AURiQ

Interactive & Real-Time Analysis
for
Billions of Rows

What is Pivot Billions?

- Software run in the cloud (AWS) designed for business users
- Analyze any amount of data in any format
- Intuitive spreadsheet-like UI
- Real-time and interactive
- Sorts, Filters, Distributions, Calculations and Charting are all done in seconds.

Data Source

NYC Taxi & Limousine Trip Sheet Data

http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml

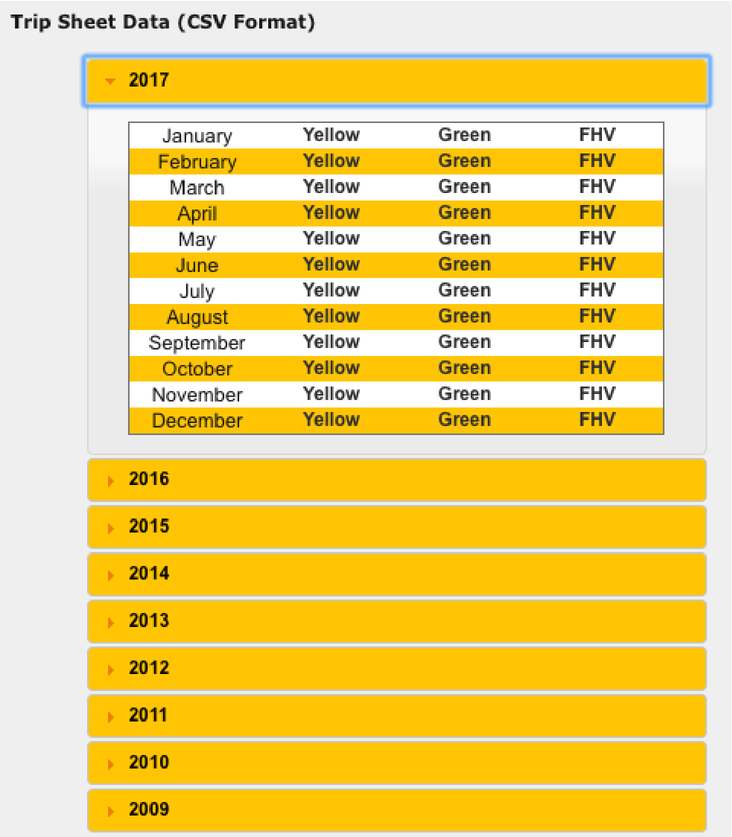
208 csv files

270 gigabytes

1.5 billions rows

Yellow taxi: 1.42 billion rows
Green taxi: 64 million rows
Uber: 18 million rows

Downloaded all files, compressed and then uploaded to Pivotbillions.com. ~ Approx. 2 nights to complete



Trip Sheet Data (CSV Format)

2017

January	Yellow	Green	FHV
February	Yellow	Green	FHV
March	Yellow	Green	FHV
April	Yellow	Green	FHV
May	Yellow	Green	FHV
June	Yellow	Green	FHV
July	Yellow	Green	FHV
August	Yellow	Green	FHV
September	Yellow	Green	FHV
October	Yellow	Green	FHV
November	Yellow	Green	FHV
December	Yellow	Green	FHV

2016

2015

2014

2013

2012

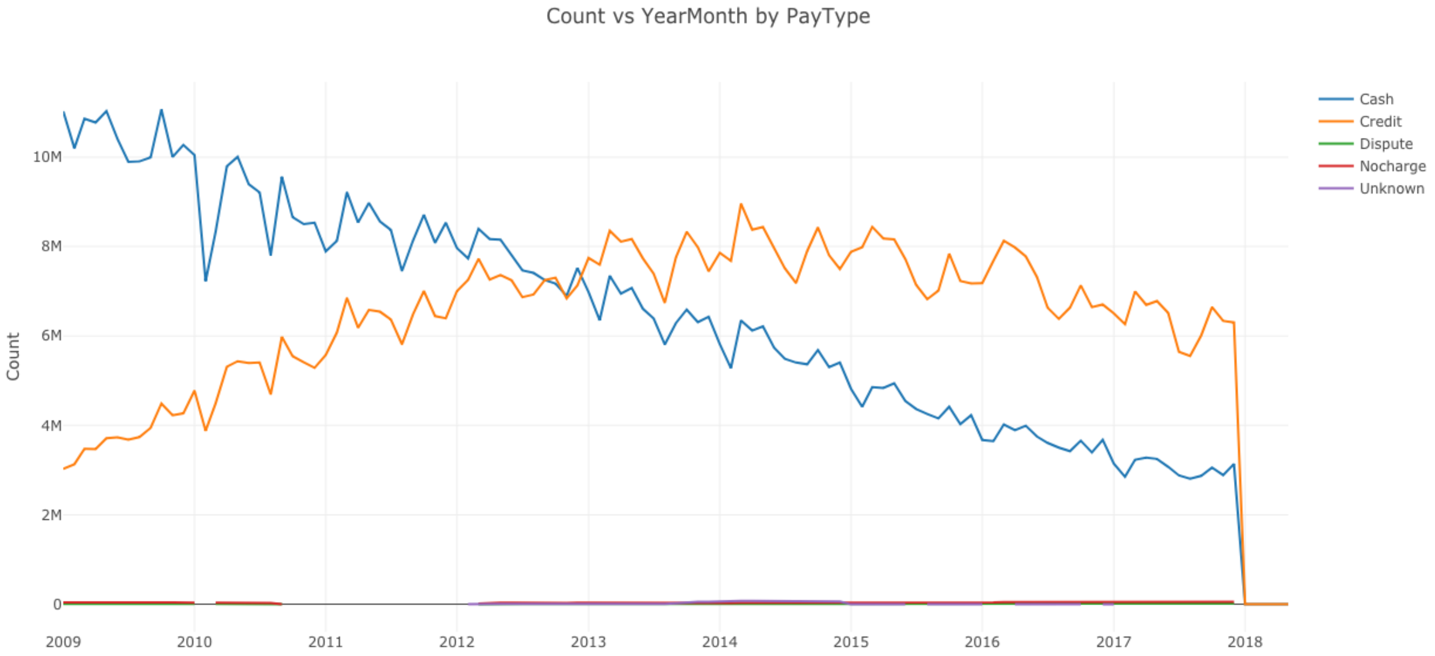
2011

2010

2009

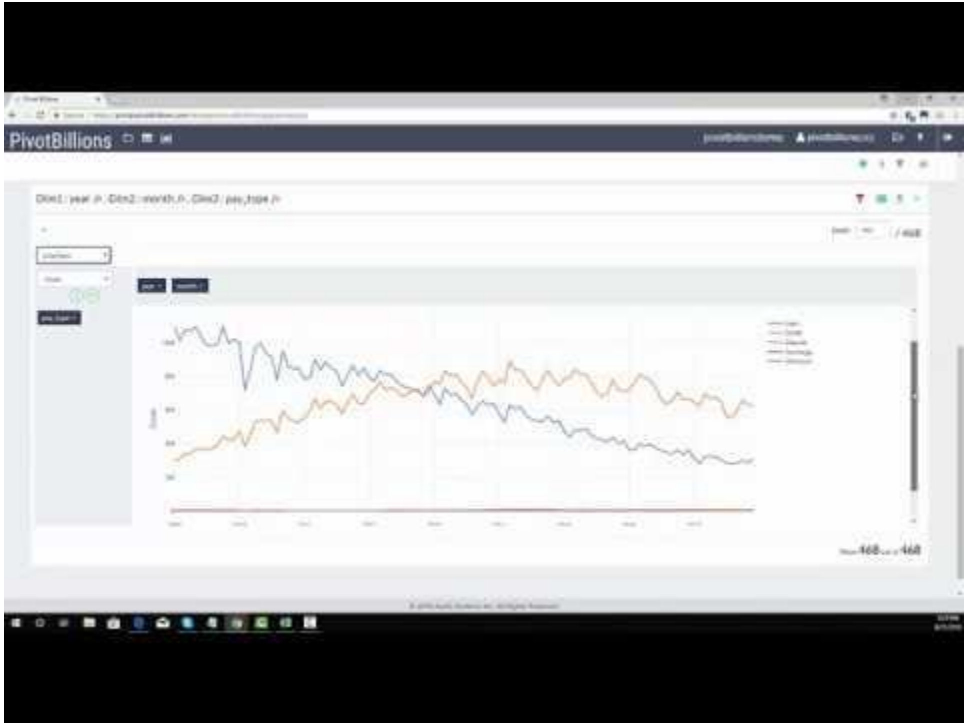
Analyzing 1.5 billion data in 10 seconds

Monthly taxi usage counts by payment type



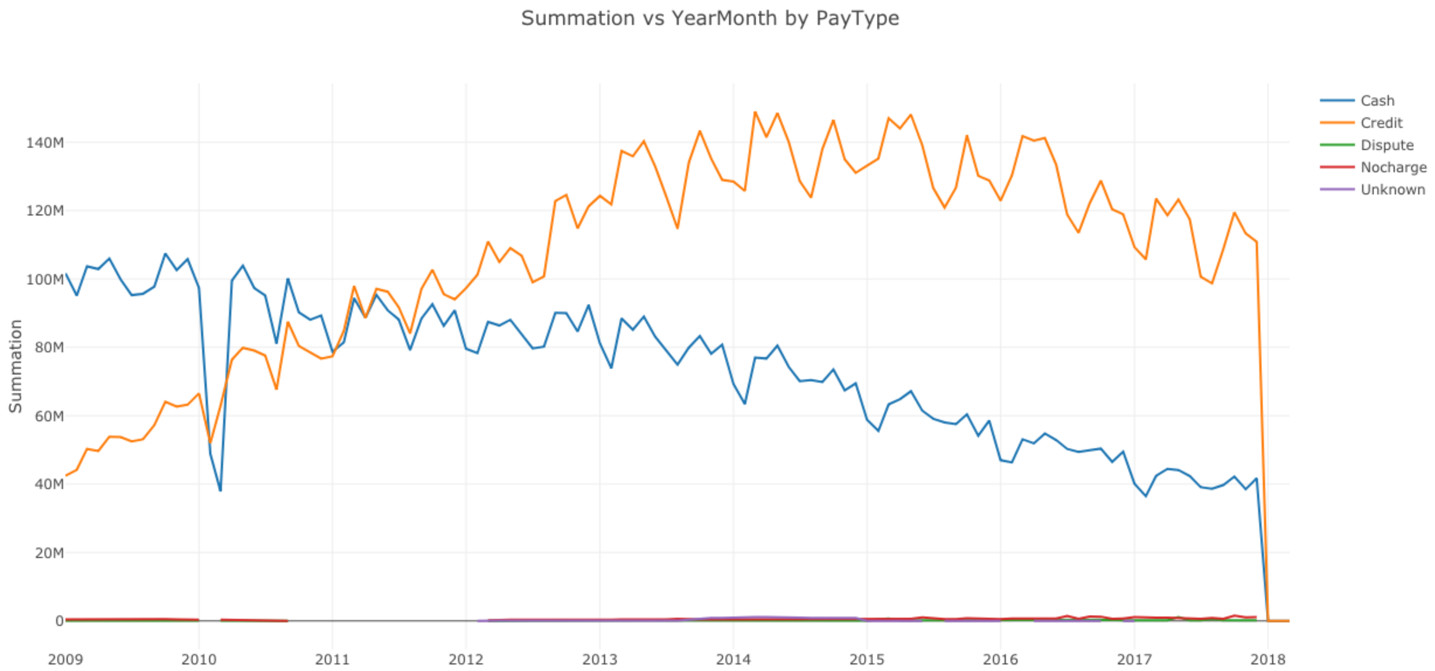
Analyzing 1.5 billion data in 10 seconds

Monthly taxi usage counts by payment type



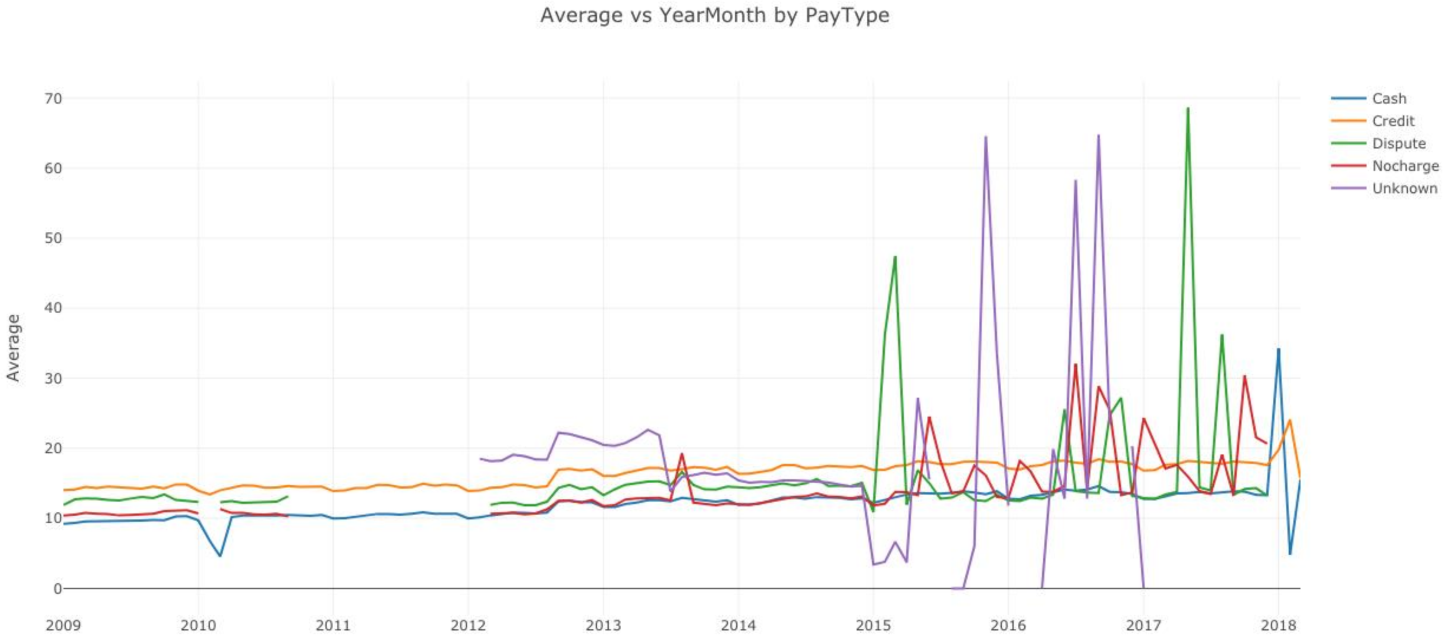
Analyzing 1.5 billion data in 10 seconds

Monthly taxi sales amount by payment type



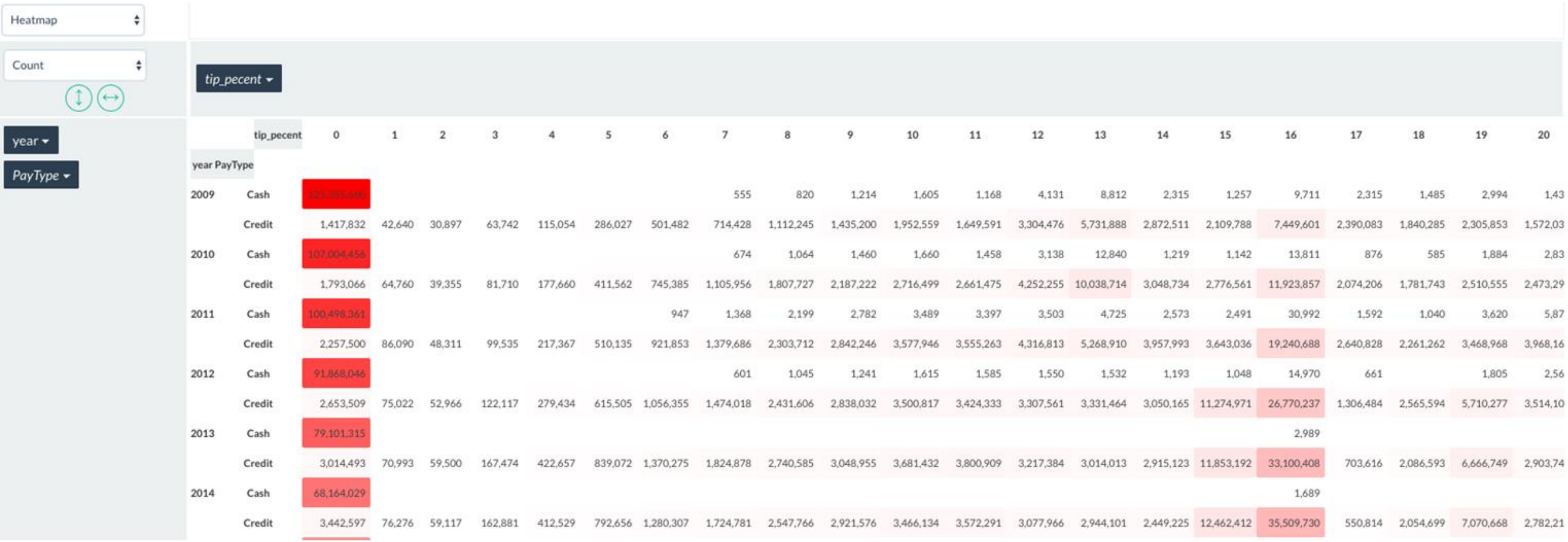
Analyzing 1.5 billion data in 10 seconds

Monthly taxi average sales by payment type



Analyzing 1.5 billion In 15 seconds

Trend of tip% by payment type

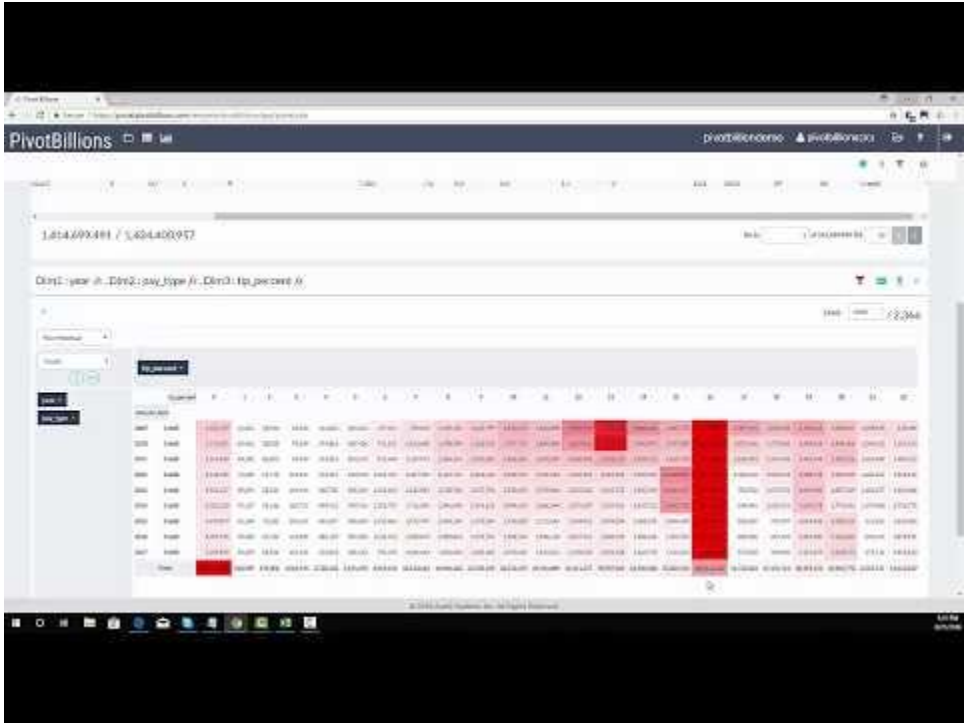


	tip_pecent	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
year	PayType																					
2009	Cash	121,355,660							555	820	1,214	1,605	1,168	4,131	8,812	2,315	1,257	9,711	2,315	1,485	2,994	1,43
	Credit	1,417,832	42,640	30,897	63,742	115,054	286,027	501,482	714,428	1,112,245	1,435,200	1,952,559	1,649,591	3,304,476	5,731,888	2,872,511	2,109,788	7,449,601	2,390,083	1,840,285	2,305,853	1,572,03
2010	Cash	107,004,456							674	1,064	1,460	1,660	1,458	3,138	12,840	1,219	1,142	13,811	876	585	1,884	2,83
	Credit	1,793,066	64,760	39,355	81,710	177,660	411,562	745,385	1,105,956	1,807,727	2,187,222	2,716,499	2,661,475	4,252,255	10,038,714	3,048,734	2,776,561	11,923,857	2,074,206	1,781,743	2,510,555	2,473,29
2011	Cash	100,498,361						947	1,368	2,199	2,782	3,489	3,397	3,503	4,725	2,573	2,491	30,992	1,592	1,040	3,620	5,87
	Credit	2,257,500	86,090	48,311	99,535	217,367	510,135	921,853	1,379,686	2,303,712	2,842,246	3,577,946	3,555,263	4,316,813	5,268,910	3,957,993	3,643,036	19,240,688	2,640,828	2,261,262	3,468,968	3,968,16
2012	Cash	91,868,046							601	1,045	1,241	1,615	1,585	1,550	1,532	1,193	1,048	14,970	661		1,805	2,56
	Credit	2,653,509	75,022	52,966	122,117	279,434	615,505	1,056,355	1,474,018	2,431,606	2,838,032	3,500,817	3,424,333	3,307,561	3,331,464	3,050,165	11,274,971	26,770,237	1,306,484	2,565,594	5,710,277	3,514,10
2013	Cash	79,101,315																2,989				
	Credit	3,014,493	70,993	59,500	167,474	422,657	839,072	1,370,275	1,824,878	2,740,585	3,048,955	3,681,432	3,800,909	3,217,384	3,014,013	2,915,123	11,853,192	33,100,408	703,616	2,086,593	6,666,749	2,903,74
2014	Cash	68,164,029																1,689				
	Credit	3,442,597	76,276	59,117	162,881	412,529	792,656	1,280,307	1,724,781	2,547,766	2,921,576	3,466,134	3,572,291	3,077,966	2,944,101	2,449,225	12,462,412	35,509,730	550,814	2,054,699	7,070,668	2,782,21

Most people paying in cash did not pay tip or a driver did not report. Most paying by credit card pays 16% maybe due to a touch panel payment system.

Analyzing 1.5 billion In 15 seconds

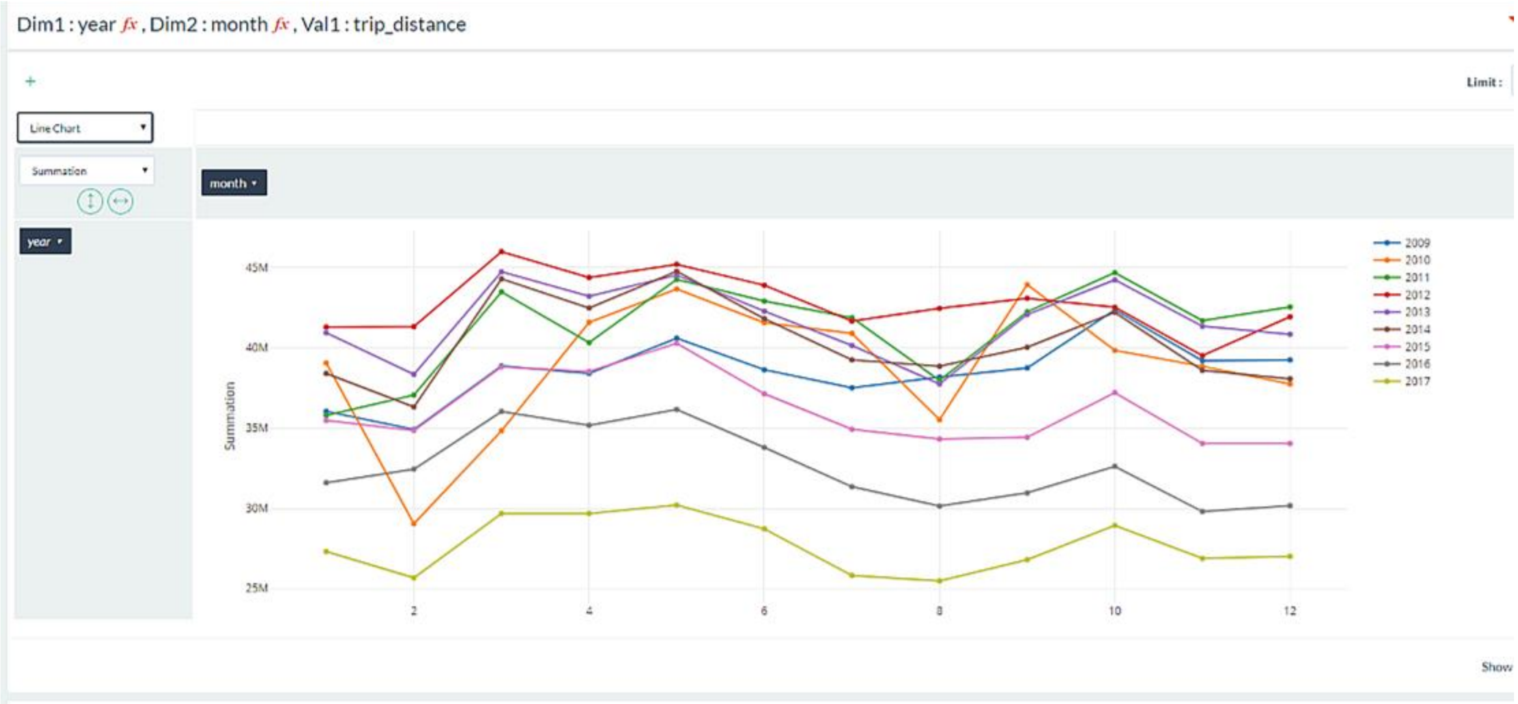
Trend of tip% by payment type



1,414,699,491 / 1,424,400,957

Dim1 : year /r . Dim2 : pay_type /r . Dim3 : tip_percent /r

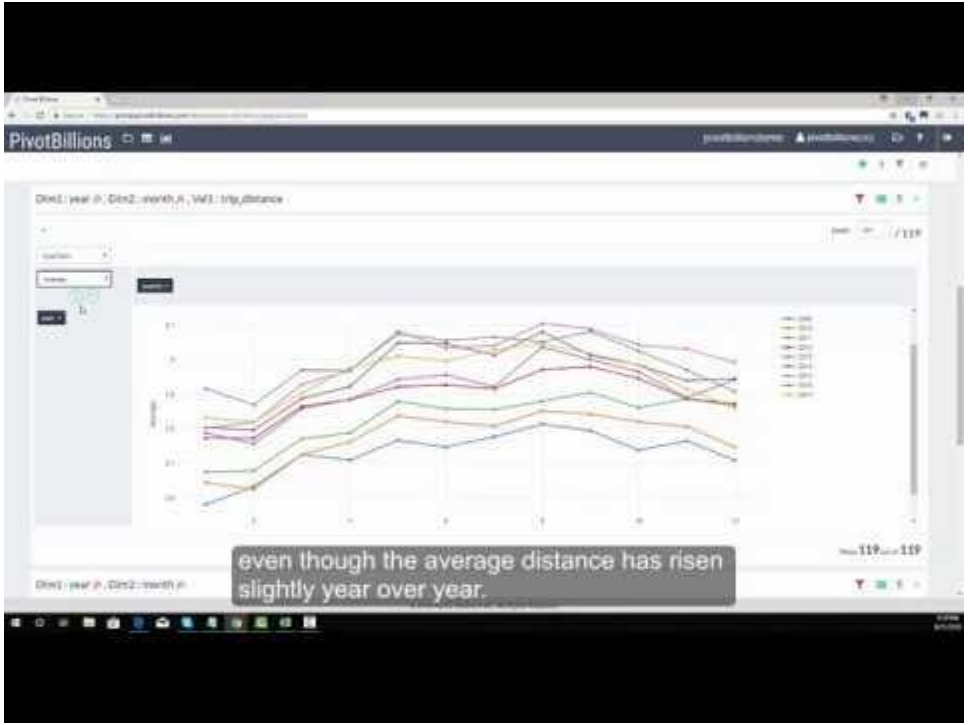
Analyzing 1.5 billion: Yearly trip distance trends



Dim1 : year fx , Dim2 : month fx , Val1 : trip_distance

Rideshare has resulted in significant reduction in total trip distances logged year over year since 2013.

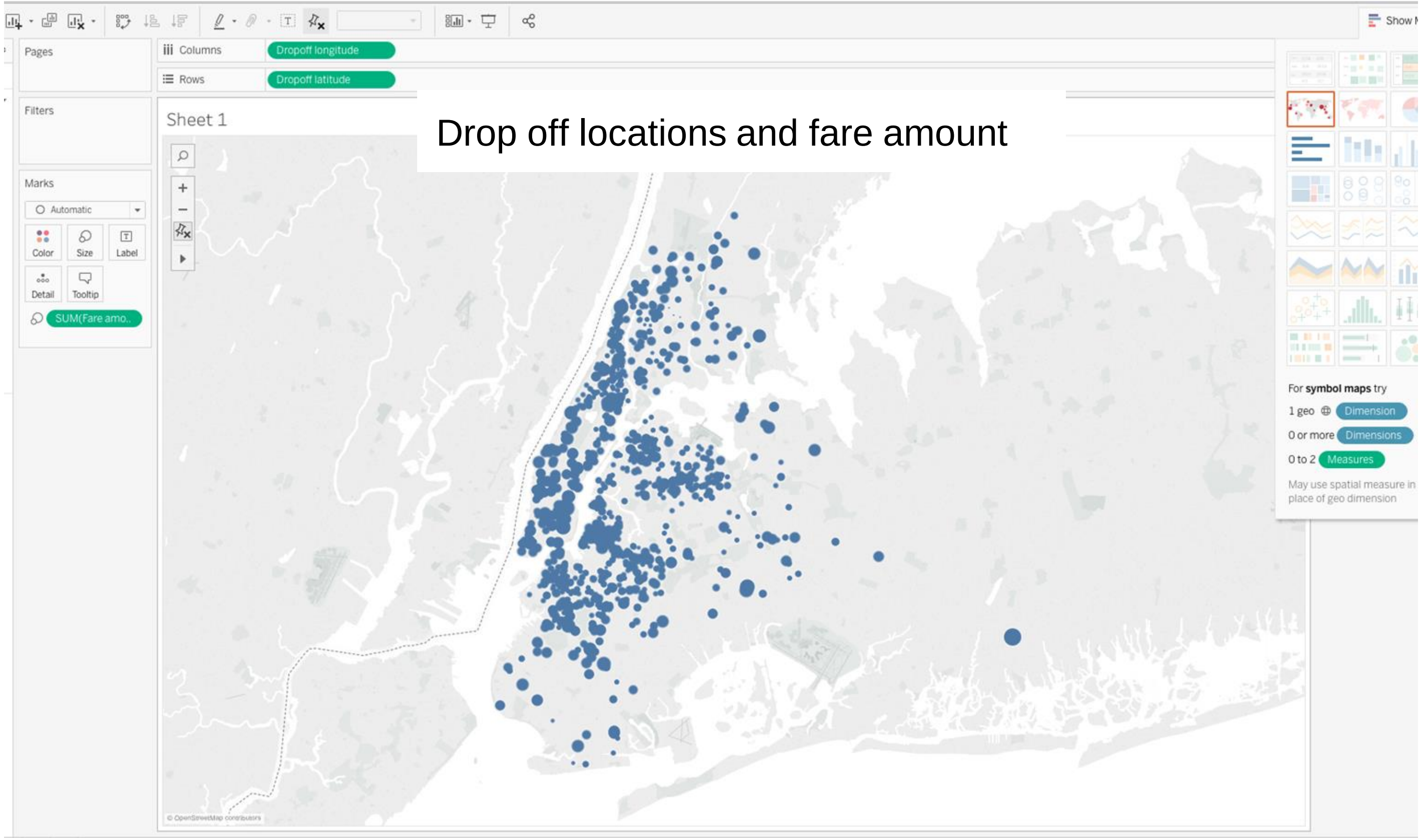
Analyzing 1.5 billion: Yearly trip distance trends



PivotBillions
Dim1 : year_n , Dim2 : month_n , Val1 : trip_distance
/ 119
119
119
Dim1 : year_n , Dim2 : month_n

even though the average distance has risen slightly year over year.

PivotBillions + Tableau example



Drop off locations and fare amount

Columns: Dropoff longitude

Rows: Dropoff latitude

Marks: Automatic

SUM(Fare amo..

For **symbol maps** try

1 geo Dimension

0 or more Dimensions

0 to 2 Measures

May use spatial measure in place of geo dimension

Why Pivot Billions?

- Business users don't want an underpowered, batch based, SQL heavy tool to be able to analyze all the data at their disposal.

- They want an agile, real-time, interactive solution capable of handling any amount of data thrown at it.

- Instead of 3-6 months for an EDW project, they want to get the answers they need within days or weeks.

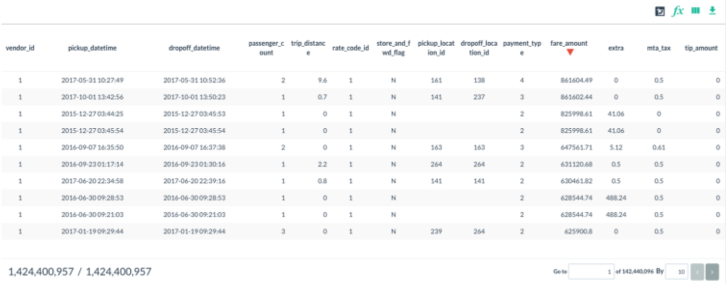
- Not just a preview or sampling of the data, but the entire thing.

Win Win for All

- Fast to implement, scalable and highly adaptable makes IT’s job easier to provide solutions for end users within days.
- Business users have little to no learning curve; just use the skills they already have from using Excel all these years.
- Cost effective cloud based solution. Only use as much resources as needed when its needed.
- Secure, role-based management with available data encryption.

How it Works

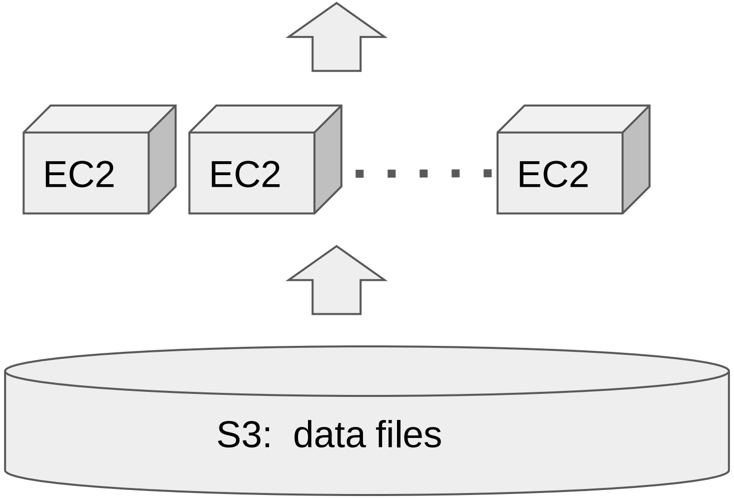
PivotBillions in Amazon Web Service



vendor_id	pickup_datetime	dropoff_datetime	passenger_count	trip_distance	rate_code_id	store_and_fwd_flag	pickup_location_id	dropoff_location_id	payment_type	fare_amount ▼	extra	mta_tax	tip_amount
1	2017-05-31 10:27:49	2017-05-31 10:52:36	2	9.6	1	N	161	138	4	861604.49	0	0.5	0
1	2017-10-01 13:42:56	2017-10-01 13:50:23	1	0.7	1	N	141	237	3	861602.44	0	0.5	0
1	2015-12-27 03:44:25	2015-12-27 03:45:53	1	0	1	N			2	825998.61	41.06	0	0
1	2015-12-27 03:45:54	2015-12-27 03:45:54	1	0	1	N			2	825998.61	41.06	0	0
1	2016-09-07 16:35:50	2016-09-07 16:37:38	2	0	1	N	163	163	3	647561.71	5.12	0.61	0
1	2016-09-23 01:17:14	2016-09-23 01:30:16	1	2.2	1	N	264	264	2	631120.68	0.5	0.5	0
1	2017-06-20 22:34:58	2017-06-20 22:39:16	1	0.8	1	N	141	141	2	630461.82	0.5	0.5	0
1	2016-06-30 09:28:53	2016-06-30 09:28:53	1	0	1	N			2	628544.74	488.24	0.5	0
1	2016-06-30 09:21:03	2016-06-30 09:21:03	1	0	1	N			2	628544.74	488.24	0.5	0
1	2017-01-19 09:29:44	2017-01-19 09:29:44	3	0	1	N	239	264	2	625900.8	0	0.5	0

1,424,400,957 / 1,424,400,957

Go to 1 of 142,440,096 By 10



EC2 EC2 · · · · · EC2

S3: data files

View and analyze entire 1.5 billion data in few seconds

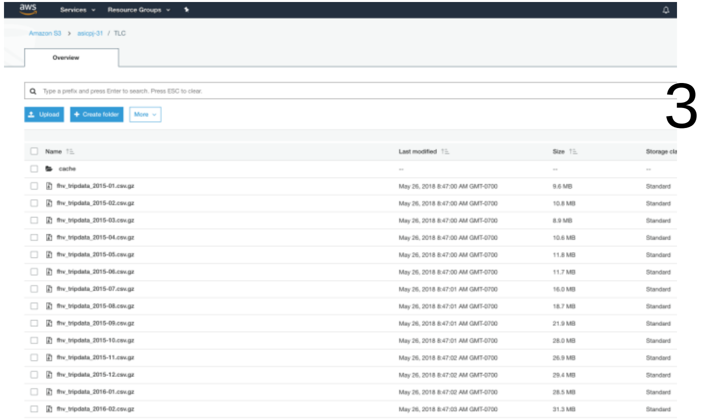
Launch 170 x c4.large and then load 1.5 billion data into Pivotbillion in 3 minutes

Number of instances is configurable
Scale up for bigger data or faster processing

Data files are in S3 (in original format)

208 files in csv & compressed
Can deal with virtually any data format any size.

Loading 1.5 billion data from 208 files in S3 to PivotBillions

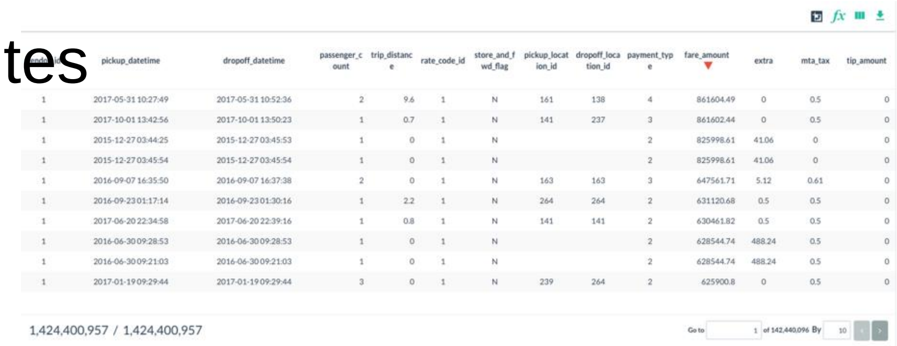

aws
Services
Resource Groups

Amazon S3 › asicpj-31 / TLC

Overview
Type a prefix and press Enter to search. Press ESC to clear.
Upload
Create folder
More

Name	Last modified	Size	Storage cla
cache	--	--	--
fhv_tripdata_2015-01.csv.gz	May 26, 2018 8:47:00 AM GMT-0700	9.6 MB	Standard
fhv_tripdata_2015-02.csv.gz	May 26, 2018 8:47:00 AM GMT-0700	10.8 MB	Standard
fhv_tripdata_2015-03.csv.gz	May 26, 2018 8:47:00 AM GMT-0700	8.9 MB	Standard
fhv_tripdata_2015-04.csv.gz	May 26, 2018 8:47:00 AM GMT-0700	10.6 MB	Standard
fhv_tripdata_2015-05.csv.gz	May 26, 2018 8:47:00 AM GMT-0700	11.8 MB	Standard
fhv_tripdata_2015-06.csv.gz	May 26, 2018 8:47:00 AM GMT-0700	11.7 MB	Standard
fhv_tripdata_2015-07.csv.gz	May 26, 2018 8:47:01 AM GMT-0700	16.0 MB	Standard
fhv_tripdata_2015-08.csv.gz	May 26, 2018 8:47:01 AM GMT-0700	18.7 MB	Standard
fhv_tripdata_2015-09.csv.gz	May 26, 2018 8:47:01 AM GMT-0700	21.9 MB	Standard
fhv_tripdata_2015-10.csv.gz	May 26, 2018 8:47:01 AM GMT-0700	28.0 MB	Standard
fhv_tripdata_2015-11.csv.gz	May 26, 2018 8:47:02 AM GMT-0700	26.9 MB	Standard
fhv_tripdata_2015-12.csv.gz	May 26, 2018 8:47:02 AM GMT-0700	29.4 MB	Standard
fhv_tripdata_2016-01.csv.gz	May 26, 2018 8:47:02 AM GMT-0700	28.5 MB	Standard
fhv_tripdata_2016-02.csv.gz	May 26, 2018 8:47:03 AM GMT-0700	31.3 MB	Standard

3 minutes



endo id	pickup_datetime	dropoff_datetime	passenger_count	trip_distance	rate_code_id	store_and_fwd_flag	pickup_location_id	dropoff_location_id	payment_type	fare_amount ▼	extra	mta_tax	tip_amount
1	2017-05-31 10:27:49	2017-05-31 10:52:36	2	9.6	1	N	161	138	4	861604.49	0	0.5	0
1	2017-10-01 13:42:56	2017-10-01 13:50:23	1	0.7	1	N	141	237	3	861602.44	0	0.5	0
1	2015-12-27 03:44:25	2015-12-27 03:45:53	1	0	1	N			2	825998.61	41.06	0	0
1	2015-12-27 03:45:54	2015-12-27 03:45:54	1	0	1	N			2	825998.61	41.06	0	0
1	2016-09-07 16:35:50	2016-09-07 16:37:38	2	0	1	N	163	163	3	647561.71	5.12	0.61	0
1	2016-09-23 01:17:14	2016-09-23 01:30:16	1	2.2	1	N	264	264	2	631120.68	0.5	0.5	0
1	2017-06-20 22:34:58	2017-06-20 22:39:16	1	0.8	1	N	141	141	2	630461.82	0.5	0.5	0
1	2016-06-30 09:28:53	2016-06-30 09:28:53	1	0	1	N			2	628544.74	488.24	0.5	0
1	2016-06-30 09:21:03	2016-06-30 09:21:03	1	0	1	N			2	628544.74	488.24	0.5	0
1	2017-01-19 09:29:44	2017-01-19 09:29:44	3	0	1	N	239	264	2	625900.8	0	0.5	0

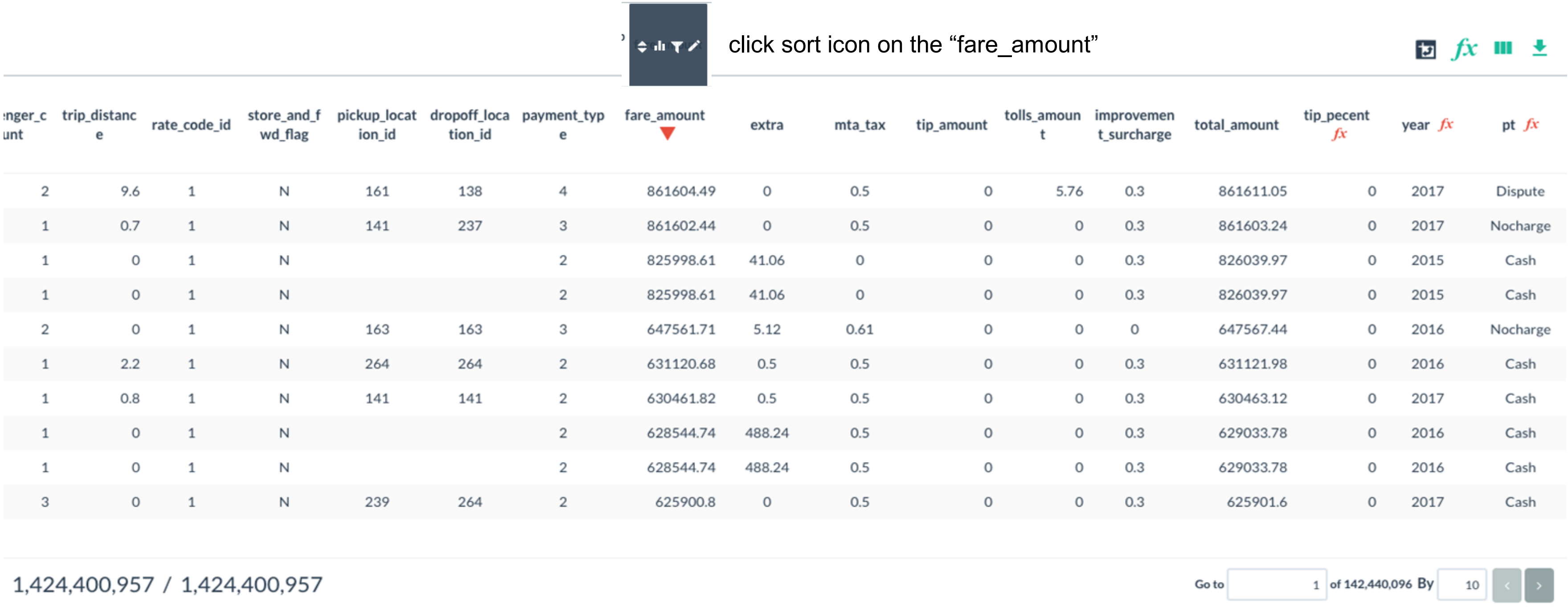
1,424,400,957 / 1,424,400,957

Go to 1 of 142,440,096 By 10

- *Data files in S3 (as-is)*
- *208 compressed files*
- *270 GB original size*

PivotBillions reads all data from all files, apply ETL rule to extract, transform and load entire 1.5 billion rows into excel-like table for real-time analysis.

Sort 1.4 billion rows like Excel in 5 seconds



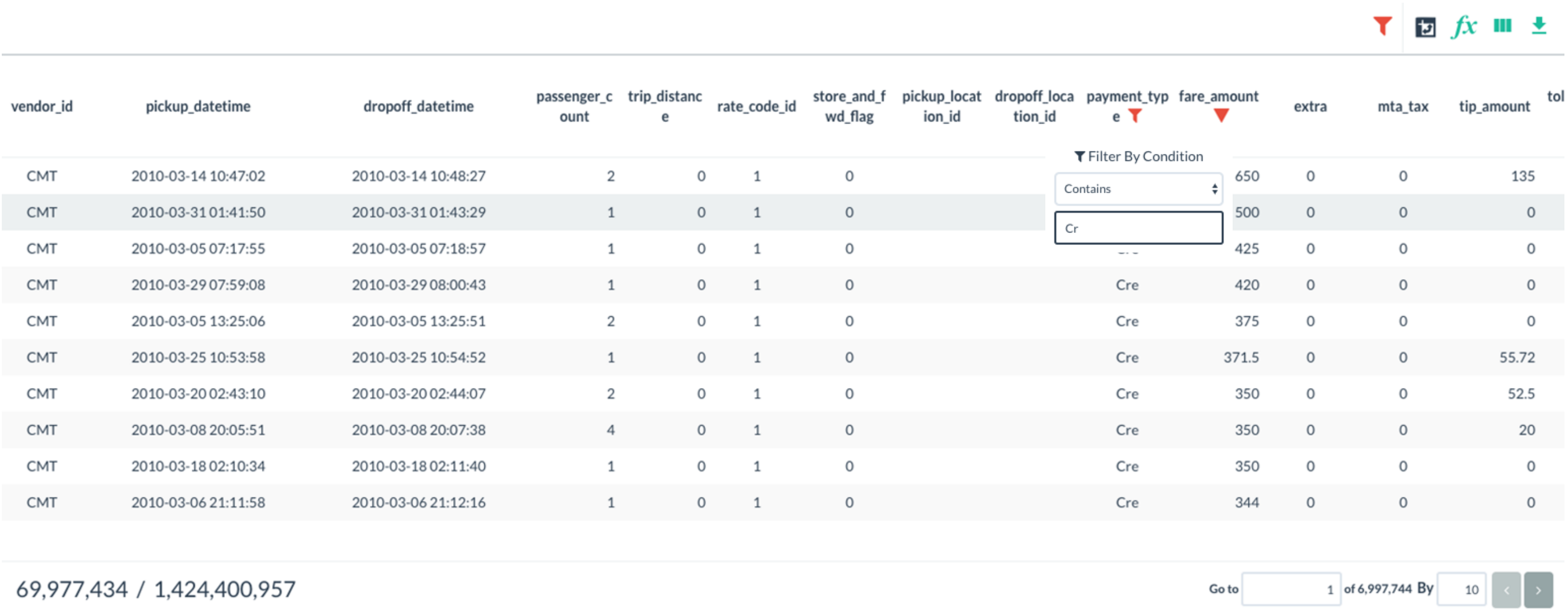
click sort icon on the “fare_amount”

nger_count	trip_distance	rate_code_id	store_and_fwd_flag	pickup_location_id	dropoff_location_id	payment_type	fare_amount ▼	extra	mta_tax	tip_amount	tolls_amount	improvement_surcharge	total_amount	tip_pecent *fx*	year *fx*	pt *fx*
2	9.6	1	N	161	138	4	861604.49	0	0.5	0	5.76	0.3	861611.05	0	2017	Dispute
1	0.7	1	N	141	237	3	861602.44	0	0.5	0	0	0.3	861603.24	0	2017	Nocharge
1	0	1	N			2	825998.61	41.06	0	0	0	0.3	826039.97	0	2015	Cash
1	0	1	N			2	825998.61	41.06	0	0	0	0.3	826039.97	0	2015	Cash
2	0	1	N	163	163	3	647561.71	5.12	0.61	0	0	0	647567.44	0	2016	Nocharge
1	2.2	1	N	264	264	2	631120.68	0.5	0.5	0	0	0.3	631121.98	0	2016	Cash
1	0.8	1	N	141	141	2	630461.82	0.5	0.5	0	0	0.3	630463.12	0	2017	Cash
1	0	1	N			2	628544.74	488.24	0.5	0	0	0.3	629033.78	0	2016	Cash
1	0	1	N			2	628544.74	488.24	0.5	0	0	0.3	629033.78	0	2016	Cash
3	0	1	N	239	264	2	625900.8	0	0.5	0	0	0.3	625901.6	0	2017	Cash

1,424,400,957 / 1,424,400,957

Go to 1 of 142,440,096 By 10

Filtering in 3 seconds



vendor_id	pickup_datetime	dropoff_datetime	passenger_count	trip_distance	rate_code_id	store_and_fwd_flag	pickup_location_id	dropoff_location_id	payment_type	fare_amount	extra	mta_tax	tip_amount	tol
CMT	2010-03-14 10:47:02	2010-03-14 10:48:27	2	0	1	0				650	0	0	135	
CMT	2010-03-31 01:41:50	2010-03-31 01:43:29	1	0	1	0				500	0	0	0	
CMT	2010-03-05 07:17:55	2010-03-05 07:18:57	1	0	1	0				425	0	0	0	
CMT	2010-03-29 07:59:08	2010-03-29 08:00:43	1	0	1	0			Cre	420	0	0	0	
CMT	2010-03-05 13:25:06	2010-03-05 13:25:51	2	0	1	0			Cre	375	0	0	0	
CMT	2010-03-25 10:53:58	2010-03-25 10:54:52	1	0	1	0			Cre	371.5	0	0	55.72	
CMT	2010-03-20 02:43:10	2010-03-20 02:44:07	2	0	1	0			Cre	350	0	0	52.5	
CMT	2010-03-08 20:05:51	2010-03-08 20:07:38	4	0	1	0			Cre	350	0	0	20	
CMT	2010-03-18 02:10:34	2010-03-18 02:11:40	1	0	1	0			Cre	350	0	0	0	
CMT	2010-03-06 21:11:58	2010-03-06 21:12:16	1	0	1	0			Cre	344	0	0	0	

Filter By Condition

Contains

Cr

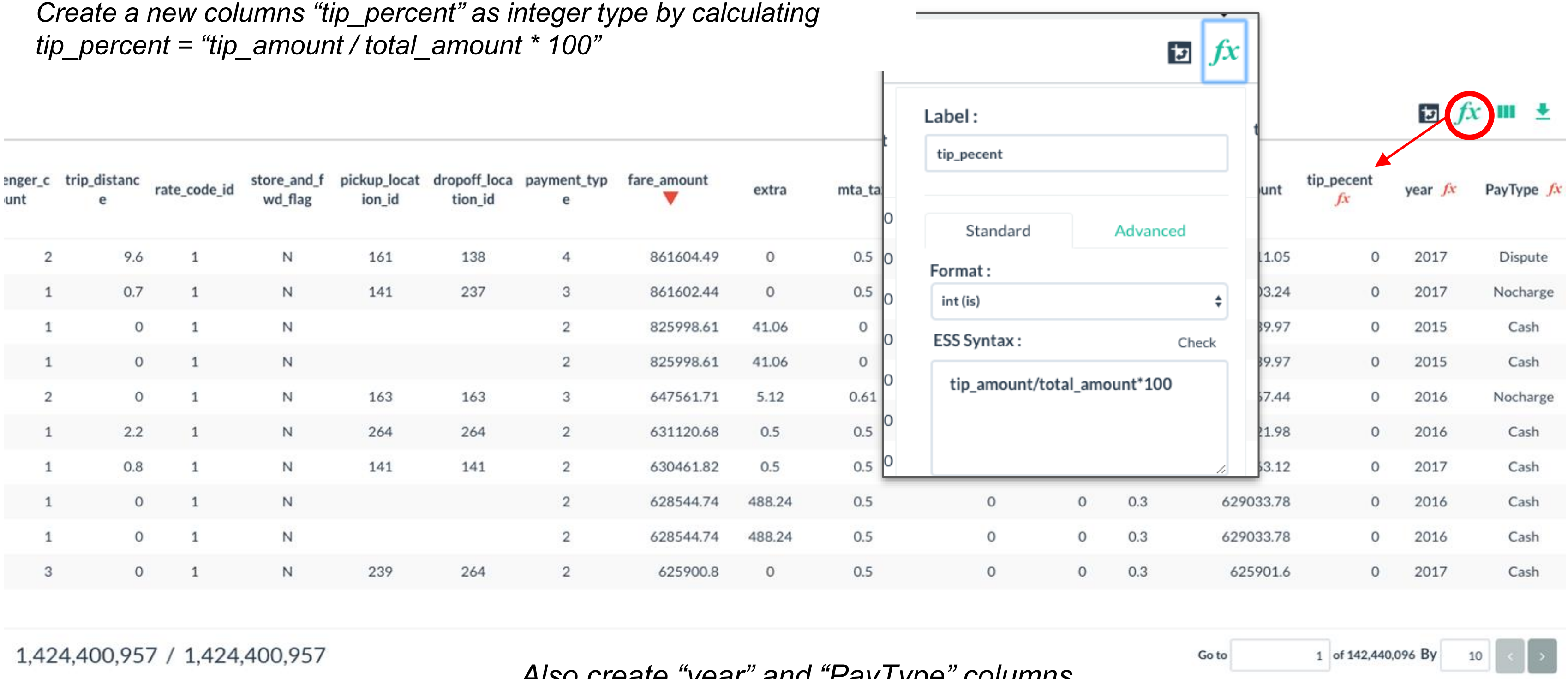
69,977,434 / 1,424,400,957

Go to 1 of 6,997,744 By 10

69 million out of 1.4 billion contains “Cr” in payment_type

Create new columns 1.4 billion rows in 4 seconds

*Create a new columns “tip_percent” as integer type by calculating tip_percent = “tip_amount / total_amount * 100”*



Label : tip_pecent

Standard | Advanced

Format : int (is)

ESS Syntax : Check

```
tip_amount/total_amount*100
```

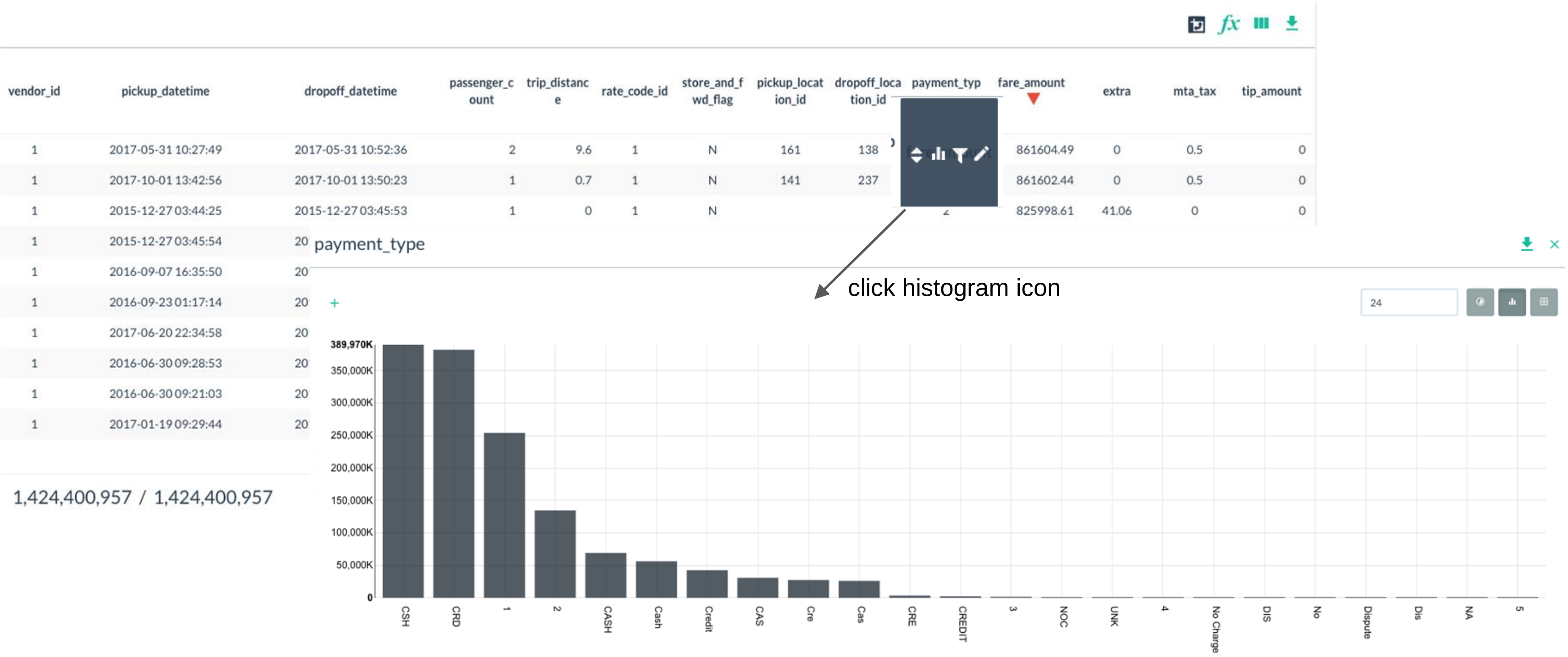
enger_count	trip_distance	rate_code_id	store_and_fwd_flag	pickup_location_id	dropoff_location_id	payment_type	fare_amount ▼	extra	mta_ta				unt	tip_pecent *fx*	year *fx*	PayType *fx*
2	9.6	1	N	161	138	4	861604.49	0	0.5				1.05	0	2017	Dispute
1	0.7	1	N	141	237	3	861602.44	0	0.5				3.24	0	2017	Nocharge
1	0	1	N			2	825998.61	41.06	0				9.97	0	2015	Cash
1	0	1	N			2	825998.61	41.06	0				9.97	0	2015	Cash
2	0	1	N	163	163	3	647561.71	5.12	0.61				7.44	0	2016	Nocharge
1	2.2	1	N	264	264	2	631120.68	0.5	0.5				1.98	0	2016	Cash
1	0.8	1	N	141	141	2	630461.82	0.5	0.5				3.12	0	2017	Cash
1	0	1	N			2	628544.74	488.24	0.5	0	0	0.3	629033.78	0	2016	Cash
1	0	1	N			2	628544.74	488.24	0.5	0	0	0.3	629033.78	0	2016	Cash
3	0	1	N	239	264	2	625900.8	0	0.5	0	0	0.3	625901.6	0	2017	Cash

1,424,400,957 / 1,424,400,957

Go to 1 of 142,440,096 By 10 ‹ ›

Also create “year” and “PayType” columns

Check data distribution in 14 seconds



vendor_id	pickup_datetime	dropoff_datetime	passenger_count	trip_distance	rate_code_id	store_and_fwd_flag	pickup_location_id	dropoff_location_id	payment_type	fare_amount	extra	mta_tax	tip_amount
1	2017-05-31 10:27:49	2017-05-31 10:52:36	2	9.6	1	N	161	138		861604.49	0	0.5	0
1	2017-10-01 13:42:56	2017-10-01 13:50:23	1	0.7	1	N	141	237		861602.44	0	0.5	0
1	2015-12-27 03:44:25	2015-12-27 03:45:53	1	0	1	N			2	825998.61	41.06	0	0
1	2015-12-27 03:45:54	20											
1	2016-09-07 16:35:50	20											
1	2016-09-23 01:17:14	20											
1	2017-06-20 22:34:58	20											
1	2016-06-30 09:28:53	20											
1	2016-06-30 09:21:03	20											
1	2017-01-19 09:29:44	20											

1,424,400,957 / 1,424,400,957

payment_type

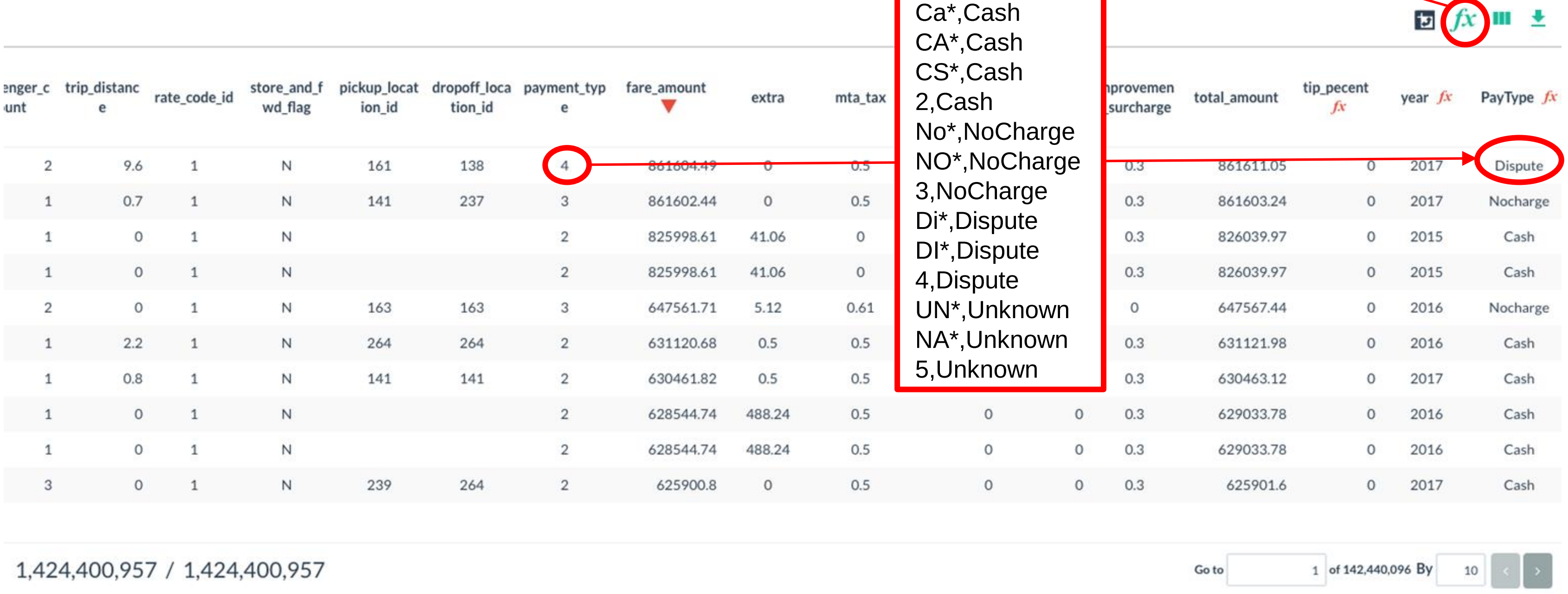
click histogram icon

Looks like payment codes were different each year

Real-time data transformation

A transformation rule for “payment_type

Cr*,Credit
CR*,Credit
1,Credit
Ca*,Cash
CA*,Cash
CS*,Cash
2,Cash
No*,NoCharge
NO*,NoCharge
3,NoCharge
Di*,Dispute
DI*,Dispute
4,Dispute
UN*,Unknown
NA*,Unknown
5,Unknown

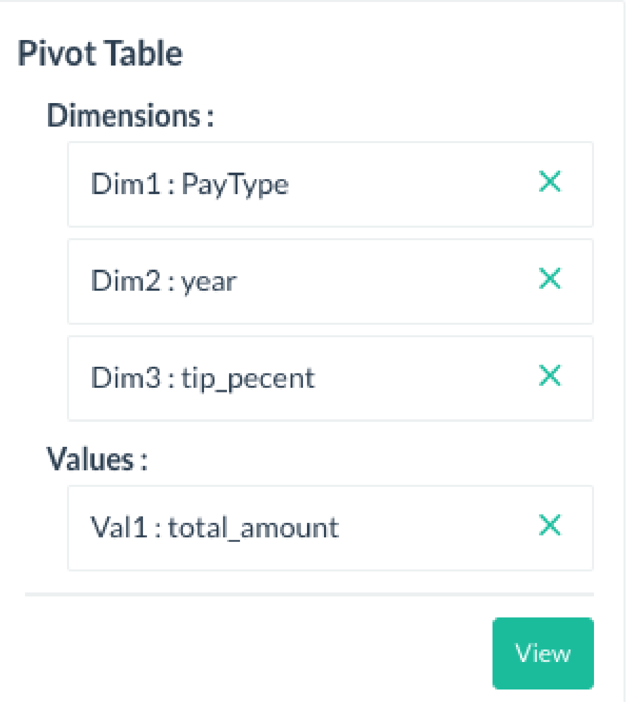


enger_count	trip_distance	rate_code_id	store_and_fwd_flag	pickup_location_id	dropoff_location_id	payment_type	fare_amount ▼	extra	mta_tax			nprovemen_surcharge	total_amount	tip_pecent *fx*	year *fx*	PayType *fx*
2	9.6	1	N	161	138	4	861604.49	0	0.5			0.3	861611.05	0	2017	Dispute
1	0.7	1	N	141	237	3	861602.44	0	0.5			0.3	861603.24	0	2017	Nocharge
1	0	1	N			2	825998.61	41.06	0			0.3	826039.97	0	2015	Cash
1	0	1	N			2	825998.61	41.06	0			0.3	826039.97	0	2015	Cash
2	0	1	N	163	163	3	647561.71	5.12	0.61			0	647567.44	0	2016	Nocharge
1	2.2	1	N	264	264	2	631120.68	0.5	0.5			0.3	631121.98	0	2016	Cash
1	0.8	1	N	141	141	2	630461.82	0.5	0.5			0.3	630463.12	0	2017	Cash
1	0	1	N			2	628544.74	488.24	0.5	0	0	0.3	629033.78	0	2016	Cash
1	0	1	N			2	628544.74	488.24	0.5	0	0	0.3	629033.78	0	2016	Cash
3	0	1	N	239	264	2	625900.8	0	0.5	0	0	0.3	625901.6	0	2017	Cash

1,424,400,957 / 1,424,400,957

Go to 1 of 142,440,096 By 10 ‹ ›

Pivoting 1.5 billion rows



Pivot Table

Dimensions :

Dim1 : PayType

Dim2 : year

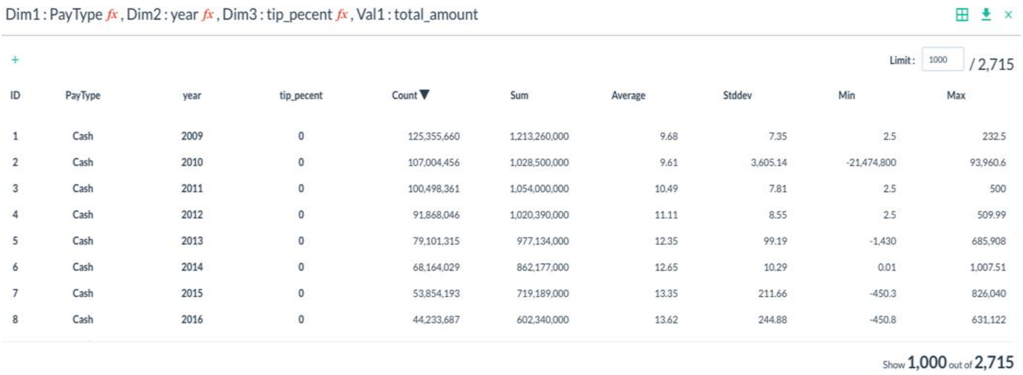
Dim3 : tip_pecent

Values :

Val1 : total_amount

in 15 seconds

For each PayType, year and tip_percent dimension calculate counts, total sales amount, average amount, standard diviation, min and max.



Dim1 : PayType fx , Dim2 : year fx , Dim3 : tip_pecent fx , Val1 : total_amount

Limit : 1000 / 2,715

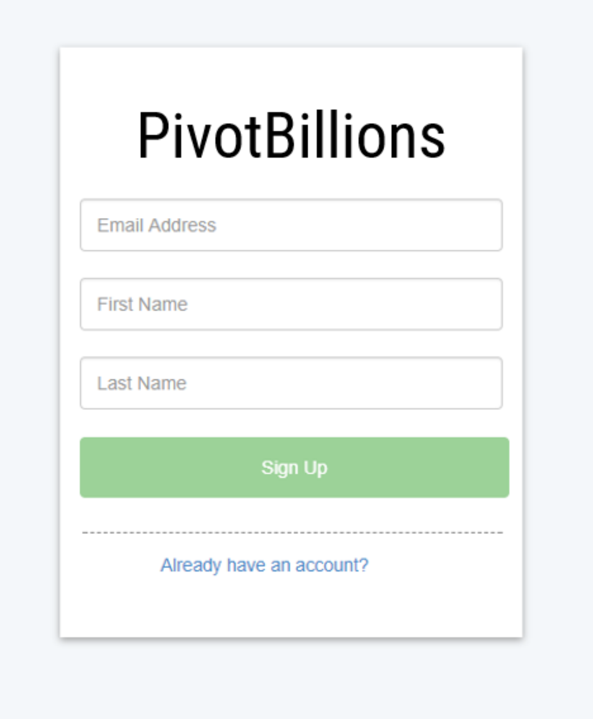
ID	PayType	year	tip_pecent	Count ▼	Sum	Average	Stddev	Min	Max
1	Cash	2009	0	125,355,660	1,213,260,000	9.68	7.35	2.5	232.5
2	Cash	2010	0	107,004,456	1,028,500,000	9.61	3,605.14	-21,474,800	93,960.6
3	Cash	2011	0	100,498,361	1,054,000,000	10.49	7.81	2.5	500
4	Cash	2012	0	91,868,046	1,020,390,000	11.11	8.55	2.5	509.99
5	Cash	2013	0	79,101,315	977,134,000	12.35	99.19	-1,430	685,908
6	Cash	2014	0	68,164,029	862,177,000	12.65	10.29	0.01	1,007.51
7	Cash	2015	0	53,854,193	719,189,000	13.35	211.66	-450.3	826,040
8	Cash	2016	0	44,233,687	602,340,000	13.62	244.88	-450.8	631,122

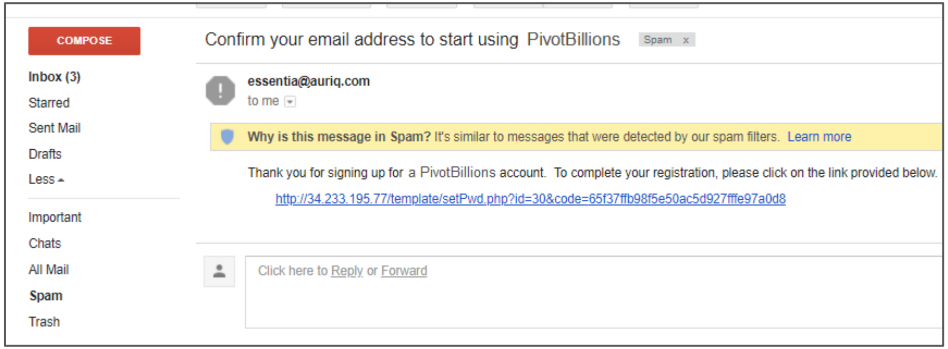
Show 1,000 out of 2,715

More

Sign Up

start analyzing your billions in real-time today.





Confirm your email address to start using PivotBillions

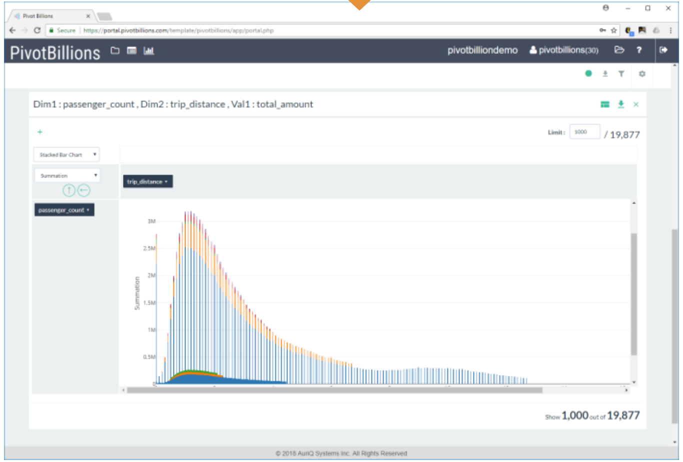
essentia@auriq.com
to me

Why is this message in Spam? It's similar to messages that were detected by our spam filters. Learn more

Thank you for signing up for a PivotBillions account. To complete your registration, please click on the link provided below.

http://34.233.195.77/template/setPwd.php?id=30&code=65f37ffb98f5e50ac5d927fffe97a0d8

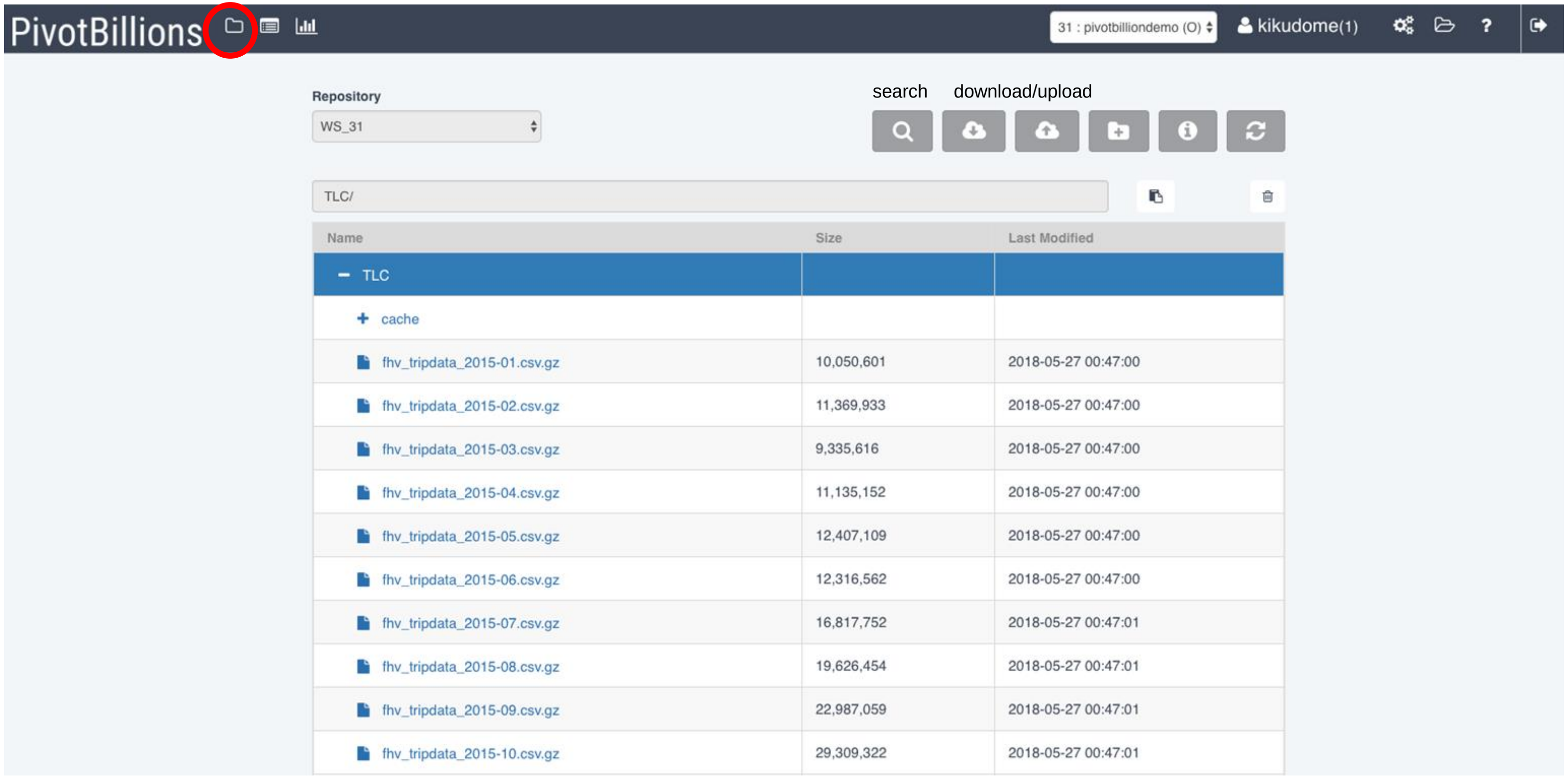
Click here to Reply or Forward



Dim1 : passenger_count , Dim2 : trip_distance , Val1 : total_amount

AURiQ

Explore: manage files in S3

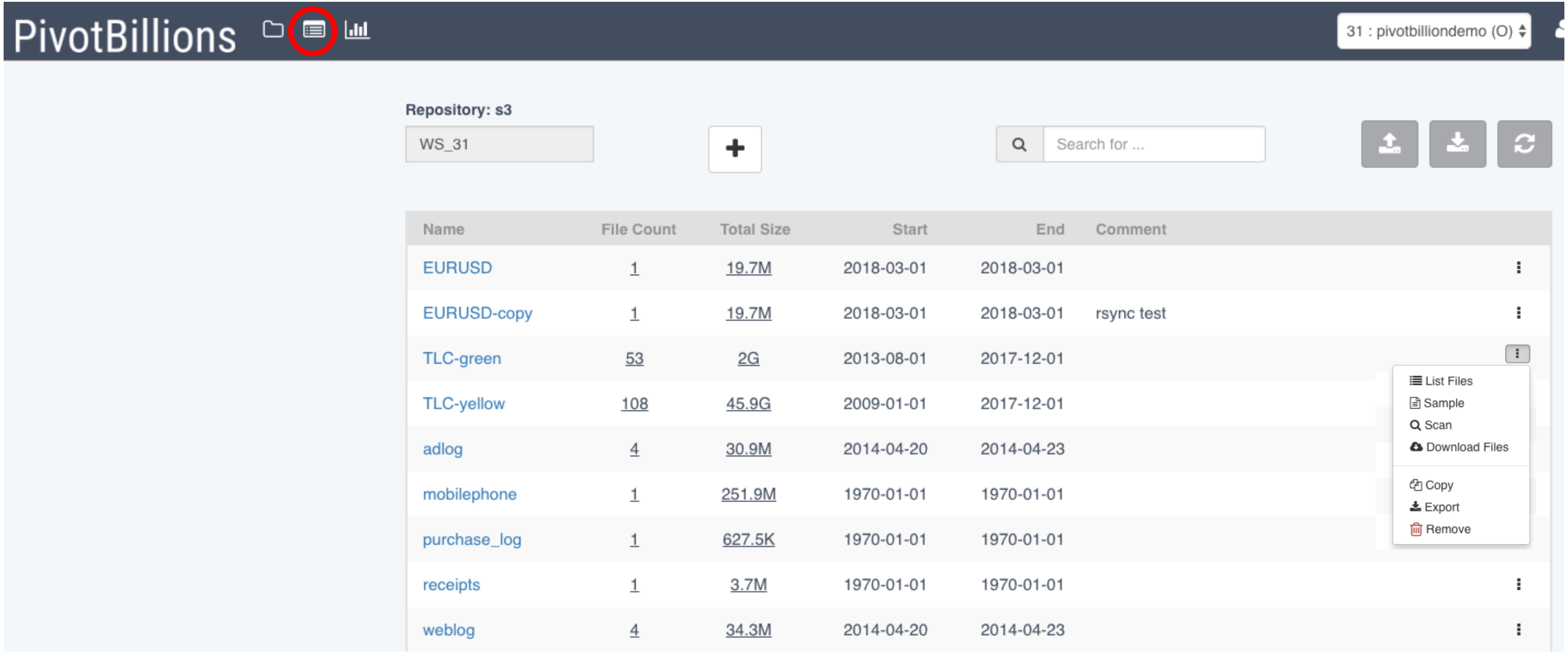


Repository: WS_31

TLC/

Name	Size	Last Modified
− TLC		
+ cache		
fhv_tripdata_2015-01.csv.gz	10,050,601	2018-05-27 00:47:00
fhv_tripdata_2015-02.csv.gz	11,369,933	2018-05-27 00:47:00
fhv_tripdata_2015-03.csv.gz	9,335,616	2018-05-27 00:47:00
fhv_tripdata_2015-04.csv.gz	11,135,152	2018-05-27 00:47:00
fhv_tripdata_2015-05.csv.gz	12,407,109	2018-05-27 00:47:00
fhv_tripdata_2015-06.csv.gz	12,316,562	2018-05-27 00:47:00
fhv_tripdata_2015-07.csv.gz	16,817,752	2018-05-27 00:47:01
fhv_tripdata_2015-08.csv.gz	19,626,454	2018-05-27 00:47:01
fhv_tripdata_2015-09.csv.gz	22,987,059	2018-05-27 00:47:01
fhv_tripdata_2015-10.csv.gz	29,309,322	2018-05-27 00:47:01

Category: data catalog



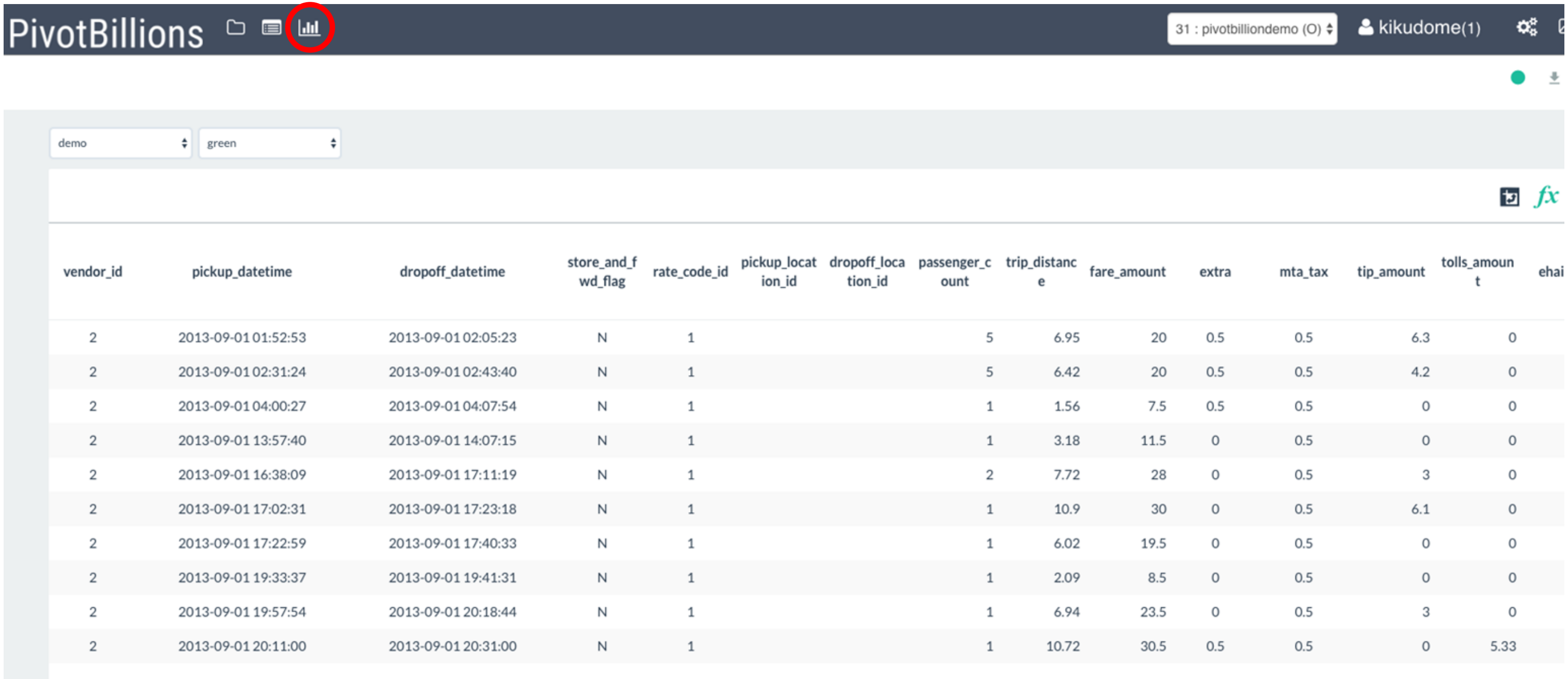
Repository: s3

WS_31

Name	File Count	Total Size	Start	End	Comment
EURUSD	1	19.7M	2018-03-01	2018-03-01	
EURUSD-copy	1	19.7M	2018-03-01	2018-03-01	rsync test
TLC-green	53	2G	2013-08-01	2017-12-01	
TLC-yellow	108	45.9G	2009-01-01	2017-12-01	
adlog	4	30.9M	2014-04-20	2014-04-23	
mobilephone	1	251.9M	1970-01-01	1970-01-01	
purchase_log	1	627.5K	1970-01-01	1970-01-01	
receipts	1	3.7M	1970-01-01	1970-01-01	
weblog	4	34.3M	2014-04-20	2014-04-23	

AURiQ

Pivot



31 : pivotbilliondemo (O)

kikudome(1)

demo

green

vendor_id	pickup_datetime	dropoff_datetime	store_and_fwd_flag	rate_code_id	pickup_location_id	dropoff_location_id	passenger_count	trip_distance	fare_amount	extra	mta_tax	tip_amount	tolls_amount	ehai
2	2013-09-01 01:52:53	2013-09-01 02:05:23	N	1			5	6.95	20	0.5	0.5	6.3	0	
2	2013-09-01 02:31:24	2013-09-01 02:43:40	N	1			5	6.42	20	0.5	0.5	4.2	0	
2	2013-09-01 04:00:27	2013-09-01 04:07:54	N	1			1	1.56	7.5	0.5	0.5	0	0	
2	2013-09-01 13:57:40	2013-09-01 14:07:15	N	1			1	3.18	11.5	0	0.5	0	0	
2	2013-09-01 16:38:09	2013-09-01 17:11:19	N	1			2	7.72	28	0	0.5	3	0	
2	2013-09-01 17:02:31	2013-09-01 17:23:18	N	1			1	10.9	30	0	0.5	6.1	0	
2	2013-09-01 17:22:59	2013-09-01 17:40:33	N	1			1	6.02	19.5	0	0.5	0	0	
2	2013-09-01 19:33:37	2013-09-01 19:41:31	N	1			1	2.09	8.5	0	0.5	0	0	
2	2013-09-01 19:57:54	2013-09-01 20:18:44	N	1			1	6.94	23.5	0	0.5	3	0	
2	2013-09-01 20:11:00	2013-09-01 20:31:00	N	1			1	10.72	30.5	0.5	0.5	0	5.33	

A sample “import script”

```
#######################################################################################
# tlc.sh:  import TLC logs
#######################################################################################
yellow_schema_2017_h1="s:vendor_id s:pickup_datetime s:dropoff_datetime i:passenger_count f:trip_distance s:rate_code_id s:store_and_fwd_flag s:pickup_location_id s:dropoff_location_id s:payment_type f:fare_amount s:extra s:mta_tax f:tip_amount f:tolls_amount s:improvement_surcharge f:total_amount"
udbopt=",ddef"

create_category () {
                ess category add green "/TLC/green_*.csv.gz" --overwrite
                ess category add yellow "/TLC/yellow_*.csv.gz" --overwrite
}

createdb () {
        ess server reset
        ess create database demo --port 0
        ess create table yellow s:vendor_id S,pkey:pickup_datetime S:dropoff_datetime I:passenger_count  F:trip_distance s:rate_code_id S:store_and_fwd_flag s:pickup_location_id s:dropoff_location_id s:payment_type \
                                F:fare_amount s:extra s:mta_tax F:tip_amount f:tolls_amount s:improvement_surcharge F:total_amount
         ess server commit
}

import_yellow() {
                ess stream yellow 2008 2014-12-31 "aq_pp -f+1,eok,qui - -d $yellow_schema_pre_2015  -imp$udbopt demo:yellow"
                ess stream yellow 2015 2016-06-30 "aq_pp -f+1,eok,qui - -d $yellow_schema_2015_2016_h1  -imp$udbopt demo:yellow"
                ess stream yellow 2016-07 2016-12-31 "aq_pp -f+1,eok,qui - -d $yellow_schema_2016_h2  -imp$udbopt demo:yellow"
                ess stream yellow 2017-01 2017-12-31 "aq_pp -f+1,eok,qui - -d $yellow_schema_2017_h1  -imp$udbopt demo:yellow"
}

import () {
                createdb
                import_green
                import_yellow
}
```

A sample “config.inc”

```
:
ISCLOUD=1  # Whether to use clusters
CLUSTERNUM=170 # How many EC2 instances to use
CLUSTERTYPE=c4.large # Specify instance type for clusters
:
```

Free trial

@

https://www.pivotbillions.com/

References

Data source: http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml

Blogs:

http://toddwschneider.com/posts/analyzing-1-1-billion-nyc-taxi-and-uber-trips-with-a-vengeance/

https://www.kdnuggets.com/2017/02/data-science-nyc-taxi-trips.html

https://nycdatascience.com/blog/student-works/analysis-of-nyc-yellow-taxi-data/

https://www.ocf.berkeley.edu/~dlevitt/2015/12/13/final-project-nyc-taxi-and-uber-data/

http://egr.uri.edu/wp-uploads/asee2016/42-150-1-DR.pdf

https://github.com/pavelk2/NYC-taxi-tips