Tableau Day 1

Professor Ernesto Lee

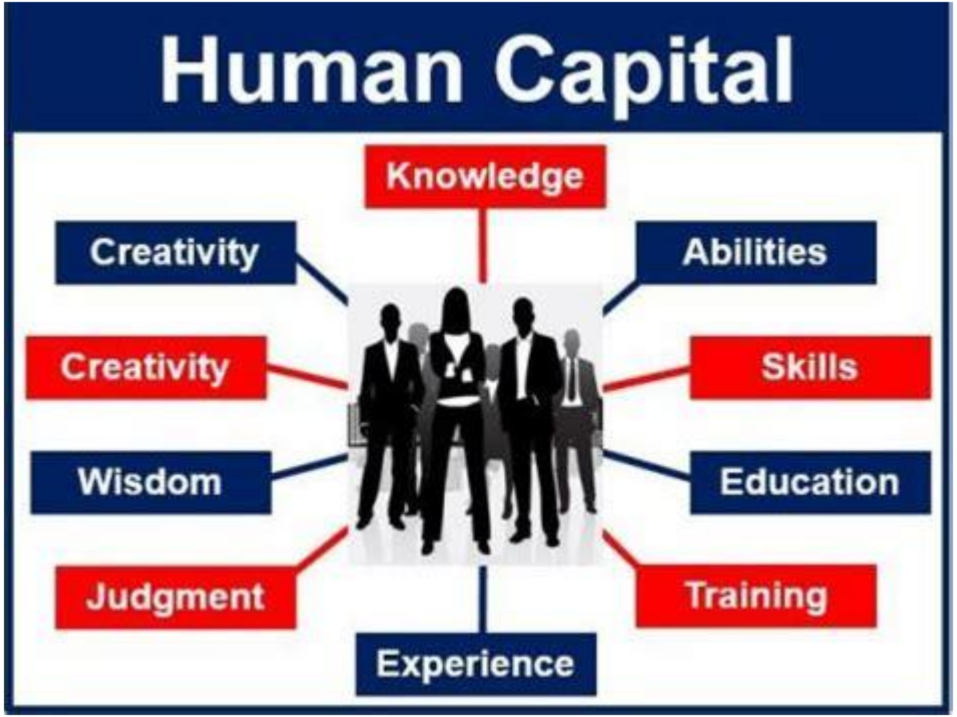
Introduction

Professor Ernesto Lee

ISM 4547: Data Analytics Management

Tooling: Tableau

Human Capital



Openness to Learning

- Skill development
- Job Readiness
- Portfolio Development

You can’t send a Duck to Eagle School

Make sure you are here for the right reasons...

or

Your Journey

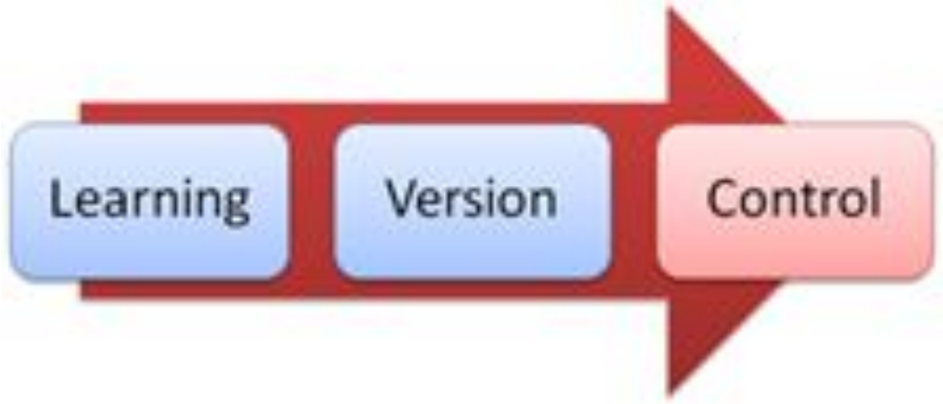
As your Professor... I am more like a coach...

I am here to **inspire** change in a direction. **Not force it.** I am here to tell you where to look but not tell you what to see.

The Syllabus and Expectations

HOW ARE YOU DOING ON YOUR UNSPOKEN OBJECTIVES?
MY WHAT?
Dilbert.com DilbertCartoonist@gmail.com
I'M REFERRING TO THE GOALS I HAVE IN MY MIND THAT I'VE NEVER MENTIONED.
5-29-12 © 2012 Scott Adams, Inc. Dist. by Universal Uclick
HOW ARE THOSE GOING?
I'M TOTALLY NAILING THEM.

Software Version Control



So Why Do We Need Version Control?

- **Backup and Restore.**
- **Synchronization.**
- **Short-term undo.**
- **Long-term undo.**
- **Track Changes**.
- **Track Ownership.**
- **Sandboxing**
- **Branching and merging**.

Half of BEING good… is SOUNDING good...

Basic Setup

- **Repository (repo)**: The database storing the files.
- **Server**: The computer storing the repo.
- **Client**: The computer connecting to the repo.
- **Working Set/Working Copy**: Your local directory of files, where you make changes.
- **Trunk/Main**: The primary location for code in the repo. Think of code as a family tree — the trunk is the main line.

You gotta fake it… to make it...

Basic Actions

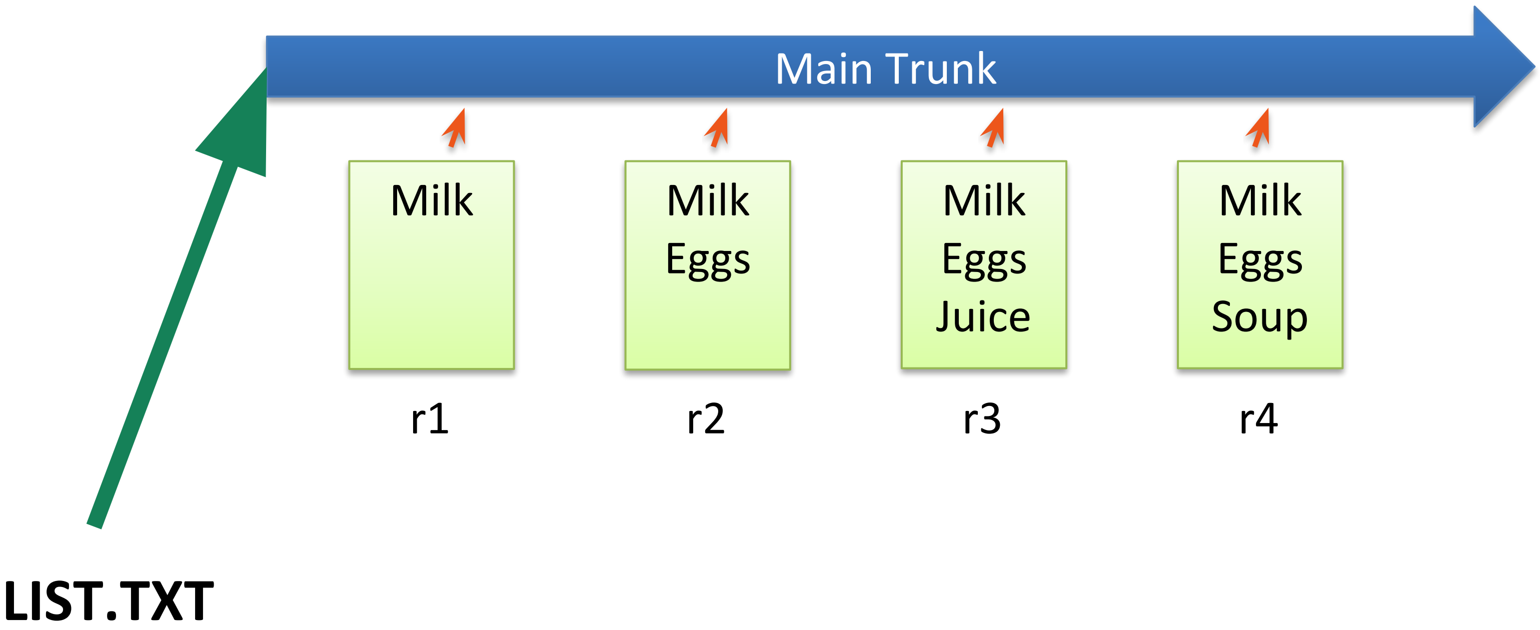
- **Add**: Put a file into the repo for the first time, i.e. begin tracking it with Version Control.
- **Revision**: What version a file is on (v1, v2, v3, etc.).
- **Head**: The latest revision in the repo.
- **Check out**: Download a file from the repo.
- **Check in**: Upload a file to the repository (if it has changed). The file gets a new revision number, and people can “check out” the latest one.
- **Checkin Message**: A short message describing what was changed.
- **Changelog/History**: A list of changes made to a file since it was created.
- **Update/Sync**: Synchronize your files with the latest from the repository. This lets you grab the latest revisions of all files.
- **Revert**: Throw away your local changes and reload the latest version from the repository.

Just Learn the Lingo...

Advanced Actions

- **Branch**: Create a separate copy of a file/folder for private use (bug fixing, testing, etc). Branch is both a verb (“branch the code”) and a noun (“Which branch is it in?”).
- **Diff/Change/Delta**: Finding the differences between two files. Useful for seeing what changed between revisions.
- **Merge (or patch)**: Apply the changes from one file to another, to bring it up-to-date. For example, you can merge features from one branch into another. (At Microsoft this was called Reverse Integrate and Forward Integrate)
- **Conflict**: When pending changes to a file contradict each other (both changes cannot be applied).
- **Resolve**: Fixing the changes that contradict each other and checking in the correct version.
- **Locking**: Taking control of a file so nobody else can edit it until you unlock it. Some version control systems use this to avoid conflicts.
- **Breaking the lock**: Forcibly unlocking a file so you can edit it. It may be needed if someone locks a file and goes on vacation (or “calls in sick” the day Halo 3 comes out).
- **Check out for edit**: Checking out an “editable” version of a file. Some VCSes have editable files by default, others require an explicit command.

Basic Checkins



Main Trunk

Milk

r1

Milk
Eggs

r2

Milk
Eggs
Juice

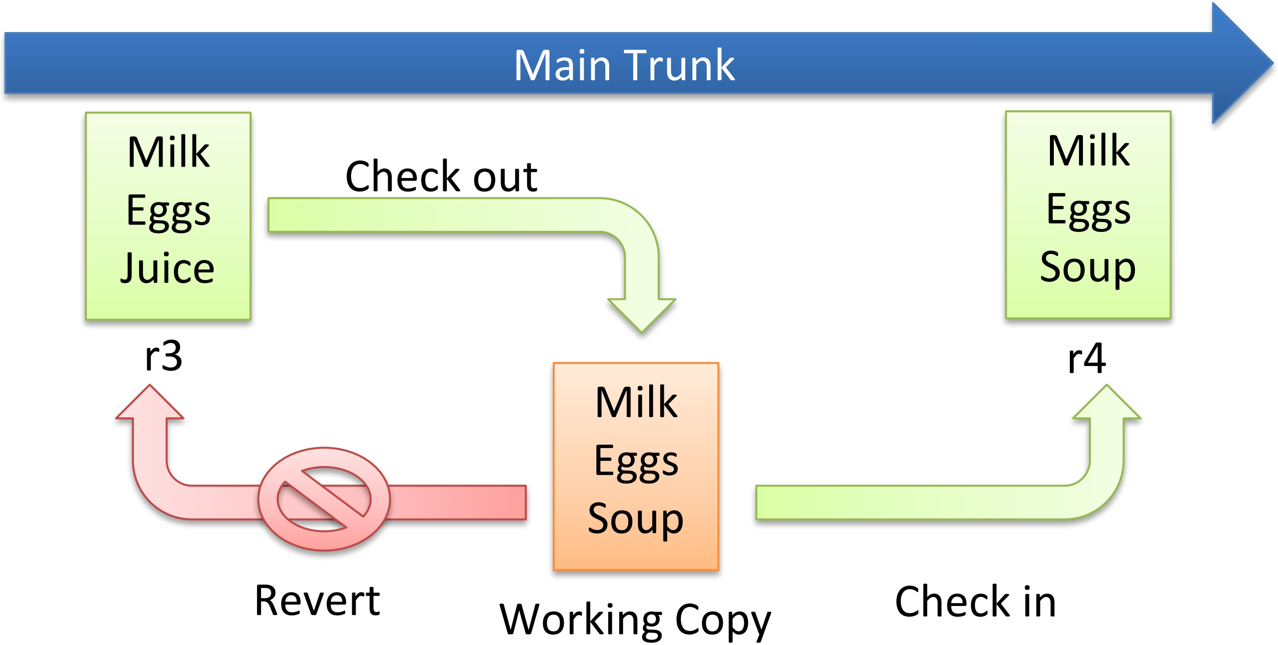
r3

Milk
Eggs
Soup

r4

LIST.TXT

Checkout and Edit



Main Trunk

Milk
Eggs
Juice

r3

Check out

Milk
Eggs
Soup

Working Copy

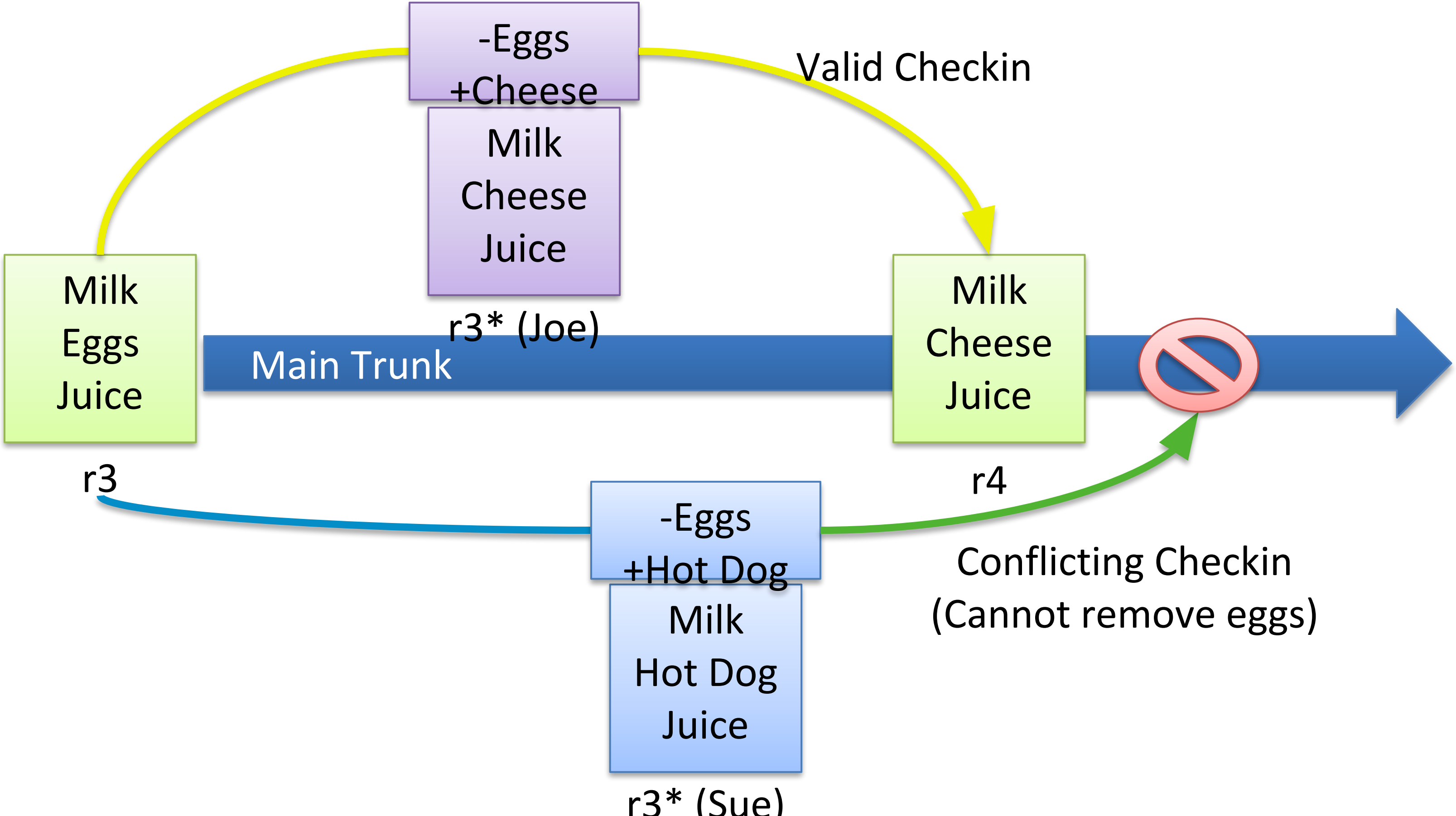
Revert

Check in

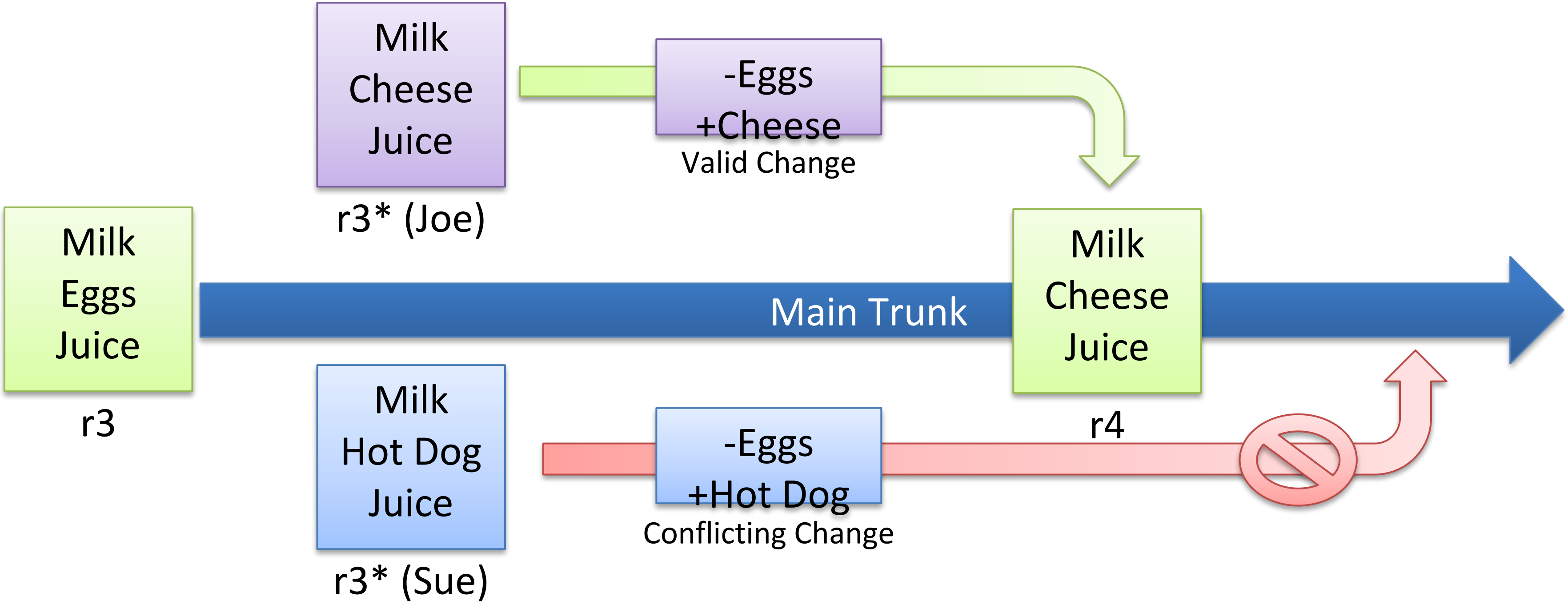
Milk
Eggs
Soup

r4

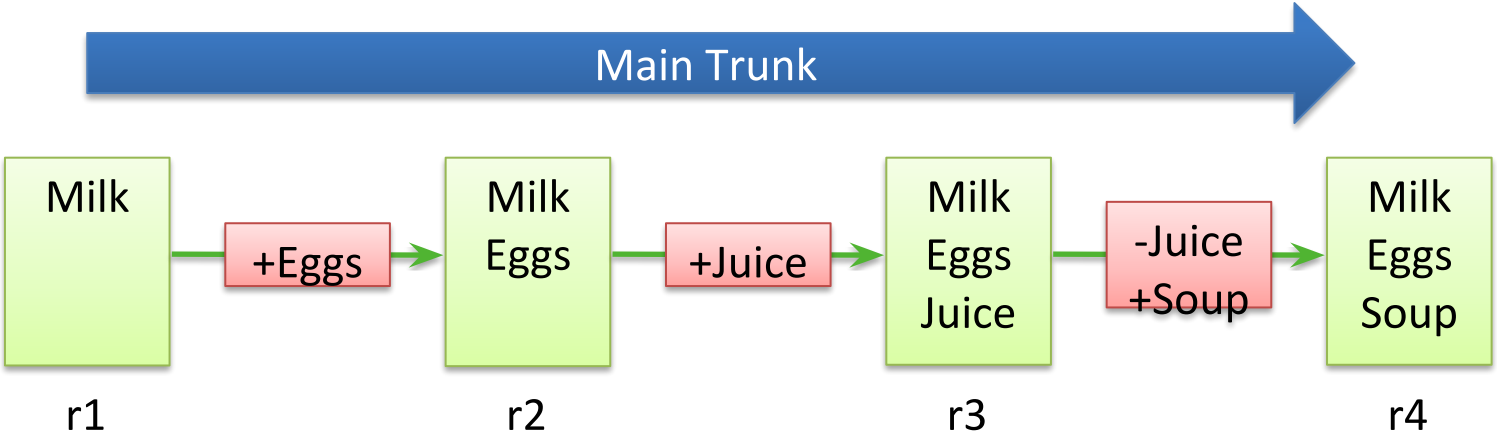
Conflicts


-Eggs
+Cheese
Milk
Cheese
Juice
r3* (Joe)
Valid Checkin
Milk
Eggs
Juice
r3
Main Trunk
Milk
Cheese
Juice
r4
-Eggs
+Hot Dog
Milk
Hot Dog
Juice
r3* (Sue)
Conflicting Checkin
(Cannot remove eggs)

Conflicts (Alt)

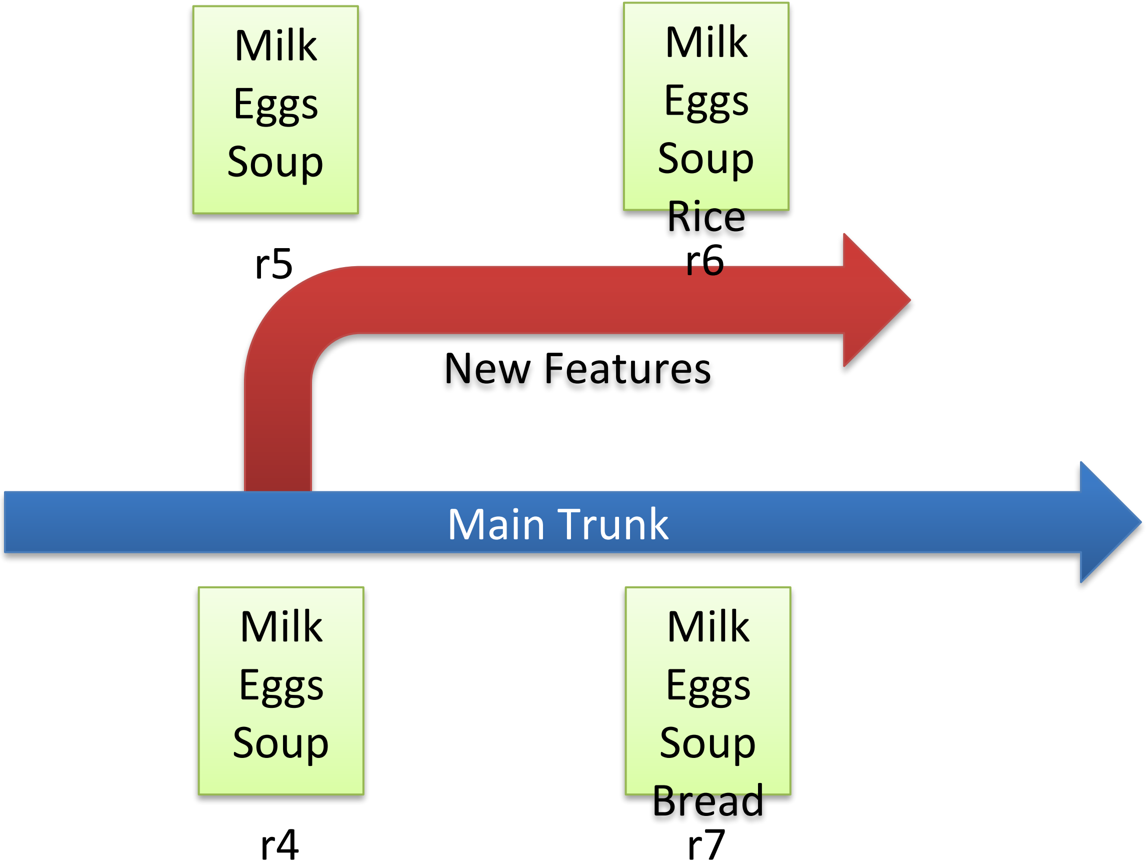


Basic Diffs



NOTE AVAILABLE IN GIT… ONLY IN SVN

Branching



Milk
Eggs
Soup

r5

Milk
Eggs
Soup
Rice

r6

New Features

Main Trunk

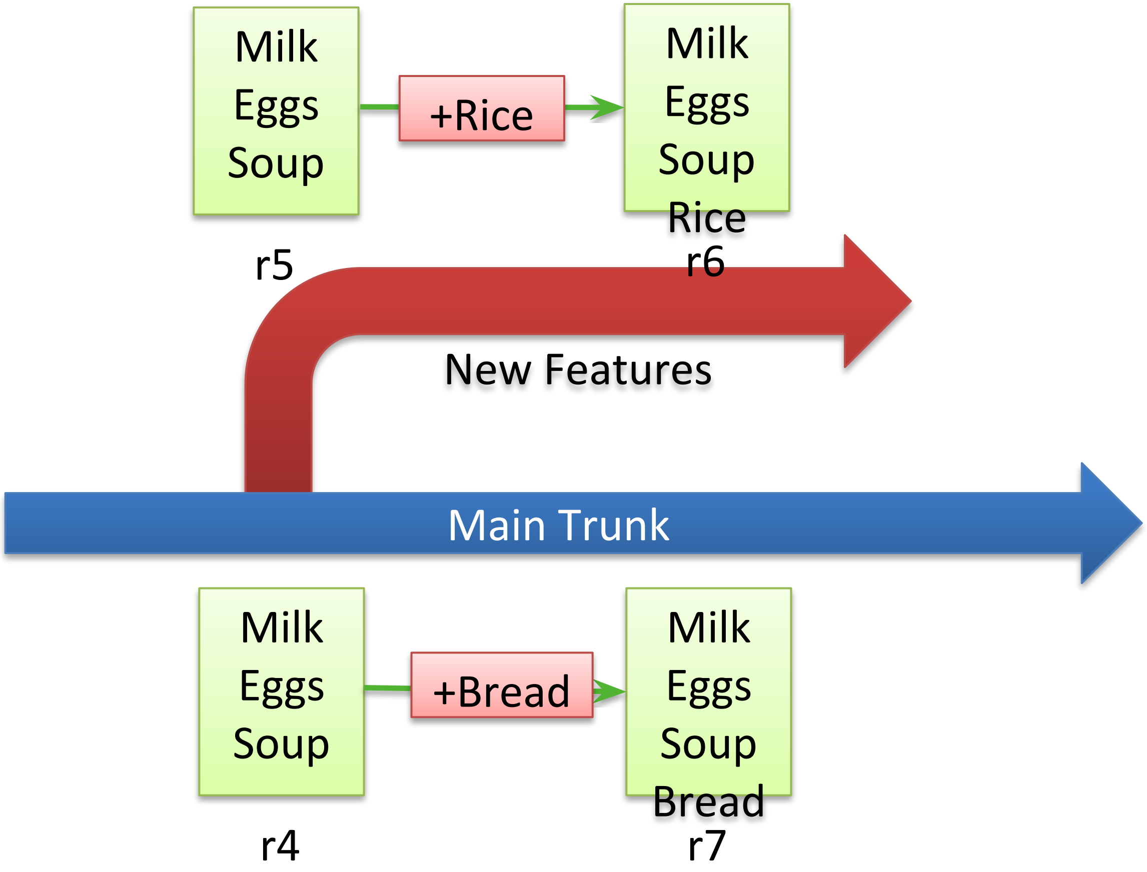
Milk
Eggs
Soup

r4

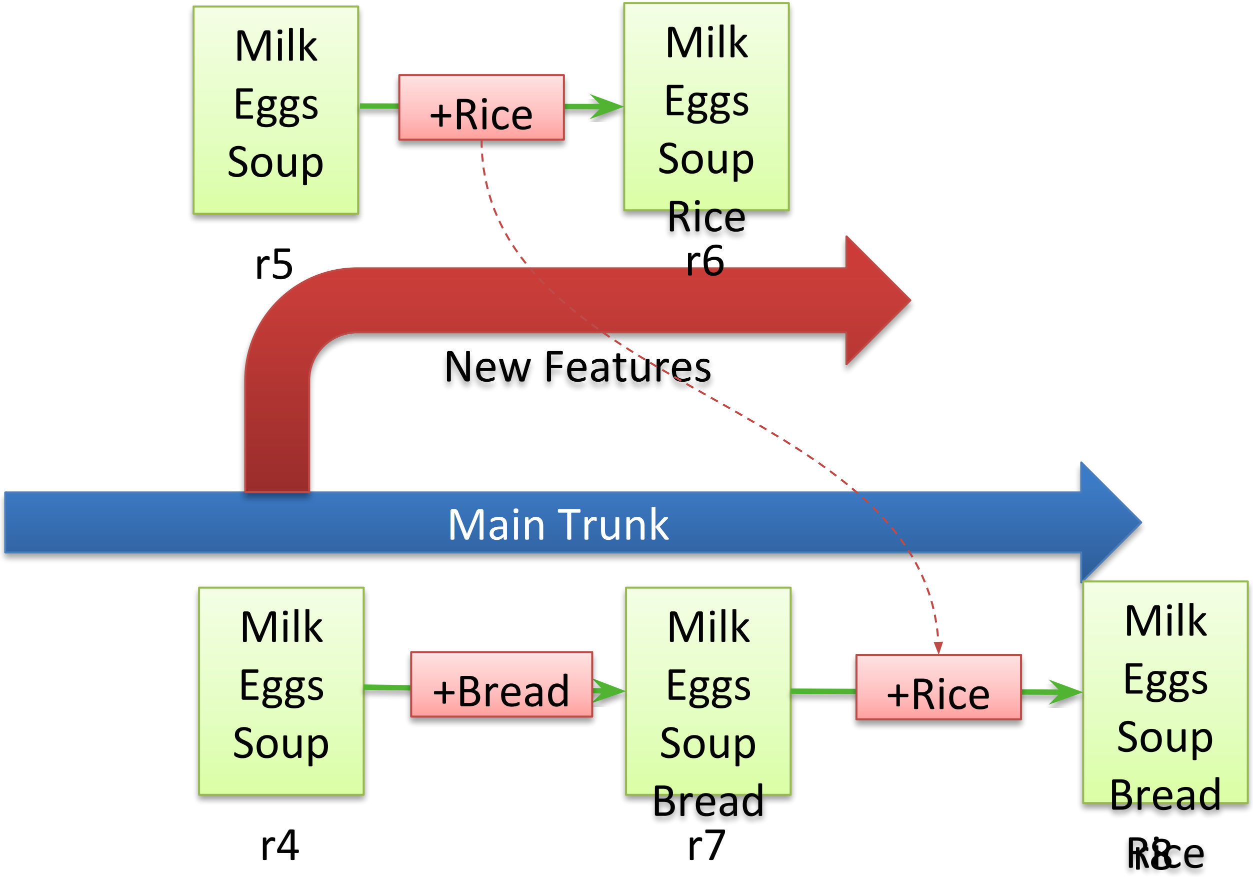
Milk
Eggs
Soup
Bread

r7

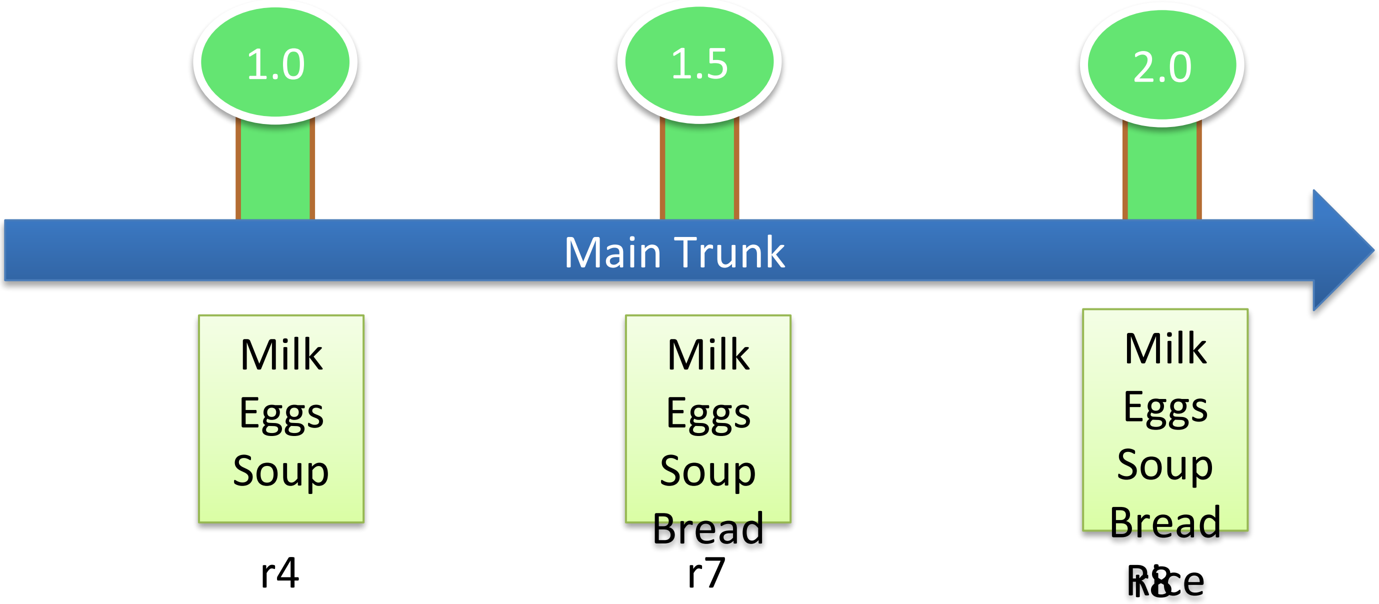
Branch Changes



Merging


Milk
Eggs
Soup
+Rice
Milk
Eggs
Soup
Rice
r5
r6
New Features
Main Trunk
Milk
Eggs
Soup
+Bread
Milk
Eggs
Soup
Bread
+Rice
Milk
Eggs
Soup
Bread
Rice
r4
r7
r8

Tagging



1.0

1.5

2.0

Main Trunk

Milk
Eggs
Soup

r4

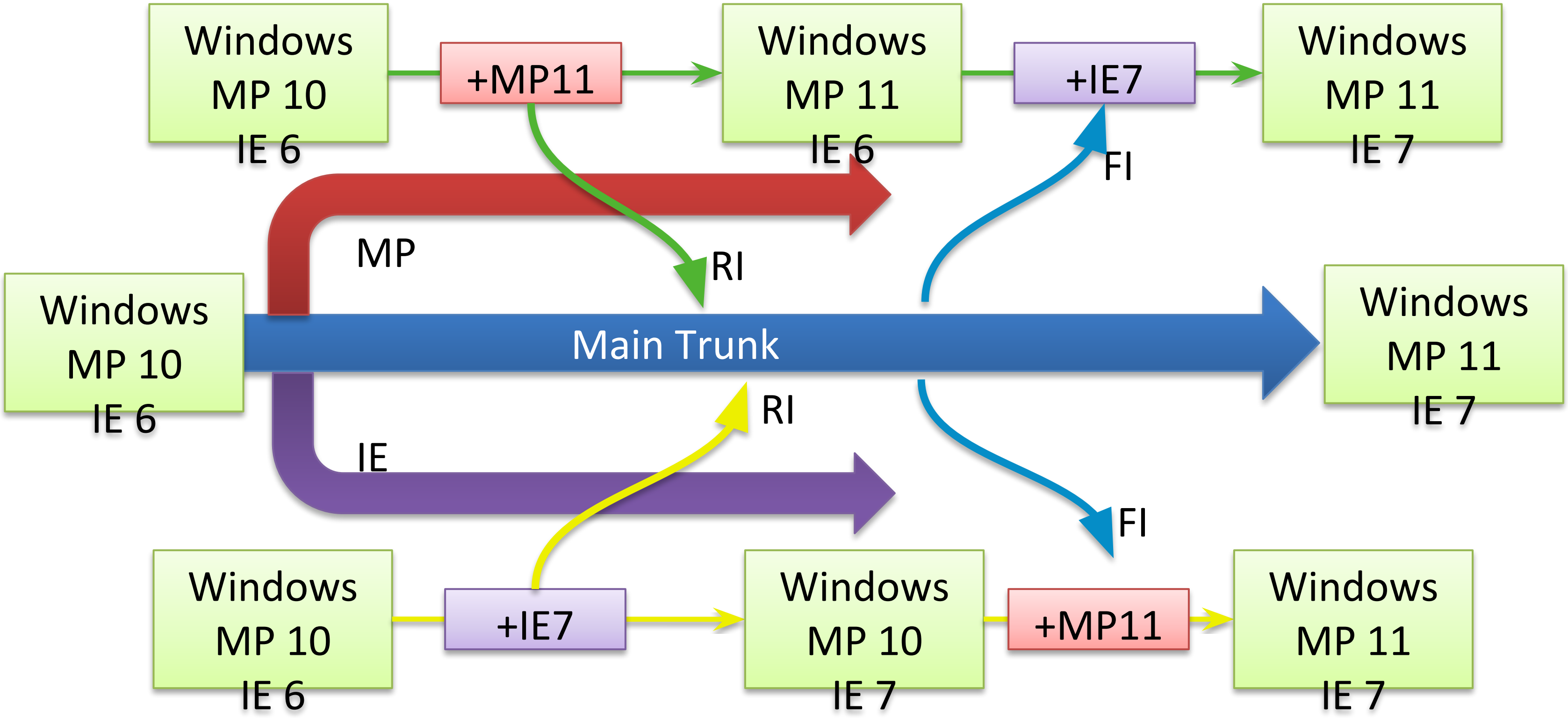
Milk
Eggs
Soup
Bread

r7

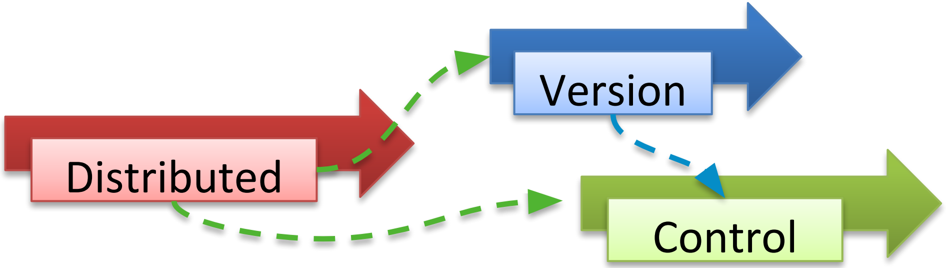
Milk
Eggs
Soup
Bread
Rice

r8

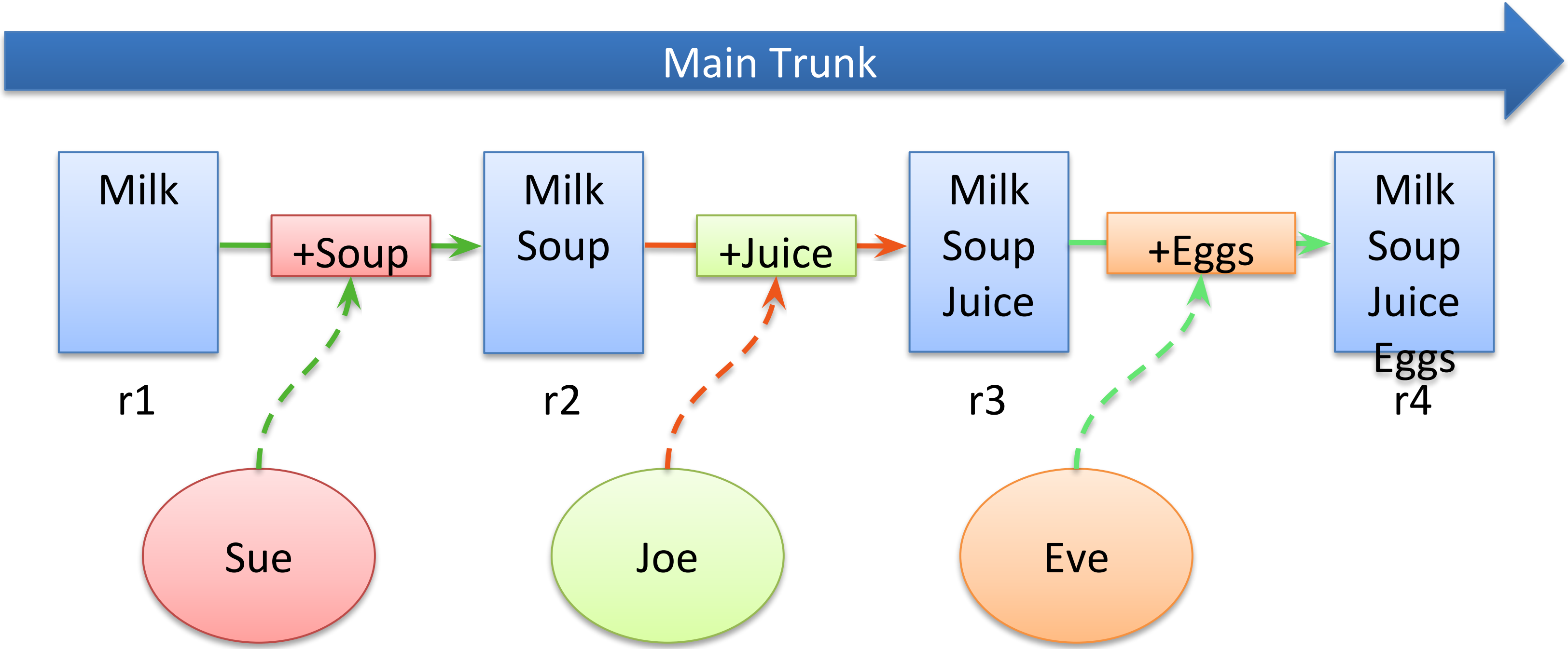
Managing Windows



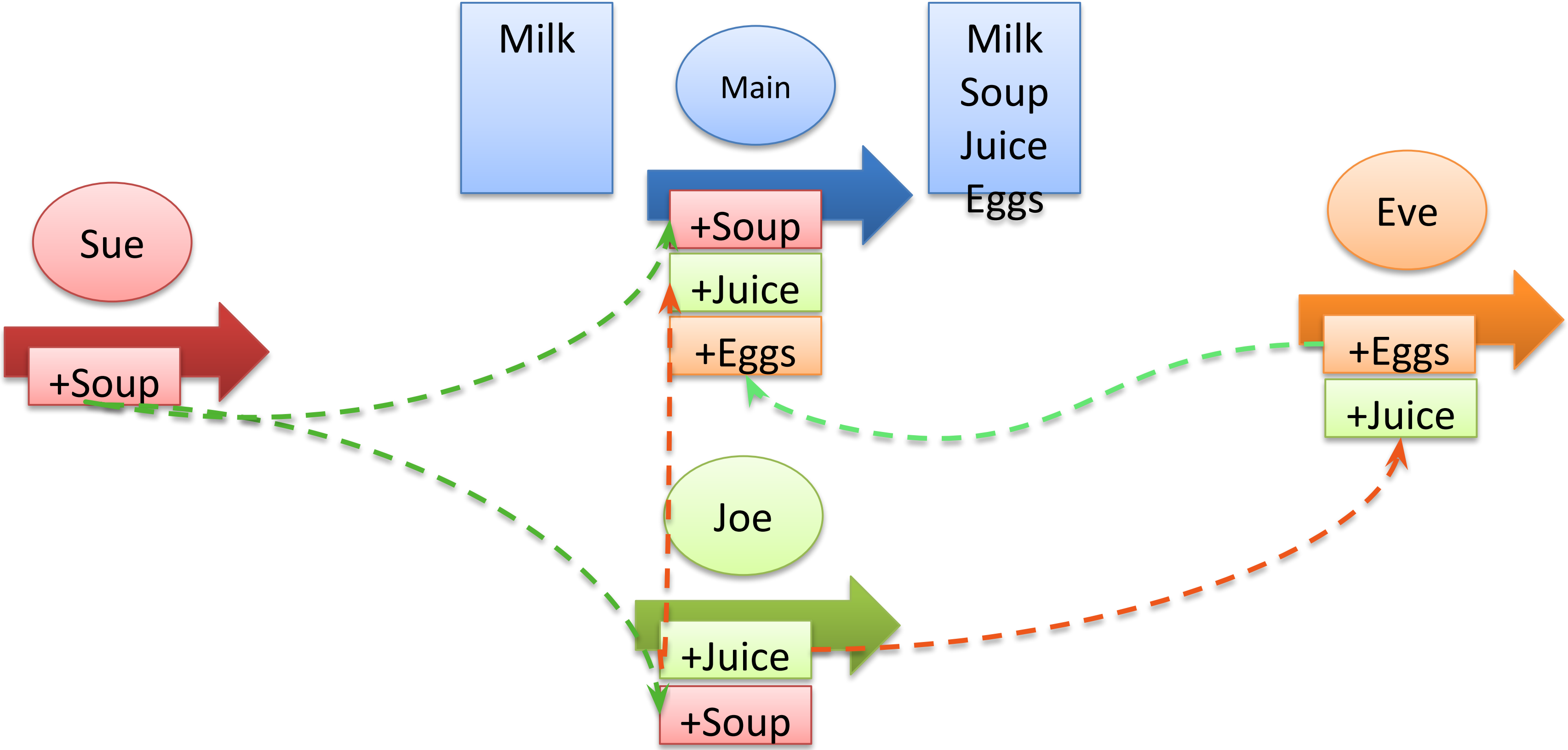
Distributed Version Control



Centralized VCS



Distributed VCS



Core Concepts

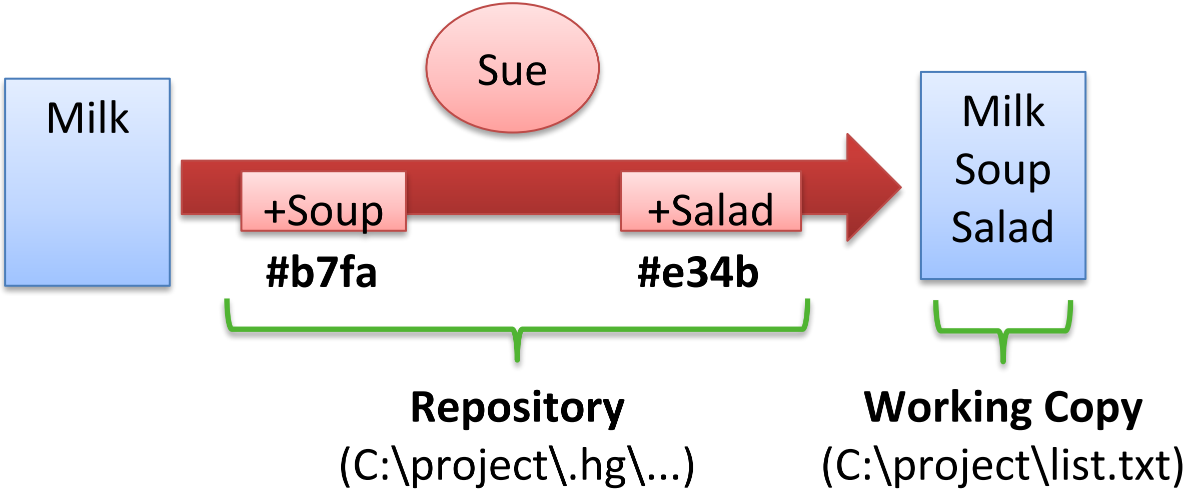
Core Concepts

- Centralized version control focuses on **synchronizing, tracking, and backing up files.**
- Distributed version control focuses on **sharing changes**; every change has a guid or unique id.
- **Recording/Downloading** and **applying** a change are separate steps (in a centralized system, they happen together).
- **Distributed systems have no forced structure**. You can create “centrally administered” locations or keep everyone as peers.

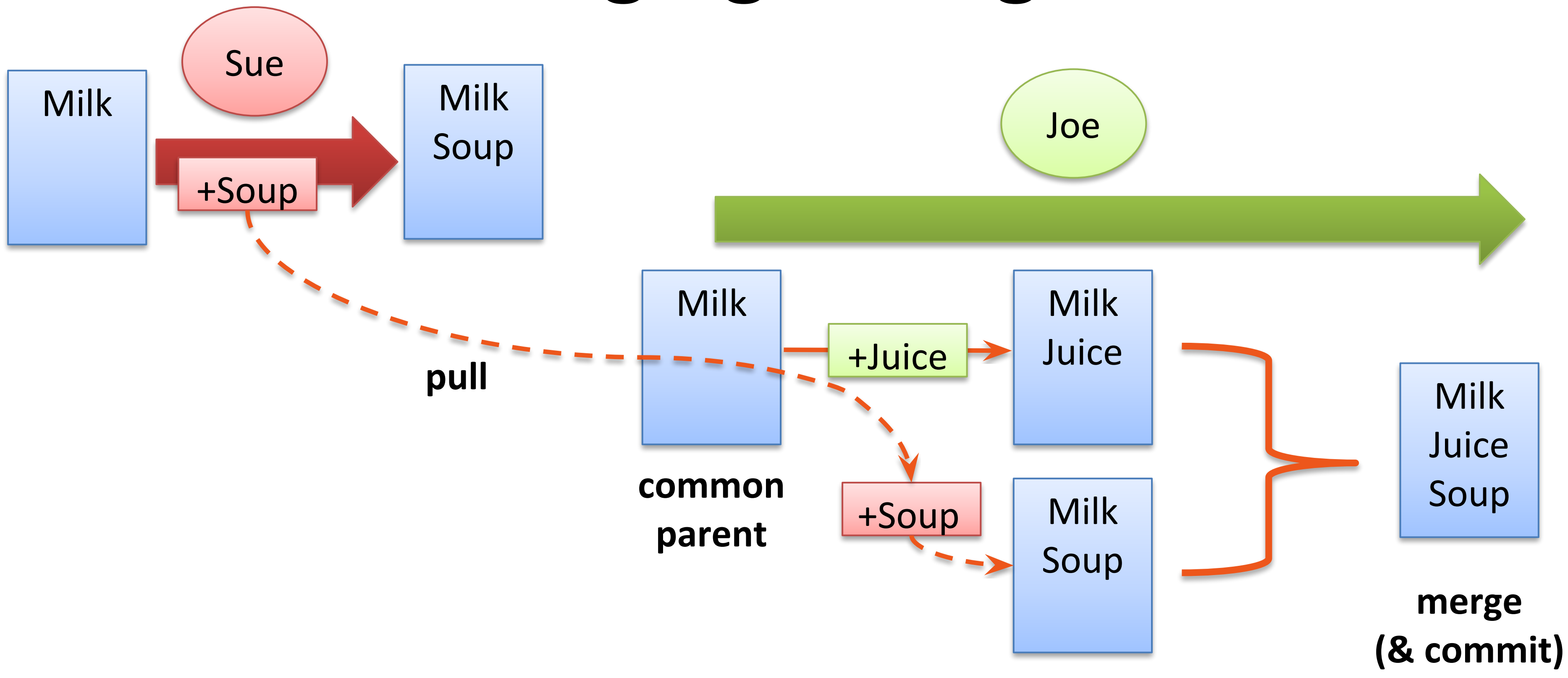
New Terminology

- **push**: send a change to another repository (may require permission)
- **pull**: grab a change from a repository

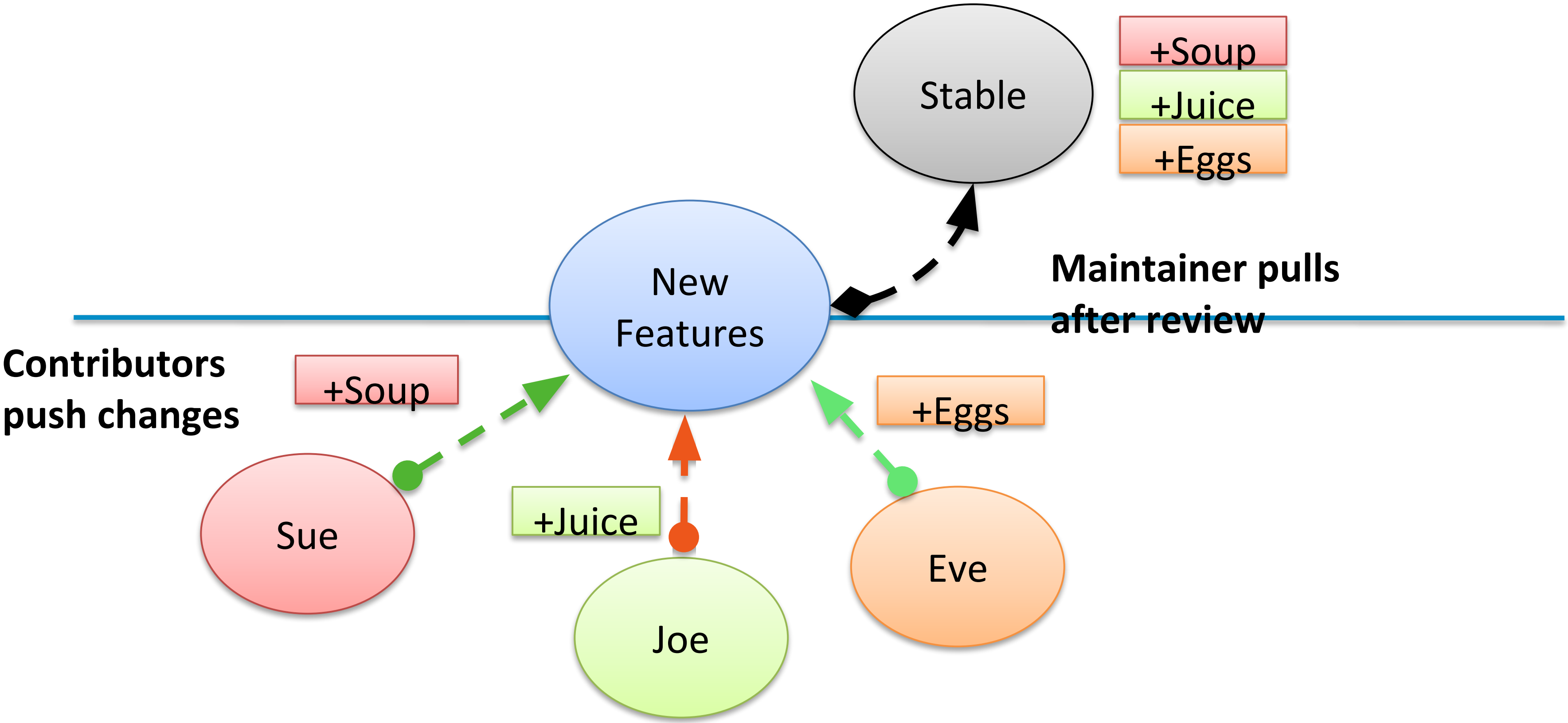
Repository Layout



Merging Changes



Distributed Push/Pull Model



What is the Slingshot?

- **Use version control.**
- **Take it slow.**
- **Keep Learning.**

DEMO

Create your GitHub Account
Create your first Repository
Create your first GitHub WebSite (Bonus)

Git “Aha’s”

Git has a staging area. **Git has a staging area!!!**

Did this ever confuse me. There's both a repo ("object database") and a staging area (called "index"). Checkins have two steps:

- `git add foo.txt`
  - Add foo.txt to the index. It's not checked in yet!
- `git commit -m "message"`
  - Put staged files in the repo; they're now tracked
  - You can "`git add --update`" to stage all tracked, modified files

Branching is a “Save As”

Branches are like "Save as..." on a directory. Best of all:

- Easily merge changes with the original (changes tracked and never applied twice)
- No wasted space (common files only stored once)

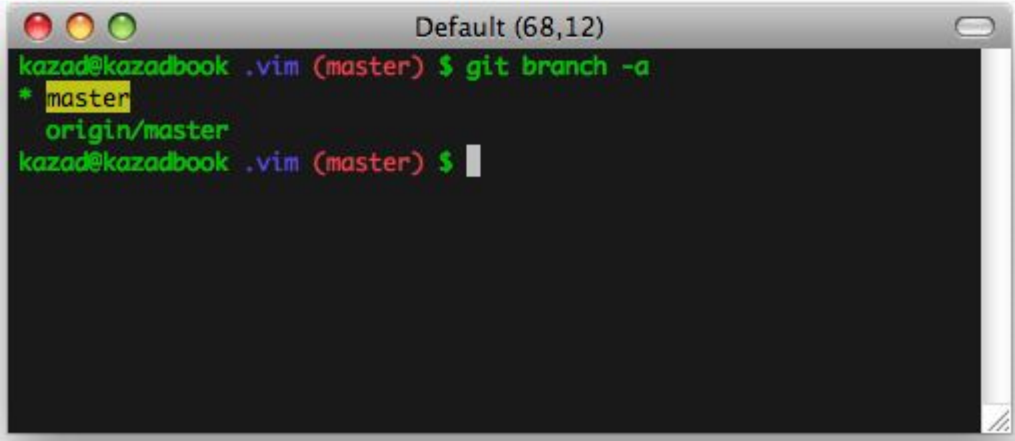
I see branches as "virtual directories" in the .git folder. While inside a physical directory (c:\project or ~/project), you traverse virtual directories with a checkout.

- `git checkout master`
  - switch to master branch ("cd master")
- `git branch dev`
  - create new branch from existing ("cp * dev")
  - you still need to "cd" with "git checkout dev"
- `git merge dev`
  - (when in master) pull in changes from dev ("cp dev/* .")
- `git branch`
  - list all branches ("ls")

Know the Current Branch

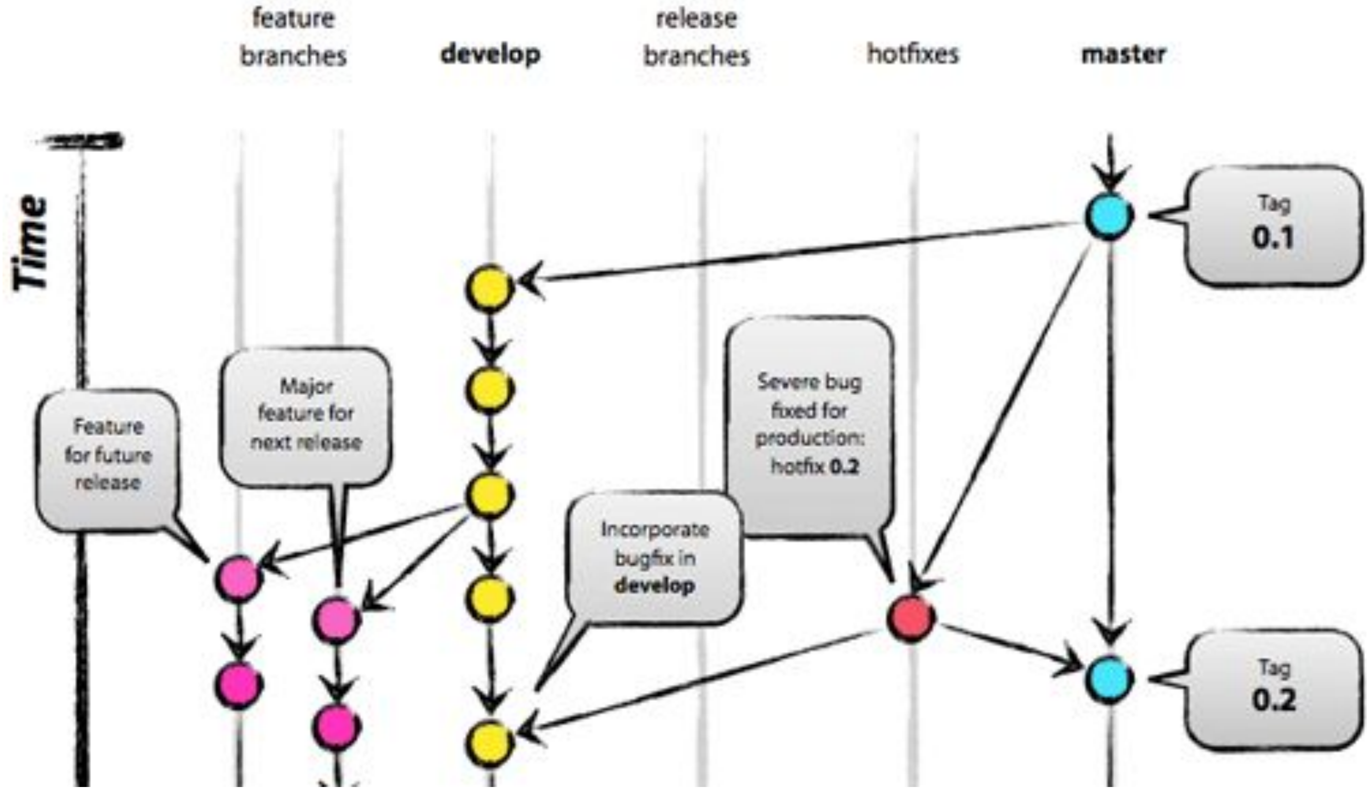
In my .bash_profile:

```
parse_git_branch() {
    git branch 2> /dev/null | sed -e '/^[^*]/d' -e 's/* (.*)/(1)/'
}

export PS1="[33[00m]u@h[33[01;34m] W [33[31m]$(parse_git_branch)

[33[00m]$[33[00m] "
```



```
kazad@kazadbook .vim (master) $ git branch -a
* master
  origin/master
kazad@kazadbook .vim (master) $ 
```

Always Visualize your Branches


feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Major feature for next release
Feature for future release
Severe bug fixed for production: hotfix 0.2
Incorporate bugfix in develop
Tag 0.2

Git has Local AND Remote

Local data

- git init
  - create local repo
  - use git add/commit/branch to work locally

Remote data

- git remote add name path-to-repo
  - track a remote repo (usually "origin") from an existing repo
  - remote branches are "origin/master", "origin/dev" etc.
- git branch -a
  - list all branches (remote and local)
- git clone path-to-repo
  - create a new local git repo copied from a remote one
  - local master tracks remote master
- git pull
  - merge changes from tracked remote branch (if in dev, pull from origin/dev)
- git push
  - send changes to tracked remote branch (if in dev, push to origin/dev)

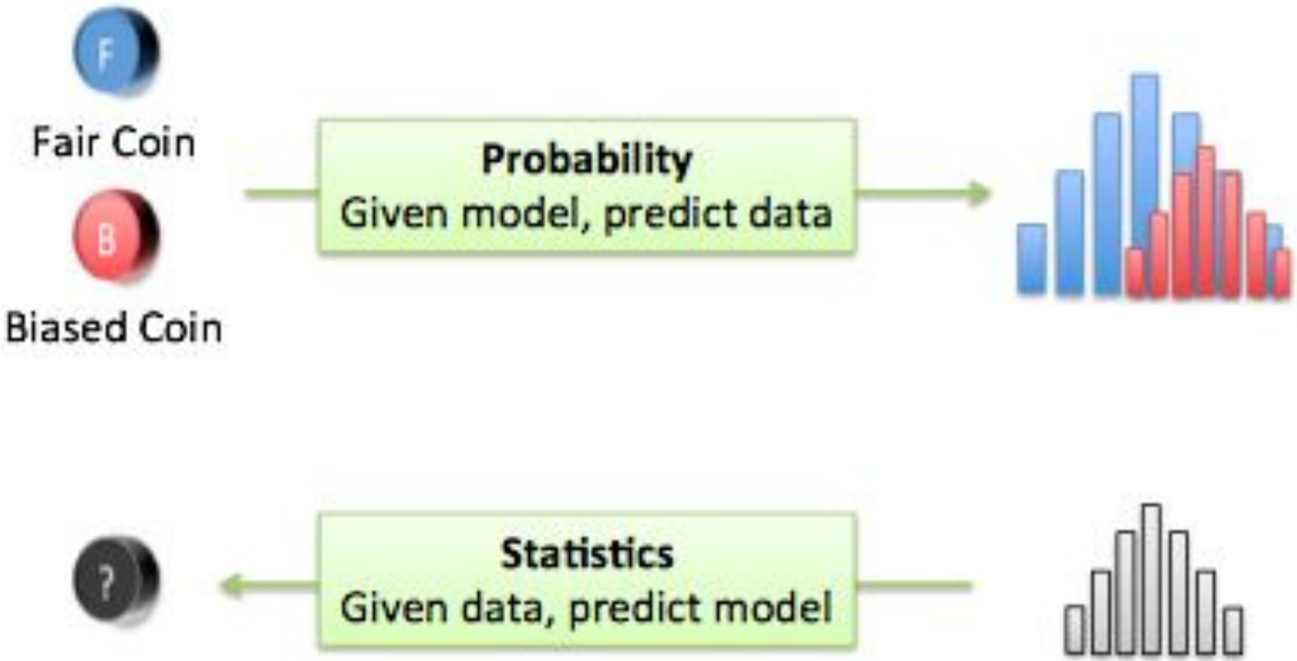
GUIDs

Statistics

Difference Between Probability & Statistics

- **Probability** is starting with an animal, and figuring out what footprints it will make.
- **Statistics** is seeing a footprint, and guessing the animal.

Probability & Statistics



Probability
Given model, predict data

Statistics
Given data, predict model

Find the Animal

Get the tracks.

Measure the basic characteristics.

Find the species.

Look up the specific animal.

Make additional predictions.

Slingshot

"Statistics is the study of the collection, organization, analysis, and interpretation of data".

- What are the most common species? (Common distributions)
- Are new ones being discovered?
- Can we predict the next footprint? (Extrapolation)
- Are the tracks following a path? (Regression / trend line)
- Here's two tracks, which animal was faster? Bigger? (Data from two drug trials: which was more effective?)
- Is one animal moving in the same direction as another? (Correlation)
- Are two animals tracking a common source? (Causation: two bears chasing the same rabbit)

Get the Git

Fork the Repo:

https://github.com/fenago/tableau2019

What is Tableau?

Skills that I want to go over...

Connecting to the data

Building a bar chart using Show Me

Building a text table

Adding filters

Adding color

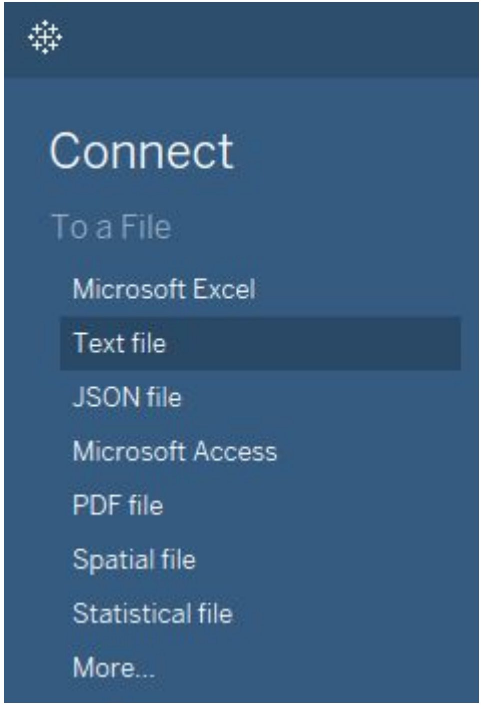
Building a tree map

Building a map

Customizing tooltips

Building a dual axis map

Connecting to Data



Connect

To a File

Microsoft Excel

Text file

JSON file

Microsoft Access

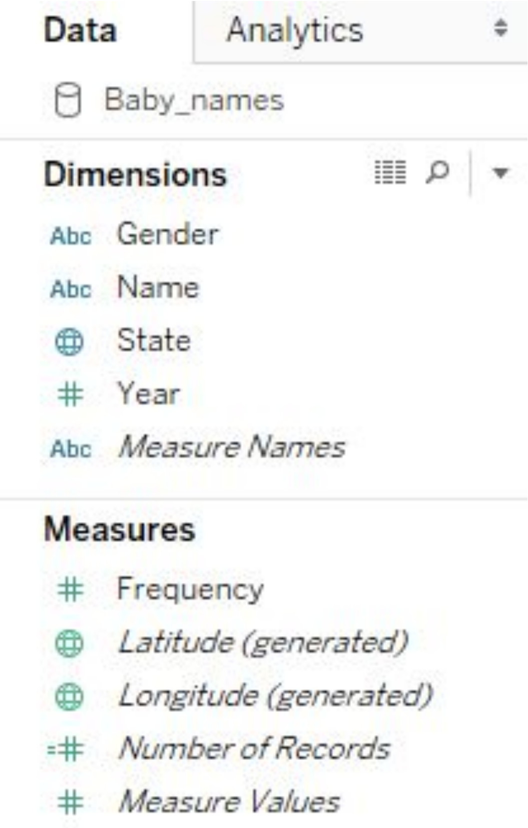
PDF file

Spatial file

Statistical file

More...

What did we do?



Data | Analytics

Baby_names

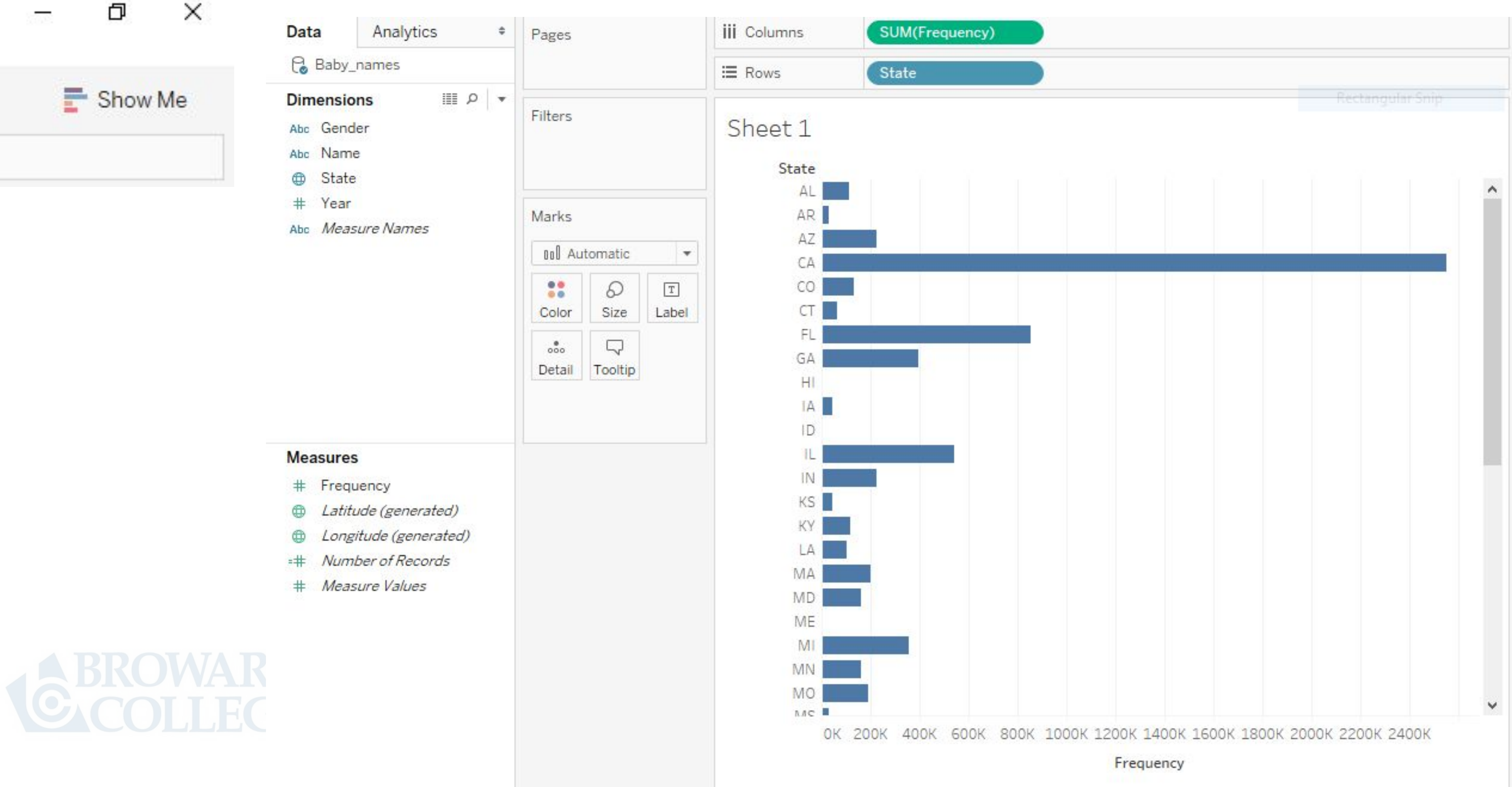
Dimensions

- Abc Gender
- Abc Name
- State
- # Year
- Abc *Measure Names*

Measures

- # Frequency
- *Latitude (generated)*
- *Longitude (generated)*
- =# *Number of Records*
- # *Measure Values*

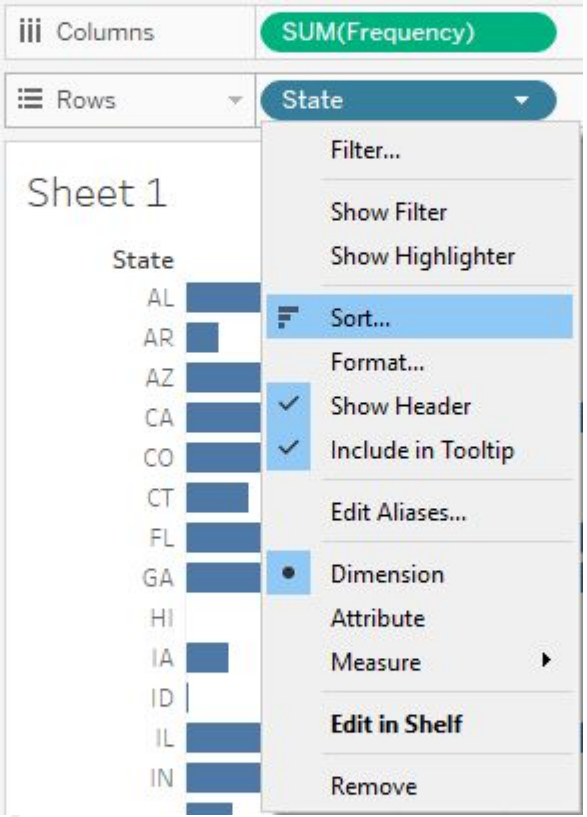
Build a Bar Chart



Columns: SUM(Frequency)

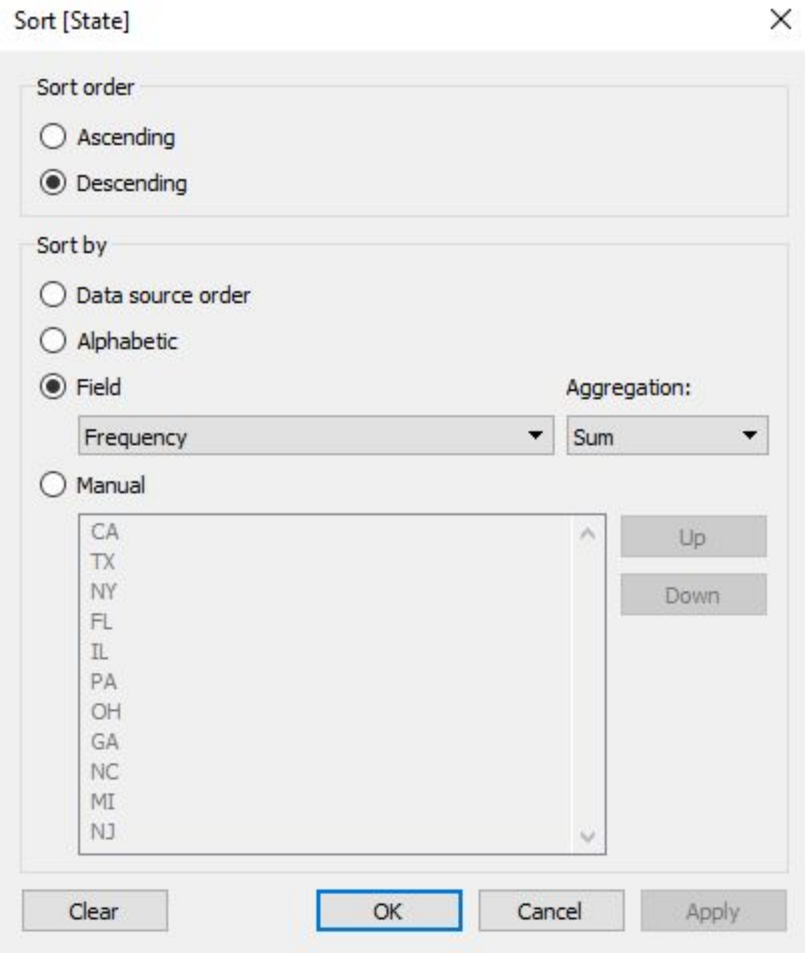
Rows: State

Sorting



Columns: SUM(Frequency)

Rows: State



Sort [State]

Sort order

○ Ascending

● Descending

Sort by

○ Data source order

○ Alphabetic

● Field

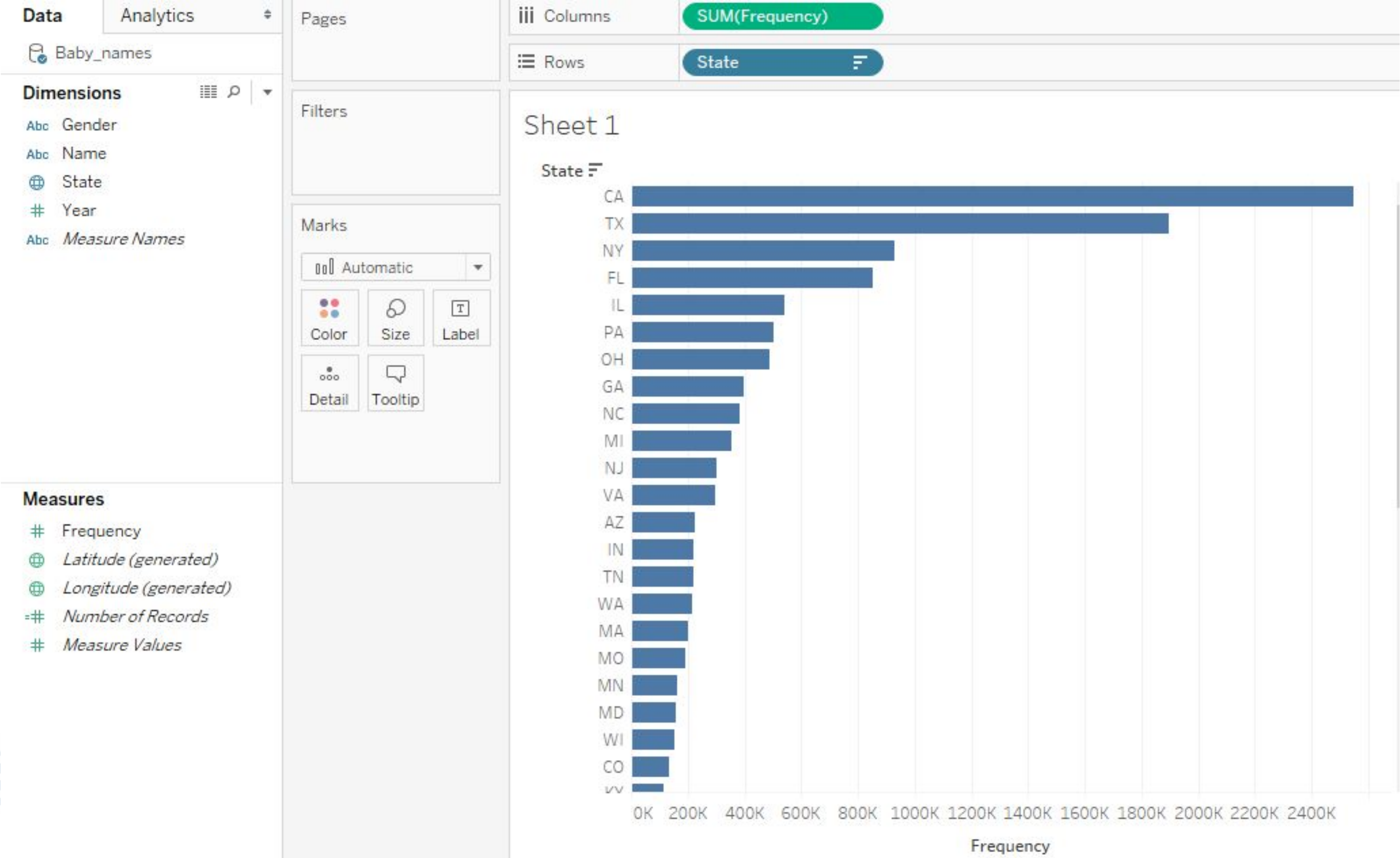
Aggregation:

Frequency

Sum

○ Manual

CA
TX
NY
FL
IL
PA
OH
GA
NC
MI
NJ



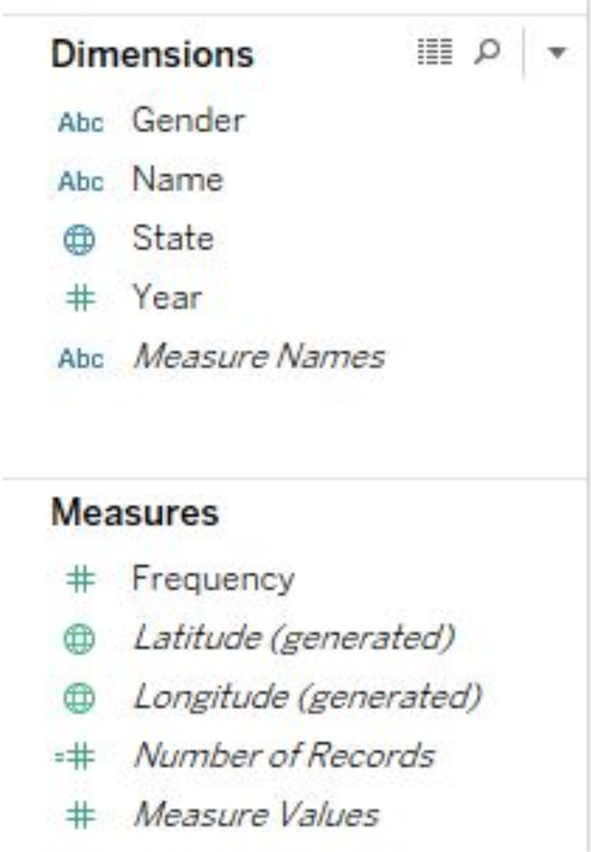
Columns: SUM(Frequency)

Rows: State

What did we do?

Dimensions: This section includes qualitative and categorical values, such as dates, strings, or geographical data

Measures: This section includes quantitative and numerical values



Dimensions

Abc Gender

Abc Name

State

# Year

Abc *Measure Names*

Measures

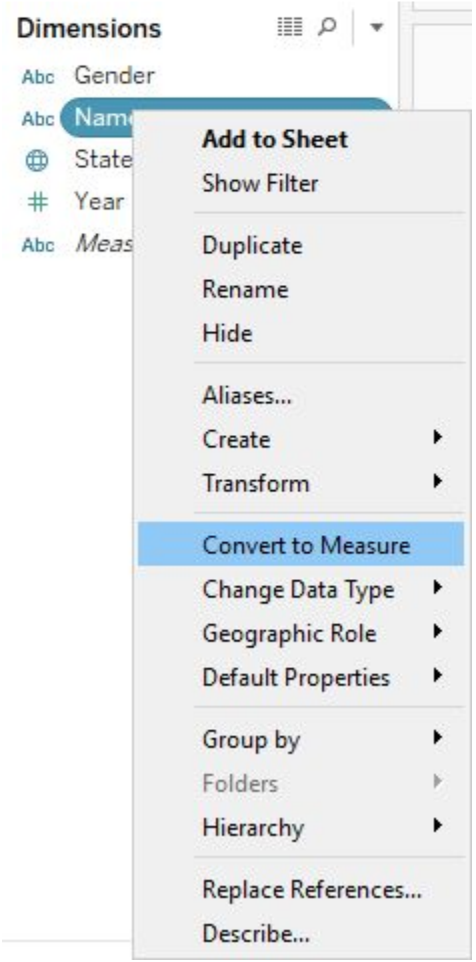
# Frequency

Latitude (generated)

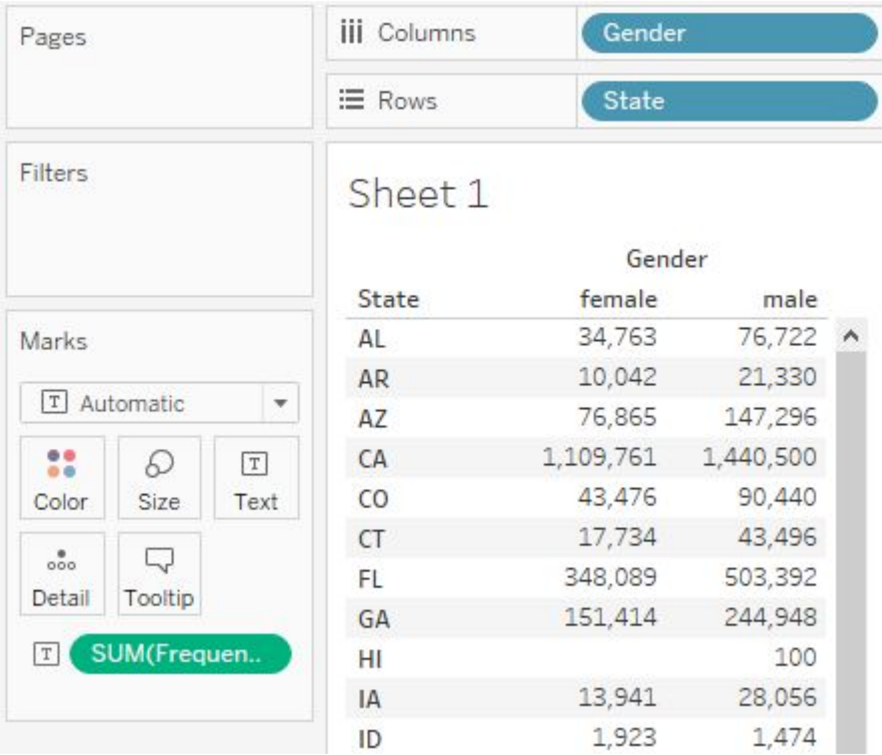
Longitude (generated)

=# *Number of Records*

# *Measure Values*


Dimensions
Abc Gender
Abc Name
State
Year
Abc Meas
Add to Sheet
Show Filter
Duplicate
Rename
Hide
Aliases...
Create
Transform
Convert to Measure
Change Data Type
Geographic Role
Default Properties
Group by
Folders
Hierarchy
Replace References...
Describe...

Text Tables



Columns: Gender

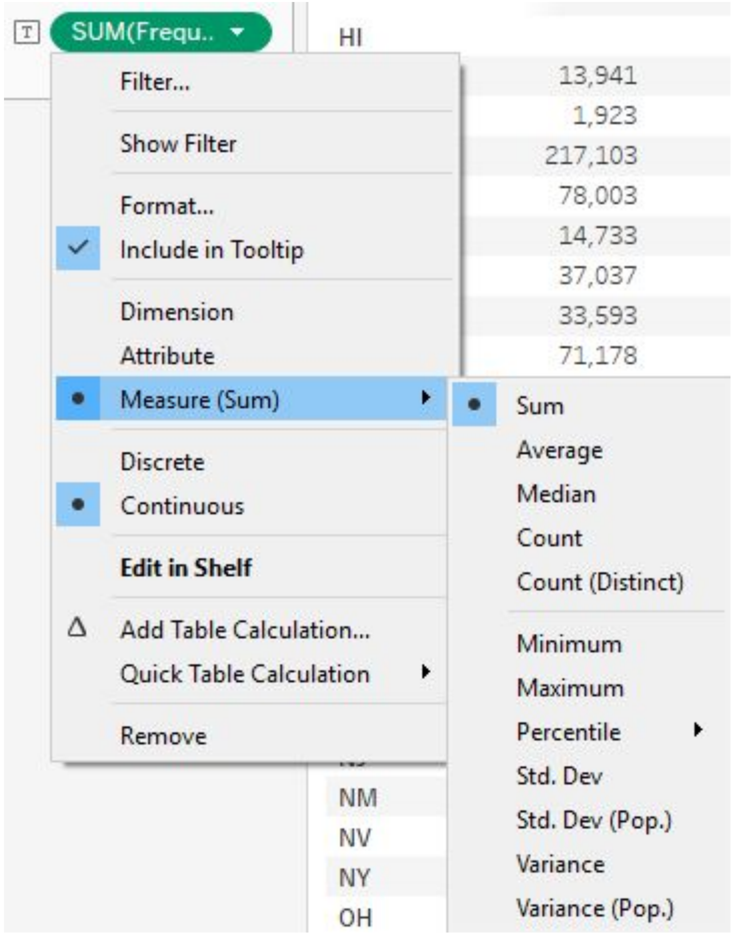
Rows: State

SUM(Frequen..

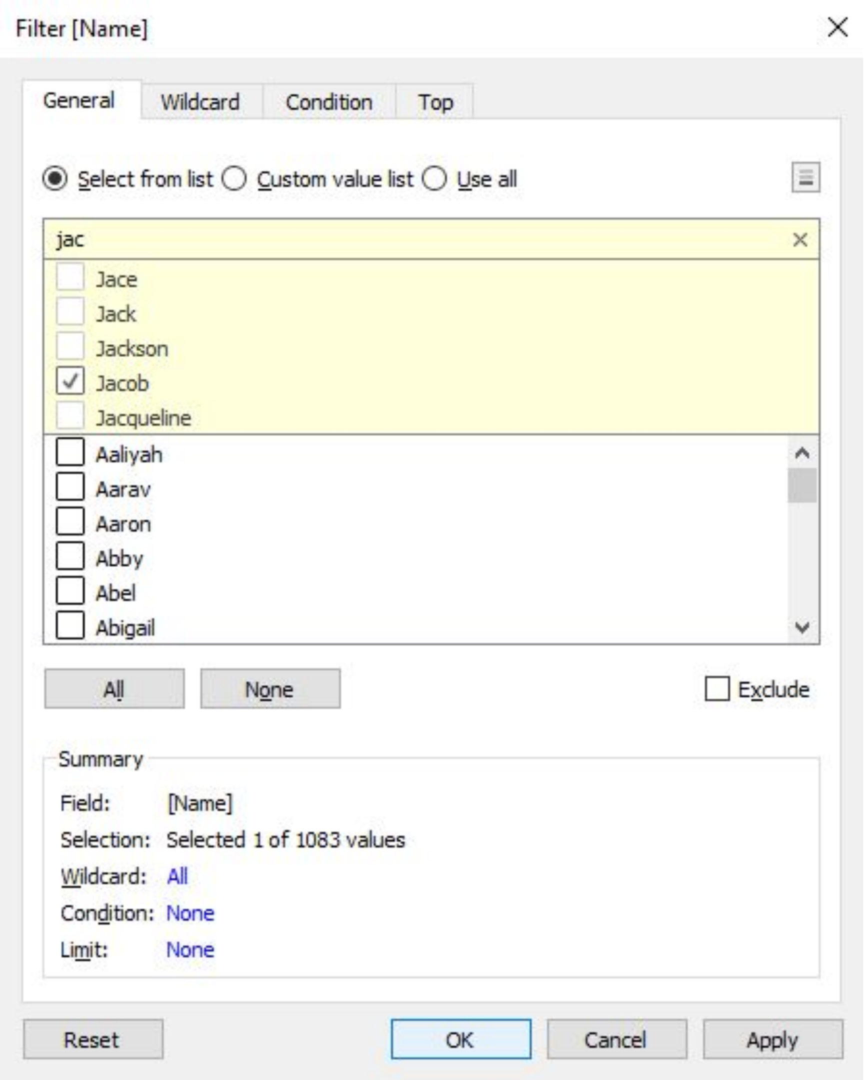
Sheet 1

	Gender	
State	female	male
AL	34,763	76,722
AR	10,042	21,330
AZ	76,865	147,296
CA	1,109,761	1,440,500
CO	43,476	90,440
CT	17,734	43,496
FL	348,089	503,392
GA	151,414	244,948
HI		100
IA	13,941	28,056
ID	1,923	1,474

Summing...


SUM(Frequ..
HI
13,941
1,923
217,103
78,003
14,733
37,037
33,593
71,178
Filter...
Show Filter
Format...
✓ Include in Tooltip
Dimension
Attribute
● Measure (Sum)
Discrete
● Continuous
Edit in Shelf
Δ Add Table Calculation...
Quick Table Calculation
Remove
● Sum
Average
Median
Count
Count (Distinct)
Minimum
Maximum
Percentile
Std. Dev
Std. Dev (Pop.)
Variance
Variance (Pop.)
NM
NV
NY
OH

Filters



● Select from list ○ Custom value list ○ Use all

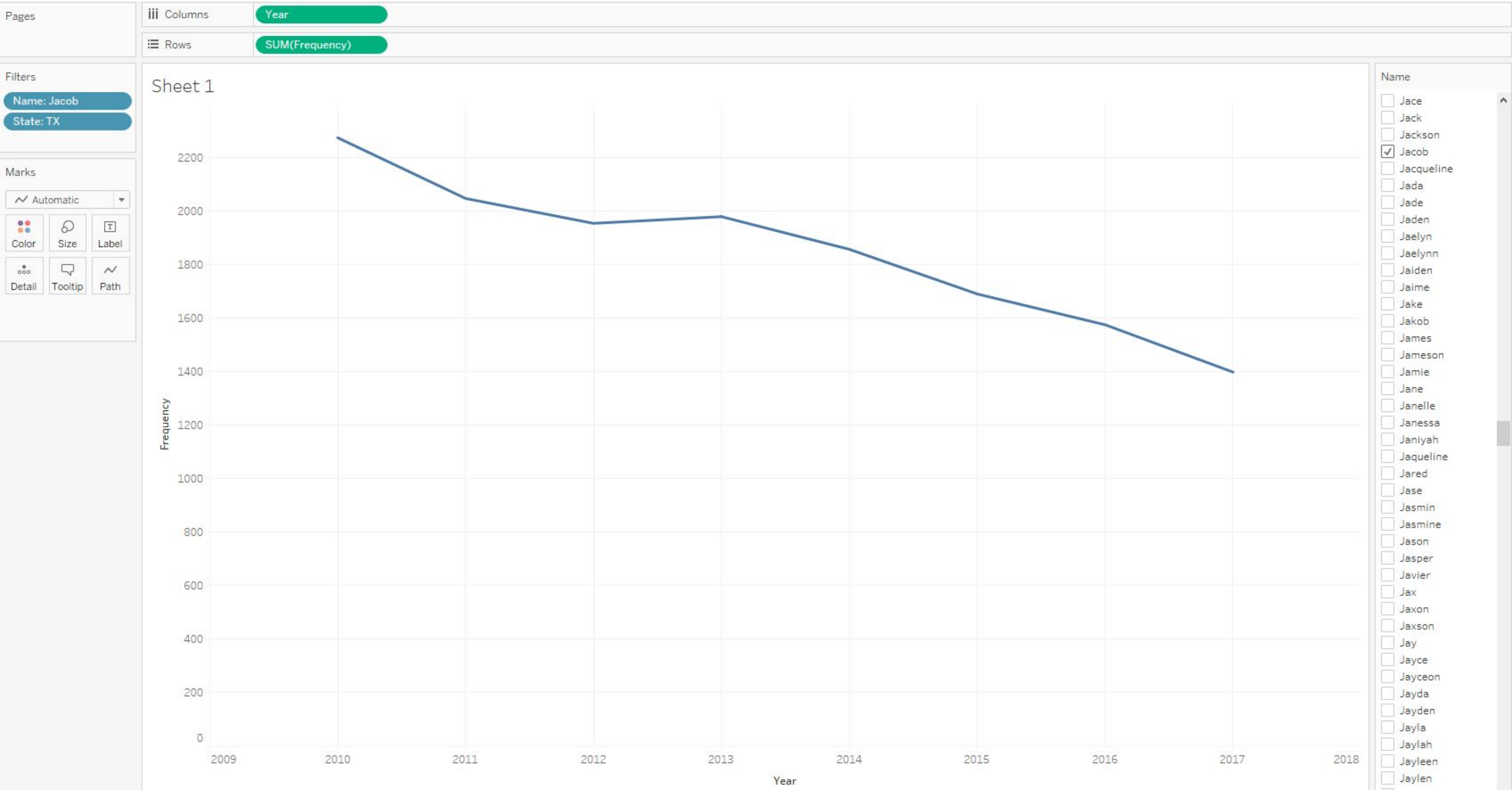
jac

- □ Jace
- □ Jack
- □ Jackson
- ☑ Jacob
- □ Jacqueline
- □ Aaliyah
- □ Aarav
- □ Aaron
- □ Abby
- □ Abel
- □ Abigail

□ Exclude

Summary

Field:	[Name]
Selection:	Selected 1 of 1083 values
Wildcard:	All
Condition:	None
Limit:	None



Columns: Year

Rows: SUM(Frequency)

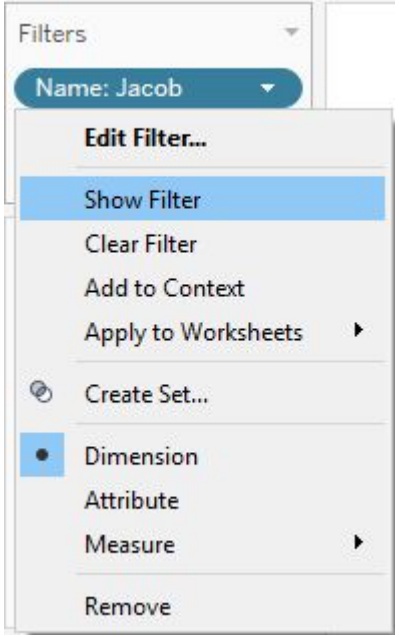
Filters:
Name: Jacob
State: TX

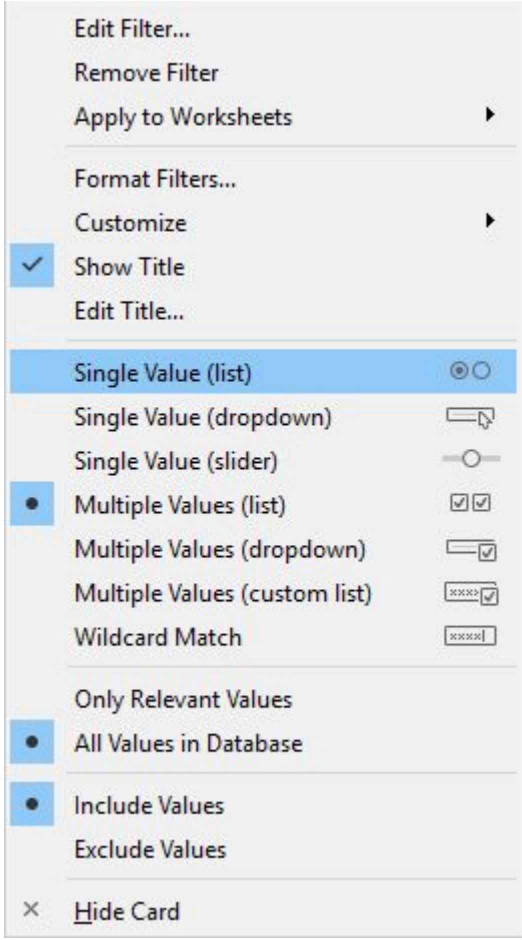
Sheet 1

Name

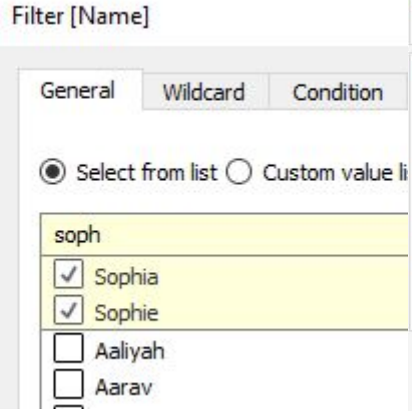
□ Jace
□ Jack
□ Jackson
☑ Jacob
□ Jacqueline
□ Jada
□ Jade
□ Jaden
□ Jaelyn
□ Jaelynn
□ Jaiden
□ Jaime
□ Jake
□ Jakob
□ James
□ Jameson
□ Jamie
□ Jane
□ Janelle
□ Janessa
□ Janiyah
□ Jaqueline
□ Jared
□ Jase
□ Jasmin
□ Jasmine
□ Jason
□ Jasper
□ Javier
□ Jax
□ Jaxon
□ Jaxson
□ Jay
□ Jayce
□ Jayceon
□ Jayda
□ Jayden
□ Jayla
□ Jaylah
□ Jayleen
□ Jaylen

Filtering...


Filters
Name: Jacob
Edit Filter...
Show Filter
Clear Filter
Add to Context
Apply to Worksheets
Create Set...
● Dimension
Attribute
Measure
Remove


Edit Filter...
Remove Filter
Apply to Worksheets
Format Filters...
Customize
✓ Show Title
Edit Title...
Single Value (list)
Single Value (dropdown)
Single Value (slider)
● Multiple Values (list)
Multiple Values (dropdown)
Multiple Values (custom list)
Wildcard Match
Only Relevant Values
● All Values in Database
● Include Values
Exclude Values
× Hide Card

Color...



Filter [Name]

General | Wildcard | Condition

● Select from list ○ Custom value li

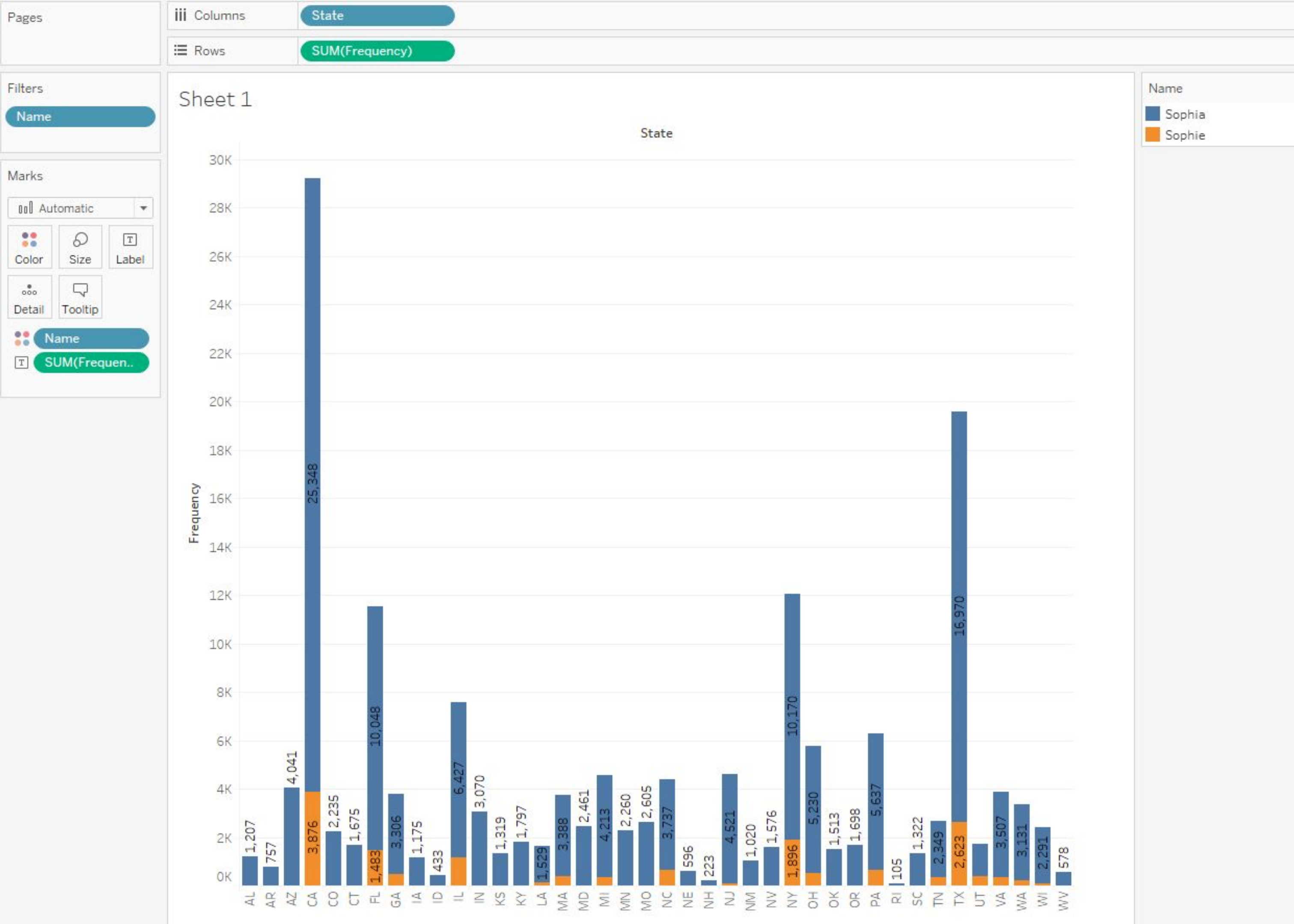
soph

☑ Sophia

☑ Sophie

□ Aaliyah

□ Aarav



Pages

Columns: State

Rows: SUM(Frequency)

Filters: Name

Marks: Automatic

Name

SUM(Frequen..

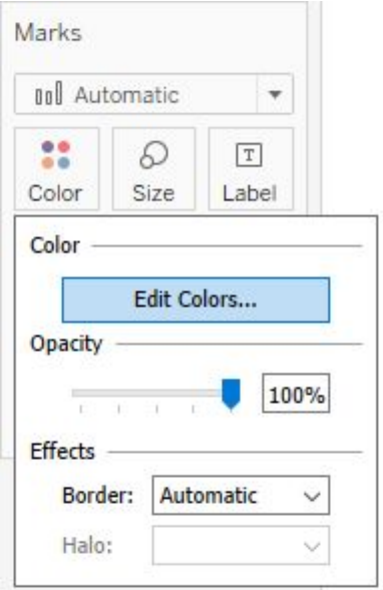
Sheet 1

Name

🟦 Sophia

🟧 Sophie

Color



Marks

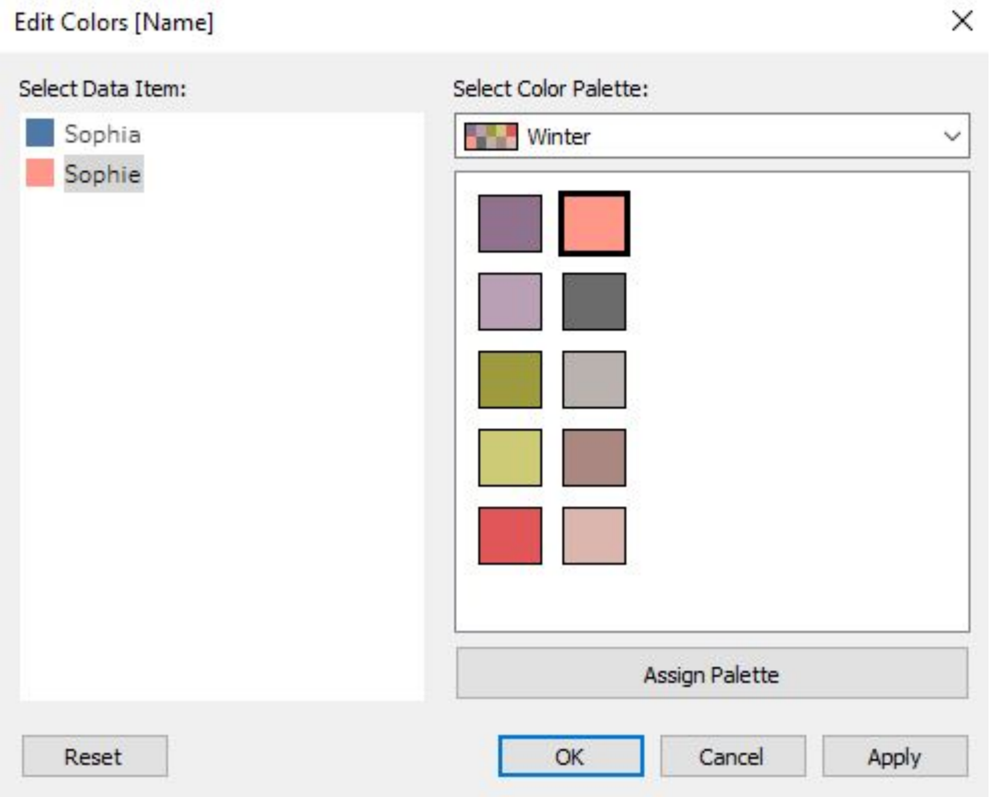
Automatic

Opacity: 100%

Effects

Border: Automatic

Halo:



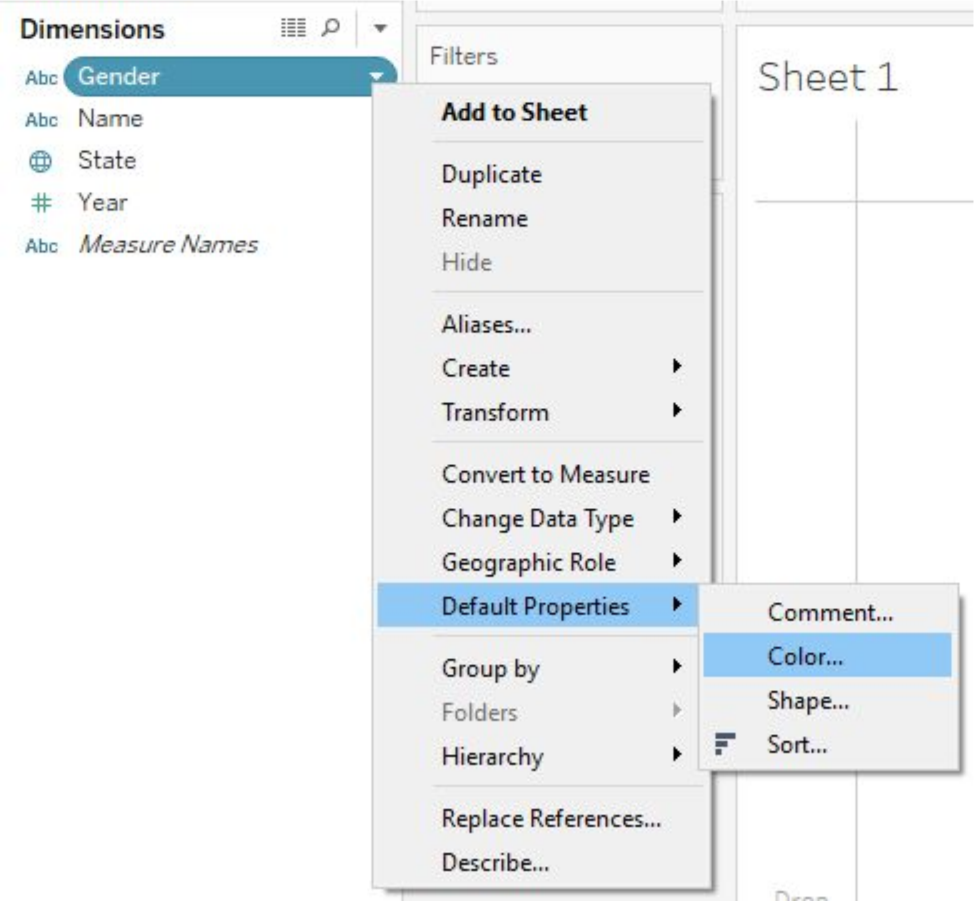
Edit Colors [Name]

Select Data Item:

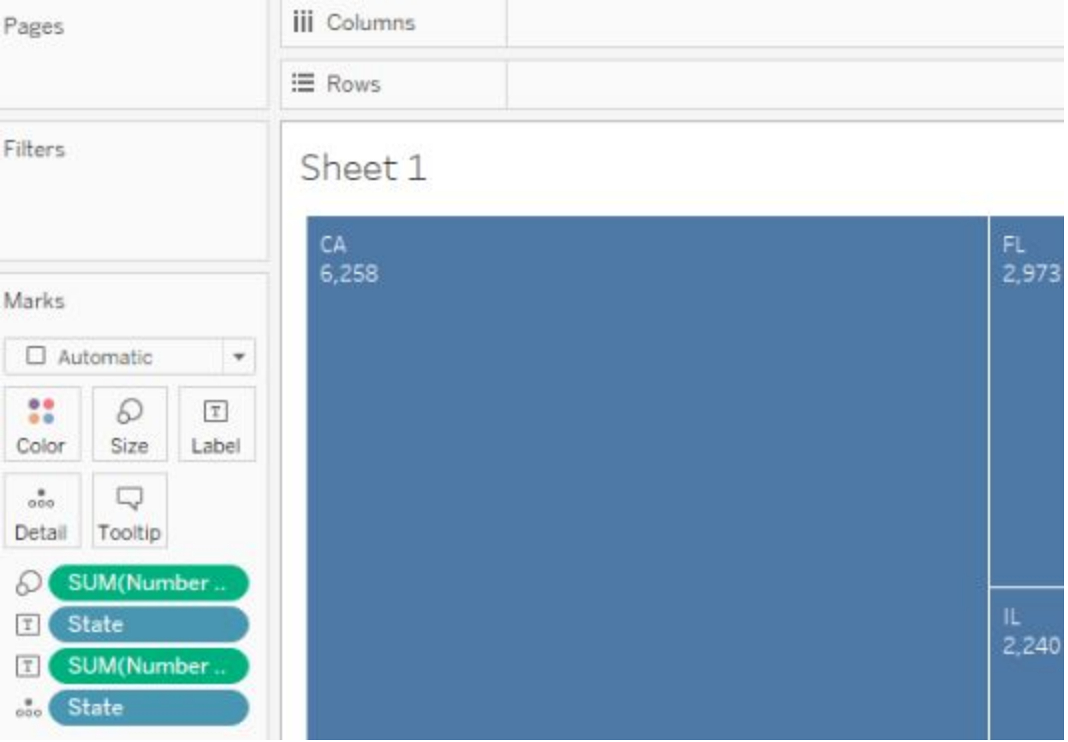
- 🟦 Sophia
- 🟥 Sophie

Select Color Palette: Winter

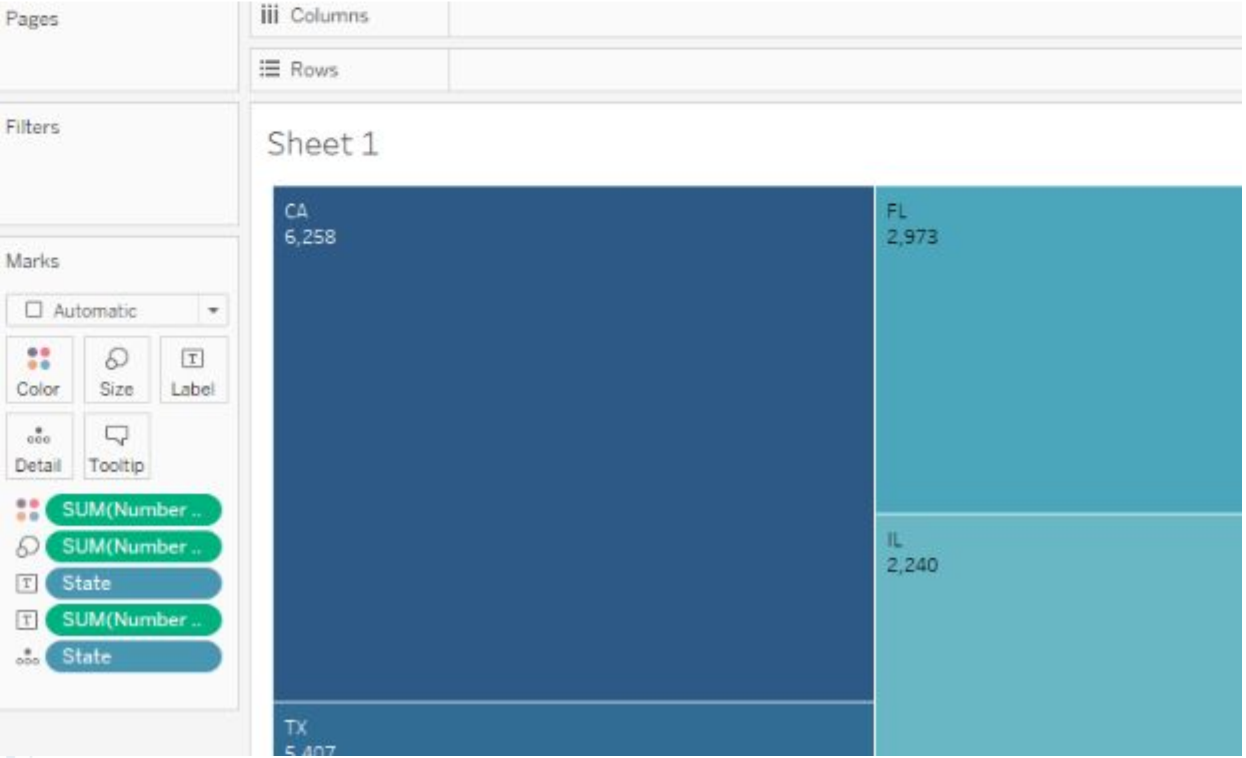
Color


Dimensions
Abc Gender
Abc Name
State
# Year
Abc Measure Names
Filters
Sheet 1
Add to Sheet
Duplicate
Rename
Hide
Aliases...
Create
Transform
Convert to Measure
Change Data Type
Geographic Role
Default Properties
Group by
Folders
Hierarchy
Replace References...
Describe...
Comment...
Color...
Shape...
Sort...

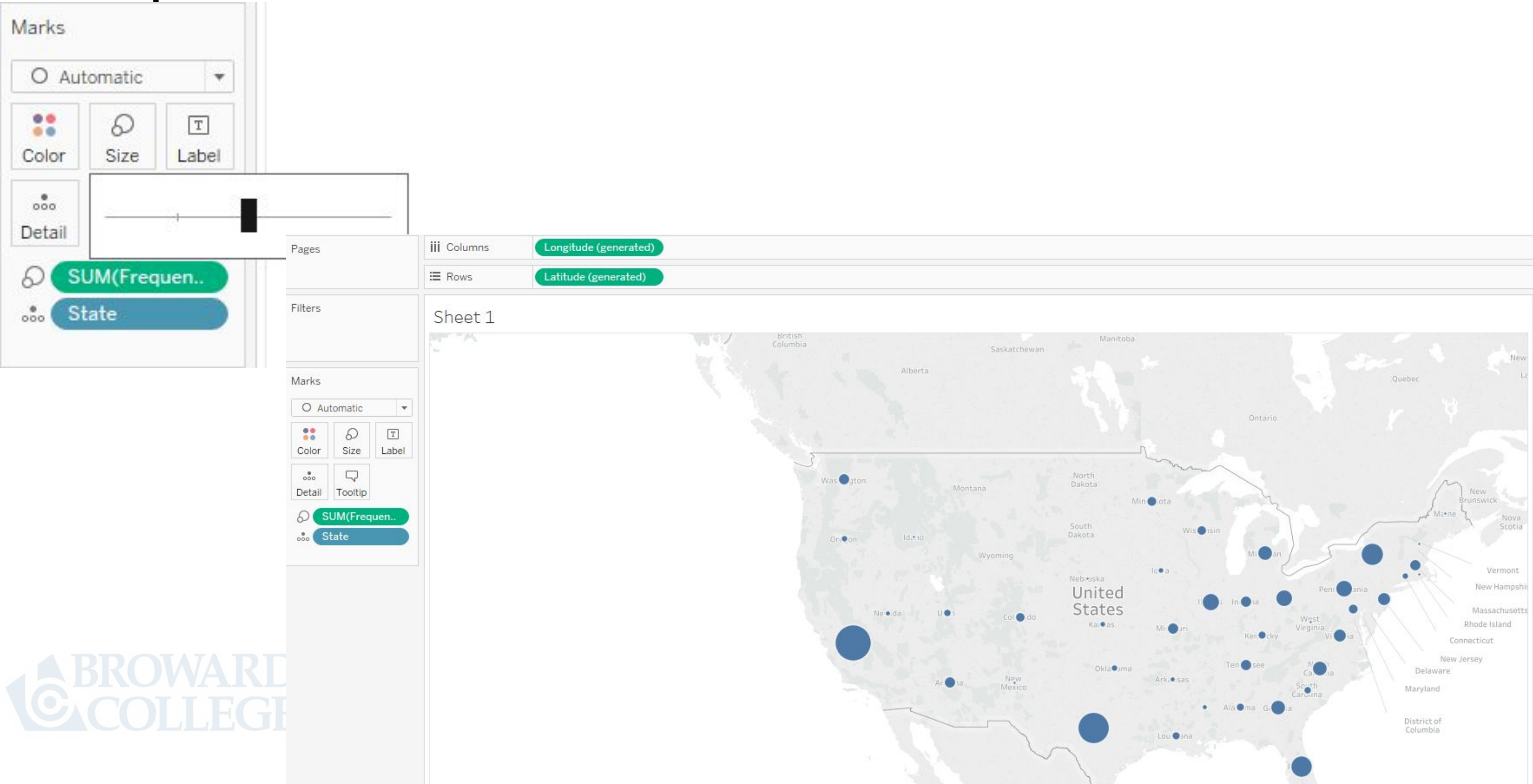
Tree Map


Pages
Columns
Rows
Filters
Marks
Automatic
Color
Size
Label
Detail
Tooltip
SUM(Number ..
State
SUM(Number ..
State
Sheet 1
CA
6,258
FL
2,973
IL
2,240

Tree Map


Pages
Columns
Rows
Filters
Sheet 1
Marks
Automatic
Color
Size
Label
Detail
Tooltip
SUM(Number ..
SUM(Number ..
State
SUM(Number ..
State
CA
6,258
FL
2,973
IL
2,240
TX
5,407

Map



Marks

○ Automatic

SUM(Frequen..

State

Pages

Filters

Columns: Longitude (generated)

Rows: Latitude (generated)

Sheet 1

Pages

Columns: Longitude (generated)

Rows: Latitude (generated)

Filters

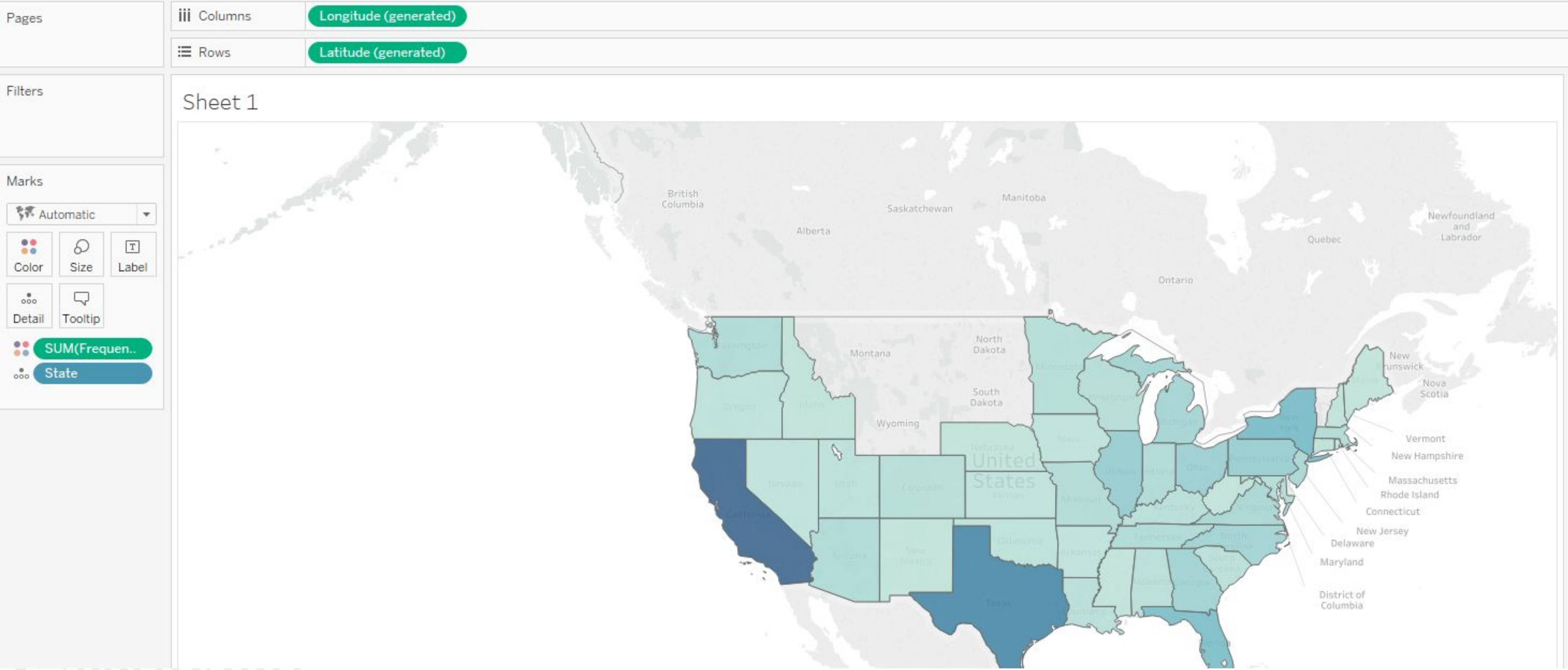
Marks

Automatic

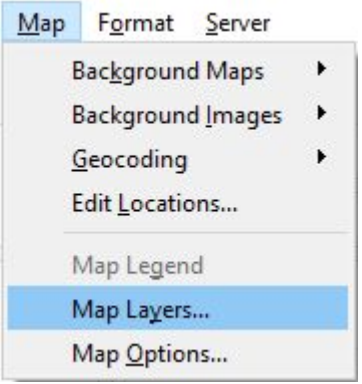
SUM(Frequen..

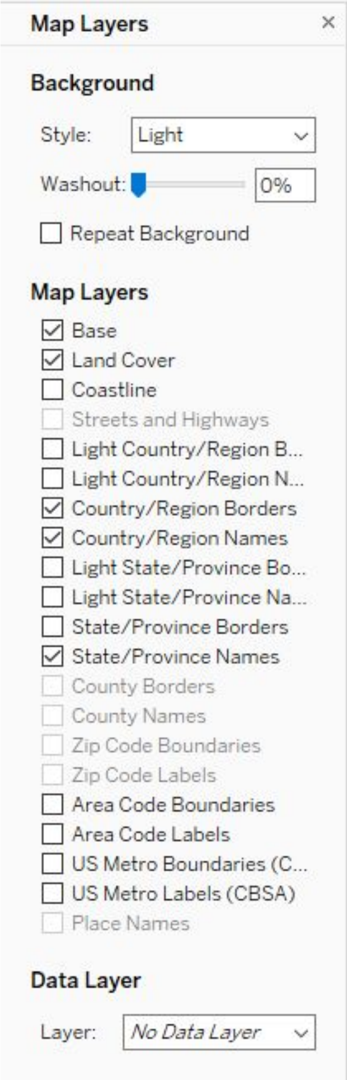
State

Sheet 1



More Maps...




Map Layers ×

Background

Style: Light

Washout: 0%

□ Repeat Background

Map Layers

☑ Base
☑ Land Cover
□ Coastline
□ Streets and Highways
□ Light Country/Region B...
□ Light Country/Region N...
☑ Country/Region Borders
☑ Country/Region Names
□ Light State/Province Bo...
□ Light State/Province Na...
□ State/Province Borders
☑ State/Province Names
□ County Borders
□ County Names
□ Zip Code Boundaries
□ Zip Code Labels
□ Area Code Boundaries
□ Area Code Labels
□ US Metro Boundaries (C...
□ US Metro Labels (CBSA)
□ Place Names

Data Layer

Layer: *No Data Layer*

The End...

Look up Dual Axis Maps

And

Customizing Tool tips