Leveraging Data Science to Facilitate Insightful, Reproducible, and Trustworthy I-O

SIOP 2019
Thursday, April 4th
1:30 PM to 2:50 PM
Room Maryland 4-6

Today's Tutorial

- Purpose:
  - Demonstrate how data science techniques can benefit IO Psychology:
    - Improved reproducibility of research
    - Better insights into human behavior at work
    - Increase credibility and impact of research
- Intended audience:
  - Already have working knowledge of R
  - Interested in using data science techniques
- Available materials: **http://bit.ly/SIOPDataScienceTutorial**

Presenters

- **Rachel Callan, PhD**
  - People Scientist at Humu
- **Andrew Collmus, MS**
  - People Data Scientist at Flex
  - Graduate Student at Old Dominion University
- **Elena Auer, MS**
  - Graduate Student at University of Minnesota
- **Sebastian Marin**
  - Graduate Student at University of Minnesota
- **Richard Landers, PhD**
  - Associate Professor, John P. Campbell Distinguished Professor of Industrial and Organizational Psychology at University of Minnesota

What to expect in this session

What we will cover:

- Accessible demonstration for later implementation:
  - Open source statistical tools, version control, and code repositories
  - Data sourcing and analytical techniques
- How these methods can improve IO research

What we won't cover:

- Installation of R/Py
- Basic coding in R
- Deep dive into specific data science techniques

How to follow along

- Live demonstration:
  - Connect to Twitter API to source data
  - Process and clean text data
  - Perform some analysis on that data
- Feel free to follow along, but it’s not necessary
  - Ask questions as we go along; resources are also available
- After the tutorial:
  - Demonstrations can be reproduced with materials on GitHub
  - Deeper discussion can be found in book chapter
  - Additional resources provided

Learning Objective #1: Increase the reproducibility of your current and future research projects

Open-source Technologies

- Free (compare to **M**any **P**ricey **L**anguages **U**sed for **S**EM that cost 100s of $$$)
- Customizable
- Excellent Documentation
- Active Community Support
- Reproducible Workflow

Open-source Technologies - Free

Free - compare to
Some **P**rofitable
Statistical **S**oftware
that costs 100s of $$$)

Accessible - to other people and your future self

Open-source Technologies - Customizable

- Write your own functions
- Modify someone else's functions

$$d_M(x,y) = \sqrt{(x-y)^T S^{-1}(x-y)}$$

$$= \sqrt{\begin{bmatrix} x_1 - y_1 & x_2 - y_2 \end{bmatrix} \begin{bmatrix} \frac{1}{\sigma_1^2} & 0 \\ 0 & \frac{1}{\sigma_2^2} \end{bmatrix} \begin{bmatrix} x_1 - y_1 \\ x_2 - y_2 \end{bmatrix}}$$

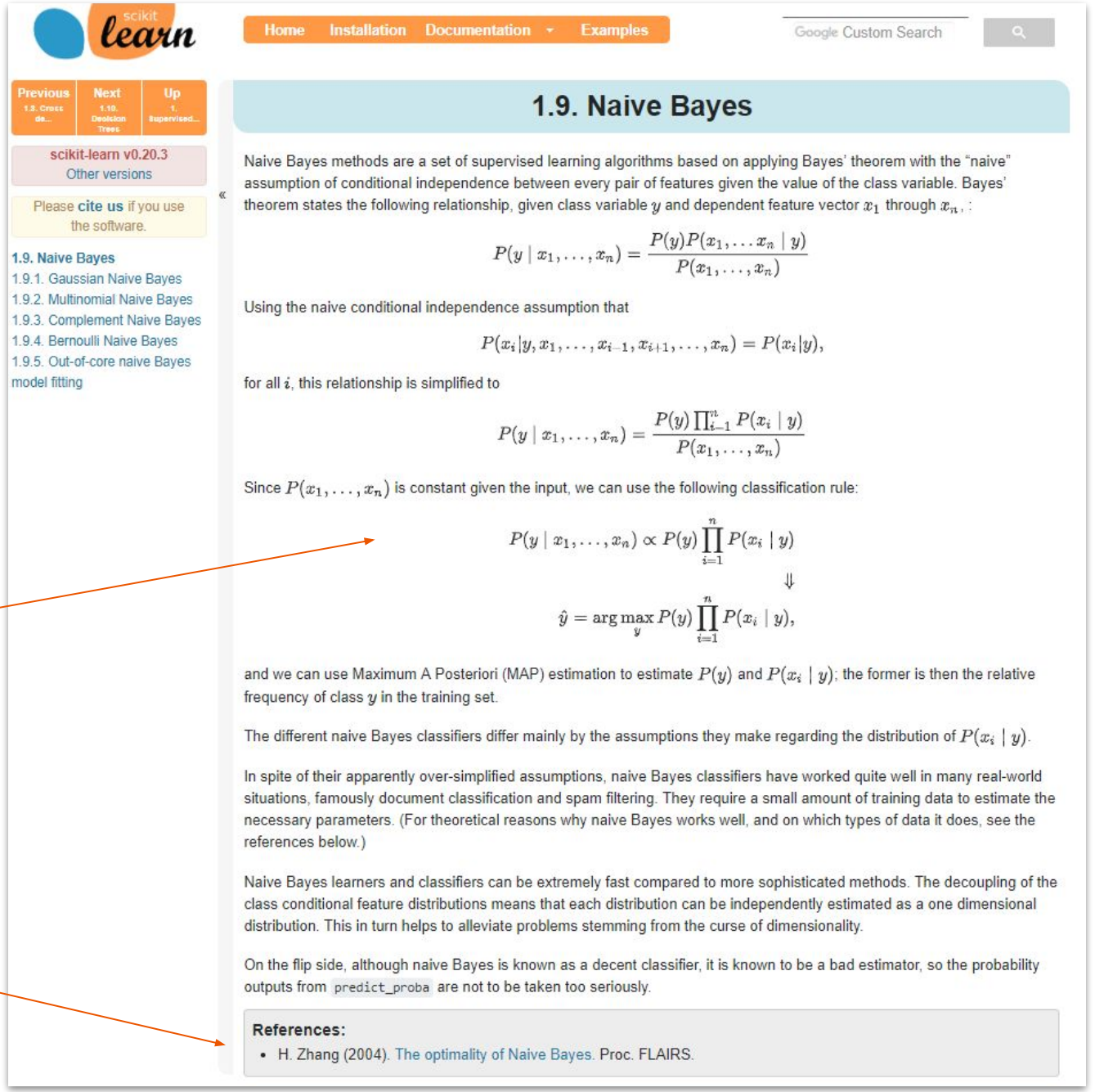
$$= \sqrt{\begin{bmatrix} \frac{x_1 - y_1}{\sigma_1^2} & \frac{x_2 - y_2}{\sigma_2^2} \end{bmatrix} \begin{bmatrix} x_1 - y_1 \\ x_2 - y_2 \end{bmatrix}}$$

$$= \sqrt{\frac{(x_1 - y_1)^2}{\sigma_1^2} + \frac{(x_2 - y_2)^2}{\sigma_2^2}}$$

Open-source Technologies - Excellent Documentation

Important Statsy Info

Hyperlinked Citation



scikit learn

Home Installation Documentation Examples

Google Custom Search

Previous 1.8. Cross de... Next 1.10. Decision Trees Up 1. Supervised...

scikit-learn v0.20.3
Other versions

Please cite us if you use the software.

1.9. Naive Bayes
1.9.1. Gaussian Naive Bayes
1.9.2. Multinomial Naive Bayes
1.9.3. Complement Naive Bayes
1.9.4. Bernoulli Naive Bayes
1.9.5. Out-of-core naive Bayes model fitting

1.9. Naive Bayes

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. Bayes' theorem states the following relationship, given class variable y and dependent feature vector x_1 through x_n, :

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots x_n \mid y)}{P(x_1, \ldots, x_n)}$$

Using the naive conditional independence assumption that

$$P(x_i|y, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = P(x_i|y),$$

for all i, this relationship is simplified to

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \ldots, x_n)}$$

Since $P(x_1, \ldots, x_n)$ is constant given the input, we can use the following classification rule:

$$P(y \mid x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

$$\Downarrow$$

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y),$$

and we can use Maximum A Posteriori (MAP) estimation to estimate $P(y)$ and $P(x_i \mid y)$; the former is then the relative frequency of class y in the training set.

The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of $P(x_i \mid y)$.

In spite of their apparently over-simplified assumptions, naive Bayes classifiers have worked quite well in many real-world situations, famously document classification and spam filtering. They require a small amount of training data to estimate the necessary parameters. (For theoretical reasons why naive Bayes works well, and on which types of data it does, see the references below.)

Naive Bayes learners and classifiers can be extremely fast compared to more sophisticated methods. The decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one dimensional distribution. This in turn helps to alleviate problems stemming from the curse of dimensionality.

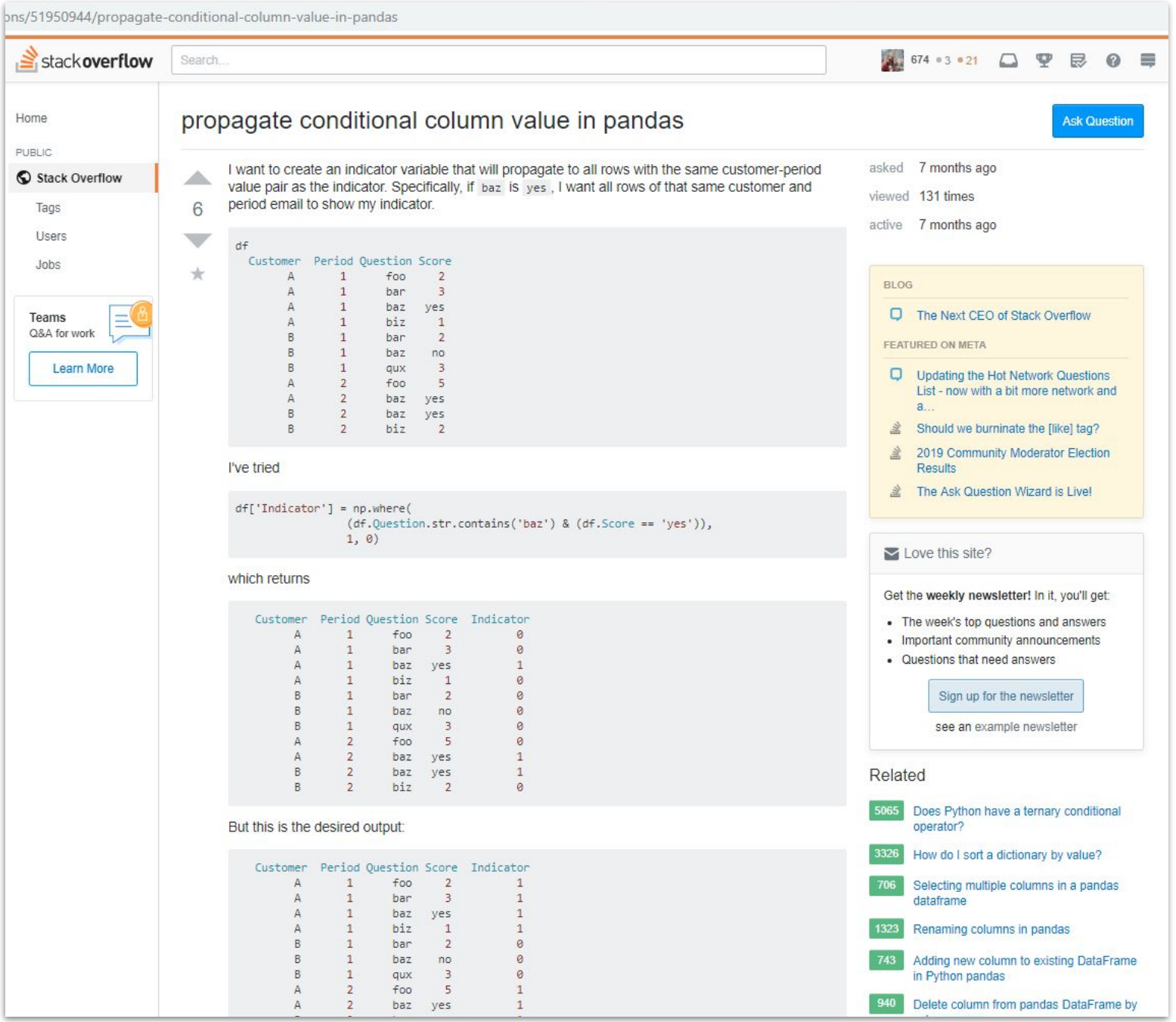
On the flip side, although naive Bayes is known as a decent classifier, it is known to be a bad estimator, so the probability outputs from `predict_proba` are not to be taken too seriously.

References:

- H. Zhang (2004). The optimality of Naive Bayes. Proc. FLAIRS.

Open-source Technologies - Community Support

Question I asked



ons/51950944/propagate-conditional-column-value-in-pandas

stackoverflow Search... 674 • 3 • 21

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

Teams
Q&A for work

Learn More

propagate conditional column value in pandas

Ask Question

I want to create an indicator variable that will propagate to all rows with the same customer-period value pair as the indicator. Specifically, if `baz` is `yes`, I want all rows of that same customer and period email to show my indicator.

```
df
  Customer  Period Question Score
         A       1      foo     2
         A       1      bar     3
         A       1      baz   yes
         A       1      biz     1
         B       1      bar     2
         B       1      baz    no
         B       1      qux     3
         A       2      foo     5
         A       2      baz   yes
         B       2      baz   yes
         B       2      biz     2
```

I've tried

```
df['Indicator'] = np.where(
                  (df.Question.str.contains('baz') & (df.Score == 'yes')),
                  1, 0)
```

which returns

```
  Customer  Period Question Score  Indicator
         A       1      foo     2          0
         A       1      bar     3          0
         A       1      baz   yes          1
         A       1      biz     1          0
         B       1      bar     2          0
         B       1      baz    no          0
         B       1      qux     3          0
         A       2      foo     5          0
         A       2      baz   yes          1
         B       2      baz   yes          1
         B       2      biz     2          0
```

But this is the desired output:

```
  Customer  Period Question Score  Indicator
         A       1      foo     2          1
         A       1      bar     3          1
         A       1      baz   yes          1
         A       1      biz     1          1
         B       1      bar     2          0
         B       1      baz    no          0
         B       1      qux     3          0
         A       2      foo     5          1
         A       2      baz   yes          1
```

asked 7 months ago
viewed 131 times
active 7 months ago

BLOG
The Next CEO of Stack Overflow
FEATURED ON META
Updating the Hot Network Questions List - now with a bit more network and a…
Should we burninate the [like] tag?
2019 Community Moderator Election Results
The Ask Question Wizard is Live!

Love this site?

Get the **weekly newsletter!** In it, you'll get:

- The week's top questions and answers
- Important community announcements
- Questions that need answers

Sign up for the newsletter

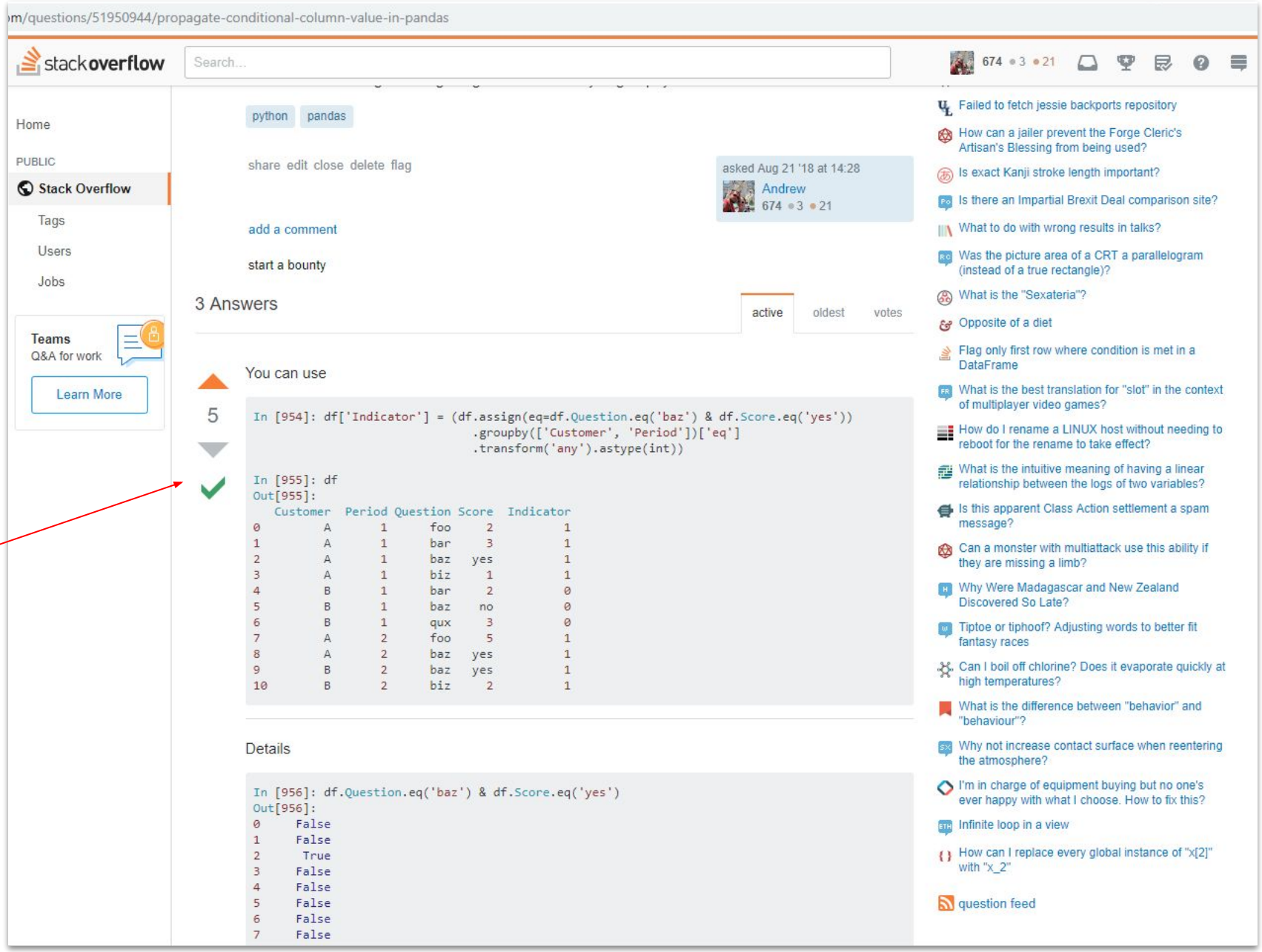
see an example newsletter

Related

5065 Does Python have a ternary conditional operator?
3326 How do I sort a dictionary by value?
706 Selecting multiple columns in a pandas dataframe
1323 Renaming columns in pandas
743 Adding new column to existing DataFrame in Python pandas
940 Delete column from pandas DataFrame by

Open-source Technologies - Community Support

Answer from Community

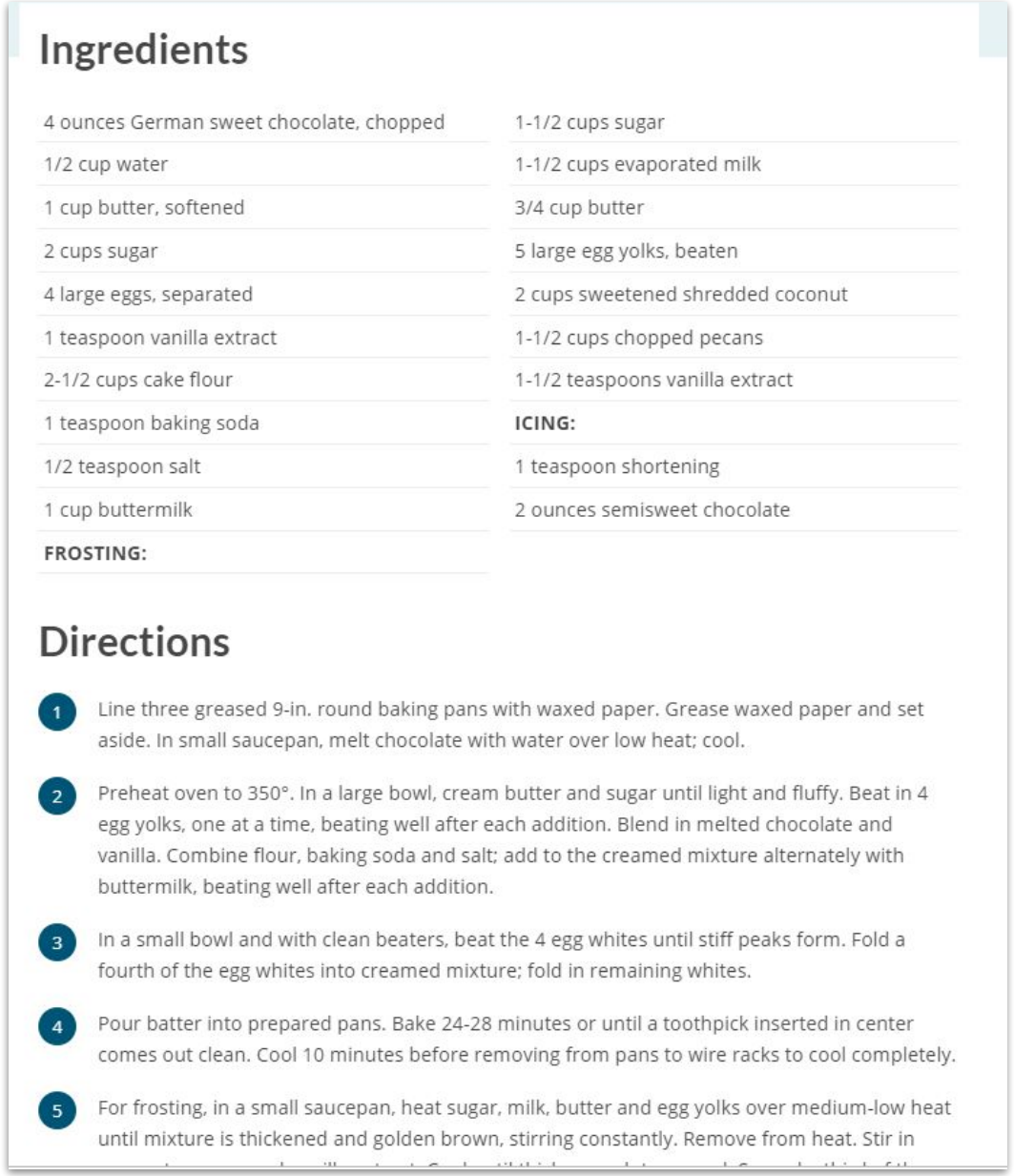


Open-source Technologies - Reproducible Workflow

Can you make this?

Open-source Technologies - Reproducible Workflow

Not without this



Ingredients

4 ounces German sweet chocolate, chopped	1-1/2 cups sugar
1/2 cup water	1-1/2 cups evaporated milk
1 cup butter, softened	3/4 cup butter
2 cups sugar	5 large egg yolks, beaten
4 large eggs, separated	2 cups sweetened shredded coconut
1 teaspoon vanilla extract	1-1/2 cups chopped pecans
2-1/2 cups cake flour	1-1/2 teaspoons vanilla extract
1 teaspoon baking soda	**ICING:**
1/2 teaspoon salt	1 teaspoon shortening
1 cup buttermilk	2 ounces semisweet chocolate
FROSTING:	

Directions

1. Line three greased 9-in. round baking pans with waxed paper. Grease waxed paper and set aside. In small saucepan, melt chocolate with water over low heat; cool.
2. Preheat oven to 350°. In a large bowl, cream butter and sugar until light and fluffy. Beat in 4 egg yolks, one at a time, beating well after each addition. Blend in melted chocolate and vanilla. Combine flour, baking soda and salt; add to the creamed mixture alternately with buttermilk, beating well after each addition.
3. In a small bowl and with clean beaters, beat the 4 egg whites until stiff peaks form. Fold a fourth of the egg whites into creamed mixture; fold in remaining whites.
4. Pour batter into prepared pans. Bake 24-28 minutes or until a toothpick inserted in center comes out clean. Cool 10 minutes before removing from pans to wire racks to cool completely.
5. For frosting, in a small saucepan, heat sugar, milk, butter and egg yolks over medium-low heat until mixture is thickened and golden brown, stirring constantly. Remove from heat. Stir in

https://www.tasteofhome.com/recipes/german-chocolate-cake

Syntax, Pipelines, and Versioning

- Syntax creates a self-documenting reproducible workflow
- Pipelines document the journey of the data
- Versioning keeps your directory clean and documents all changes

Syntax, Pipelines, and Versioning

- Syntax creates a self-documenting reproducible workflow
  - Every action is recorded, and replicable

Python

```
In [286]: df = pd.read_csv("ds_ch.csv", sep=",")
     ...: X = df.iloc[:,:-1] # get features, which make up first n columns of dataframe
     ...: y = df.iloc[:, -1] # get criterion, which is the last column of the dataframe
     ...: X_sm = sm.add_constant(X) # fits an intercept
     ...: sm_model = sm.OLS(y, X_sm).fit()
     ...: sm_model.summary()
```

R

```
> df <- read.csv('ds_ch.csv')
> names(df) <- c(paste0('X', 1:10), 'y')
>
> Rmodel <- lm(y~., data=df)
> summary(Rmodel)
```

Syntax, Pipelines, and Versioning

Python

```
Out[286]:
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                    0.1   R-squared:                       0.518
Model:                            OLS   Adj. R-squared:                  0.507
Method:                 Least Squares   F-statistic:                     46.27
Date:                Wed, 21 Mar 2018   Prob (F-statistic):           3.83e-62
Time:                        14:04:38   Log-Likelihood:                -2386.0
No. Observations:                 442   AIC:                             4794.
Df Residuals:                     431   BIC:                             4839.
Df Model:                          10
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        152.1335      2.576     59.061      0.000     147.071     157.196
0            -10.0122     59.749     -0.168      0.867    -127.448     107.424
1           -239.8191     61.222     -3.917      0.000    -360.151    -119.488
2            519.8398     66.534      7.813      0.000     389.069     650.610
3            324.3904     65.422      4.958      0.000     195.805     452.976
4           -792.1842    416.684     -1.901      0.058   -1611.169      26.801
5            476.7458    339.035      1.406      0.160    -189.621    1143.113
6            101.0446    212.533      0.475      0.635    -316.685     518.774
7            177.0642    161.476      1.097      0.273    -140.313     494.442
8            751.2793    171.902      4.370      0.000     413.409    1089.150
9             67.6254     65.984      1.025      0.306     -62.065     197.316
==============================================================================
```

R

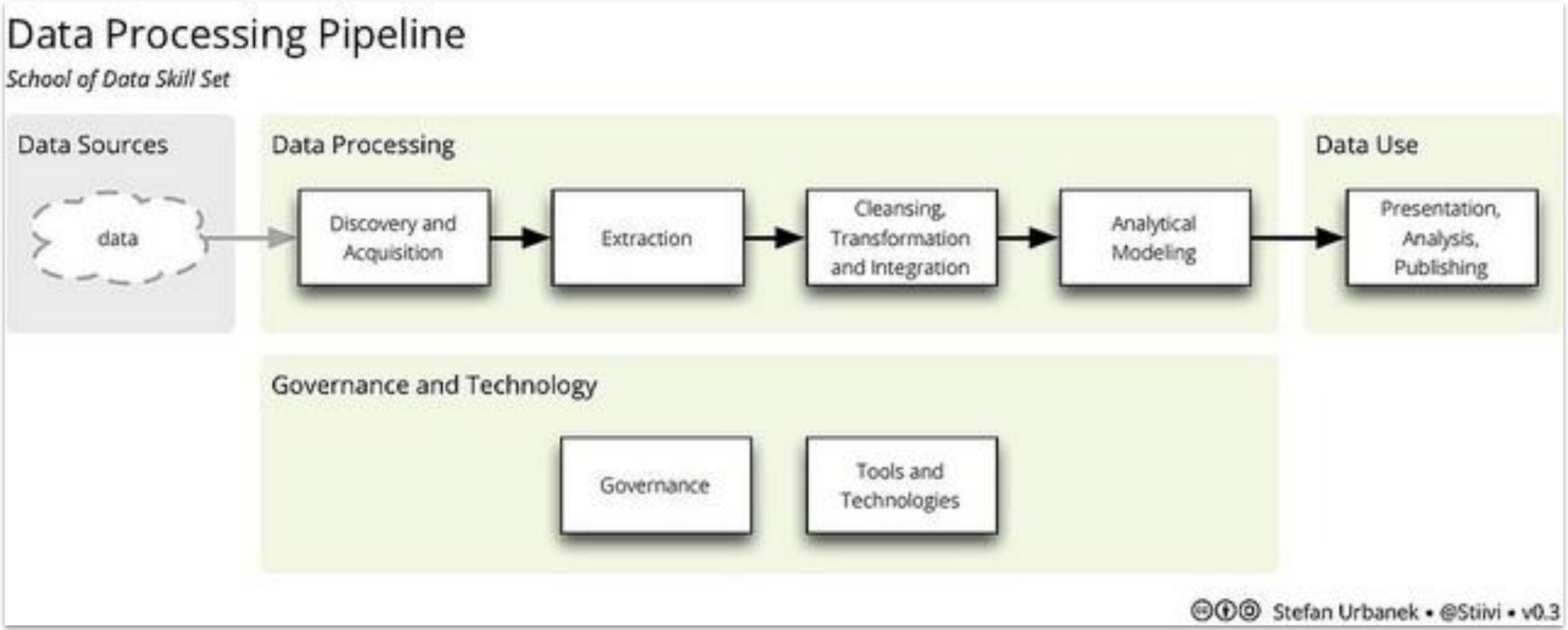
```
Call:
lm(formula = y ~ ., data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-155.829  -38.534   -0.227   37.806  151.355

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  152.133      2.576  59.061  < 2e-16 ***
X1           -10.012     59.749  -0.168 0.867000
X2          -239.819     61.222  -3.917 0.000104 ***
X3           519.840     66.534   7.813 4.30e-14 ***
X4           324.390     65.422   4.958 1.02e-06 ***
X5          -792.184    416.684  -1.901 0.057947 .
X6           476.746    339.035   1.406 0.160389
X7           101.045    212.533   0.475 0.634721
X8           177.064    161.476   1.097 0.273456
X9           751.279    171.902   4.370 1.56e-05 ***
X10           67.625     65.984   1.025 0.305998
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 54.15 on 431 degrees of freedom
Multiple R-squared:  0.5177,    Adjusted R-squared:  0.5066
F-statistic: 46.27 on 10 and 431 DF,  p-value: < 2.2e-16
```

Syntax, Pipelines, and Versioning



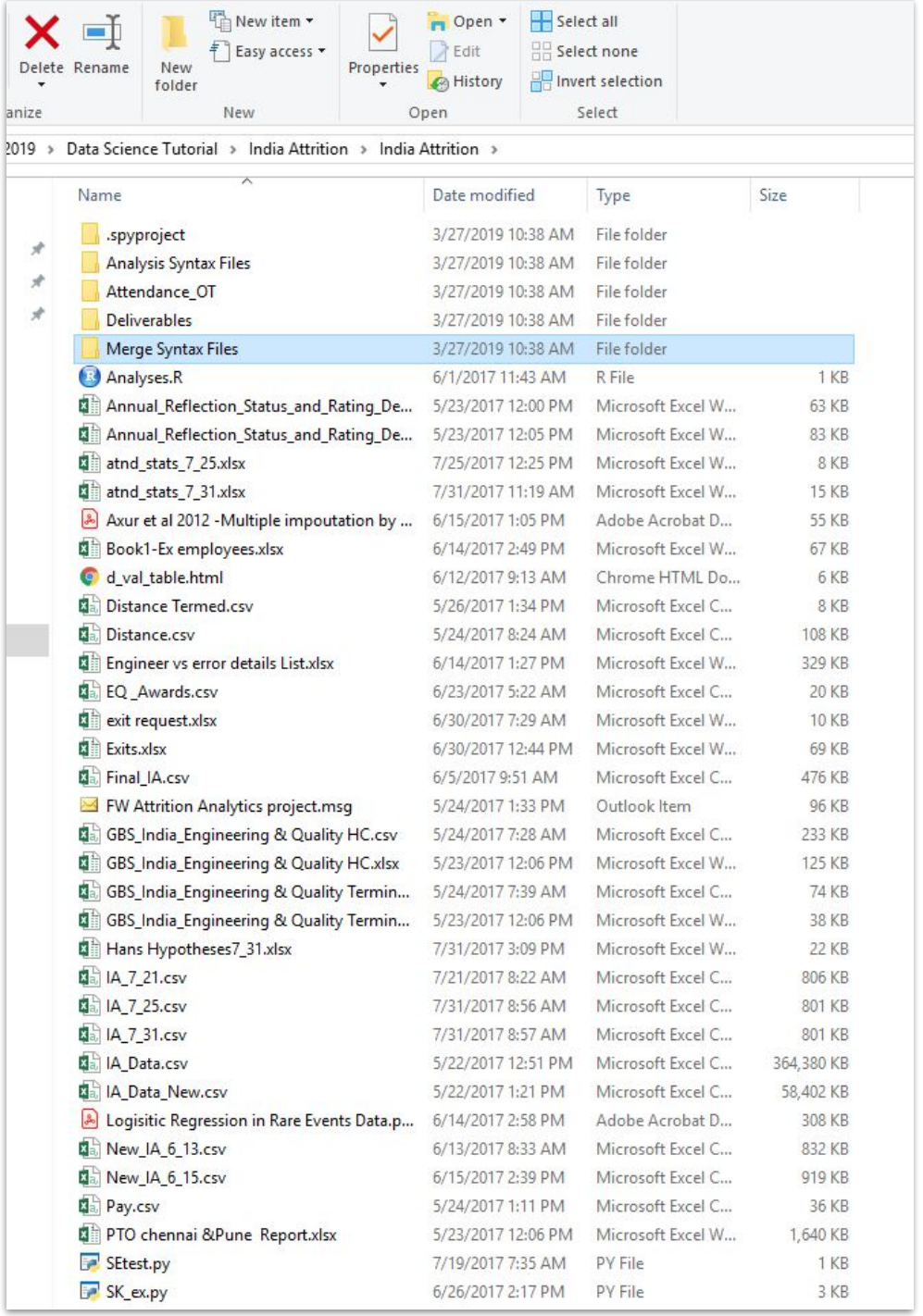
Syntax, Pipelines, and Versioning

Versioning keeps your directory clean and documents all changes

Why is this important?

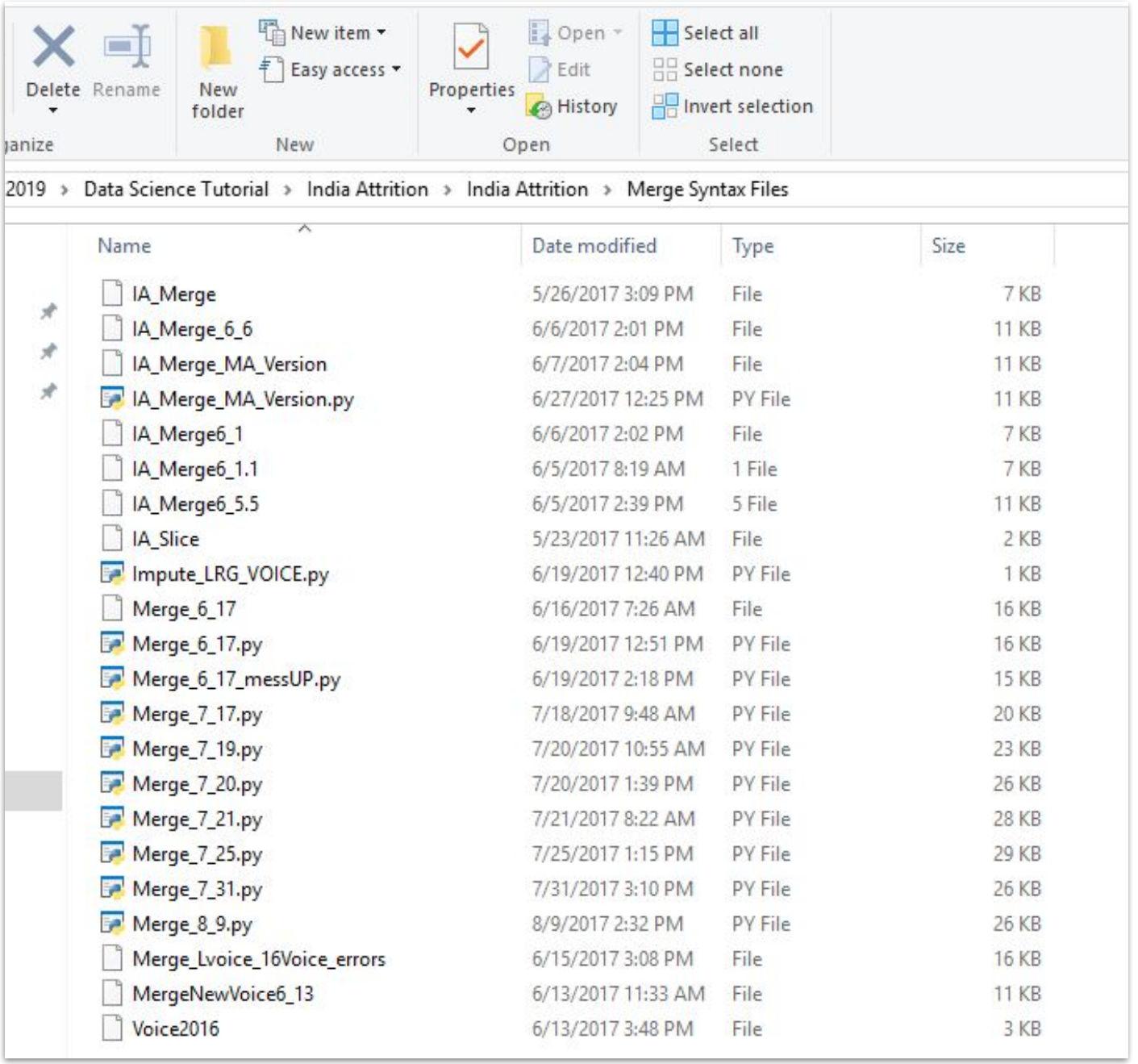
Versioning

The Enemy



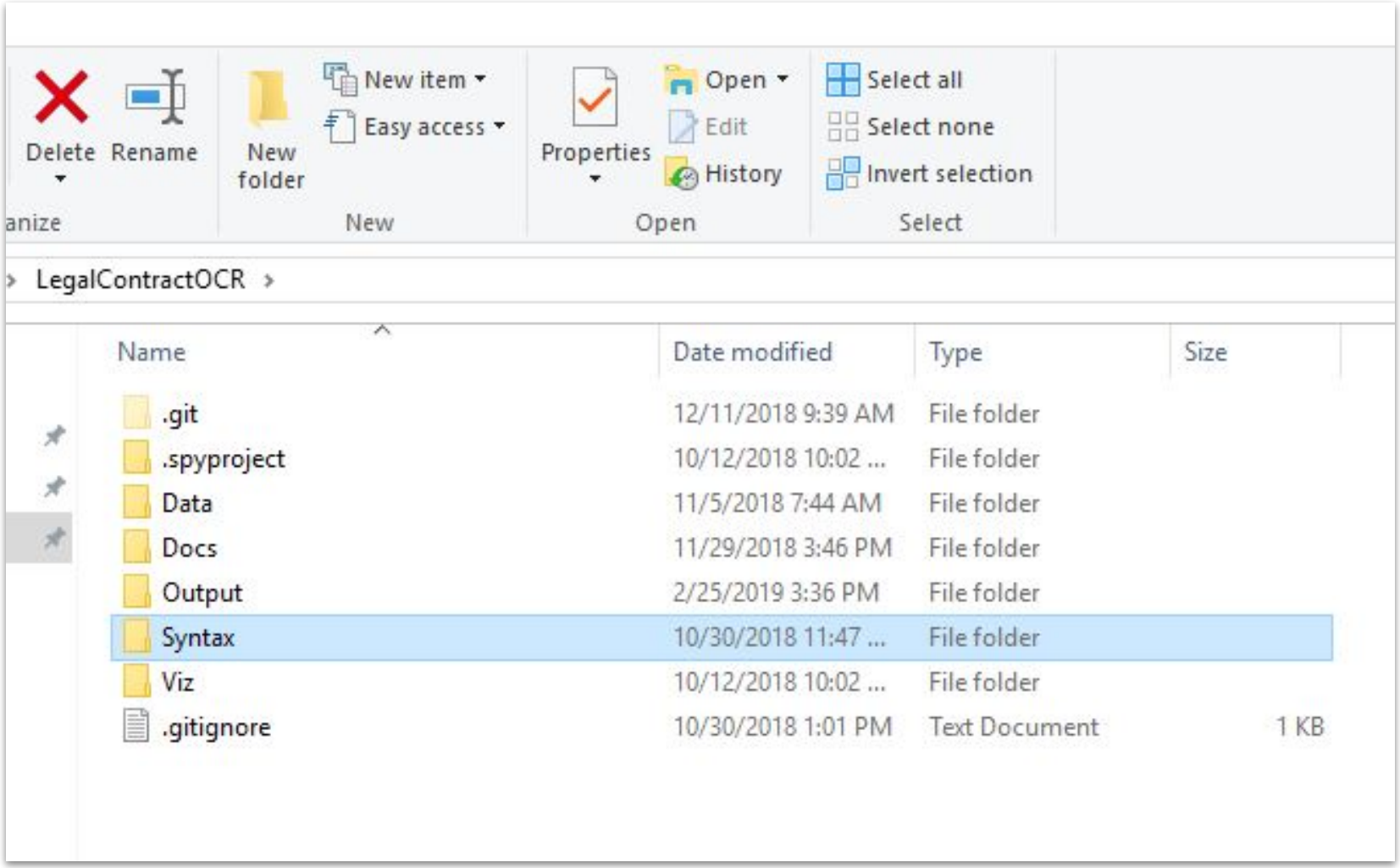
Versioning

This one does not spark joy



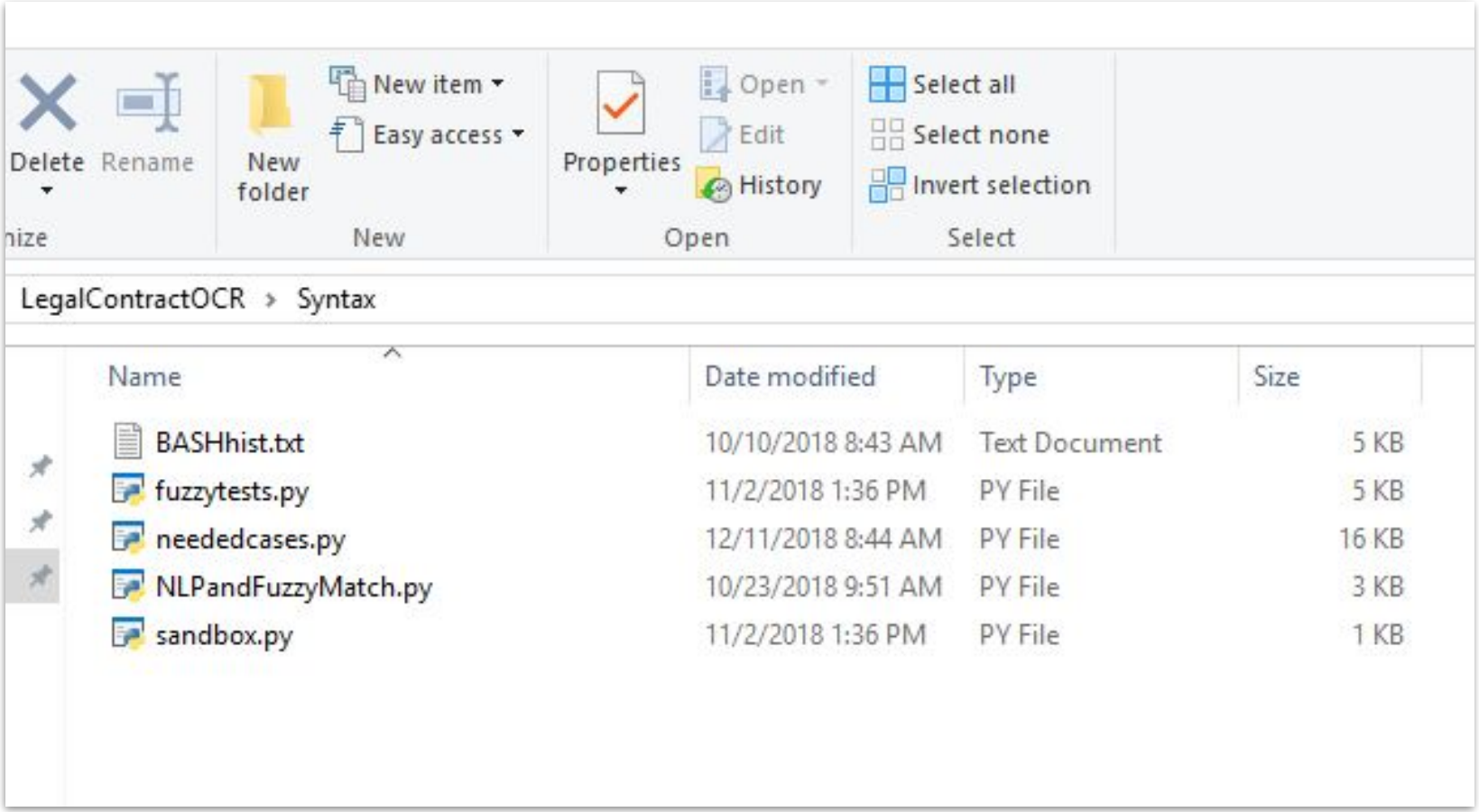
Versioning

The Hero



Versioning

This one sparks joy



Versioning

Git - Language used for versioning (keeps track of changes) Two Main Functions

1. **Stage** - make changes to a file
2. **Commit** - publish group of changes, accompanied by
    a. **Commit Message** - Brief present-tense description

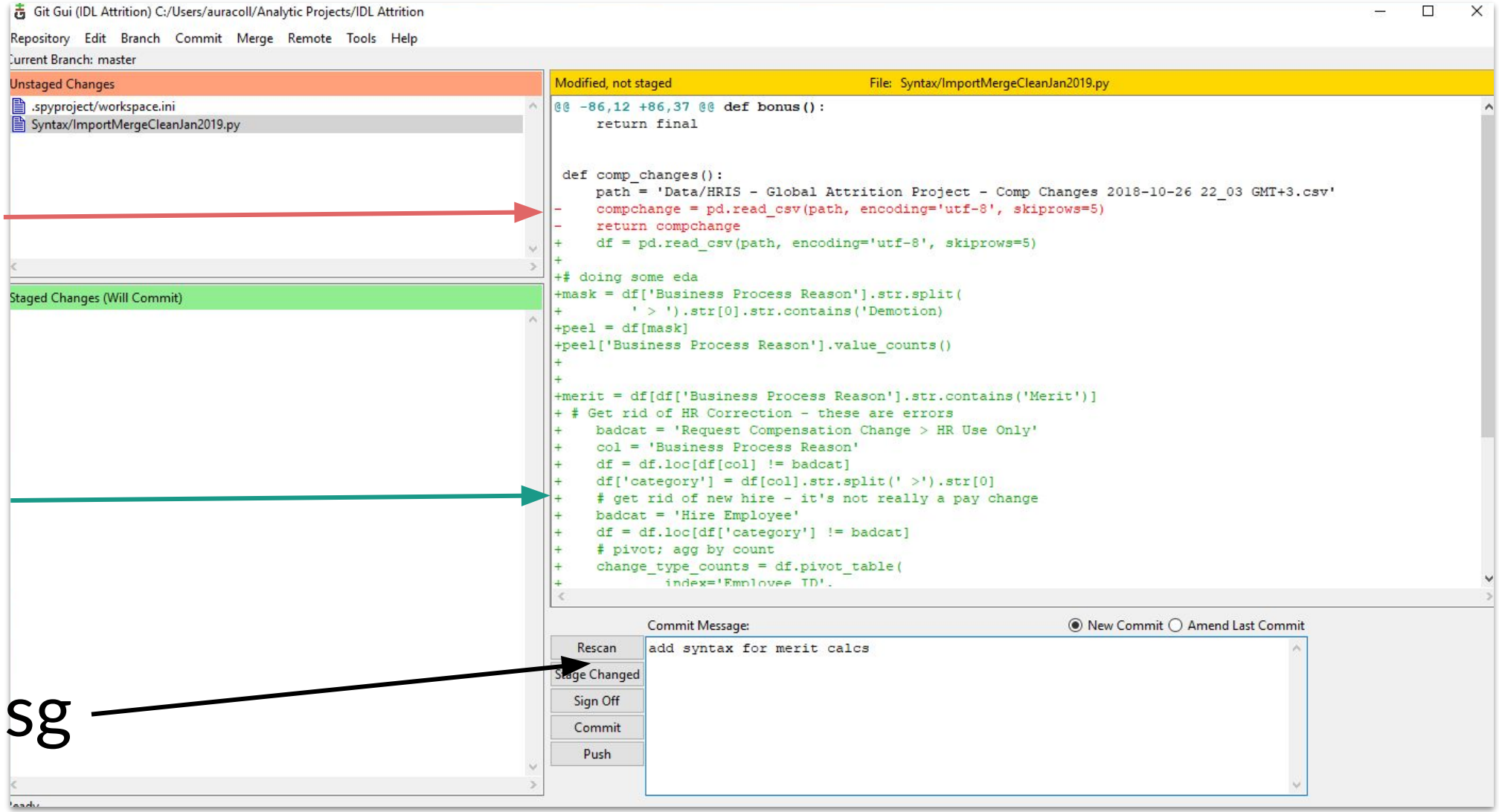
    e.g. “***create pie chart***” “***resolve duplicate IDs***”

Versioning
Git

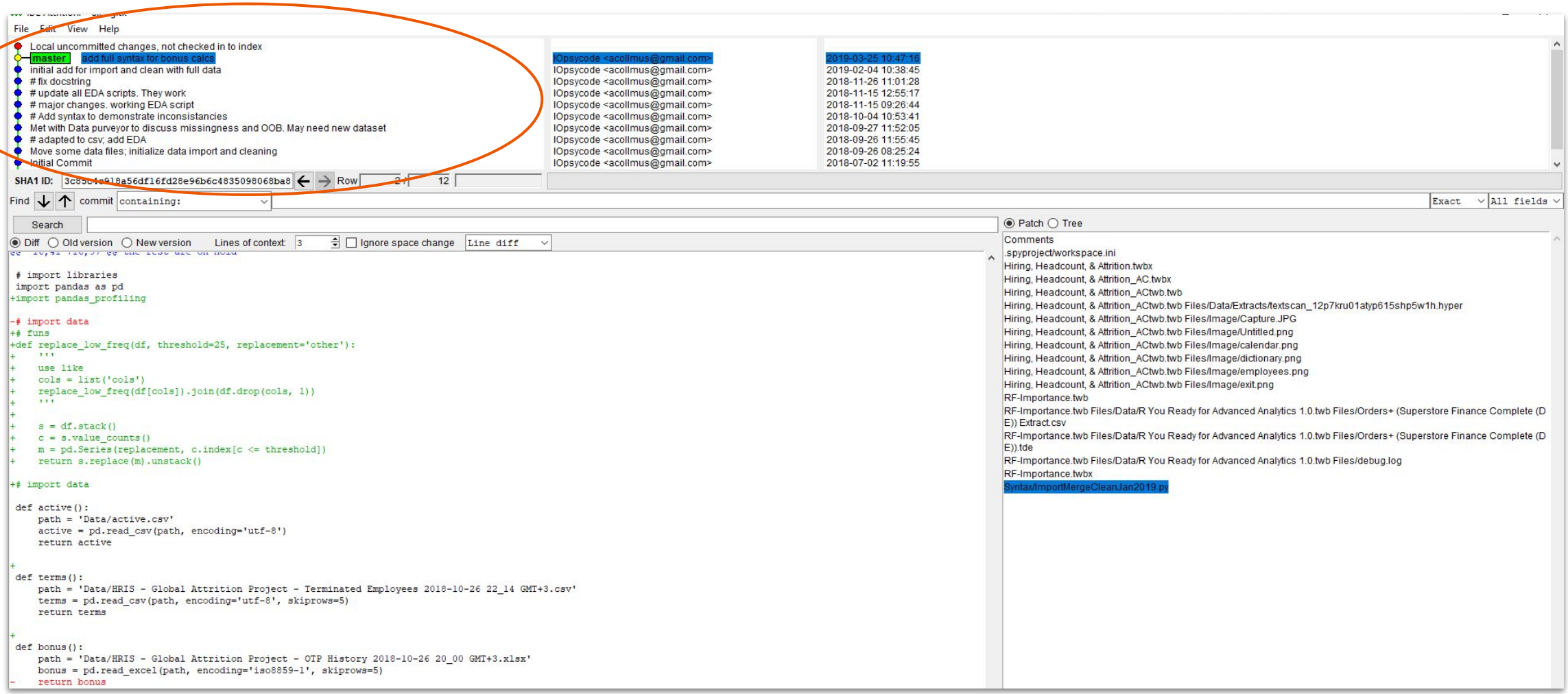
Deletions

Insertions

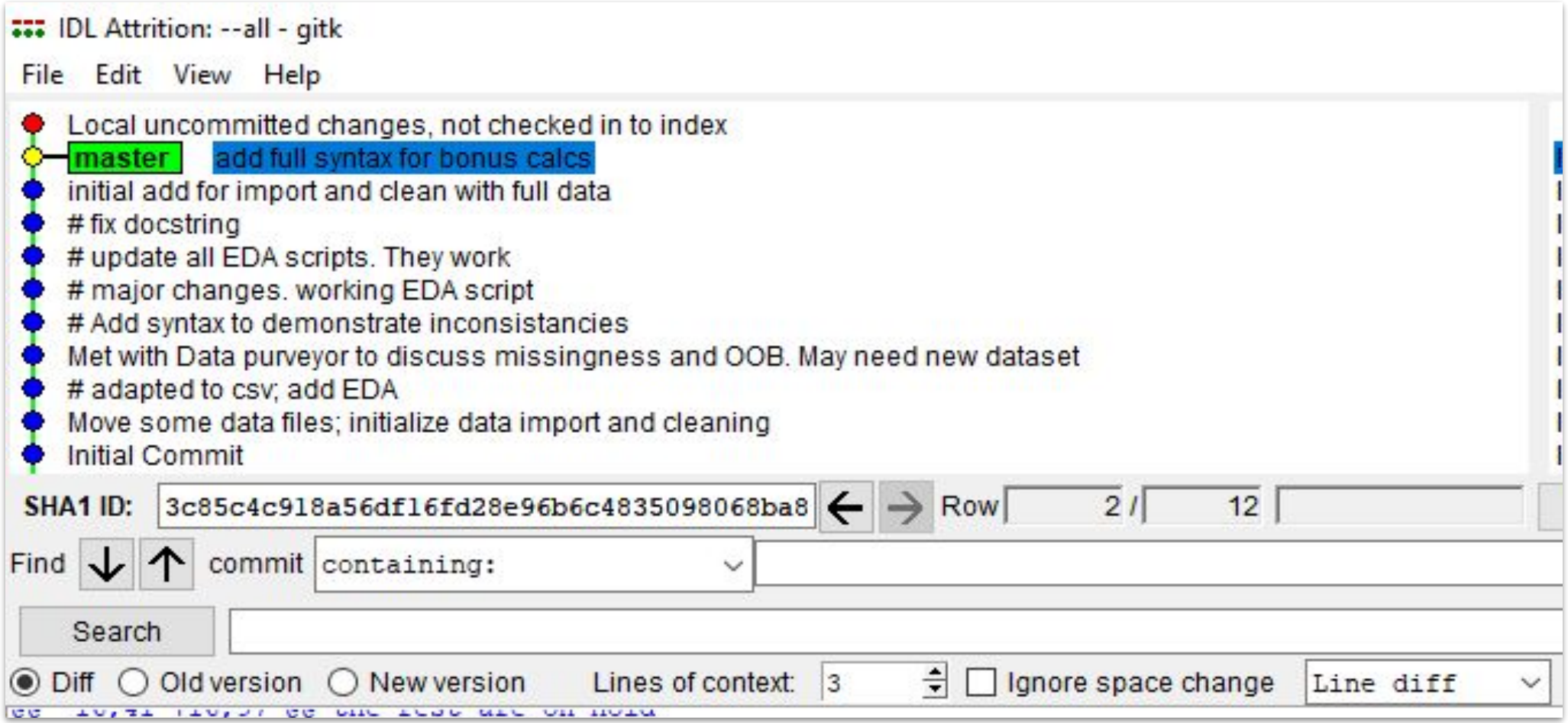
Commit Msg



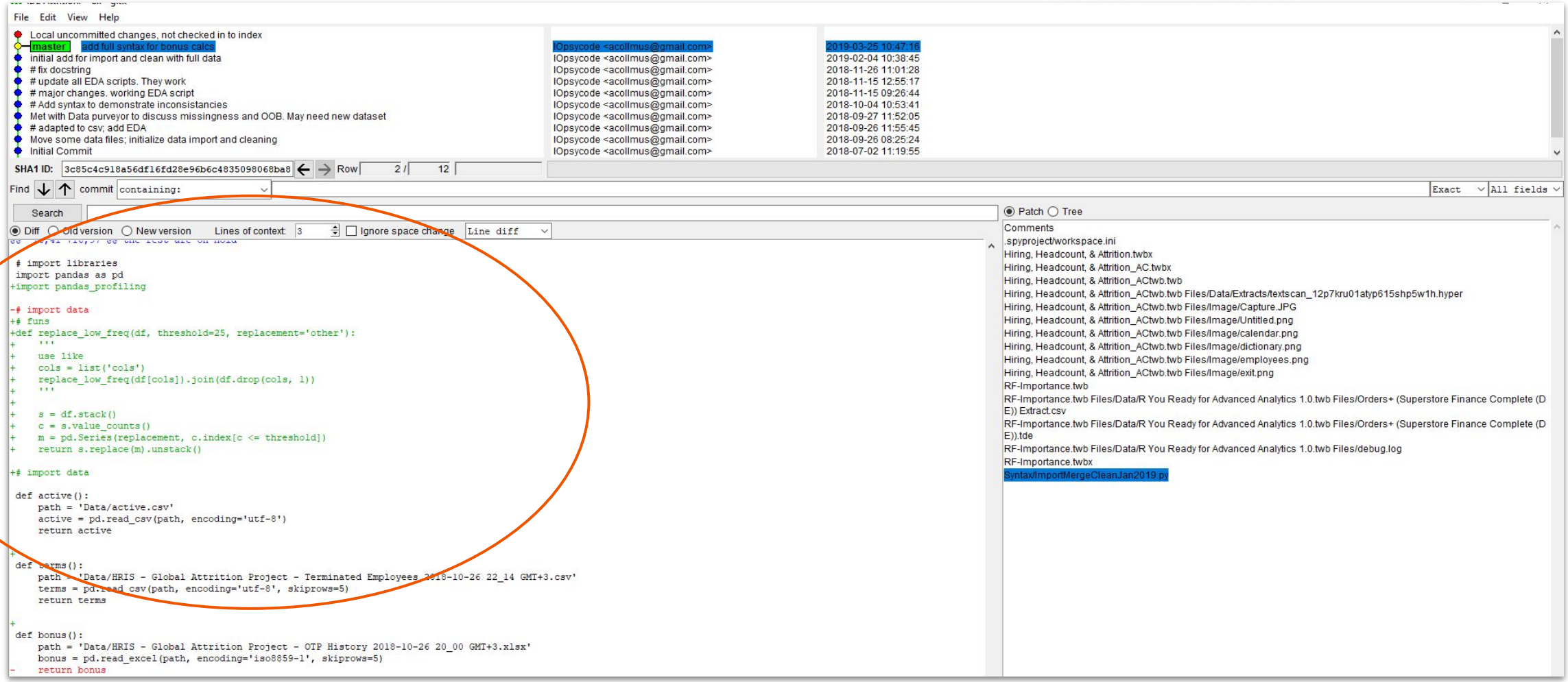
Versioning
Git Tree



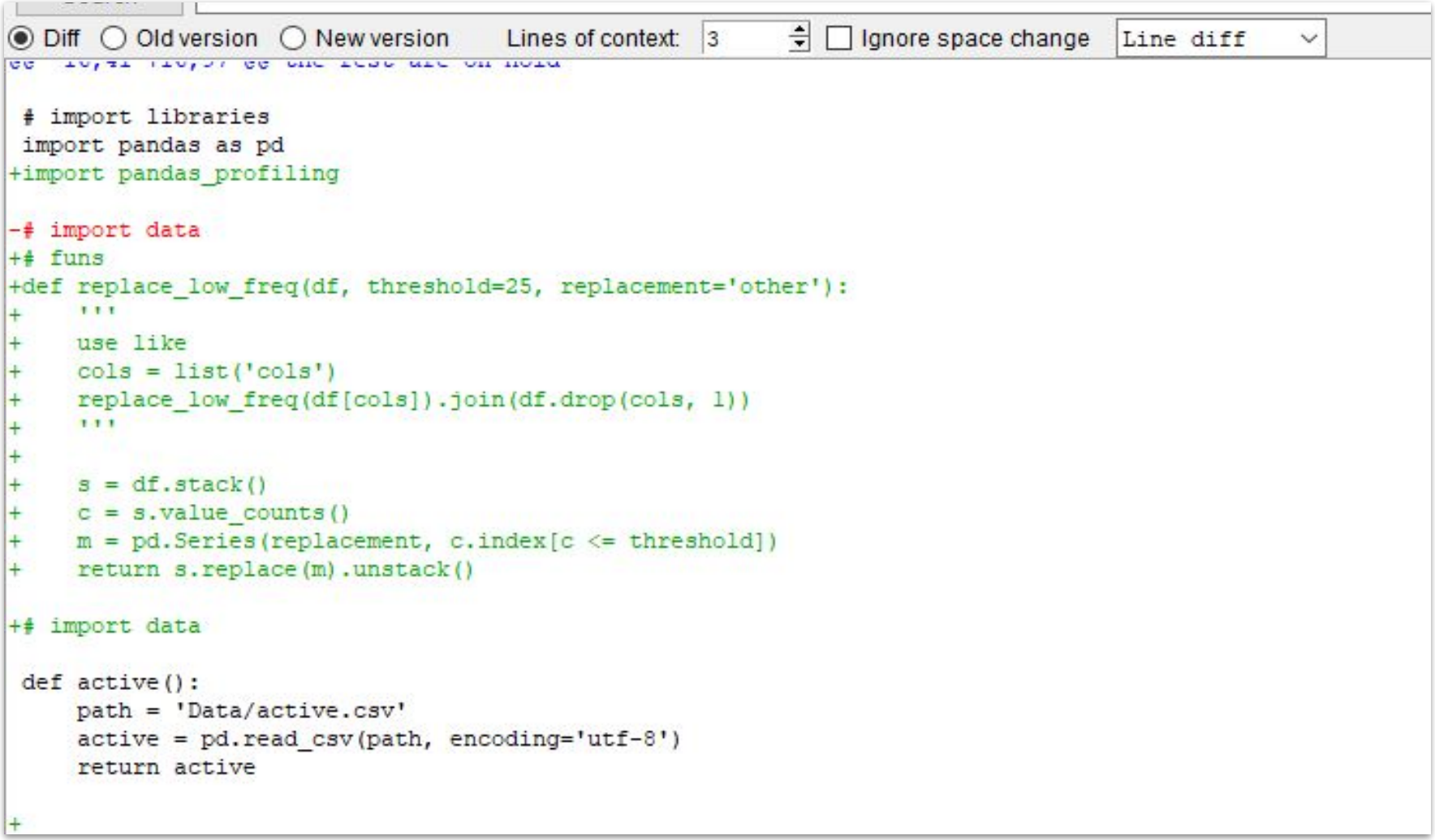
Versioning
Git Tree



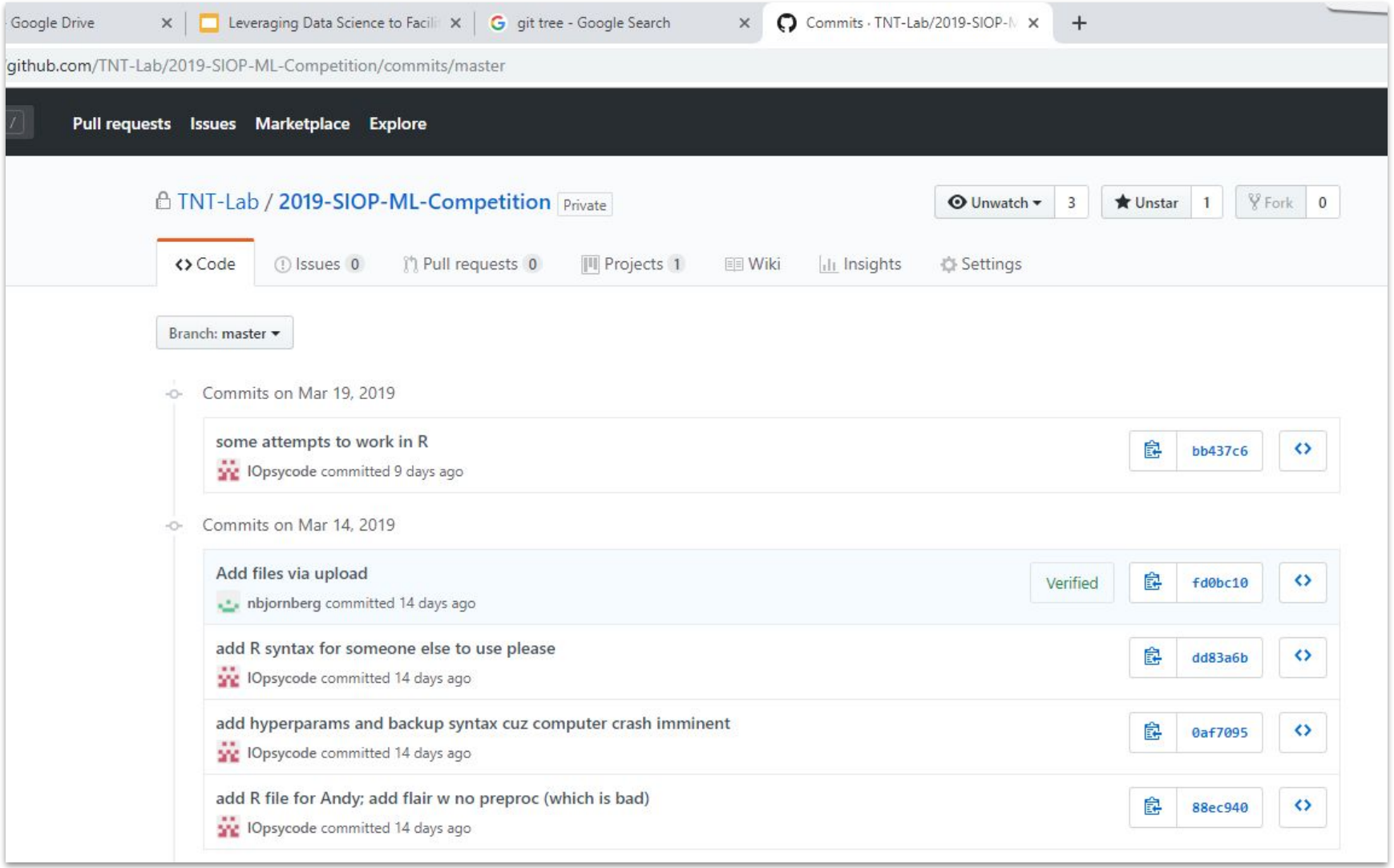
Versioning
Git Tree



Versioning
Git Tree



Versioning
GitHub

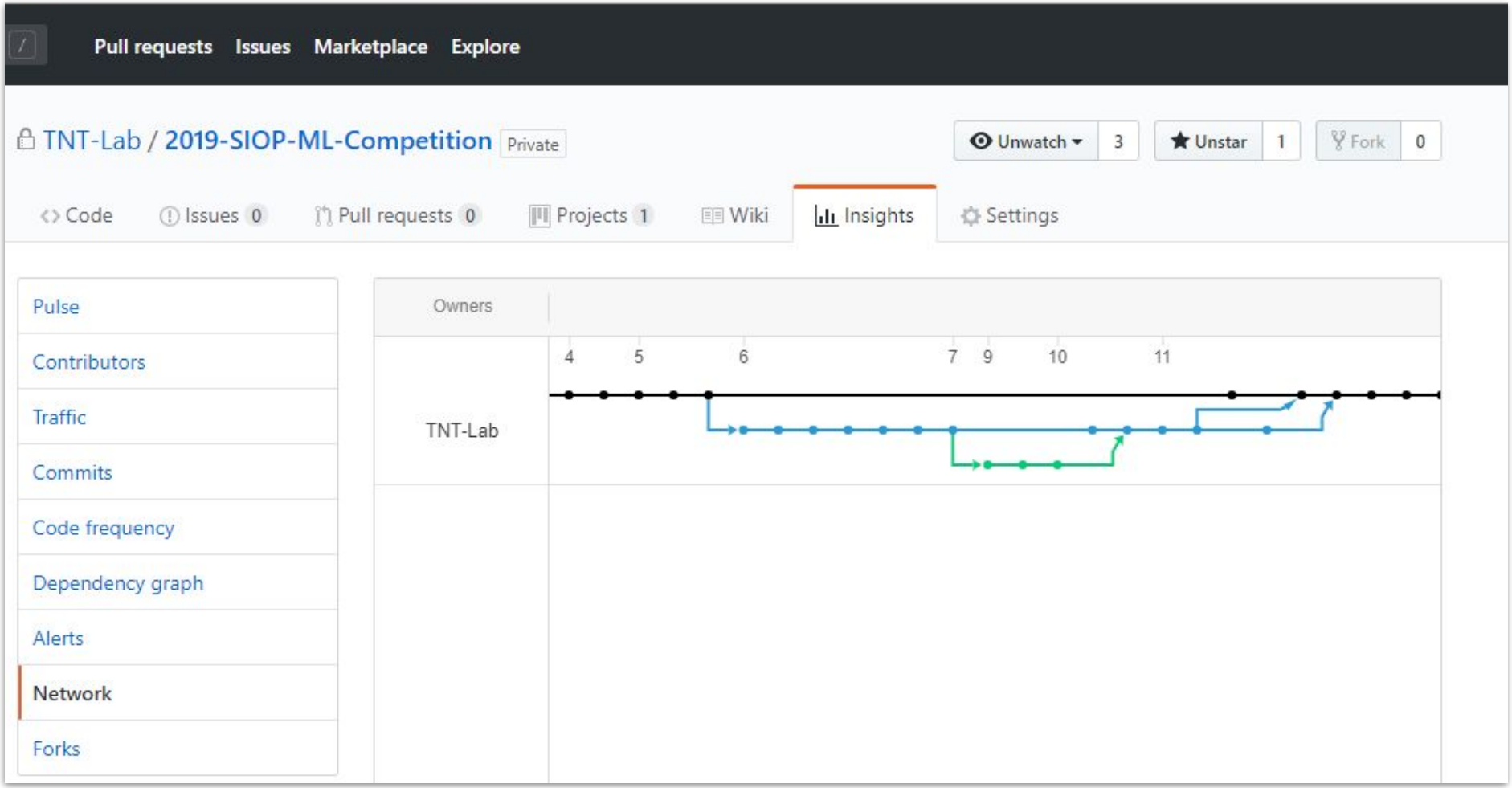


Versioning

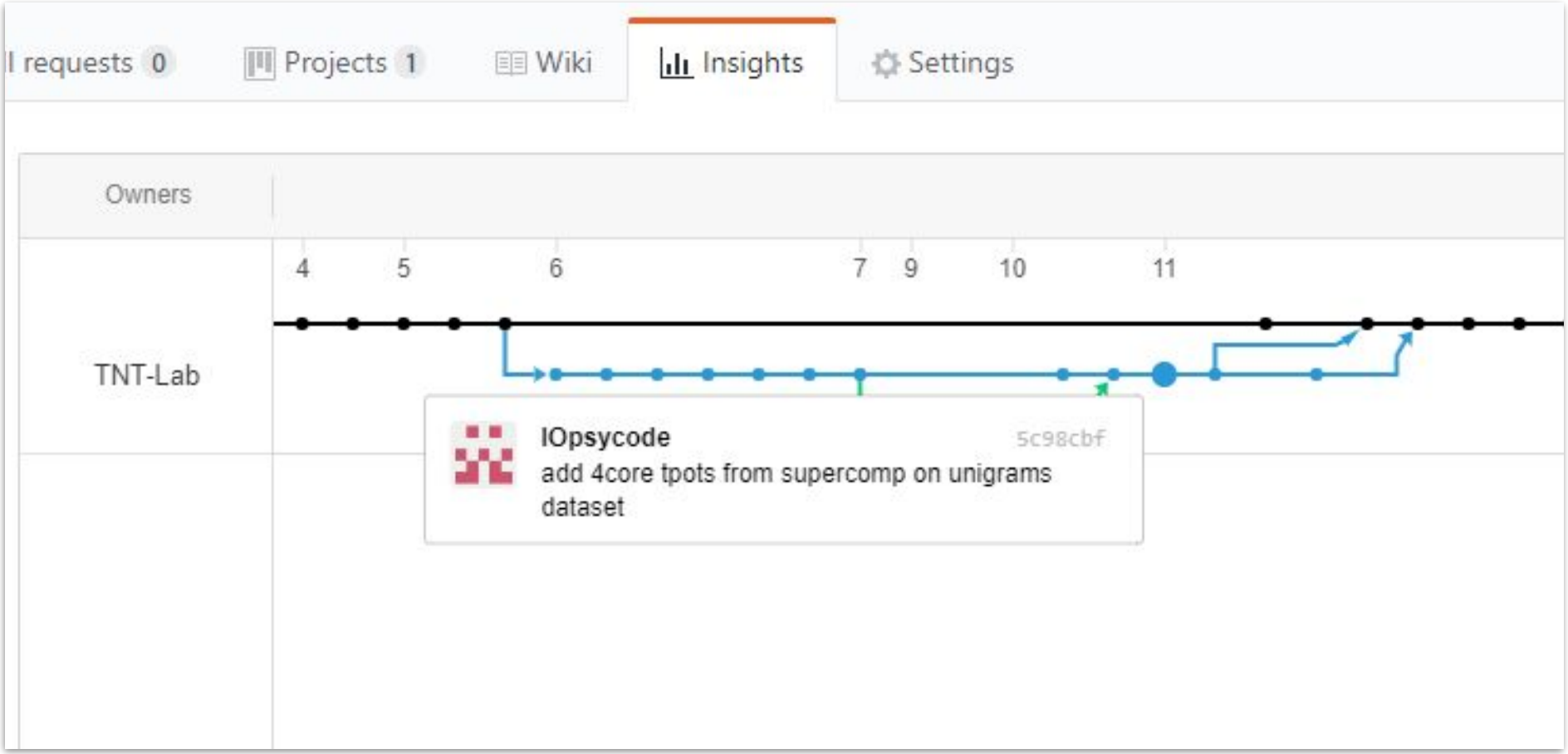
Github - Online Repo built on Git Language Two New Functions

1. **Pull** - get up-to-date version from master repository
2. **Stage** -
3. **Commit** -
    a. **Commit Message** -
4. **Push** - send your version to master repository

Versioning
GitHub



Versioning
GitHub

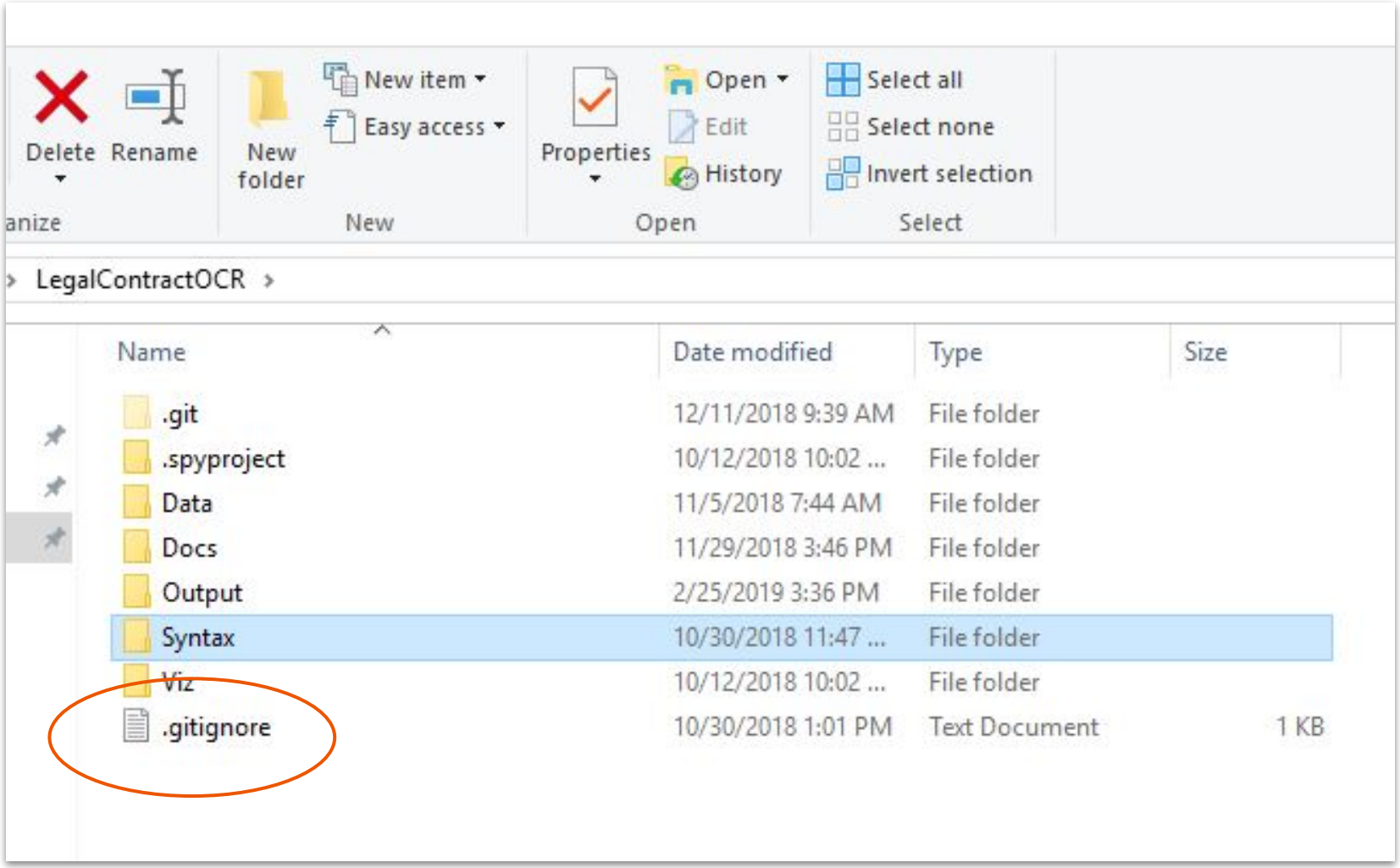


Versioning
Git Ignore

- txt file that lists what you want to ignore
    - i.e., don't commit, don’t push, don’t pull
- Use to protect sensitive data
- Use to avoid workspace conflicts
- Accepts wildcards *

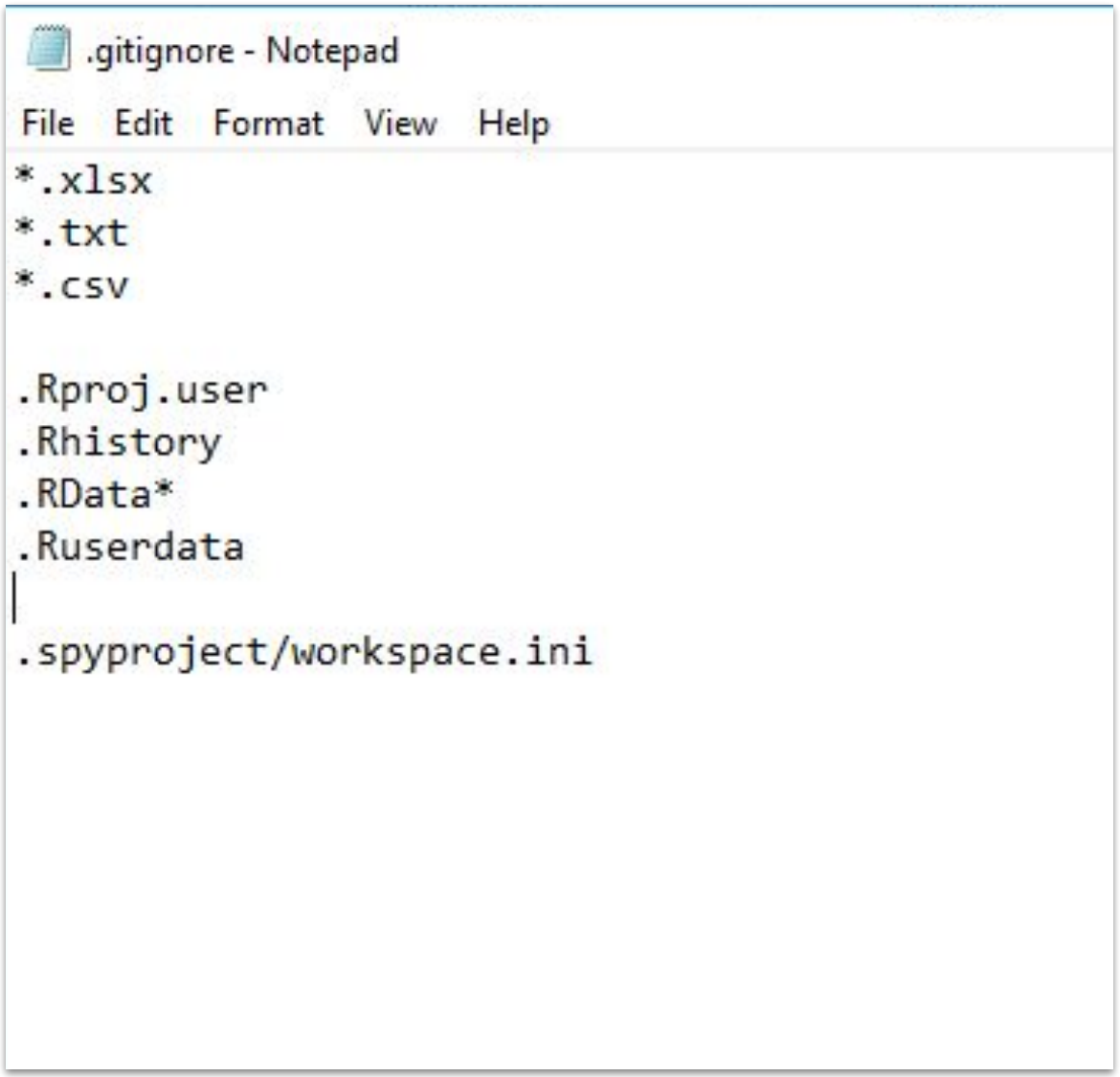
Versioning
Git Ignore

Remember
our Hero?



Versioning
Git Ignore

Example from
Recent Project



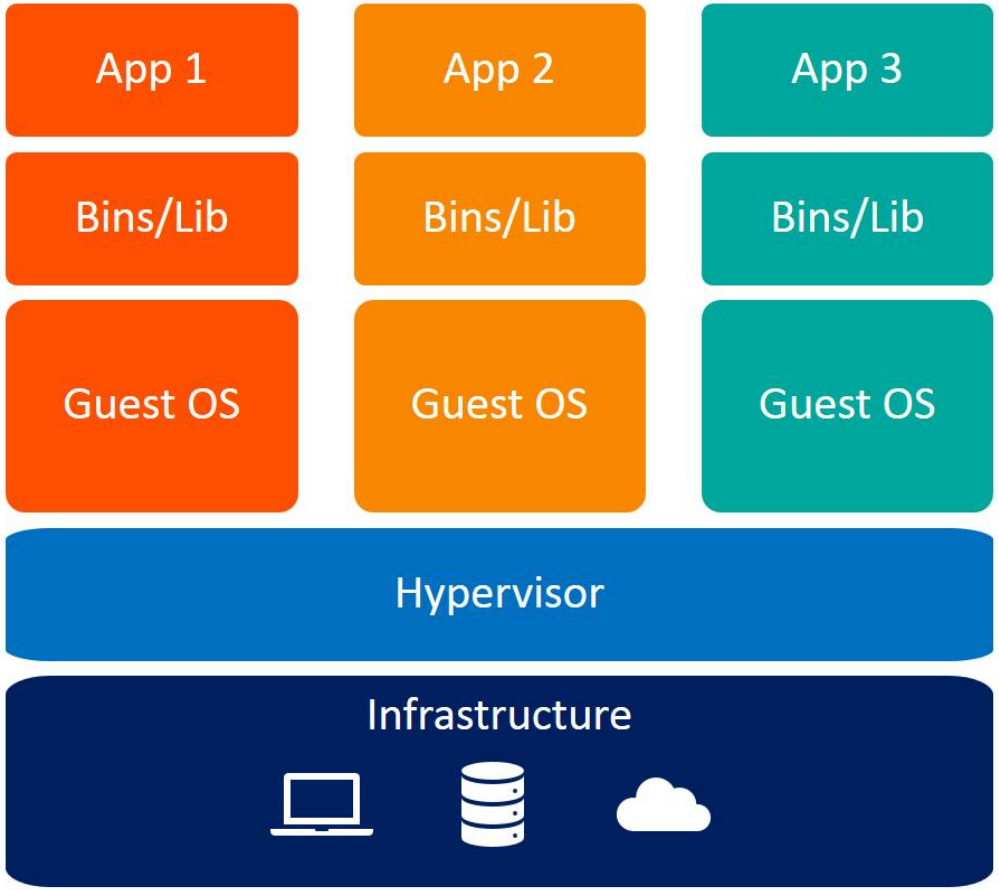
Interactive Notebooks and Containers

- Jupyter
- Docker
- Binder

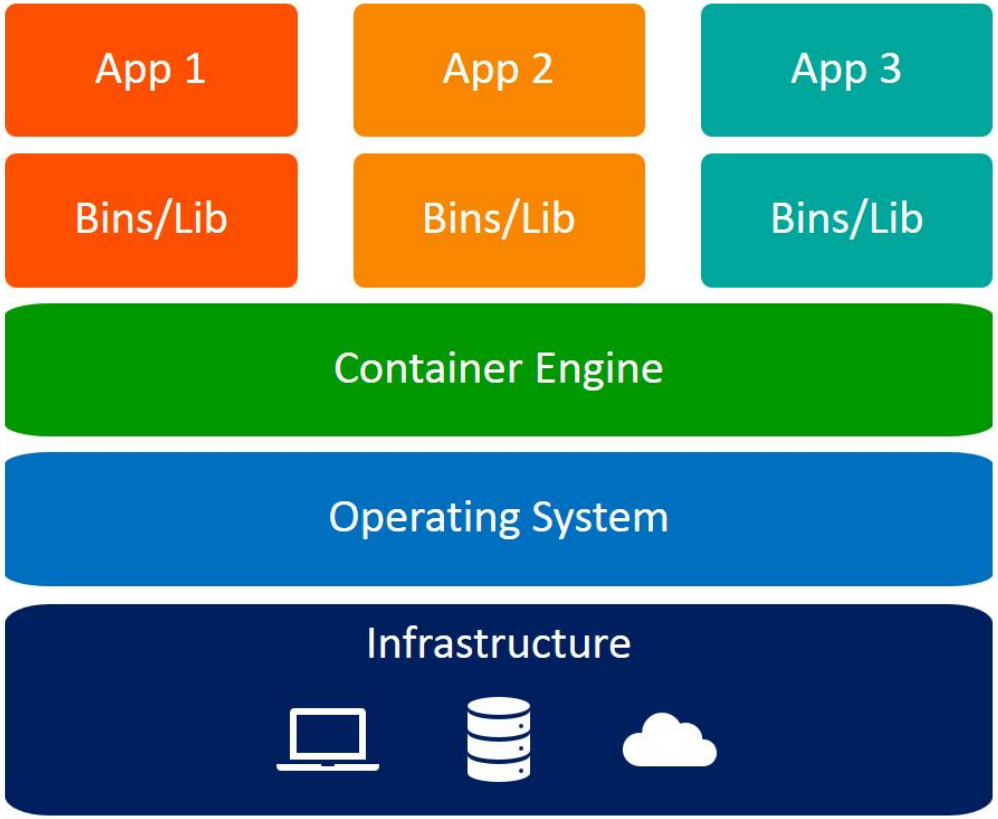
Interactive Notebooks and Containers

- Jupyter - notebook that can run R or Python and markdown in your web browser.
  - Easy to show interactive visualizations
  - Easy to demonstrate concepts to stakeholders
  - You will see example shortly

Interactive Notebooks and Containers



Virtual Machines

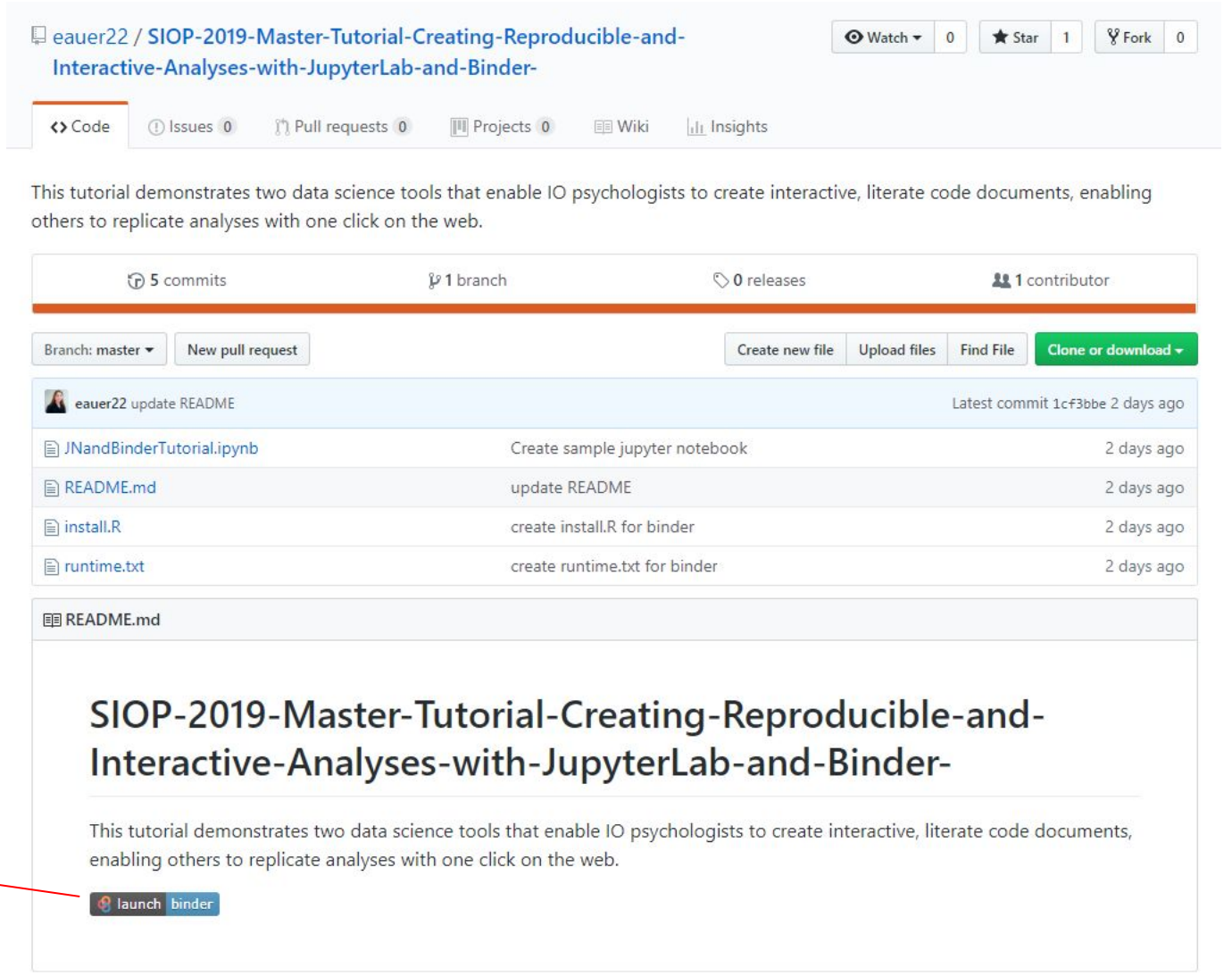


Containers

https://www.zdnet.com/article/what-is-docker-and-why-is-it-so-darn-popular/

Interactive Notebooks and Containers

Binder - easily turn github repos into shareable notebooks



Learning Objective #2: Organize and analyze new sources of data to better address old research questions and develop new research questions not previously testable

Next, we'll be moving to a Jupyter Notebook
To follow along with us:
http://bit.ly/SIOPDataScienceTutorial

Topics Discussed:

Data Sourcing Techniques:

- Trace Data
- APIs & Webscraping

Data Science Methods:

- Natural Language Processing
- Machine Learning

Demo:

1. Access Tweets from Twitter API
2. Clean Tweet Text
3. Predict Tweet Popularity using Machine Learning

Learning Objective #3: Increase the credibility of your research and IO psychology research in general to outside researchers and the public

Data Science can be leveraged to make I-O more insightful, reproducible, and trustworthy

Insightful:

- Measurement
- Triangulation

Reproducible:

- Open-source tools
- Documentation

Trustworthy:

- Increase credibility of I-O research
- Help influence public and inform public policy

An insightful I-O psychology

- Change in measurement of behavior
  - “new” data (e.g., microbehaviors, text data)
- Data science methods add to triangulation of theory
  - Improved prediction
  - New methods for theory-testing
  - Inductive theory building
- Example: dynamic phenomenon (team processes)

A reproducible I-O psychology

- Data science practices make reproducibility ***easier***
- Data science practices make reproducibility ***more explicit***
- Data science practices make reproducibility ***the default***
- Why not make verification immediate and easy for other researchers?

A trustworthy I-O psychology

- I-O's have led the way with:
  - reporting effect sizes
  - meta-analysis
  - data science practices?
- Data science tools can help reduce *p*-hacking and QRPs by increasing transparency

The public and public policy

Empower citizens directly	Influence public policy

• Publish data, code/syntax for public access
  • make comments for yourself and future learners/researchers
  • encourage inclusion
	• Rapid dissemination for evidence-based policy
	  • Research synthesis (e.g., metaBUS)
	• Good PR: better reproducibility & replication, and enhances evidence-based decisions

Additional Resources

Tutorial GitHub page:

http://bit.ly/SIOPDataScienceTutorial

Data Science for Social Scientists Online Course:

http://datascience.tntlab.org/

New Book Chapter:

Landers, R. N., Auer, E. M., Collmus, A. B., & Marin, S. (2019). Data science as a new foundation for insightful, reproducible, and trustworthy social science. In R. N. Landers (Ed.), *Cambridge Handbook of Technology and Employee Behavior* (pp. 761-789). New York, NY: Cambridge University Press.

Thank you!