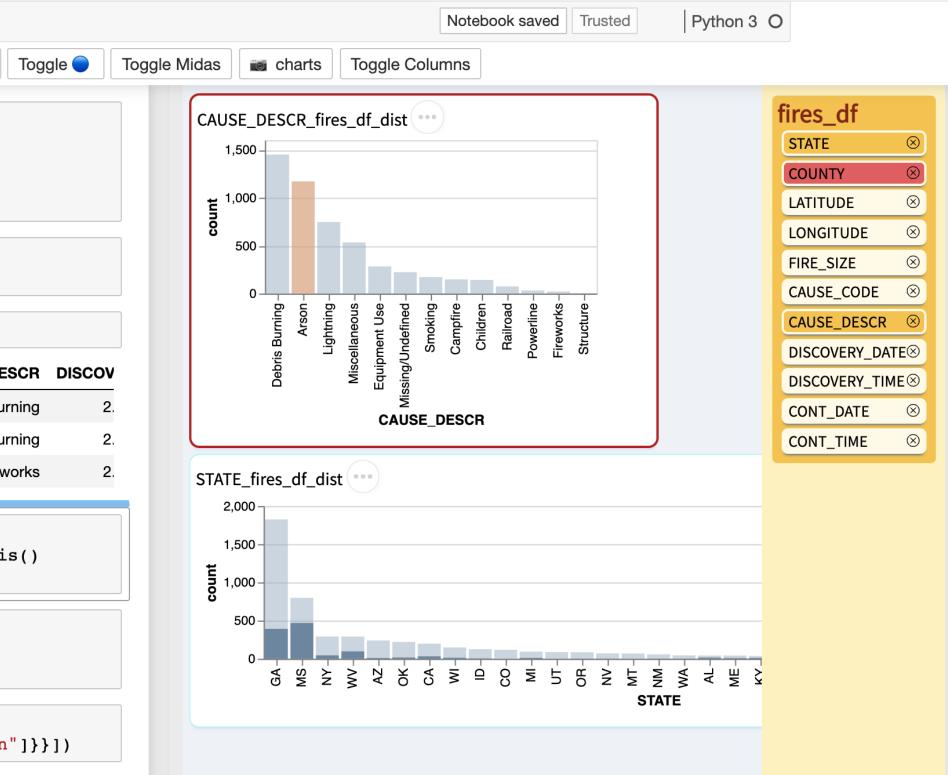
## RELATED WORK

## B2 – Wu, Hellerstein, Satyanarayan, UIST 2020

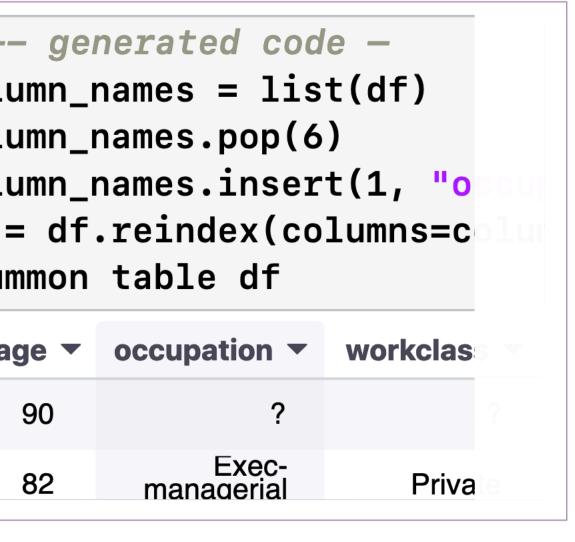
	File	Edit Vie	ew Inser	t Cell	Kernel	Widgets He	elp
	<b>B</b> +	× 4		▶ NRun	■ C >>	Code	
In [1]:	impo # In	midas <b>im</b> rt numpy <i>itiate Mi</i> Midas()	<b>as</b> np				
In [2]:			from_file	e("./data/1	fire_earl	ier.csv")	
In [6]:	fire	s_df.head	(3)				
Out[6]:	STATE	COUNTY	LATITUDE	LONGITUDE	FIRE_SIZE	CAUSE_CODE	CAUSE_DES
	GA	Baldwin	33.0514	-83.0735	6.65	5	5 Debris Burn
	WV	Monongalia	39.5665	-80.0572	5	5	5 Debris Burn
	WV	Logan	37.9409	-82.023	61	10	) Firewo
In [3]:	• # <del> </del> CAUS	<i>06:40 PM</i> E_DESCR_f		list = fire	es_df.gro	up('CAUSE_I	DESCR').vis
In [4]:		<i>06:40 PM</i> E_fires_d		fires_df.	group(' <mark>S</mark> T	ATE').vis()	
In [5]:		06:40 PM l([{"CAUS		fires_df_di	ist": {"C	AUSE_DESCR"	: ["Arson"
	·!						

## Mage – Kery et al., UIST 2020

1	mage	e : user edits	table		2
<mark>%</mark> S	ummon	table df			# -
	age 🔻	workclass 🔻	fnlwgt	e	co co
0	90	? occupation 👻	77053		co:
1	82	<b>∫</b> ∭ate	132870		df %sı
2	66	?	186061		
3	54	Private	140359		0
4	41	Private	264663		1



## mage : edits reflect in code





# THE PERSIT APPROACH

## PRINCIPLE

## Track events in interactive visualizations

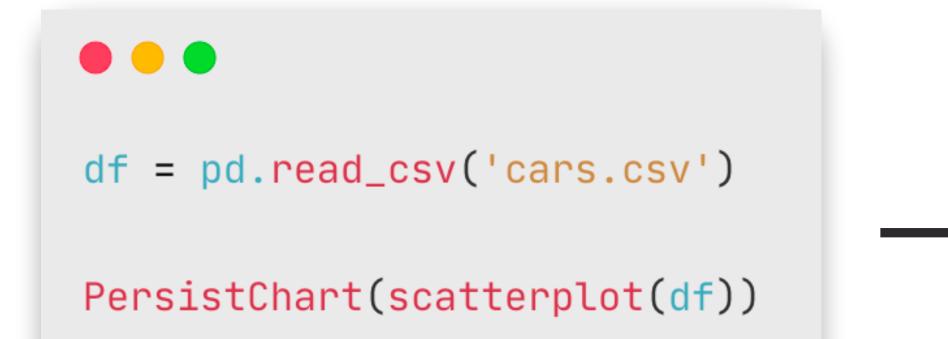
## Map them to data frame operations

## **Operations then applied to data** frame

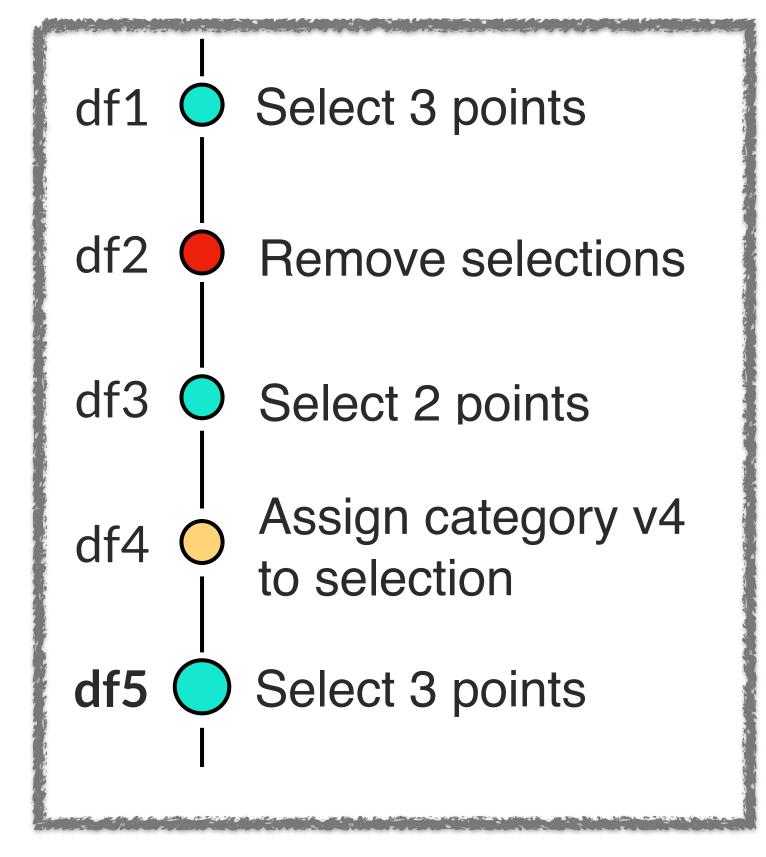


## HOW IT WORKS

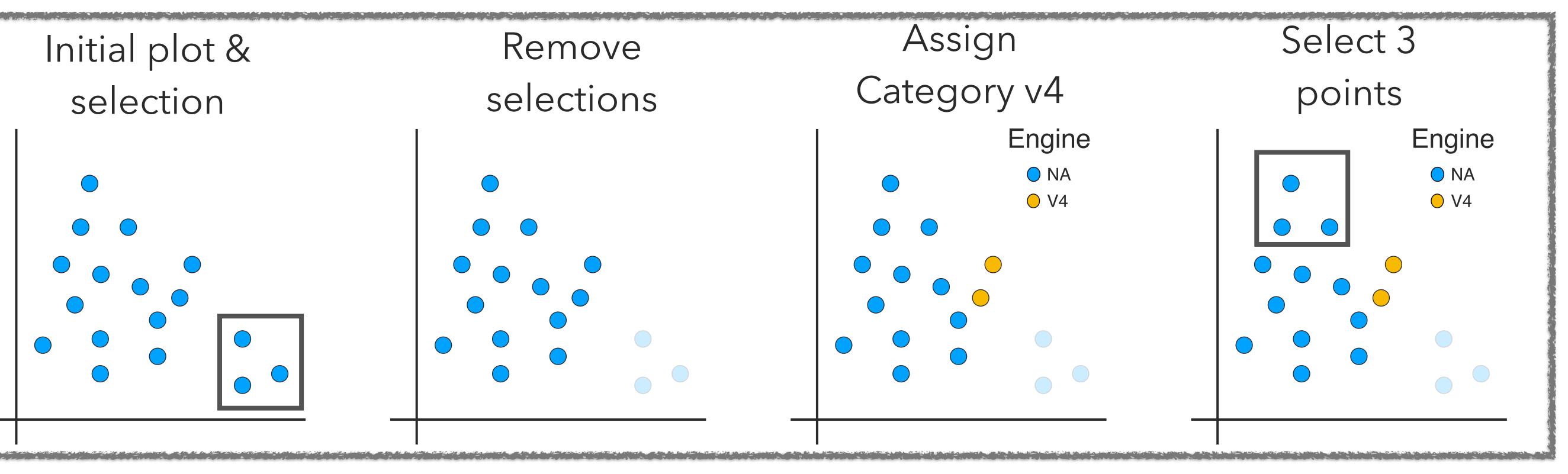
## Code to create chart

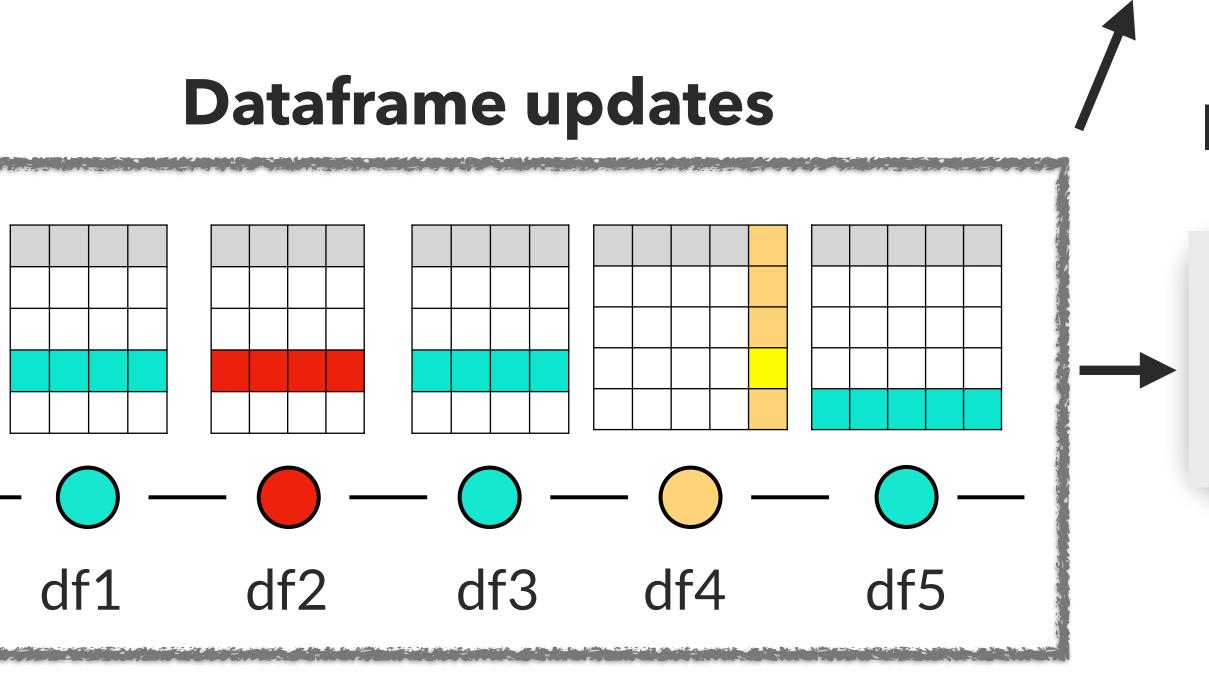


## Provenance

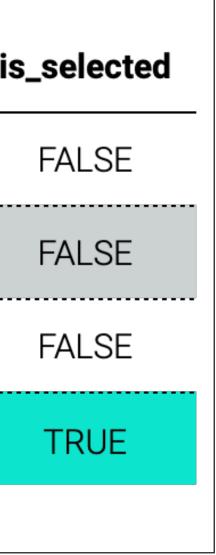


## **Interactive Visualization**





In code	Name	MPG	Cylinders	origin	Engine	is			
	ford	18	8	USA	NA				
df5.head()	dodge	15	8	USA	NA				
	volkswagen	22	4	Europe	V4				
	amc	16	8	USA	NA				
	27 rows x 8 columns								

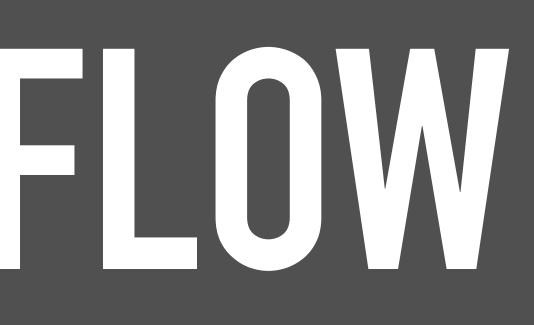


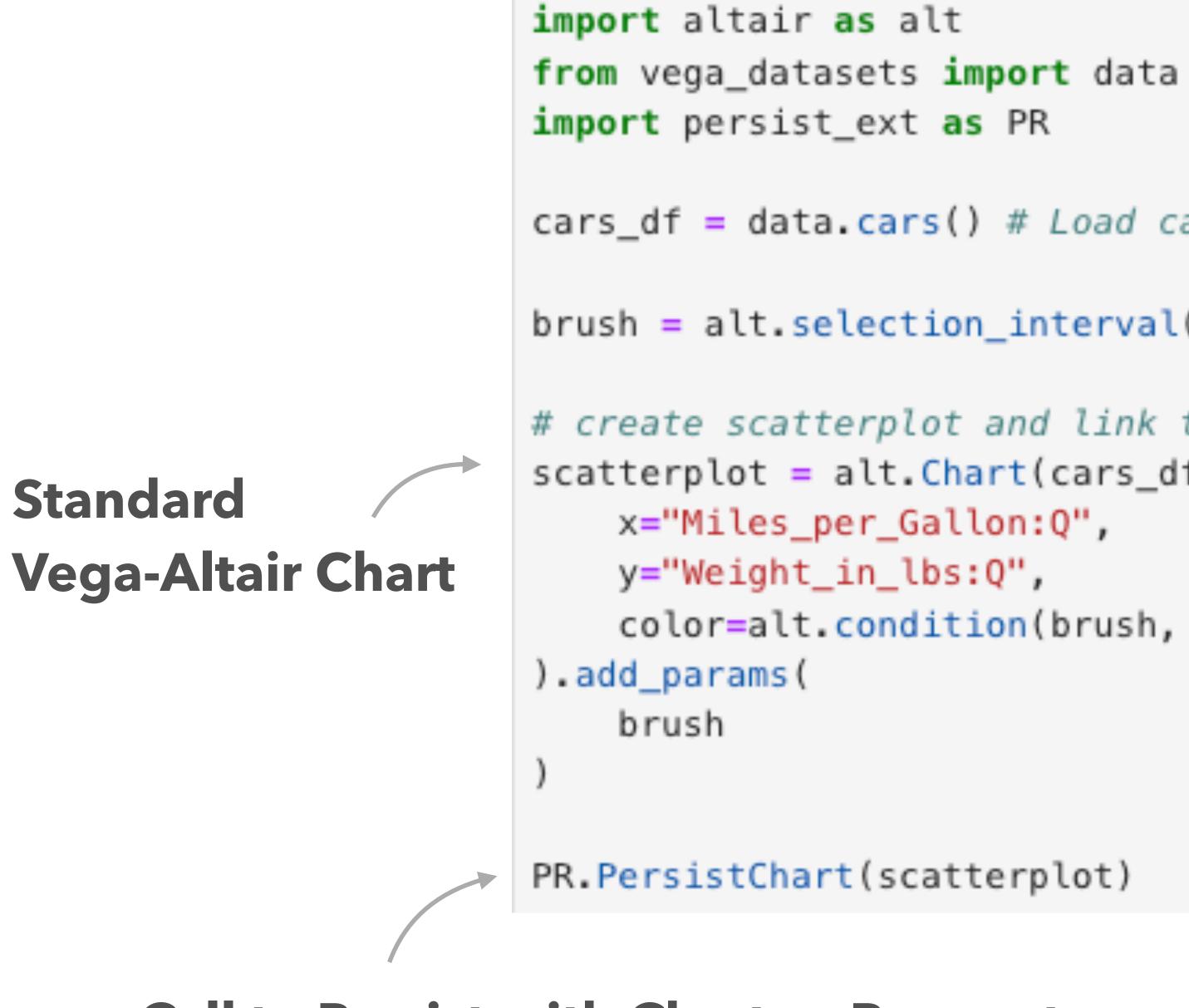
## OPERATIONS

## Selection **Edit column names, edit cells EV** Sort rows/columns **Drop columns Filter (in/out) items E** Label items **Categorize items Change data types**



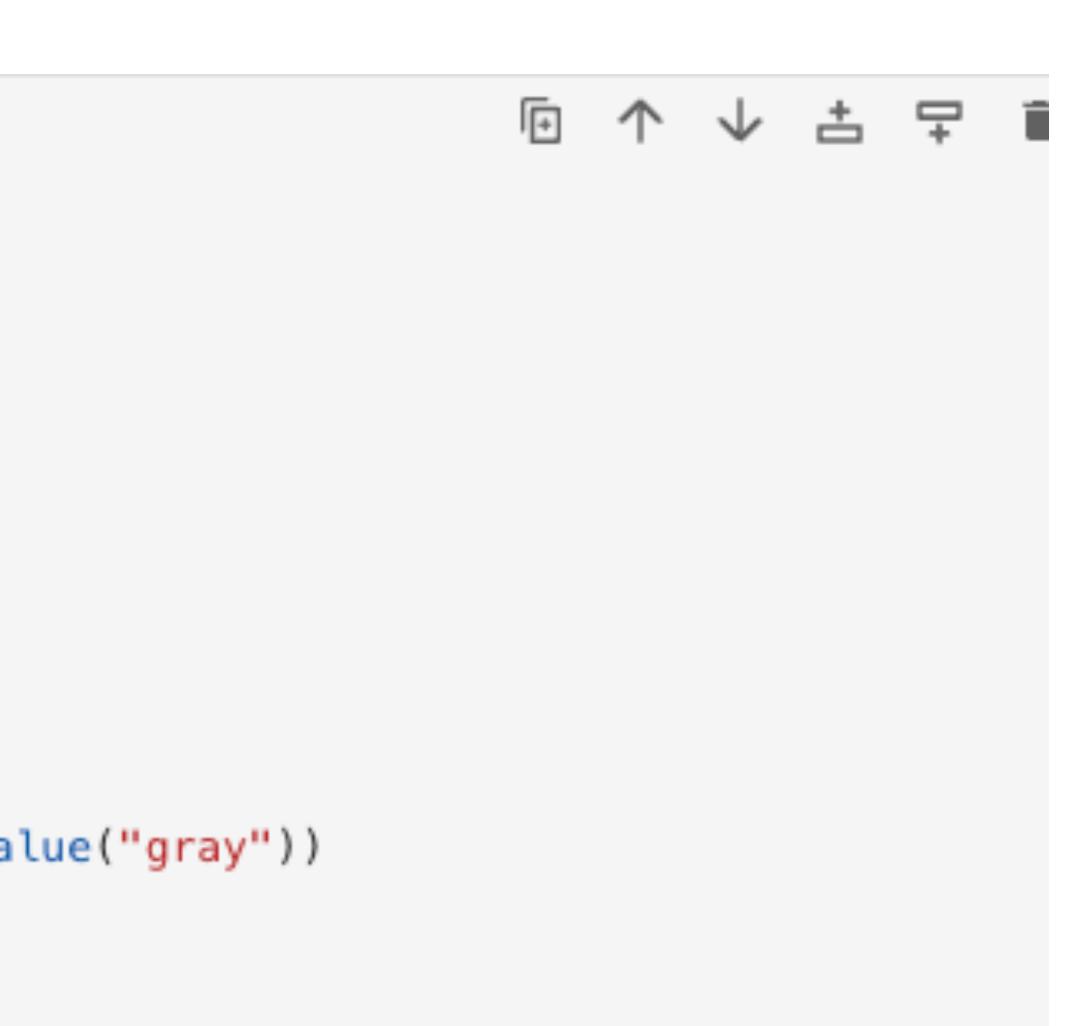
# PERSIST WORKFLOW





```
cars_df = data.cars() # Load cars dataset
brush = alt.selection_interval() # Create a 2d brush
# create scatterplot and link to brush
scatterplot = alt.Chart(cars_df).mark_point().encode(
   x="Miles_per_Gallon:Q",
    color=alt.condition(brush, alt.value("steelblue"), alt.value("gray"))
```

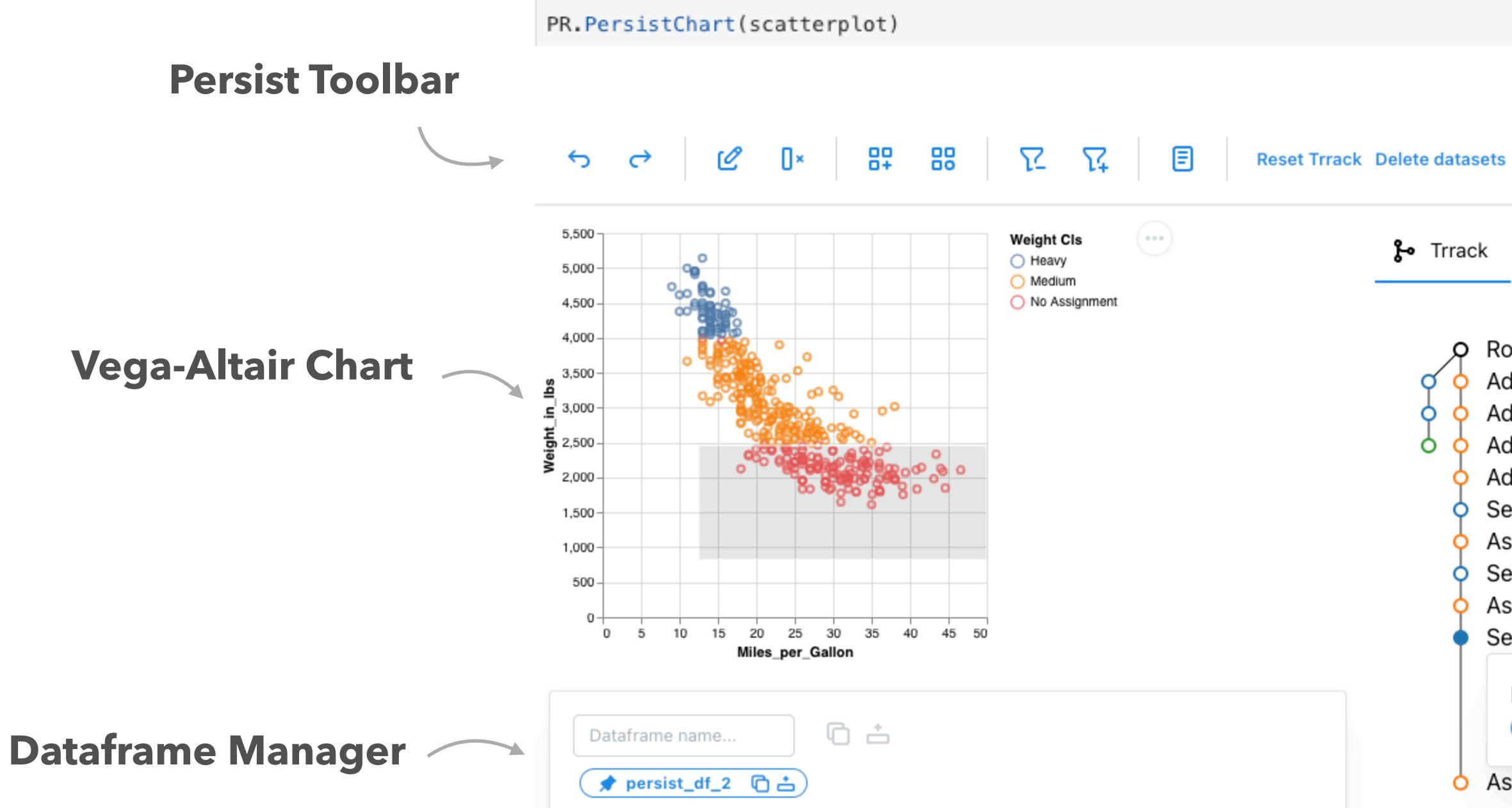
## **Call to Persist with Chart as Parameter**



```
import altair as alt
         from vega_datasets import data
         import persist_ext as PR
         cars_df = data.cars() # Load cars dataset
         brush = alt.selection_interval() # Create a 2d brush
         # create scatterplot and link to brush
         scatterplot = alt.Chart(cars_df).mark_point().encode(
             x="Miles_per_Gallon:Q",
             y="Weight_in_lbs:Q",
             color=alt.condition(brush, alt.value("steelblue"), alt.value("gray"))
         ).add_params(
             brush
         PR.PersistChart(scatterplot)
5,500 -
                                             Weight Cls
                                                        ....

    Heavy

                     0
          5,000
                    90
                                               ) Medium
                   ° 00 00 0
                                              No Assignment
          4,500-
          4,000-
                                                                                Q
                             0
          3,500-
         .⊆ 3,000-
         ີ່ຫຼັ 2,500 -
                                                                              0
          2,000-
```



## 

## ⊱ Trrack 🛛 🗄 Summary

## Provenance History

## Root

Add new category 'We...

- Add new option 'Heav...
- Add new option 'Medi...
- Add new option 'Light'...
- Selected Miles\_per\_G...
- Assign Weight Cls (He...
- Selected Miles\_per\_G...
- Assign Weight Cls (Me...
- Selected Miles\_per\_G...

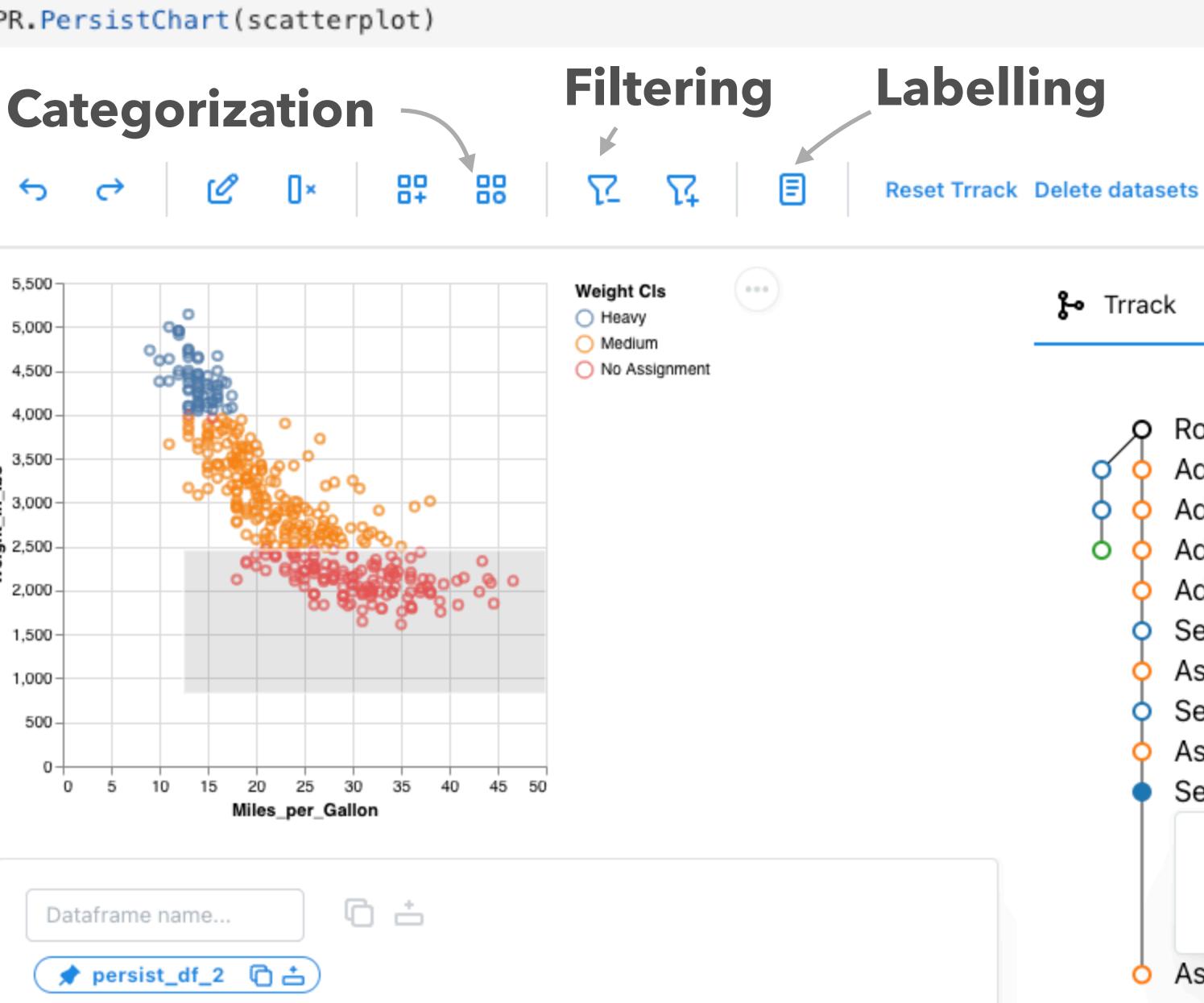
## Dataframes:

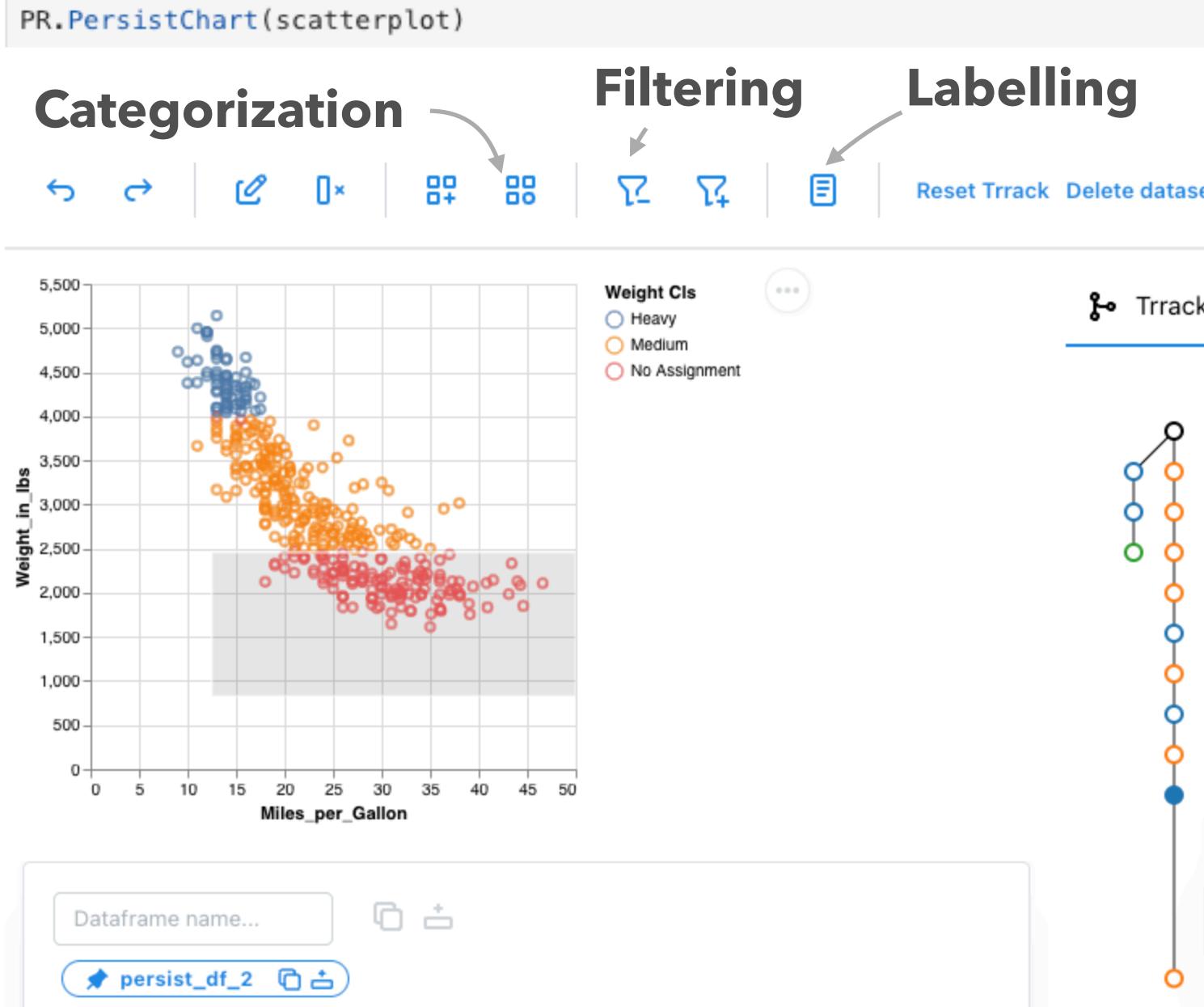
🖈 persist\_df\_2 🛛 📥

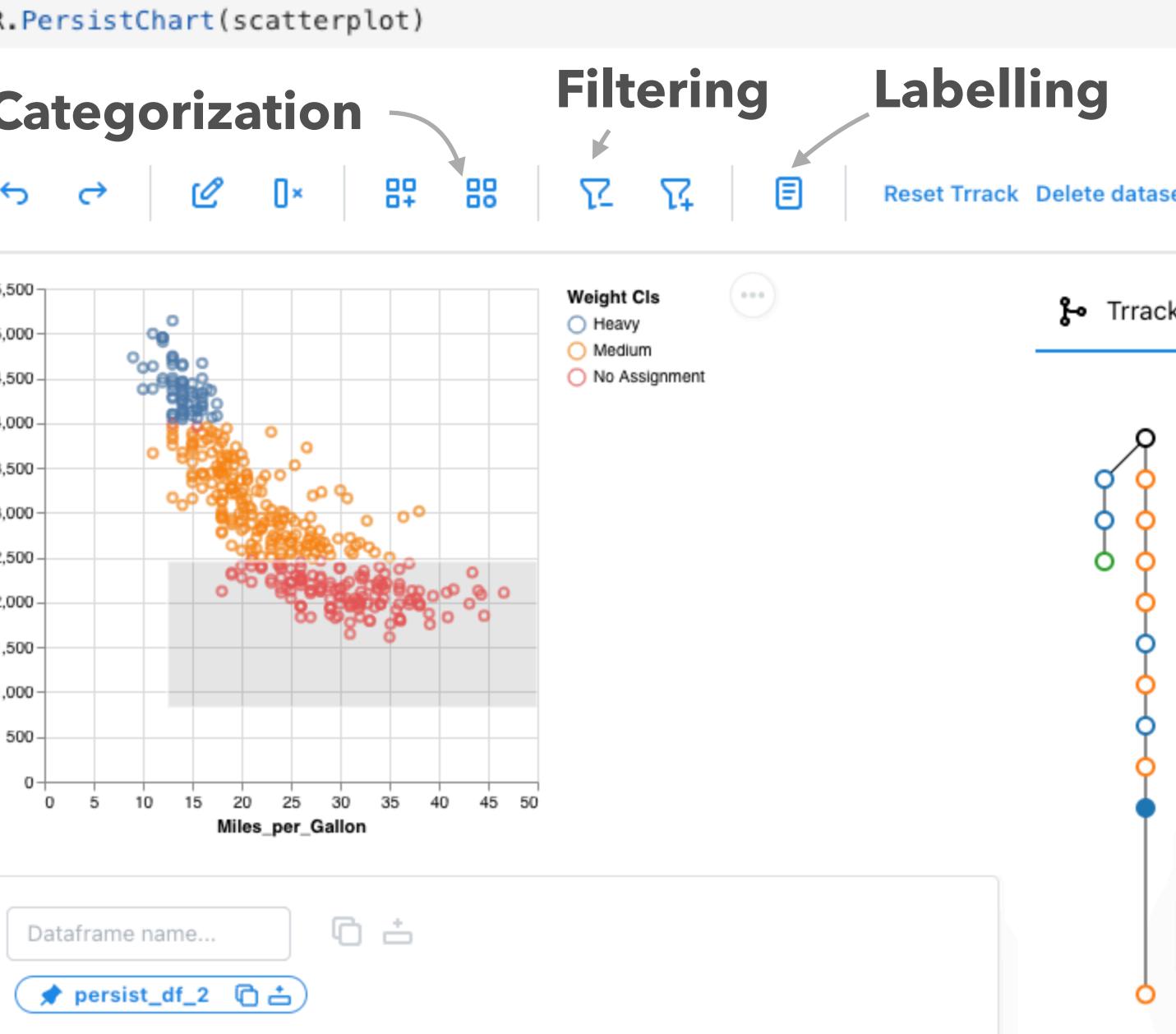
Assign Weight Cls (Lig...



```
import altair as alt
from vega_datasets import data
import persist_ext as PR
cars_df = data.cars() # Load cars dataset
brush = alt.selection_interval() # Create a 2d brush
# create scatterplot and link to brush
scatterplot = alt.Chart(cars_df).mark_point().encode(
   x="Miles_per_Gallon:Q",
   y="Weight_in_lbs:Q",
    color=alt.condition(brush, alt.value("steelblue"), alt.value("gray"))
).add_params(
    brush
```







## 

## ⊱ Trrack 🛛 🗄 Summary

## Root

Add new category 'We...

- Add new option 'Heav...
- Add new option 'Medi...
- Add new option 'Light'...
- Selected Miles\_per\_G...
- Assign Weight Cls (He...
- Selected Miles\_per\_G...
- Assign Weight Cls (Me...
- Selected Miles\_per\_G...

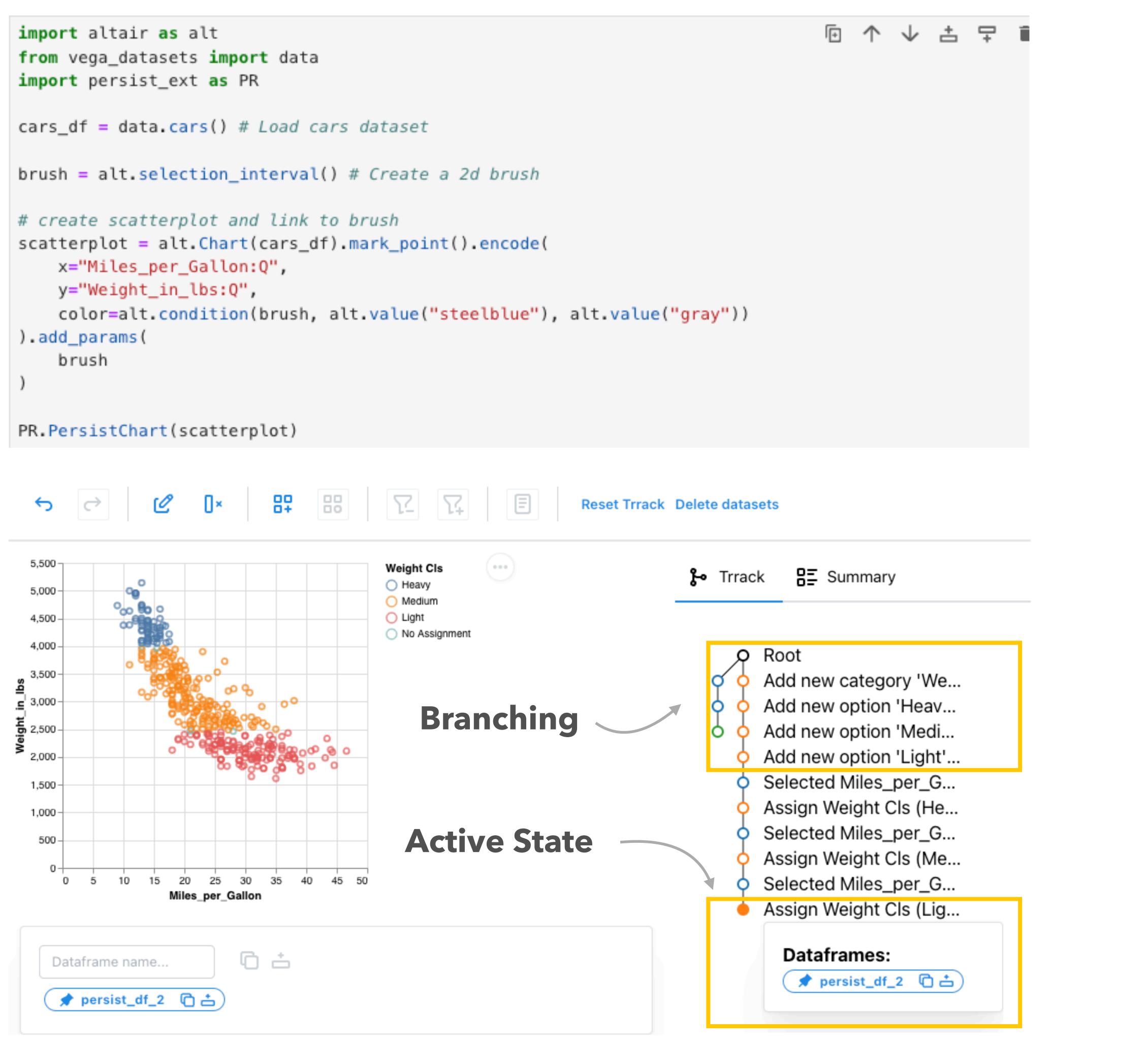
## Dataframes:



Assign Weight Cls (Lig...

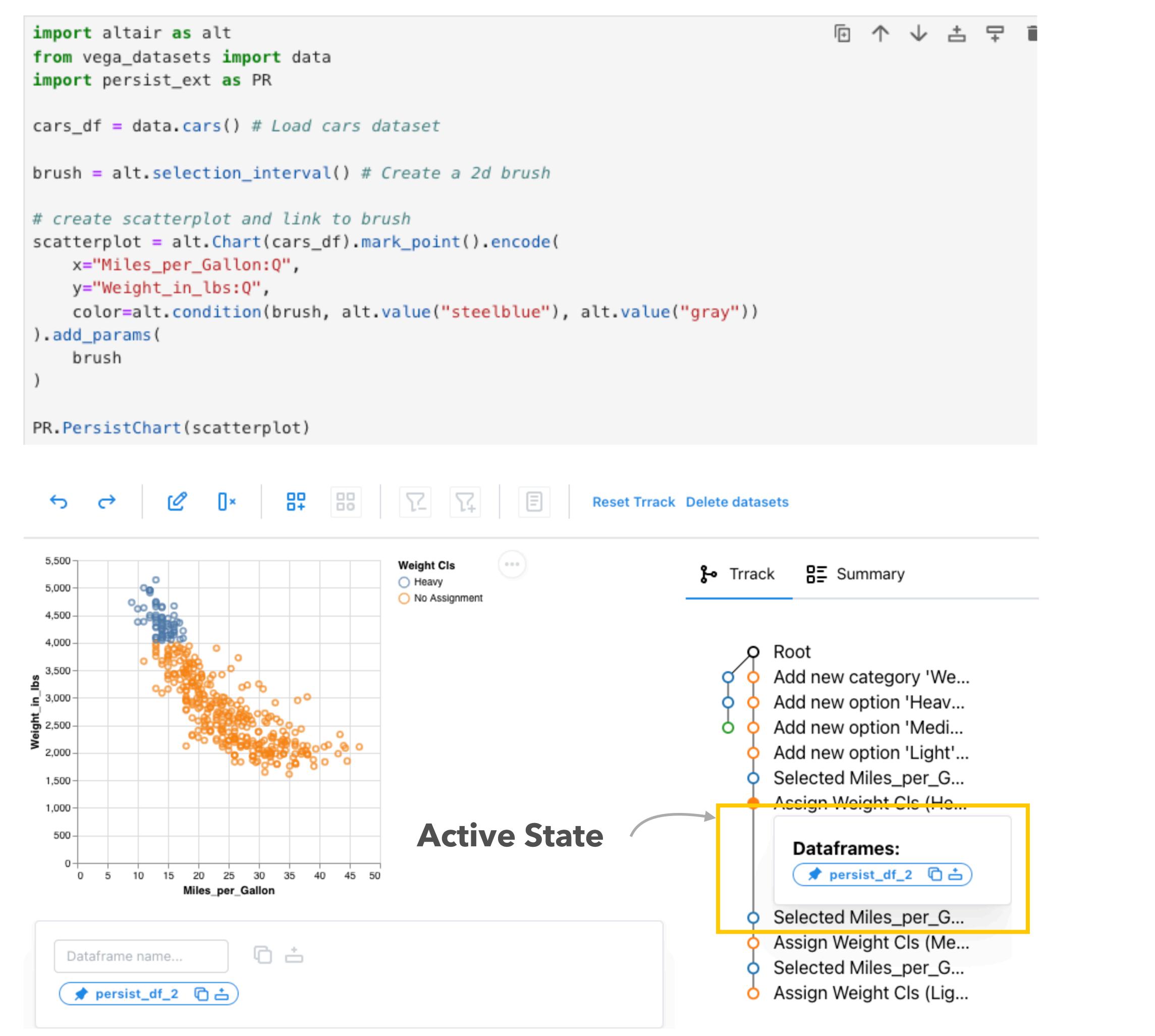
## Branches and Choosing a State in provenance support non-linear analysis, addressing the layout gap

```
import altair as alt
from vega_datasets import data
import persist_ext as PR
cars_df = data.cars() # Load cars dataset
brush = alt.selection_interval() # Create a 2d brush
# create scatterplot and link to brush
scatterplot = alt.Chart(cars_df).mark_point().encode(
   x="Miles_per_Gallon:Q",
   y="Weight_in_lbs:Q",
    color=alt.condition(brush, alt.value("steelblue"), alt.value("gray"))
).add_params(
    brush
```

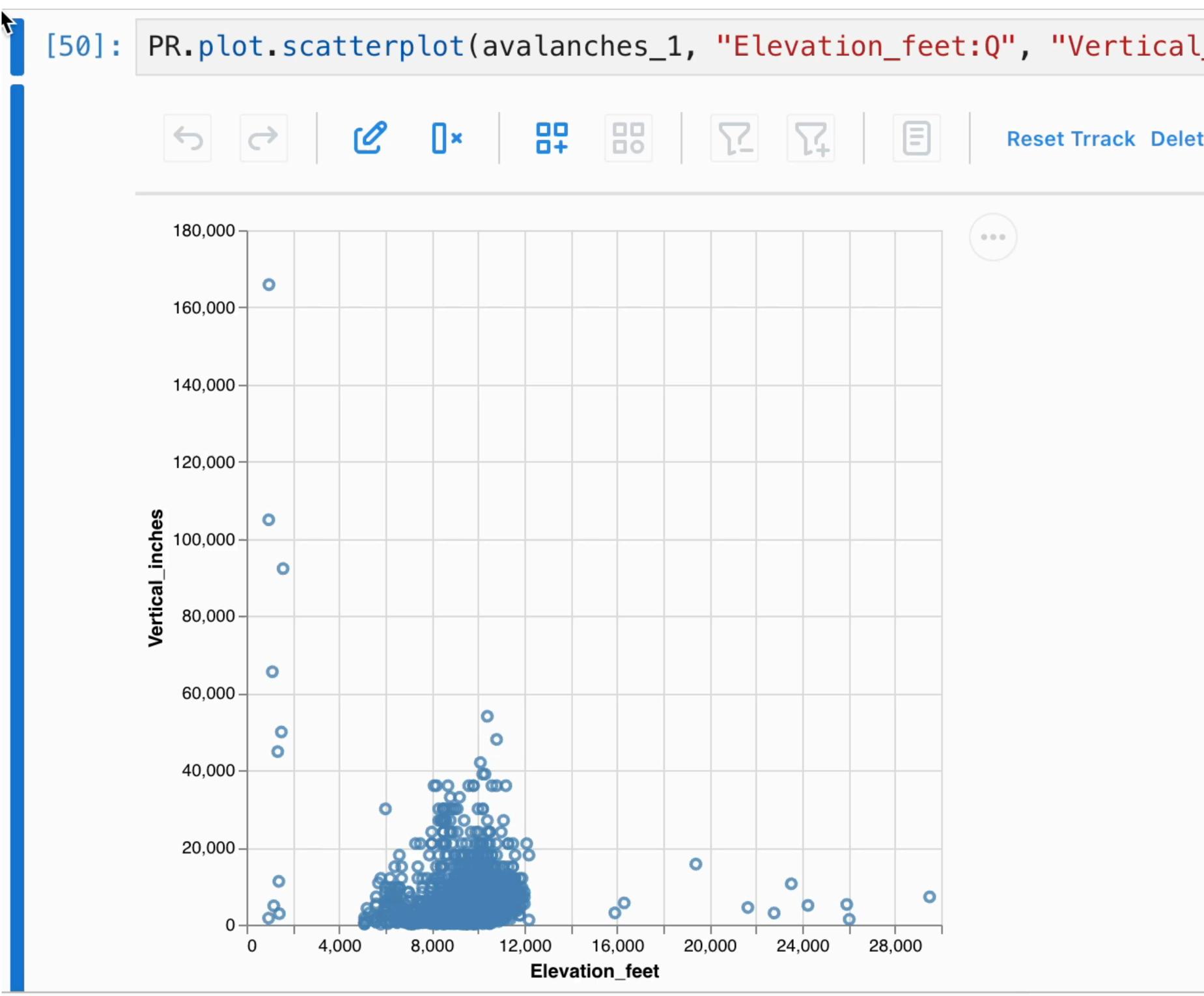


## Branches and Choosing a State in provenance support non-linear analysis, addressing the layout gap

```
import altair as alt
                                                                               ⊡ ↑ ↓
                                                                                          ±
                                                                                             7
from vega_datasets import data
import persist_ext as PR
cars_df = data.cars() # Load cars dataset
brush = alt.selection_interval() # Create a 2d brush
# create scatterplot and link to brush
scatterplot = alt.Chart(cars_df).mark_point().encode(
   x="Miles_per_Gallon:Q",
   y="Weight_in_lbs:Q",
   color=alt.condition(brush, alt.value("steelblue"), alt.value("gray"))
).add_params(
   brush
```

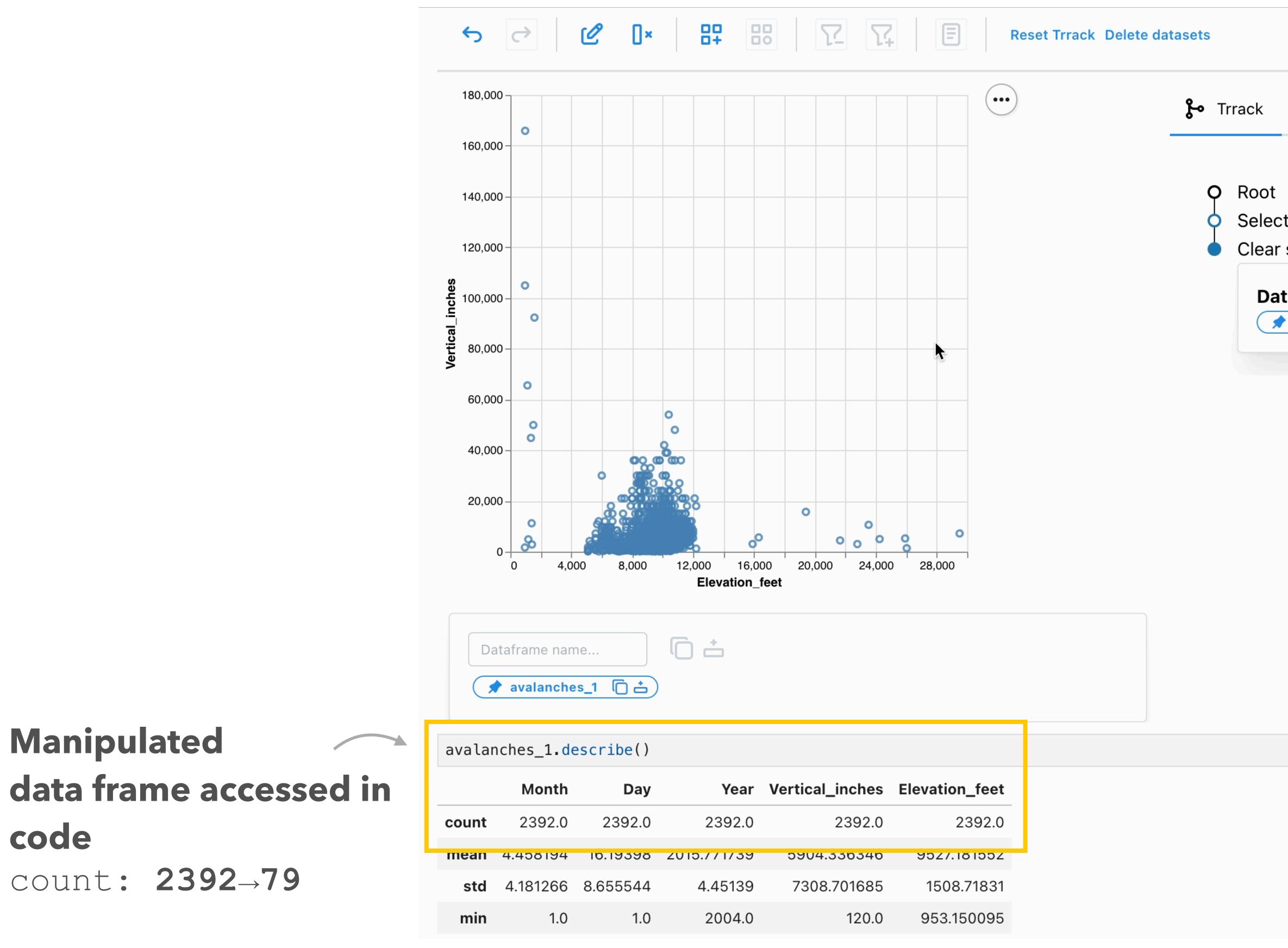


# Persist **re-runs the interactions** in the output, addressing **the temporal gap**



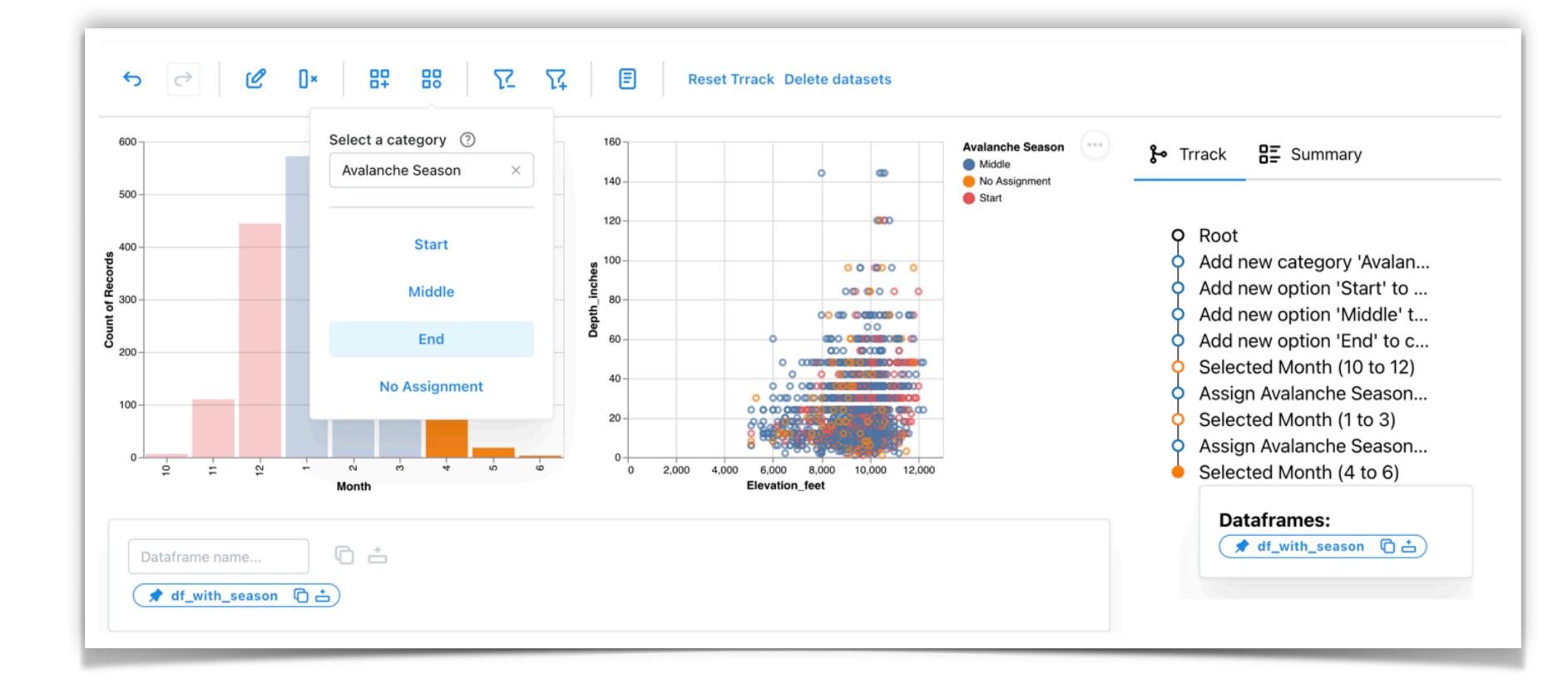
incl	hes	:Q",	df_na	ame="	aval	anche	es_2'	")
e datas	sets							
ዮ	Trra	ck	83 Sur	mmary	,			
	D R	oot						
			o from o	••				
			aframe avalanch			)		

# Persist *applies interactions to data frames* that can be accessed in code, addressing the **semantic gap**



🗄 Summary
cted Elevation_feet (10252 t selections
taframes:
🕈 avalanches_1  🗋

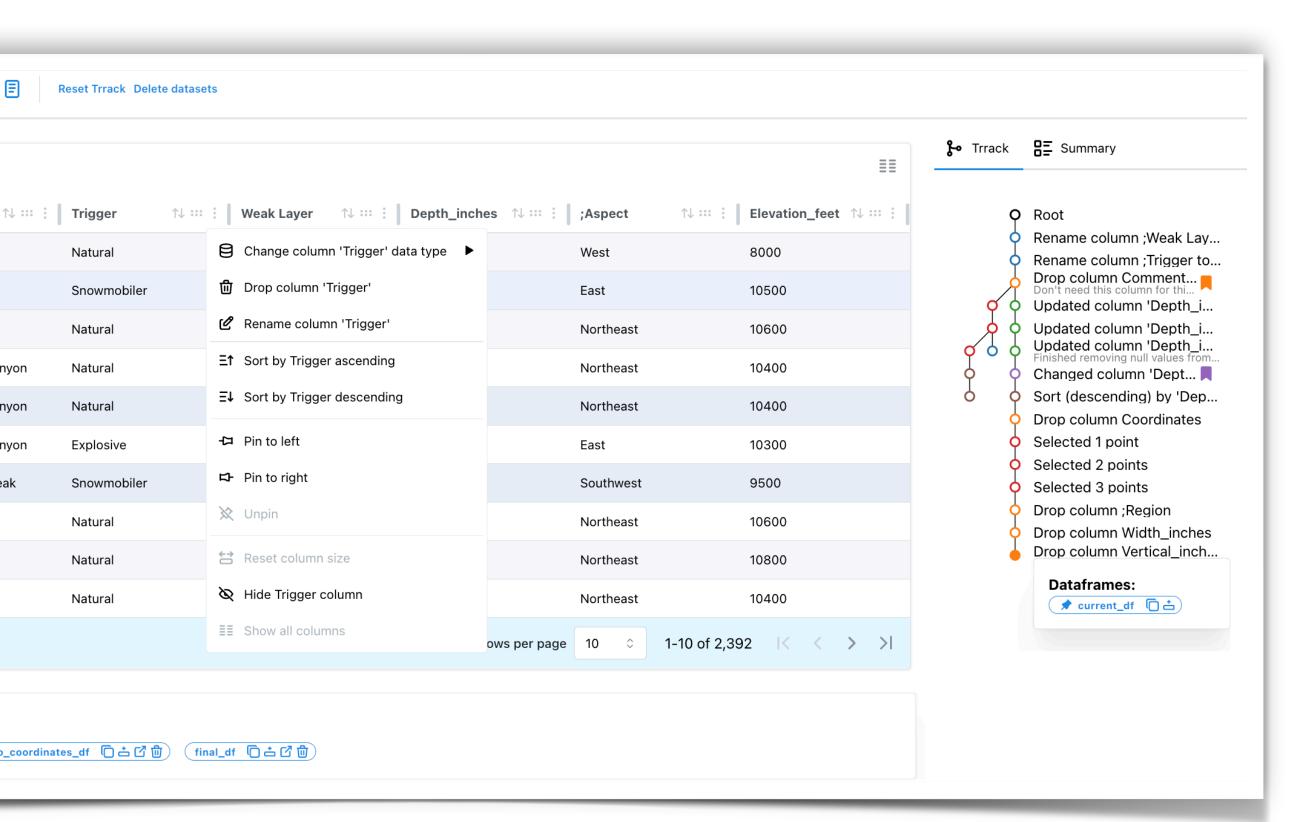
## **VISUALIZATION OPTIONS**



## **Arbitrary Vega-Altair Charts**

Q	Search		
	#	Date ↑↓ ፡፡፡ :	Place 1
	2338	3-23-2023	Dry Creek
	955	1-19-2014	Whitney Basin
	1028	2-21-2014	Chalk Creek
	1024	2-17-2014	Upper Weber Cany
	998	2-12-2014	Upper Weber Cany
	938	1-14-2014	Upper Weber Cany
	1299	1-26-2016	Currant Creek Pea
	1044	2-28-2014	Chalk Creek
	2348	3-30-2023	Bunnels
	1977	4-6-2021	Blue Ice
3 (	of 2392 rov	v(s) selected	
	aframe name current_df		「☐ ் Ґ 前) (no_





## **An Interactive Data Table**

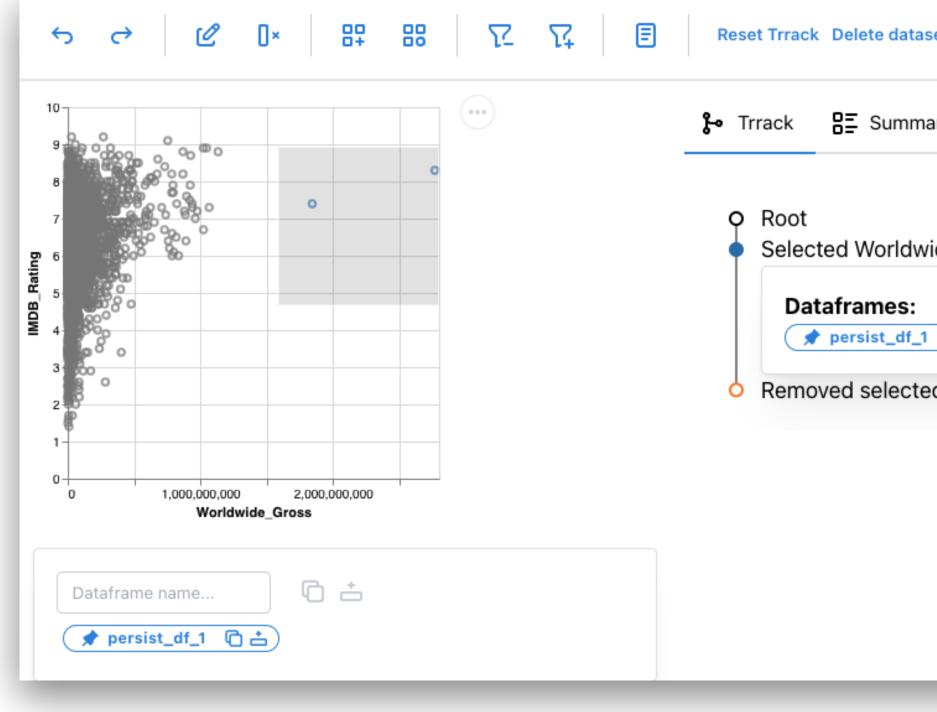
## VEGA-ALTAIR

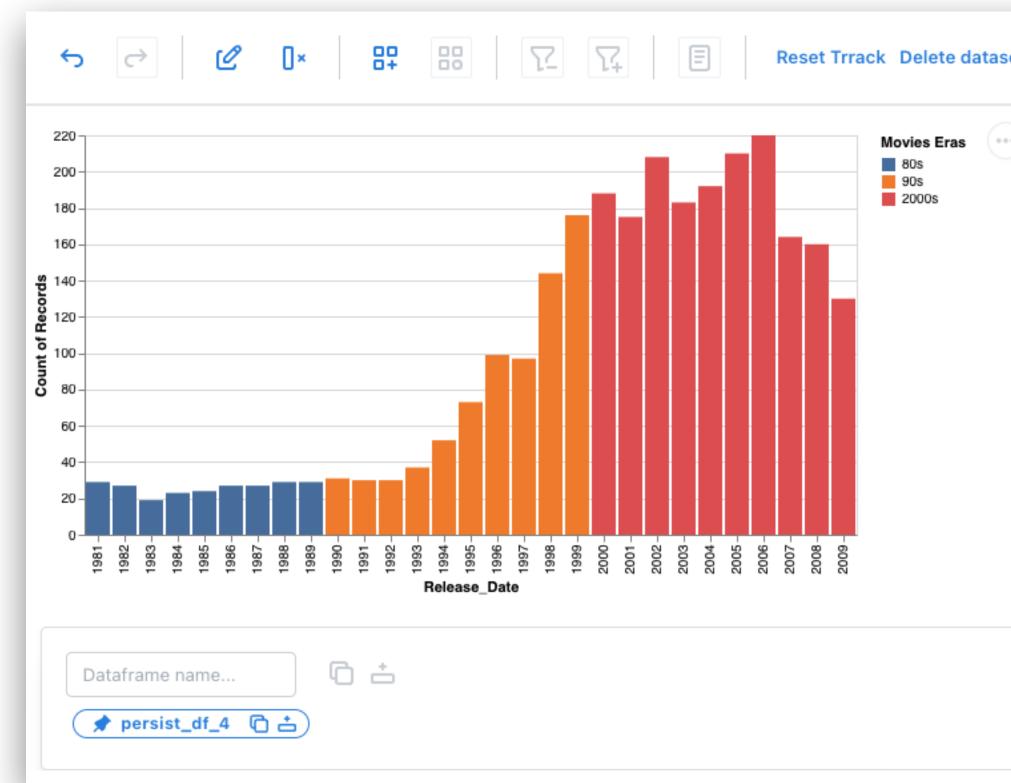
Persist works with most Vega-Altair charts Updates Vega charts:

# Use original chart spec when possible (e.g., filters) Update spec when necessary (categories, labels)

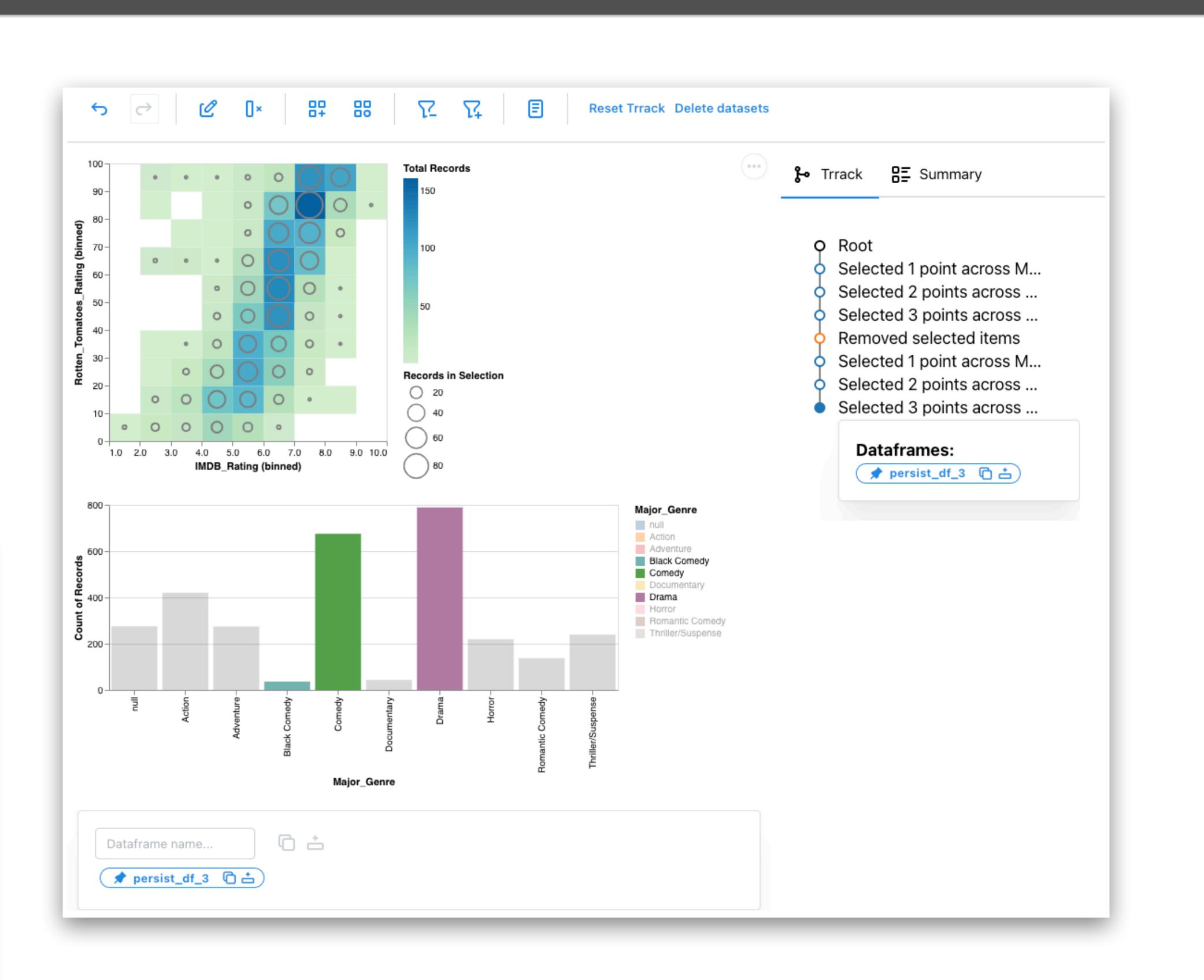
"Listens" to native operations (selections)

## EXAMPLE CHARTS





e datasets		
ummary		
orldwide_Gros		
les: t_df_1  ☐ ≟)		
elected items		
lected items		
datasets		
s	⊱ Trrack 📲 Summary	
	O Root	
	<ul> <li>Add new category 'Movie</li> <li>Add new option '80s' to c</li> </ul>	
	<ul> <li>Add new option '80s' to c</li> <li>Add new option '90s' to c</li> </ul>	
	Add new option '2000s' t	
	Selected Release_Date (1	
	<ul> <li>Assign Movies Eras (80s)</li> <li>Selected Release_Date (1</li> </ul>	
	Assign Movies Eras (90s)	
	Selected Release_Date (2	
	<ul> <li>Assign Movies Eras (200</li> </ul>	
	Dataframes:	
	📌 persist_df_4  🔂 📩	



# 

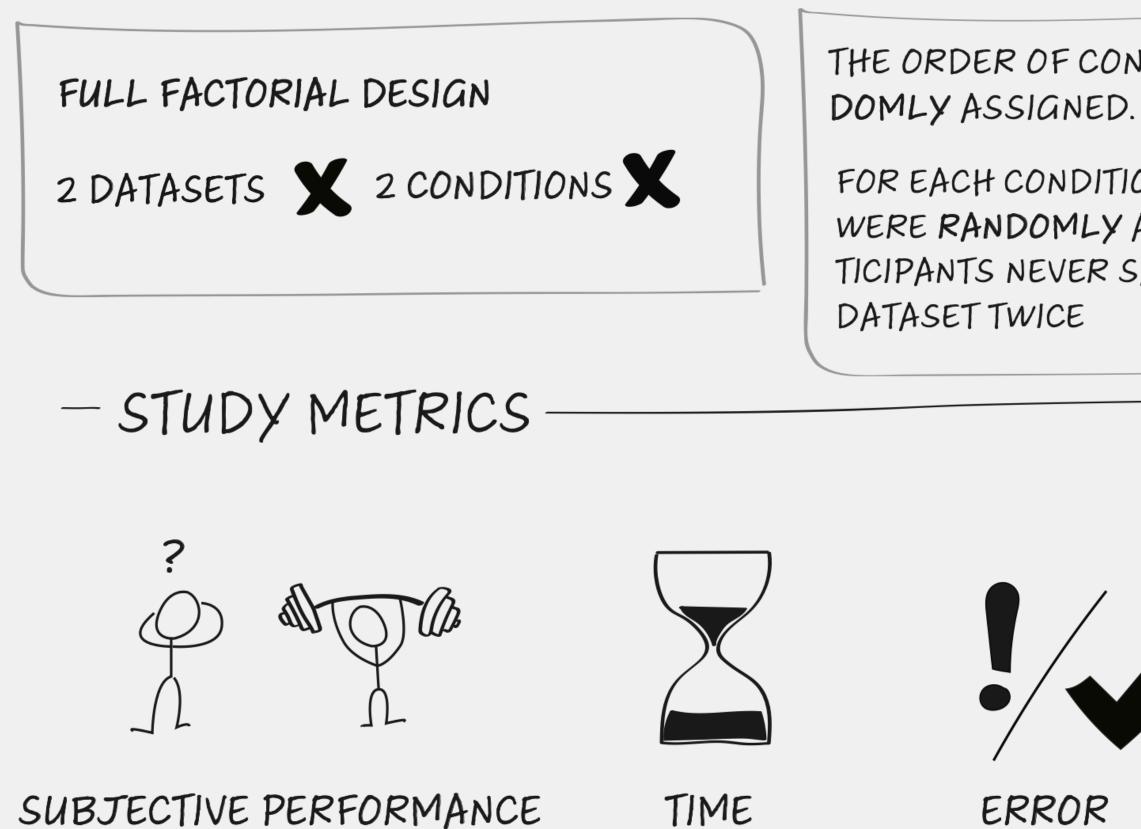
# EVALUATION



## IN-LAB STUDY

### III STUDY DESIGN

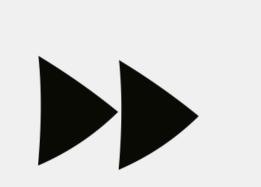
WE RECRUITED ELEVEN PARTICIPANTS FOR THE STUDY. PARTICIPANTS ALL HAD PRIOR EXPERIENCE WITH PYTHON AND PANDAS.



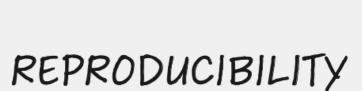
SUBJECTIVE PERFORMANCE

THE ORDER OF CONDITIONS WAS RAN-

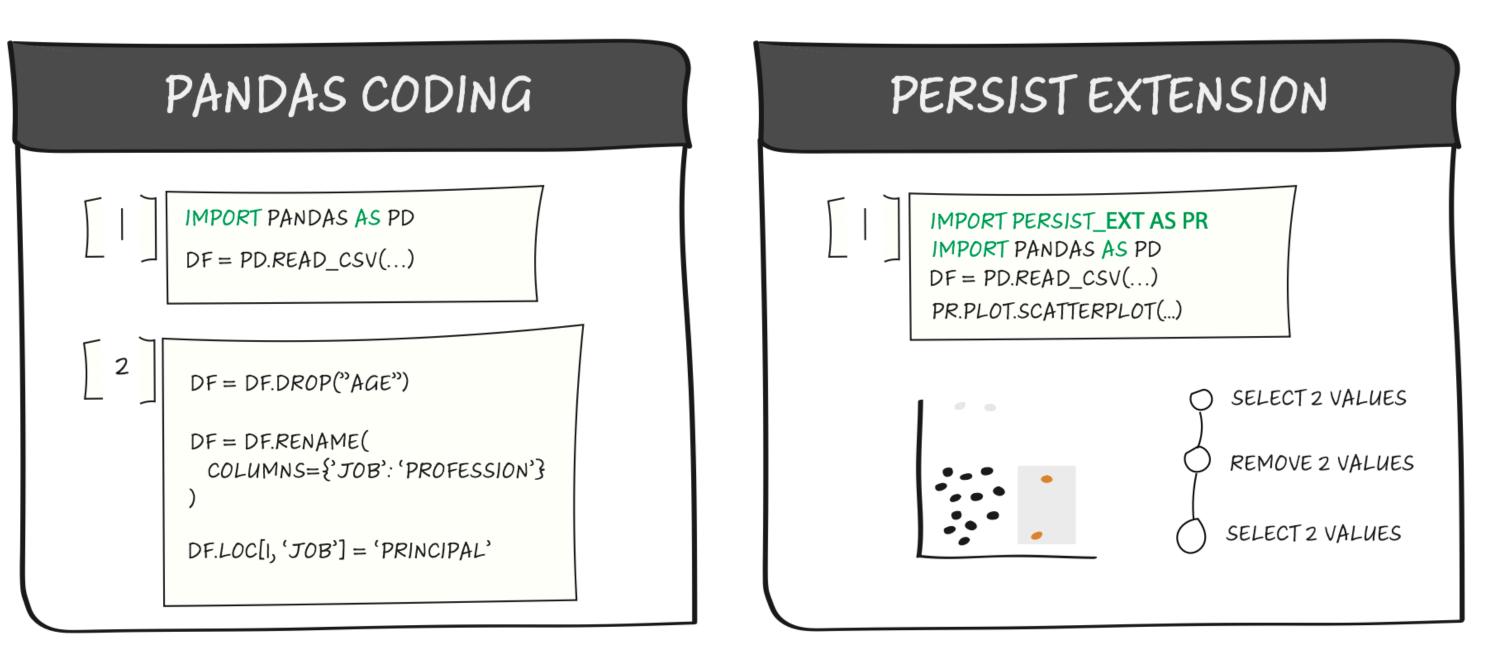
FOR EACH CONDITION, DATASETS WERE RANDOMLY ASSIGNED. PAR-TICIPANTS NEVER SAW THE SAME

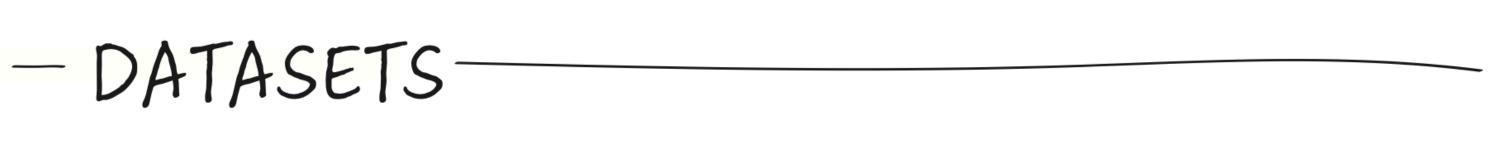


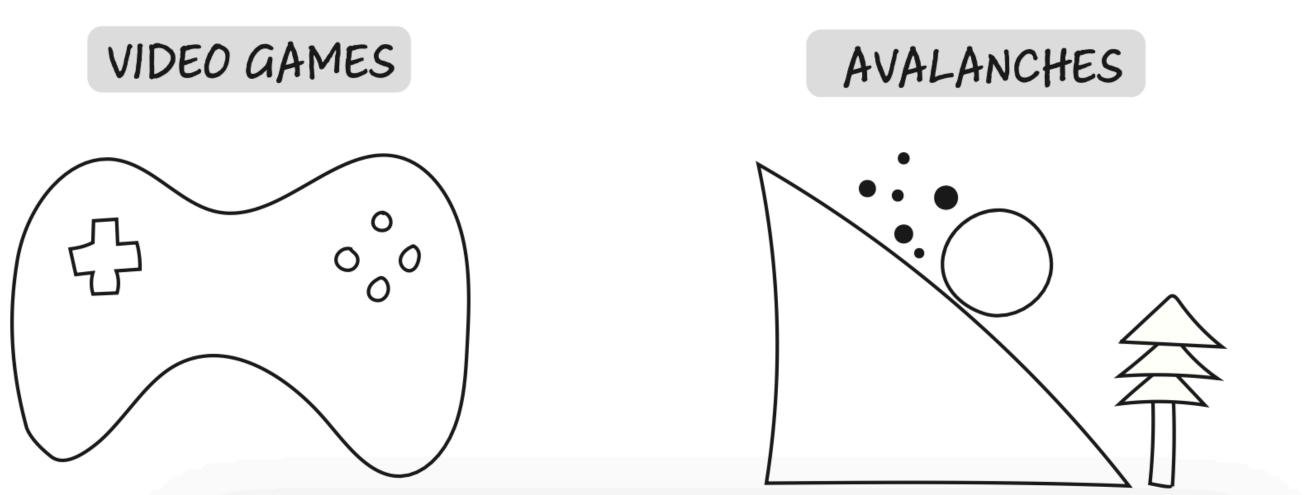
ERROR



### - CONDITIONS



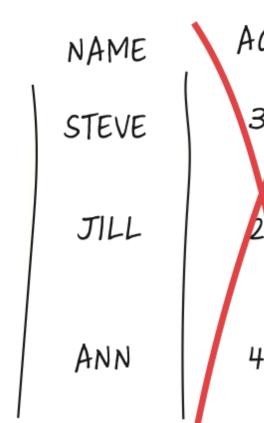






## IN-LAB STUDY

### - TASKS PARTICIPANTS MADE THE FOLLOWING CHANGES TO A DATASET

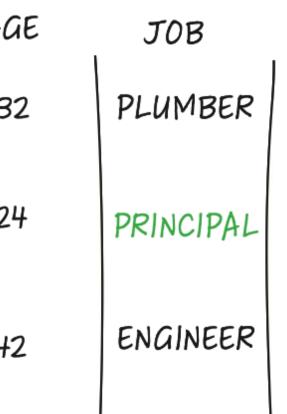


NAME	AC
STEVE	3:
JILL	24
ANN	4:

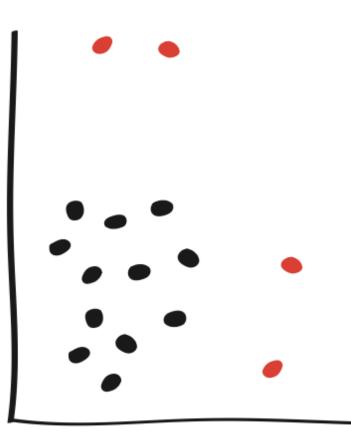
AGE	JOB	NAME	AGE	PROFESSION	
32	PLUMBER	STEVE	32	PLUMBER	
24	TEACHER	JILL	24	TEACHER	
42	ENGINEER	Ann	42	ENGINEER	
				1	

REMOVE COLUMNS

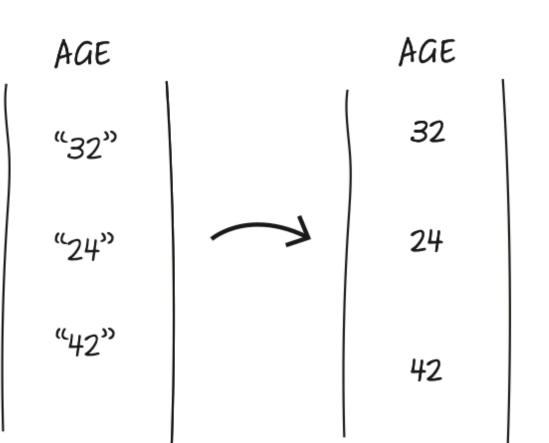
RENAME COLUMNS



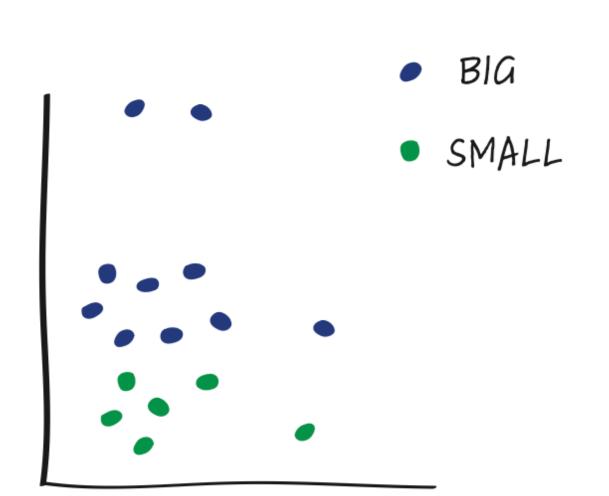
EDIT VALUES



FILTER DATA



### CHANGE DATA TYPE



### ADD CATEGORICAL COLUMAN

## RESULTS



### tasks correctly using Persist, compared to 85% for Pandas



### times faster with Persist

# 1111

notebooks using Persist were **reproducible** 

only 7/11 using pandas were



## 

"easier as compared to the code and everything was visible [...] and it didn't take much time."

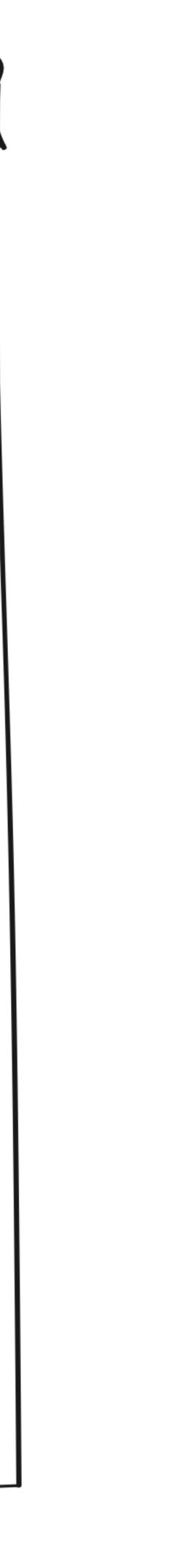
"Changing the category type, or adding new categories or removing anomalies from data, they were very much easier in [Persist] than coding." - M7

"The thing I really liked about is version control, which shows the history of all operations [...] and also saves the changes [...] into a data frame."

"so much easier than manually coding." - M4

- M14

- M2



# DISCUSSION



## GENERATING CODE VS **PROVENANCE TRACKING**

Provenance better for most cases No code clutter Undo/redo Consistent semantics But code generation might be more robust works outside of notebooks works w/o the library

## GENERATE CODE ON DEMAND [Beta]

Ś	ightarrow	C	<b>0</b> ×	87	72	74	E

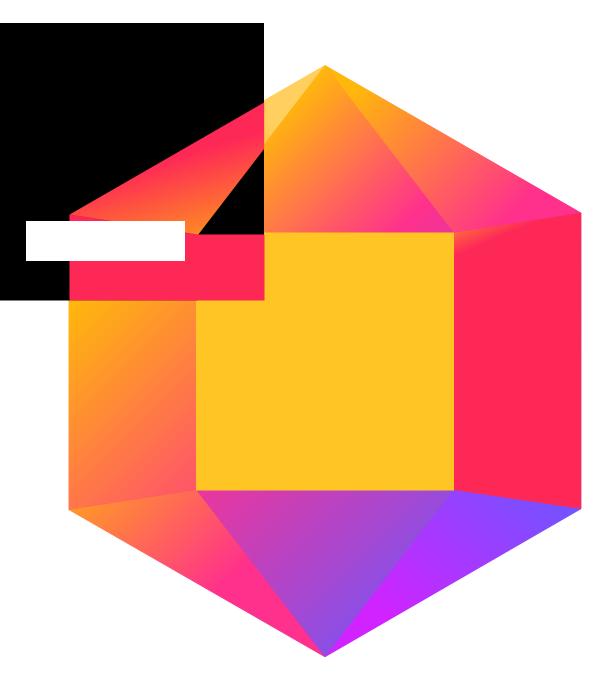
Q Search						
#	Region ↑↓ ፡፡፡ : Yea	ar î↓ ∷ : <mark> </mark> ;Tri	igger ↑↓ ፡፡፡ : ¦;Wea	k Layer î↓ ःः : Der	Q	Root
3	Salt Lake	2012	Skier	Facets		Rename column ;Region to Re
4	Salt Lake	2012	Skier	New Snow	J J	Selected 1 point
5	Salt Lake	2012	Skier	Facets	7 7	Selected 2 points
6	Salt Lake	2012	Skier New	Snow/Old Snow Interface	🚽 🖕 🛉	Removed selected items
7	Salt Lake	2012	Skier	Facets		
8	Salt Lake	2012	Skier	Facets		Dataframes:
9	Salt Lake	2012	Skier	Facets		🗩 persist_df_1  🔂
10	Salt Lake	2012	Skier	Facets	6	
11	Salt Lake	2012	Skier	Facets		
12	Salt Lake	2012	Unknown	Ground Interface		
Dataframe na	ame			390  < < > >		
Dataframe na						
Dataframe na						
<pre> persist_c  def creat df =  # Add df.in df[" # Add </pre>	<pre>df_1 () : df_1 () : df.copy(deep=True) I "id_column" as the isert(0, "id_column", id_column"] = df["i I selection column</pre>	df.index + 1)	pe(str)			
<pre> persist_c  def creat df =  # Add df.in df[" # Add df.in df[" </pre>	<pre>df_1 (df): ce_persist_df_1(df): df.copy(deep=True) "id_column" as the sert(0, "id_column", id_column"] = df["i selection column selected"] = False</pre>	df.index + 1)	<pre>pe(str)</pre>			
<pre> persist_c  def creat df =  # Add df.in df[" # Add df.in df[" # Add df[" # Add </pre>	<pre>df_1 () : df_1 () : df.copy(deep=True) I "id_column" as the isert(0, "id_column", id_column"] = df["i I selection column</pre>	df.index + 1) d_column"].asty				
<pre> persist_c  def creat df =  # Add df.in df[" # Add df.in df[" # Add df[" # Add df[" # Ren df = </pre>	<pre>df_1 () : df_1 () : df.copy(deep=True) "id_column" as the sert(0, "id_column", id_column"] = df["i selection column selected"] = False ame column</pre>	df.index + 1) d_column"].asty	• } )			
<pre> persist_c  def creat df =  # Add df.in  df.in  # Add df.in  # Add df.in  df["</pre>	<pre>df_1 (df): ce_persist_df_1(df): df.copy(deep=True) "id_column" as the sert(0, "id_column", id_column"] = df["i selection column selected"] = False name column df.rename(columns={';F</pre>	<pre>df.index + 1) d_column"].asty Region': 'Region sin(['2']), "s</pre>	<pre>'}) selected"] = True</pre>			



## **BEYOND JUPYTER**

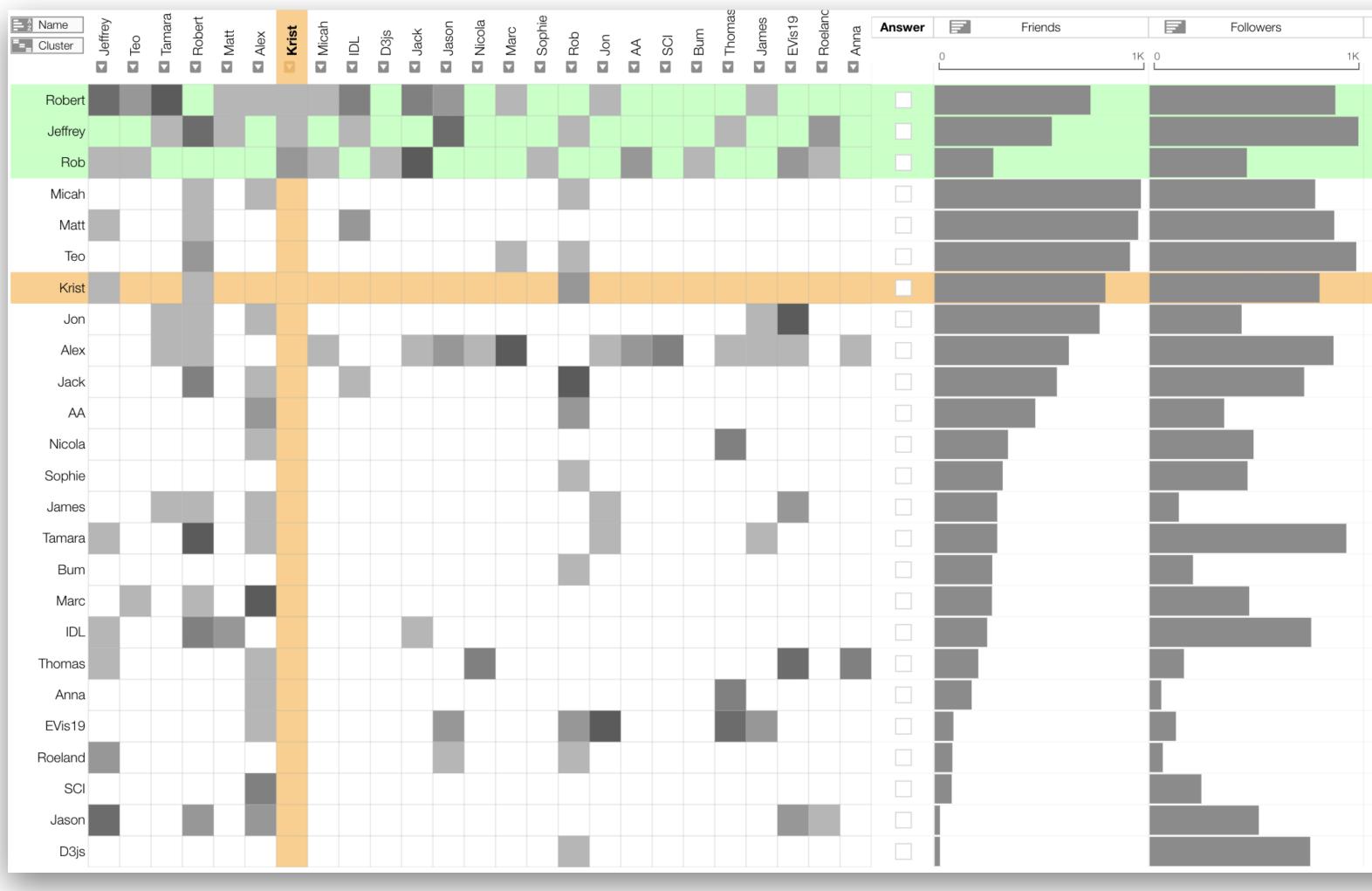






### **DIFFERENT DATA TYPES** DIFFERENT CHARTING LIBRARIES

## Other Interactive Plotting Libraries Plot.ly, Bokeh Other data types Maps, Networks, ...





### [Nobre et al., 2020]



=_ Pers	Inst.	
ł		

## TRY OUT PERSIST!

## Persist is available now! https://vdl.sci.utah.edu/persist/ Documentation & examples

### Feedback / bug reports appreciated!

### **Getting Started**

### Requirements

- pandas >= 0.25
- altair >= 5
- ipywidgets
- anywidget

### Install

To install the extension, execute:

pip install persist\_ext

If the Jupyter server was already running, you might have to reload the browser page and restart the kernel.

### Uninstall

To remove the extension, execute:

pip uninstall persist\_ext

### Usage

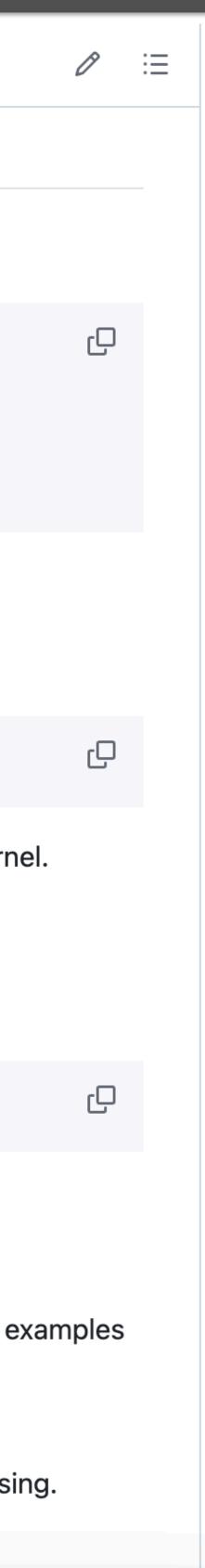
Persist supports two types of interactive outputs — a custom data table and Vega-Altair (>=5.0.0, see requirements and caveats) charts. The following examples will walk you through creating each one. The examples are also available as notebooks in the examples folder of the repository. Each section will link to the corresponding notebook as well as a binder link for the notebook.

Persist currently works with pandas dataframes, so load/convert the data to pandas dataframe before using.

### Examples

▲ BSD-3-Clause license

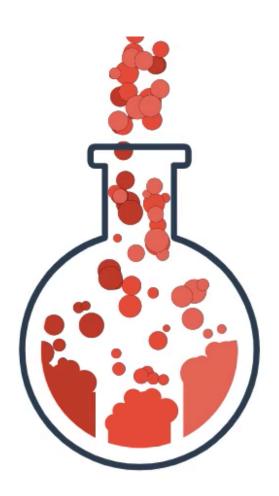
```
– JupyterLab >= 4.0.0 or Jupyter Notebook >= 7.0.0
```



### Kiran Gadhave, Zach Cutler, Alexander Lex http://vdl.sci.utah.edu

# **Persist:** Persistent and Reusable Interactions in Computational Notebooks





### visualization design lab





**Thanks to Jake Wagoner Thanks to the NSF for Funding** (IIS 1751238, CNS 213756, and CNS-2313998)

### THE UNIVERSITY OF UTAH



