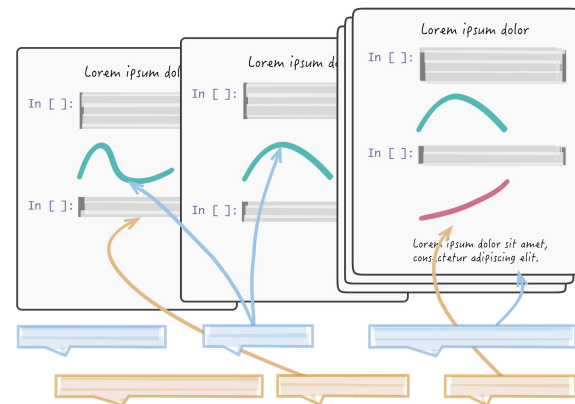


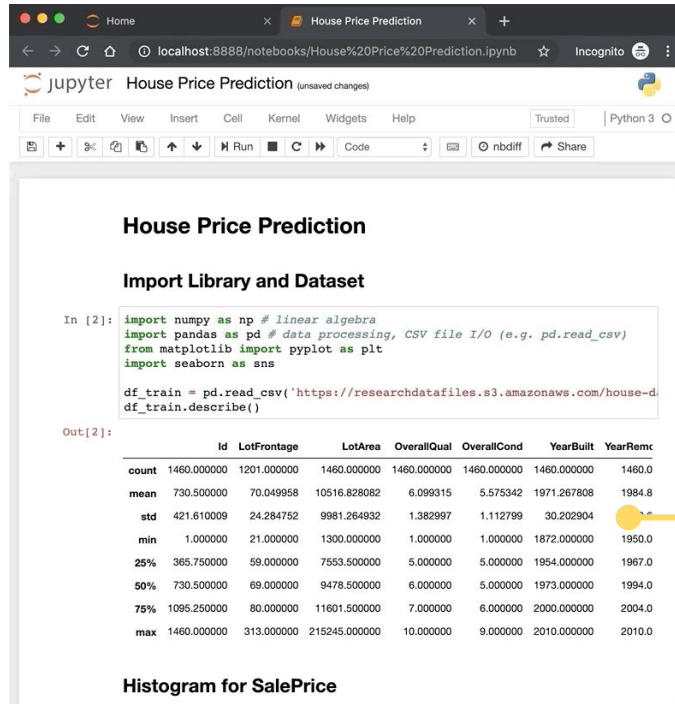
# Callisto: Capturing the “Why” by Connecting Conversations with Computational Narratives

April Yi Wang<sup>1</sup>, Zihan Wu<sup>2</sup>, Chris Brooks<sup>1</sup>, Steve Oney<sup>1</sup>

<sup>1</sup>University of Michigan

<sup>2</sup>Tsinghua University



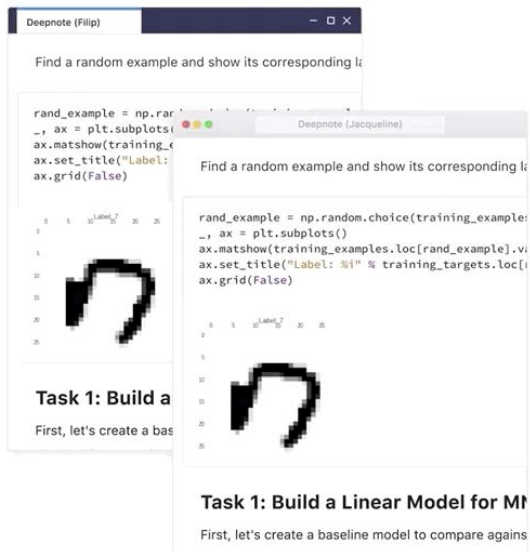


Computational notebooks allow data scientists to document and replicate the exploration process.

Code  
Explanatory text  
Intermediate output

# Collaborative Data Science

## Motivation



Deepnote (Filip)

Find a random example and show its corresponding label

```
rand_example = np.random.choice(training_examples)
_, ax = plt.subplots(1, 1)
ax.imshow(training_examples[rand_example].v)
ax.set_title("Label: %i" % training_targets.loc[rand_example])
ax.grid(False)
```

Deepnote (Jacqueline)

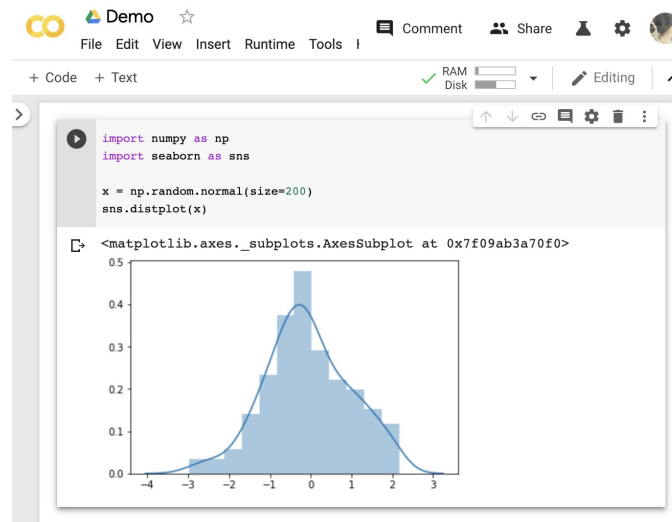
Find a random example and show its corresponding label

```
rand_example = np.random.choice(training_examples)
_, ax = plt.subplots(1, 1)
ax.imshow(training_examples.loc[rand_example].v)
ax.set_title("Label: %i" % training_targets.loc[rand_example])
ax.grid(False)
```

**Task 1: Build a Linear Model for MNIST**

First, let's create a baseline model to compare against

Deepnote



Demo

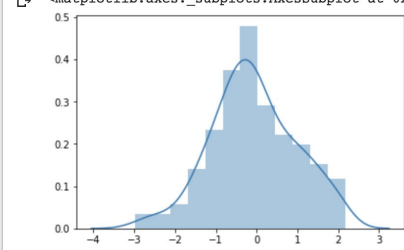
File Edit View Insert Runtime Tools Help

+ Code + Text

```
import numpy as np
import seaborn as sns

x = np.random.normal(size=200)
sns.distplot(x)
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f09ab3a70f0>



Google colab

# Challenges in Collaboration

## Motivation

Remove Outliers by Stats



```
In [ ]: #data = data[(np.abs(stats.zscore(data)) < 5).all(axis=

In [ ]: ##### TODO #####
# any additional analysis

In [ ]: #print(np.unique(df_train['YrSold'], return_counts=True)
#np.where(df_train['YrSold'] == 2007)
abc = np.where(df_train['YrSold'] == 2008)[0]
df_subset = df_train['SalePrice']
useful_subset = dict(df_subset)
for x in abc:
    useful_subset[x]
    x

In [ ]: # print(np.unique(df_train['YearBuilt'], return_counts=
abc = np.where(df_train['YrSold'] == 2008)
abc = list(abc)[0]
prices = []
for x in abc:
    prices.append(df_train['SalePrice'][x])

print(np.mean(prices) - np.mean(df_train['SalePrice']))

In [18]: np.unique(df_train['YrSold'])
Out[18]: array([2006, 2007, 2008, 2009, 2010])
```

Alice and Bob joined the shared notebook.  
They decided to explore outliers together.



# Challenges in Collaboration

## Motivation

Remove Outliers by Stats

```
In [ ]: #data = data[(np.abs(stats.zscore(data)) < 5).all(axis=
```



```
In [ ]: ##### TODO #####  
# any additional analysis
```

```
In [ ]: #print(np.unique(df_train['YrSold'], return_counts=True)  
#np.where(df_train['YrSold'] == 2007)  
abc = np.where(df_train['YrSold'] == 2008)[0]  
df_subset = df_train['SalePrice']  
useful_subset = dict(df_subset)  
for x in abc:  
    useful_subset[x]  
    x
```

```
In [ ]: # print(np.unique(df_train['YearBuilt'], return_counts=  
abc = np.where(df_train['YrSold'] == 2008)  
abc = list(abc)[0]  
prices = []  
for x in abc:  
    prices.append(df_train['SalePrice'][x])  
  
print(np.mean(prices) - np.mean(df_train['SalePrice']))
```

```
In [18]: np.unique(df_train['YrSold'])
```

```
Out[18]: array([2006, 2007, 2008, 2009, 2010])
```

Alice and Bob joined the shared notebook.    
They decided to explore outliers together.

Bob edited his analysis in cell #10. 

# Challenges in Collaboration

## Motivation

Remove Outliers by Stats

```
In [ ]: #data = data[(np.abs(stats.zscore(data)) < 5).all(axis=
```

```
In [ ]: ##### TODO #####  
# any additional analysis
```

```
In [ ]: #print(np.unique(df_train['YrSold'], return_counts=True)  
#np.where(df_train['YrSold'] == 2007)  
abc = np.where(df_train['YrSold'] == 2008)[0]  
df_subset = df_train['SalePrice']  
useful_subset = dict(df_subset)  
for x in abc:  
    useful_subset[x]  
    x
```

```
In [ ]: # print(np.unique(df_train['YearBuilt'], return_counts=  
abc = np.where(df_train['YrSold'] == 2008)  
abc = list(abc)[0]  
prices = []  
for x in abc:  
    prices.append(df_train['SalePrice'][x])  
  
print(np.mean(prices) - np.mean(df_train['SalePrice']))
```

```
In [18]: np.unique(df_train['YrSold'])
```

```
Out[18]: array([2006, 2007, 2008, 2009, 2010])
```

Alice and Bob joined the shared notebook. They decided to explore outliers together.



Bob edited his analysis in cell #10.



Alice edited her analysis in cell #11.



# Challenges in Collaboration

## Motivation

Remove Outliers by Stats

```
In [ ]: #data = data[(np.abs(stats.zscore(data)) < 5).all(axis=

In [ ]: ##### TODO #####
# any additional analysis

In [ ]: #print(np.unique(df_train['YrSold'], return_counts=True)
#np.where(df_train['YrSold'] == 2007)
abc = np.where(df_train['YrSold'] == 2008)[0]
df_subset = df_train['SalePrice']
useful_subset = dict(df_subset)
for x in abc:
    useful_subset[x]
    x

In [ ]: # print(np.unique(df_train['YearBuilt'], return_counts=
abc = np.where(df_train['YrSold'] == 2008)
abc = list(abc)[0]
prices = []
for x in abc:
    prices.append(df_train['SalePrice'][x])

print(np.mean(prices) - np.mean(df_train['SalePrice']))

In [18]: np.unique(df_train['YrSold'])
Out[18]: array([2006, 2007, 2008, 2009, 2010])
```

Alice and Bob joined the shared notebook. They decided to explore outliers together.



Bob edited his analysis in cell #10.



Alice edited her analysis in cell #11.



After discussing, they agreed to use Alice's code and moved on to the next step.



# Challenges in Collaboration

## Motivation

Remove Outliers by Stats

```
In [ ]: #data = data[(np.abs(stats.zscore(data)) < 5).all(axis=
```

```
In [ ]: ##### TODO #####  
# any additional analysis
```

```
In [ ]: #print(np.unique(df_train['YrSold'], return_counts=True)  
#np.where(df_train['YrSold'] == 2007)  
abc = np.where(df_train['YrSold'] == 2008)[0]  
df_subset = df_train['SalePrice']  
useful_subset = dict(df_subset)  
for x in abc:  
    useful_subset[x]  
    x
```

```
In [ ]: # print(np.unique(df_train['YearBuilt'], return_counts=  
abc = np.where(df_train['YrSold'] == 2008)  
abc = list(abc)[0]  
prices = []  
for x in abc:  
    prices.append(df_train['SalePrice'][x])  
  
print(np.mean(prices) - np.mean(df_train['SalePrice']))
```

```
In [18]: np.unique(df_train['YrSold'])
```

```
Out[18]: array([2006, 2007, 2008, 2009, 2010])
```

Alice and Bob joined the shared notebook. They decided to explore outliers together.



Bob edited his analysis in cell #10.



Alice edited her analysis in cell #11.



After discussing, they agreed to use Alice's code and moved on to the next step.



2 days later...

Another colleague, Charlie joined the shared notebook.





# Challenges in Collaboration

*Remove Outliers by Stats*

```
In [ ]: #data = data[(np.abs(stats.zscore(data)) < 5).all(axis=
```

```
In [ ]: ##### TODO #####  
# any additional analysis
```

```
In [ ]: #print(np.unique(df_train['YrSold'], return_counts=True)  
#np.where(df_train['YrSold'] == 2007)  
abc = np.where(df_train['YrSold'] == 2008)[0]  
df_subset = df_train['SalePrice']  
useful_subset = dict(df_subset)  
for x in abc:  
    useful_subset[x]  
    x
```

```
In [ ]: # print(np.unique(df_train['YearBuilt'], return_counts=  
abc = np.where(df_train['YrSold'] == 2008)  
abc = list(abc)[0]  
prices = []  
for x in abc:  
    prices.append(df_train['SalePrice'][x])  
  
print(np.mean(prices) - np.mean(df_train['SalePrice']))
```

```
In [18]: np.unique(df_train['YrSold'])
```

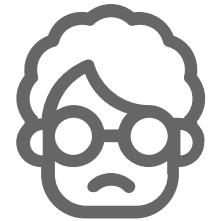
```
Out[18]: array([2006, 2007, 2008, 2009, 2010])
```

## Motivation

What the hell  
is going on?

Unclear code  
Random variable names  
Useless documentation  
Ambiguous cell orders

.....



- Difficult to maintain an updated explanation and a clean notebook during the exploration process  
(Rule et al. CHI18)
- Write lower quality code, change the execution order, or accidentally overwrite important analyses while iterating on different ideas  
(Kery et al., CHI19, Head et al., CHI19)
- Can be amplified in a collaborative setting where it is important to keep a shared understanding of past design decisions across team members  
(Wang et al., CSCW19, Koesten et al., CHI19, Kery et al., VL/HCC17)

# Challenges in Collaboration

## Motivation

Remove Outliers by Stats

```
In [ ]: #data = data[(np.abs(stats.zscore(data)) < 5).all(axis=
```

```
In [ ]: ##### TODO #####  
# any additional analysis
```

```
In [ ]: #print(np.unique(df_train['YrSold'], return_counts=True)  
#np.where(df_train['YrSold'] == 2007)  
abc = np.where(df_train['YrSold'] == 2008)[0]  
df_subset = df_train['SalePrice']  
useful_subset = dict(df_subset)  
for x in abc:  
    useful_subset[x]  
    x
```

```
In [ ]: # print(np.unique(df_train['YearBuilt'], return_counts=  
abc = np.where(df_train['YrSold'] == 2008)  
abc = list(abc)[0]  
prices = []  
for x in abc:  
    prices.append(df_train['SalePrice'][x])  
  
print(np.mean(prices) - np.mean(df_train['SalePrice']))
```

```
In [18]: np.unique(df_train['YrSold'])
```

```
Out[18]: array([2006, 2007, 2008, 2009, 2010])
```

The screenshot shows a Slack interface for a channel named "Data Science G...". The left sidebar lists various channels and direct messages. The "Channels" section includes "# general", "# houseprice" (which is highlighted in blue), and "# random". Below this, there is an option to "Add a channel". The "Direct Messages" section lists "Slackbot", "Peggy (you)", "Alice", and "Bob". At the bottom, there is an option to "Add apps".

#houseprice


☆ | 👤 3 | 🔔 0 | ✎ Add a topic

@Alice created this channel today. This is the very beginning of u


🔗 Set a purpose + Add an app 👤 Add people to this channel





 Alice 2:53 AM  
joined #houseprice along with Bob.

 Bob 2:58 AM  
so I think we need to solve these 6 tasks  
it looks like it's related to trying to predict house prices from a numbe


 Alice 2:59 AM  
yes

 Bob 2:59 AM  
and luckily we have lots of helpful outline code.  
let's look at this together

 Alice 2:59 AM  
oh sure

 Bob 2:59 AM  
LotArea is probably important, i put it in our list

 Alice 2:59 AM  
I think maybe GrLivArea instead? it refers to the living area..

 Bob 2:59 AM  
is it like general living area?

 Alice 3:00 AM

# Challenges in Collaboration

## Motivation

Remove Outliers by Stats

```
In [ ]: #data = data[(np.abs(stats.zscore(data)) < 5).all(axis=
```

```
In [ ]: ##### TODO #####  
# any additional analysis
```

```
In [ ]: #print(np.unique(df_train['YrSold'], return_counts=True)  
#np.where(df_train['YrSold'] == 2007)  
abc = np.where(df_train['YrSold'] == 2008)[0]  
df_subset = df_train['SalePrice']  
useful_subset = dict(df_subset)  
for x in abc:  
    useful_subset[x]  
    x
```

```
In [ ]: # print(np.unique(df_train['YearBuilt'], return_counts=  
abc = np.where(df_train['YrSold'] == 2008)  
abc = list(abc)[0]  
prices = []  
for x in abc:  
    prices.append(df_train['SalePrice'][x])  
  
print(np.mean(prices) - np.mean(df_train['SalePrice']))
```

```
In [18]: np.unique(df_train['YrSold'])
```

```
Out[18]: array([2006, 2007, 2008, 2009, 2010])
```

The screenshot shows a Slack interface for a channel named "#houseprice". The channel was created by @Alice. The conversation includes:

- Alice (2:53 AM) joined the channel along with Bob.
- Bob (2:58 AM) says, "so I think we need to solve these 6 tasks".
- Alice (2:59 AM) replies, "it looks like it's related to trying to predict house prices from a r".
- Alice (2:59 AM) says "yes".
- Bob (2:59 AM) says, "and luckily we have lots of helpful outline code. let's look at this together".
- Alice (2:59 AM) says "oh sure".
- Bob (2:59 AM) says, "LotArea is probably important, i put it in our list".
- Alice (2:59 AM) is partially visible at the bottom.



# Challenges in Collaboration

## Motivation

Remove Outliers by Stats

```
In [ ]: #data = data[(np.abs(stats.zscore(data)) < 5).all(axis=
```

```
In [ ]: ##### TODO #####  
# any additional analysis
```

```
In [ ]: #print(np.unique(df_train['YrSold'], return_counts=True)  
#np.where(df_train['YrSold'] == 2007)  
abc = np.where(df_train['YrSold'] == 2008)[0]  
df_subset = df_train['SalePrice']  
useful_subset = dict(df_subset)  
for x in abc:  
    useful_subset[x]  
    x
```

```
In [ ]: # print(np.unique(df_train['YearBuilt'], return_counts=  
abc = np.where(df_train['YrSold'] == 2008)  
abc = list(abc)[0]  
prices = []  
for x in abc:  
    prices.append(df_train['SalePrice'][x])  
  
print(np.mean(prices) - np.mean(df_train['SalePrice']))
```

```
In [18]: np.unique(df_train['YrSold'])
```

```
Out[18]: array([2006, 2007, 2008, 2009, 2010])
```

The screenshot shows a Slack interface for a channel named 'Data Science G...'. The left sidebar lists several channels: '# general', '# houseprice' (which is selected and highlighted in blue), and '# random'. Below the channels list are options to '+ Add a channel', '+ Invite people', and '+ Add apps'. The main area of the interface shows a list of direct messages with names like 'Peggy', 'Alice', and 'Bob'.

#houseprice

☆ | 👤 3 | 🔔 0 | ✎ Add a topic

@Alice created this channel today. This is the very beginning of u

✎ Set a purpose + Add an app 👤 Add people to this channel

Alice 2:53 AM  
joined #houseprice along with Bob.

Bob 2:58 AM  
so I think we need to solve these 6 tasks  
it looks like it's related to trying to predict house prices from a numbe

Alice 2:59 AM  
yes

Bob 2:59 AM  
and luckily we have lots of helpful outline code.  
let's look at this together

Alice 2:59 AM  
oh sure

Bob 2:59 AM  
LotArea is probably important, i put it in our list

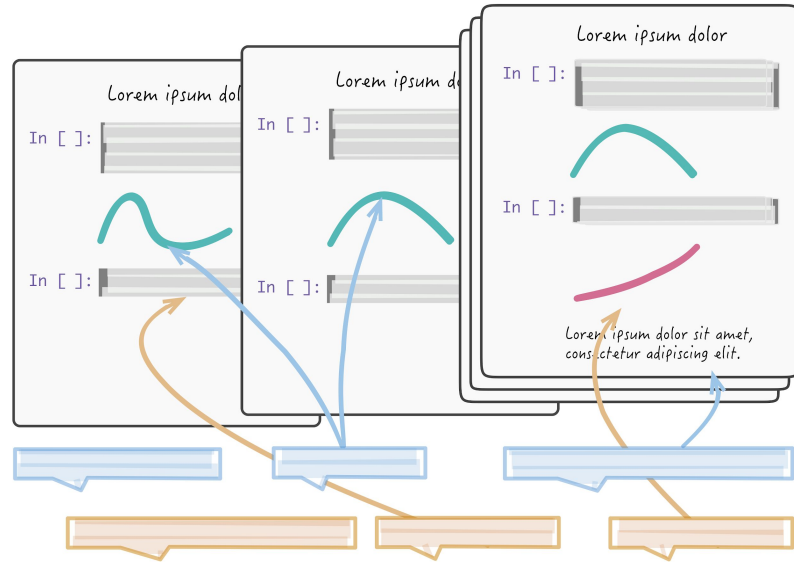
Alice 2:59 AM  
I think maybe GrLivArea instead? it refers to the living area..

Bob 2:59 AM  
is it like general living area?

Alice 3:00 AM



# We propose to improve collaborative data science by connecting discussions with computational notebooks.



### RQ: How can discussions be useful for explaining the data-exploration process?

```
Remove Outliers by Stats

In [ ]: #data = data[(np.abs(stats.zscore(data)) < 5)].all(axis=

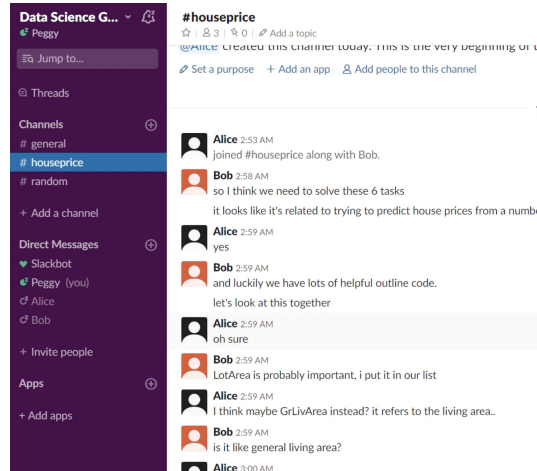
In [ ]: ##### TODO #####
# any additional analysis

In [ ]: #print(np.unique(df_train['YrSold'], return_counts=True)
#np.where(df_train['YrSold'] == 2007)
abc = np.where(df_train['YrSold'] == 2008)[0]
df_subset = df_train['SalePrice']
useful_subset = dict(df_subset)
for x in abc:
    useful_subset[x]
    x

In [ ]: # print(np.unique(df_train['YearBuilt'], return_counts=
abc = np.where(df_train['YrSold'] == 2008)
abc = list(abc)[0]
prices = []
for x in abc:
    prices.append(df_train['SalePrice'][x])

print(np.mean(prices) - np.mean(df_train['SalePrice']))

In [18]: np.unique(df_train['YrSold'])
Out[18]: array([2006, 2007, 2008, 2009, 2010])
```



- Six data science students working remotely in pairs
- Collected and analyzed 760 chat messages

## Analyzing Chat Messages

## Formative Study

### **Purpose**

1) Reflecting; 2) Planning; 3) Check-in; 4) Cooperation; 5) Out-of-scope



## Analyzing Chat Messages

## Formative Study

### **Purpose**

1) Reflecting; 2) Planning; 3) Check-in; 4) Cooperation; 5) Out-of-scope

### **Relevance**

- 1) Ideas that were only discussed but never implemented
- 2) Ideas that had not yet been implemented when the message was sent, but appeared in the notebook later
- 3) Ideas that had been implemented in the notebook when the message was sent, but did not appear in the final notebook
- 4) Ideas that had been implemented when the message was sent and appeared in the final notebook

# Analyzing Chat Messages

## Formative Study

### Purpose

1) Reflecting; 2) Planning; 3) Check-in; 4) Cooperation; 5) Out-of-scope

### Relevance

- 1) Ideas that were only discussed but never implemented
- 2) Ideas that had not yet been implemented when the message was sent, but appeared in the notebook later
- 3) Ideas that had been implemented in the notebook when the message was sent, but did not appear in the final notebook;
- 4) Ideas that had been implemented when the message was sent and appeared in the final notebook

### Granularity

- 1) Directly referred to a specific line of code
- 2) Directly referred to the output of a cell
- 3) High-level ideas across multiple cells

## **Formative Study**

---

**Chat messages are useful for explaining the exploration process.**

**Chat messages are difficult to follow.**

**Notebook elements are frequently referred to in chat messages.**

The screenshot displays a JupyterLab environment. At the top, the Jupyter logo and 'demo' are visible, along with the text 'Last Checkpoint: 01/31/2020 (unsaved changes)'. On the right, there are buttons for 'Logout' and 'Control Panel', and a Python 3 version indicator. The main toolbar includes 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. Below the toolbar, there are icons for 'Run', 'Stop', and 'Refresh', along with a 'Markdown' dropdown, a 'Share' button, and a 'Filter' dropdown. User avatars for 'aprilwang', 'bob-participant', and 'alice-participant' are shown.

The notebook content is titled 'House Price Prediction'. It contains the following text:

You will work together on Exploratory Data Analysis (EDA) for predicting house prices.

The process is described as the following:

1. **Understand the problem.** First, let's take a look at each variable and do a philosophical analysis about their meaning and impact.
2. **Univariate study.** Next, let's take a look at the dependent variable (`SalePrice`).
3. **Multivariate study.** Let's try to understand how the dependent variable and independent variables relate.
4. **Basic cleaning.** Let's come up a *plan* for cleaning the dataset and handle the missing data, outliers and categorical variables.
5. **Test on linear regression model.** (provided)

You need to finish 6 tasks in total. You will find the detailed instructions as you navigate through the notebook.

- Task 1. Philosophical analysis
- Task 2. Implement a histogram of `SalePrice`
- Task 3. Analyze the Correlation (1)
- Task 4. Analyze the Correlation (2)
- Task 5. Analyze the Correlation (3)
- Task 6. Make a Plan for Future Exploration

If you need some help with pandas and seaborn, here are two cheatsheets with basic operations for them:  
Pandas: <http://datacamp-community-prod.s3.amazonaws.com/dbed353d-2757-4617-8206-8767ab379ab3>  
Seaborn: [https://s3.amazonaws.com/assets.datacamp.com/blog\\_assets/Python\\_Seaborn\\_Cheat\\_Sheet.pdf](https://s3.amazonaws.com/assets.datacamp.com/blog_assets/Python_Seaborn_Cheat_Sheet.pdf)  
Feel free to search the internet to get any information you need to complete the task.

The notebook is currently on the section '0. Loading libraries and dataset'. The code cell shows:

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
```

On the right side, there is a 'Chat' window with a search bar and a list of messages:

- bob-participant (1:54PM): hi, I'm bob, I'm excited to solve our data science problem together.
- alice-participant (1:55PM): hi, I'm Alice!
- bob-participant (1:56PM): so I think we need to solve these 6 tasks
- bob-participant (1:56PM): it looks like it's related to trying to predict house prices from a number of variables
- alice-participant (1:56PM): write your message

The screenshot displays the Jupyter web interface. At the top, the Jupyter logo and name are visible, along with a search bar and navigation buttons for 'Share', 'Filter', 'Bob', and 'Alice'. The main content area shows a notebook titled 'Current Notebook' updated '5 minutes ago'. The notebook content includes a title 'House Price Prediction', a question 'Is there a linear relationship between GrLivArea and SalePrice?', and a code cell 'In [8]:' containing a scatter plot command. Below the code is a scatter plot of blue data points with two outliers circled in black. The left sidebar shows a list of recent cell executions by Alice and Bob. The right sidebar features a 'Chat' window with a search bar and a message history showing a conversation about outliers and a financial crash.

```
@@ -1,7 +1,7 @@
# scatter plot grlivarea/saleprice
+ data.plot.scatter(x='GrLivArea', y='SalePrice');
```

**House Price Prediction**

Is there a linear relationship between GrLivArea and SalePrice?

In [8]: # scatter plot grlivarea/saleprice  
data.plot.scatter(x='GrLivArea', y='SalePrice');

In [ ]:   
In [ ]:

**Chat**

search

Alice 3:30pm  
What about these outliers? marker

Bob 3:31pm  
Let me check their values.

Bob 3:33pm  
They were both sold in 2008 cell

Alice 3:34pm  
Financial crash?

Write your message

# Enabling Sharing and Real-Time Collaboration

## Design of Callisto

The screenshot displays the Jupyter web interface. At the top, the Jupyter logo and name are visible. A navigation bar includes a 'Share' button (highlighted with a yellow box), a 'Filter' button, and user profile buttons for 'Bob' and 'Alice'. The main content area shows a notebook titled 'Current Notebook' with a timestamp of '5 minutes ago'. The notebook title is 'House Price Prediction' and the question is 'Is there a linear relationship between GrLivArea and SalePrice?'. The code cell shows the following code:

```
In [8]: # scatter plot grlivarea/saleprice
data.plot.scatter(x='GrLivArea', y='SalePrice');
```

The output is a scatter plot of blue dots representing data points. A black hand-drawn oval highlights a cluster of points on the right side of the plot, indicating outliers. Below the plot are input fields for the next code cell.

On the left side, a chat window shows a conversation:

- Alice (3:30pm): Executed a modified cell. Code: `@@ -1,7 +1,7 @@`  
`# scatter plot grlivarea/s`  
`+ data.plot.scatter(x='GrLiv`
- Bob (3:31pm): Executed a modified cell.
- Alice (3:35pm): Executed a modified cell.

On the right side, a chat window shows a conversation:

- Alice (3:30pm): What about these outliers? `marker`
- Bob (3:31pm): Let me check their values.
- Bob (3:33pm): They were both sold in 2008 `cell`
- Alice (3:34pm): Financial crash?

At the bottom right, there is a text input field labeled 'Write your message' and a send icon.

# Enabling Sharing and Real-Time Collaboration

## Design of Callisto

The screenshot displays the Jupyter web interface. At the top, the Jupyter logo and name are visible. Below the logo, there are input fields for a URL and a search bar. To the right of these fields are buttons for 'Share' (with a checkmark icon), 'Filter' (with a dropdown arrow icon), and user selection buttons for 'Bob' (with a home icon) and 'Alice' (with a person icon). The main content area shows a notebook titled 'Current Notebook' with a timestamp of '5 minutes ago'. The notebook's title is 'House Price Prediction' and the question is 'Is there a linear relationship between GrLivArea and SalePrice?'. The code cell 'In [8]:' contains the following code: 

```
# scatter plot grlivarea/saleprice
data.plot.scatter(x='GrLivArea', y='SalePrice');
```

 Below the code is a scatter plot showing a positive correlation between GrLivArea and SalePrice. A black hand-drawn oval highlights a cluster of outliers in the upper right corner of the plot. The left sidebar shows a list of recent activity: Alice executed a modified cell at 3:30pm, Bob executed a modified cell at 3:31pm, and Alice executed a modified cell at 3:35pm. The right sidebar is a chat window titled 'Chat' with a search bar. The chat history shows: Alice (3:30pm) asking 'What about these outliers?' with a 'marker' tag; Bob (3:31pm) replying 'Let me check their values.'; Bob (3:33pm) replying 'They were both sold in 2008' with a 'cell' tag; and Alice (3:34pm) asking 'Financial crash?'. At the bottom of the chat is a text input field 'Write your message' and a send button.

The screenshot displays the Jupyter web interface. At the top, the Jupyter logo and name are visible, along with a search bar and navigation buttons for 'Share', 'Filter', 'Bob', and 'Alice'. The main workspace shows a notebook titled 'Current Notebook' updated '5 minutes ago'. The notebook content includes a title 'House Price Prediction', a question 'Is there a linear relationship between GrLivArea and SalePrice?', and a code cell 'In [8]:' containing a scatter plot command. Below the code is a scatter plot of blue data points with two outliers circled in black. A yellow box highlights the 'B A' collaboration icons next to the code cell. On the left, a sidebar shows chat messages from Alice and Bob. On the right, a 'Chat' window shows a conversation about outliers and a 'Financial crash?' question.

```
@@ -1,7 +1,7 @@
# scatter plot grlivarea/saleprice
+ data.plot.scatter(x='GrLivArea', y='SalePrice');
```

In [8]: # scatter plot grlivarea/saleprice  
data.plot.scatter(x='GrLivArea', y='SalePrice');

In [ ]:

In [ ]:



The screenshot displays the Jupyter web interface. At the top, the Jupyter logo and name are visible, along with a search bar and navigation buttons for 'Share', 'Filter', 'Bob', and 'Alice'. The main notebook area is titled 'Current Notebook' and shows a code cell with the following content:

```
Current Notebook 5 minutes ago
```

### House Price Prediction

Is there a linear relationship between GrLivArea and SalePrice?

```
In [8]: # scatter plot grlivarea/saleprice
data.plot.scatter(x='GrLivArea', y='SalePrice');
```

The code cell is followed by a scatter plot showing a positive correlation between GrLivArea and SalePrice. Two outliers are circled in black. The plot includes a pencil icon for editing and a refresh icon.

On the left side, there are three chat messages from Alice and Bob:

- Alice (3:30pm): Executed a modified cell. Code: @@ -1,7 +1,7 @@ # scatter plot grlivarea/s + data.plot.scatter(x='GrLiv
- Bob (3:31pm): Executed a modified cell.
- Alice (3:35pm): Executed a modified cell.

On the right side, a chat window is open, showing a search bar and a list of messages:

- Alice (3:30pm): What about these outliers? marker
- Bob (3:31pm): Let me check their values.
- Bob (3:33pm): They were both sold in 2008 cell
- Alice (3:34pm): Financial crash?

The chat window has a 'Write your message' input field and a send button.

# Enabling Sharing and Real-Time Collaboration

## Design of Callisto

**jupyter**

Share Filter Bob Alice

Current Notebook 5 minutes ago

### House Price Prediction

Is there a linear relationship between GrLivArea and SalePrice?

```
In [8]: # scatter plot grlivarea/saleprice
data.plot.scatter(x='GrLivArea', y='SalePrice');
```

In [ ]:

In [ ]:

**Chat**

search

Alice 3:30pm  
What about these outliers? marker

Bob 3:31pm  
Let me check their values.

Bob 3:33pm  
They were both sold in 2008 cell

Alice 3:34pm  
Financial crash?

Write your message

The screenshot displays the JupyterLab interface. At the top, the Jupyter logo and name are visible, along with a search bar and navigation buttons for 'Share', 'Filter', 'Bob', and 'Alice'. The main notebook area is titled 'House Price Prediction' and contains the question 'Is there a linear relationship between GrLivArea and SalePrice?'. Below the question is a code cell with the following code: 

```
In [8]: # scatter plot grlivarea/saleprice
data.plot.scatter(x='GrLivArea', y='SalePrice');
```

 The code cell is followed by a scatter plot showing a positive correlation between GrLivArea and SalePrice. A black circle highlights two outlier points in the upper right corner of the plot. To the left of the notebook, a sidebar shows a list of user activities: Alice at 3:30pm executed a modified cell with code including `data.plot.scatter(x='GrLiv`; Bob at 3:31pm executed a modified cell; and Alice at 3:35pm executed a modified cell. To the right, a chat window is open, showing a conversation: Alice at 3:30pm asks 'What about these outliers?' with a yellow box around the word 'marker'; Bob at 3:31pm replies 'Let me check their values.'; Alice at 3:33pm replies 'They were both sold in 2008 cell' with a yellow box around the word 'cell'; and Alice at 3:34pm asks 'Financial crash?'. A yellow box labeled 'References' is positioned over the chat window.

The screenshot displays the Jupyter web interface. At the top, the Jupyter logo and name are visible, along with navigation buttons for 'Share', 'Filter', 'Bob', and 'Alice'. The main area is divided into three sections:

- Left Panel (Activity Log):** Shows a list of recent actions by users Alice and Bob, such as 'Executed a modified cell' at various times.
- Center Panel (Notebook):** Titled 'House Price Prediction', it contains a code cell with the following code:

```
In [8]: # scatter plot grlivarea/saleprice
data.plot.scatter(x='GrLivArea', y='SalePrice'),
```

The code is followed by a scatter plot of blue data points. A yellow box highlights the variable 'SalePrice' in the code, and a yellow arrow points from this box to the 'Code References' section in the chat panel. Below the plot are empty input fields for 'In []:'.
- Right Panel (Chat):** A chat window with a search bar and a list of messages. The messages include:
  - Alice (3:30pm): 'What about these outliers? marker'
  - Bob (3:31pm): 'Let me check their values.'
  - Bob (3:33pm): 'They were both sold in 2008 cell'
  - Alice (3:34pm): 'Financial crash?'A yellow box highlights the 'Code References' icon in the chat input area, with a yellow arrow pointing from the 'SalePrice' variable in the code to this icon.

The screenshot displays the Jupyter web interface. At the top, the Jupyter logo and name are visible, along with a search bar and navigation buttons for 'Share', 'Filter', 'Bob', and 'Alice'. The main notebook area is titled 'Current Notebook' and shows a cell with the following content:

```
Current Notebook 5 minutes ago
```

### House Price Prediction

### Cell References

Is there a linear relationship between GrLivArea and SalePrice?

```
In [8]: # scatter plot grlivarea/saleprice
data.plot.scatter(x='GrLivArea', y='SalePrice');
```

The cell output shows a scatter plot of blue data points. A yellow box highlights the entire cell content. In the chat window on the right, a yellow arrow points from the chat message 'What about these outliers? marker' to the scatter plot. The chat window also shows a search bar and a 'Write your message' input field with a pencil icon.

On the left side of the interface, there are three chat messages from Alice and Bob:

- Alice 3:30pm: Executed a modified cell. Code: `@@ -1,7 +1,7 @@`
- Bob 3:31pm: Executed a modified cell.
- Alice 3:35pm: Executed a modified cell.

The screenshot displays the JupyterLab interface. At the top, the Jupyter logo and name are visible, along with a search bar and navigation buttons for 'Share', 'Filter', 'Bob', and 'Alice'. The main notebook area is titled 'House Price Prediction' and contains the question 'Is there a linear relationship between GrLivArea and SalePrice?'. Below the question is a code cell with the following code:

```
In [8]: # scatter plot grlivarea/saleprice
data.plot.scatter(x='GrLivArea', y='SalePrice');
```

The code cell is followed by a scatter plot showing a positive correlation between GrLivArea and SalePrice. A yellow box highlights a cluster of outliers in the plot, with a yellow arrow pointing to the 'marker' property in the chat window. The chat window on the right shows a conversation:

- Alice (3:30pm): Executed a modified cell
- Bob (3:31pm): Executed a modified cell
- Alice (3:35pm): Executed a modified cell
- Bob (3:31pm): Let me check their values.
- Bob (3:33pm): They were both sold in 2008 cell
- Alice (3:34pm): Financial crash?

The chat input field at the bottom right has a yellow box around the 'marker' property in the previous message, with a yellow arrow pointing to the scatter plot. The notebook interface also shows a 'Current Notebook' tab and a '5 minutes ago' timestamp.

## Snapshot References

```
Alice 3:30pm
Executed a modified cell

@@ -1,7 +1,7 @@
# scatter plot grlivarea/s
+ data.plot.scatter(x='GrLiv
```

jupyter

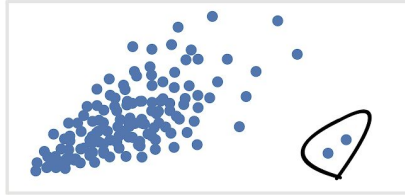
Share  Filter

</> Current Notebook  5 minutes ago

### House Price Prediction

Is there a linear relationship between GrLivArea and SalePrice?

```
In [8]: # scatter plot grlivarea/saleprice
data.plot.scatter(x='GrLivArea', y='SalePrice');
```



```
In [ ]:
In [ ]:
```

### Chat

search

Alice 3:30pm  
What about these outliers? **marker**

Bob 3:31pm  
Let me check their values.

Bob 3:33pm  
They were both sold in 2008 cell

Alice 3:34pm  
Financial crash?

Write your message

The screenshot displays the Jupyter web interface. At the top, the Jupyter logo and name are visible, along with a search bar and navigation buttons for 'Share', 'Filter', 'Bob', and 'Alice'. Below the header, the main workspace is divided into three sections:

- Left Panel (User Activity):** Shows a list of recent actions by users Alice and Bob, including timestamps and the text 'Executed a modified cell'.
- Center Panel (Notebook):** Titled 'Current Notebook', it shows a cell with the following code:

```
In [8]: # scatter plot grlivarea/saleprice
data.plot.scatter(x='GrLivArea', y='SalePrice');
```

Below the code is a scatter plot of blue data points. A yellow box highlights the '5 minutes ago' timestamp in the notebook header, with an arrow pointing to it from a yellow box labeled 'Diff References'. Another yellow box labeled 'Diff References' is positioned above the chat panel, with an arrow pointing to the 'marker' in a chat message.
- Right Panel (Chat):** A chat window with a search bar and a list of messages from Alice and Bob. The messages include: 'What about these outliers? marker', 'Let me check their values.', 'They were both sold in 2008 cell', and 'Financial crash?'. A yellow box highlights the 'marker' text in the first message, with an arrow pointing to it from the 'Diff References' box above.



The screenshot shows a Jupyter notebook interface with a chat sidebar on the left and a chat window on the right. The notebook title is "House Price Prediction" and the question is "Is there a linear relationship between GrLivArea and SalePrice?". The code cell shows a scatter plot of GrLivArea vs SalePrice. The chat window shows a conversation where Alice asks about outliers and Bob responds that they were sold in 2008. A yellow box labeled "Explicit References" highlights the chat messages and the corresponding code cell.

**Explicit References**

**Chat**

Alice 3:30pm  
Executed a modified cell  
@@ -1,7 +1,7 @@  
# scatter plot grlivarea/s  
+ data.plot.scatter(x='GrLiv

Bob 3:31pm  
Executed a modified cell

Alice 3:35pm  
Executed a modified cell

**House Price Prediction**

Is there a linear relationship between GrLivArea and SalePrice?

In [8]: # scatter plot grlivarea/saleprice  
data.plot.scatter(x='GrLivArea', y='SalePrice');

What about these outliers? marker

Bob 3:31pm  
Let me check their values.

Bob 3:33pm  
They were both sold in 2008 cell

Alice 3:34pm  
Financial crash?

Write your message

The screenshot displays a Jupyter notebook interface with a chat sidebar. The notebook title is "House Price Prediction" and the question is "Is there a linear relationship between GrLivArea and SalePrice?". The code cell shows a scatter plot of GrLivArea vs SalePrice. The plot shows a positive correlation with two outliers circled in black. The chat sidebar shows messages from Alice and Bob discussing these outliers. Yellow arrows connect the chat messages to the corresponding parts of the notebook.

**Chat Messages:**

- Alice (3:30pm): What about these outliers? marker
- Bob (3:31pm): Let me check their values.
- Bob (3:33pm): They were both sold in 2008 cell
- Alice (3:34pm): Financial crash?

**Notebook Content:**

Current Notebook | 5 minutes ago

### House Price Prediction

Is there a linear relationship between GrLivArea and SalePrice?

```
In [8]: # scatter plot grlivarea/saleprice
data.plot.scatter(x='GrLivArea', y='SalePrice');
```

In [ ]:

In [ ]:

The screenshot displays the JupyterLab interface. At the top, the Jupyter logo and name are visible, along with navigation buttons for 'Share', 'Filter', 'Home', and 'User' (Alice). The main notebook area is titled 'House Price Prediction' and contains the question 'Is there a linear relationship between GrLivArea and SalePrice?'. Below the question is a code cell (In [8]:) with the following code: 

```
# scatter plot grlivarea/saleprice
data.plot.scatter(x='GrLivArea', y='SalePrice');
```

 This code cell is highlighted with a blue border and labeled 'Current selected cell'. Below the code is a scatter plot showing a positive correlation between GrLivArea and SalePrice, with a small cluster of outliers circled in black. To the right of the notebook is a chat window titled 'Chat' with a search bar and a list of messages. The messages are: Alice (3:30pm) 'What about these outliers? marker', Bob (3:31pm) 'Let me check their values.', Bob (3:33pm) 'They were both sold in 2008 cell', and Alice (3:34pm) 'Financial crash?'. The first two messages are highlighted with yellow boxes, and the last one is highlighted with a blue box. Dashed blue arrows point from the yellow boxes to the scatter plot and from the blue box to the code cell. On the left side, a message history panel shows three messages from Alice, each with a timestamp and the text 'Executed a modified cell'. The first message is highlighted with a red border.

**Automatically inferring references from context**

**Current selected cell**

# Navigating Messages and Notebook Content

## Design of Callisto

The screenshot displays the Jupyter web interface. At the top, the Jupyter logo and name are visible, along with a search bar and navigation buttons for 'Share', 'Filter', 'Bob', and 'Alice'. The main area is a notebook titled 'Current Notebook' (updated 5 minutes ago) with the heading 'House Price Prediction' and the question 'Is there a linear relationship between GrLivArea and SalePrice?'. The notebook content includes a code cell with the following code:

```
In [8]: # scatter plot grlivarea/saleprice
data.plot.scatter(x='GrLivArea', y='SalePrice');
```

Below the code is a scatter plot showing a positive correlation between GrLivArea and SalePrice. A black hand-drawn circle highlights a cluster of outliers in the bottom right corner of the plot. The notebook interface also shows input fields for 'In []:' and 'In [ ]:'. On the left sidebar, three messages are listed:

- Alice 3:30pm: Executed a modified cell. Code: @@ -1,7 +1,7 @@ # scatter plot grlivarea/s + data.plot.scatter(x='GrLiv
- Bob 3:31pm: Executed a modified cell.
- Alice 3:35pm: Executed a modified cell.

On the right, a 'Chat' window is open, showing a conversation:

- Alice 3:30pm: What about these outliers? marker
- Bob 3:31pm: Let me check their values.
- Bob 3:33pm: They were both sold in 2008 cell
- Alice 3:34pm: Financial crash?

A yellow arrow points from the chat window to the notebook's code cell, and a green arrow points from the chat window to the scatter plot, indicating the flow of information and interaction between the chat and the notebook content.

# Navigating Messages and Notebook Content

From messages to notebook content -- “what changes were made”

Design of Callisto

The screenshot displays the Jupyter web interface. At the top, the Jupyter logo and name are visible, along with a search bar and navigation buttons for 'Share', 'Filter', 'Bob', and 'Alice'. The main content area is a notebook titled 'Current Notebook' (last updated 5 minutes ago). The notebook content includes a title 'House Price Prediction', a question 'Is there a linear relationship between GrLivArea and SalePrice?', and a code cell 'In [8]:' containing a scatter plot command. Below the code is a scatter plot of blue data points with a black circle highlighting two outliers. The notebook interface also shows 'In []:' and 'In [ ]:' input fields. On the left, a message history sidebar shows three messages from Alice and Bob. On the right, a 'Chat' window is open, showing a conversation where Alice asks about outliers, Bob checks their values, and Alice asks about a financial crash. A green arrow points from the chat window to the notebook's top bar.

**Jupyter**

Share Filter Bob Alice

Current Notebook 5 minutes ago

### House Price Prediction

Is there a linear relationship between GrLivArea and SalePrice?

```
In [8]: # scatter plot grlivarea/saleprice
data.plot.scatter(x='GrLivArea', y='SalePrice');
```

In []:

In [ ]:

**Chat**

search

Alice 3:30pm  
What about these outliers? marker

Bob 3:31pm  
Let me check their values.

Bob 3:33pm  
They were both sold in 2008 cell

Alice 3:34pm  
Financial crash?

Write your message

# Navigating Messages and Notebook Content

From messages to notebook content -- “what changes were made”

Design of Callisto

The screenshot displays the Jupyter interface with the following components:

- Header:** Jupyter logo, a search bar, and navigation buttons for 'Share', 'Filter', 'Bob', and 'Alice'.
- Message History (Left):**
  - Alice (3:30pm): Executed a modified cell. Code: `@@ -1,7 +1,7 @@`  
`# scatter plot grlivarea/s`  
`+ data.plot.scatter(x='GrLiv`
  - Bob (3:31pm): Executed a modified cell.
  - Alice (3:35pm): Executed a modified cell.
- Notebook (Center):**
  - Tab: `</> Current Notebook` | Refresh: `5 minutes ago`
  - Title: **House Price Prediction**
  - Text: **Is there a linear relationship between GrLivArea and SalePrice?**
  - Code Cell (In [8]):

```
# scatter plot grlivarea/saleprice
data.plot.scatter(x='GrLivArea', y='SalePrice');
```
  - Figure: A scatter plot of blue dots showing a positive correlation. A yellow circle highlights two outlier points, with a green arrow pointing from the chat to them.
  - Input fields: `In [ ]:` and `In [ ]:`
- Chat (Right):**
  - Search bar.
  - Alice (3:30pm): "What about these outlier? marker" (The word "marker" is highlighted in a green box).
  - Bob (3:31pm): "Let me check their values."
  - Bob (3:33pm): "They were both sold in 2008 cell"
  - Alice (3:34pm): "Financial crash?"
  - Input field: "Write your message"

# Navigating Messages and Notebook Content

From messages to notebook content -- “what changes were made”

Design of Callisto

The screenshot displays the Jupyter web interface. At the top, the Jupyter logo and name are visible, along with navigation buttons for 'Share', 'Filter', 'Bob', and 'Alice'. The main area shows a notebook titled 'Current Notebook' with a timestamp of '5 minutes ago'. The notebook content includes a title 'House Price Prediction', a question 'Is there a linear relationship between GrLivArea and SalePrice?', and a code cell 'In [8]: # scatter plot grlivarea/saleprice data.plot.scatter(x='GrLivArea', y='SalePrice');'. Below the code is a scatter plot of blue data points with a black oval highlighting a cluster of outliers. A green callout box points to the code cell with the text 'Highlight relevant cells in current notebook'. On the left, a sidebar shows a list of messages from Alice and Bob. On the right, a 'Chat' window is open, showing a message from Alice: 'What about these outliers? marker'. A green box highlights this message, and a white callout box with the text 'Select one message' points to it. At the bottom of the chat window, there are buttons for 'Snapshot', 'Edit Link (1)', and 'Cancel (1)'.

# Navigating Messages and Notebook Content

From messages to notebook content -- “what changes were made”

Design of Callisto

The screenshot displays the Jupyter web interface. At the top, the Jupyter logo and name are visible, along with navigation buttons for 'Share', 'Filter', 'Home (Bob)', and 'Profile (Alice)'. The main area is titled 'Current Notebook' and shows a notebook titled 'House Price Prediction'. The notebook content includes a question: 'Is there a linear relationship between GrLivArea and SalePrice?' followed by a code cell (In [8]:) containing the following code: 

```
# scatter plot grlivarea/saleprice
data.plot.scatter(x='GrLivArea', y='SalePrice');
```

 Below the code is a scatter plot showing a positive correlation between GrLivArea and SalePrice, with a small cluster of outliers circled in black. The notebook interface also shows a 'Snapshot' button and an 'Edit Link (1)' button. On the left side, there is a chat window with messages from Alice and Bob. A green box highlights a message from Alice: 'What about these outliers? marker'. A green callout box points to this message with the text 'Select one message'. Another green callout box points to the 'Snapshot' button with the text 'View the snapshot of the notebook'. The bottom right corner of the interface shows a 'Cancel (1)' button.



# Navigating Messages and Notebook Content

From messages to notebook content -- “what changes were made”

Design of Callisto

The screenshot displays the Jupyter web interface. At the top, the Jupyter logo and name are visible, along with a search bar and navigation buttons for 'Share', 'Filter', 'Bob', and 'Alice'. The main content area shows a notebook titled 'House Price Prediction' with the question 'Is there a linear relationship between GrLivArea and SalePrice?'. Below the question is a code cell (In [8]:) containing the following code: 

```
# scatter plot grlivarea/saleprice
data.plot.scatter(x='GrLivArea', y='SalePrice');
```

 The output of this cell is a scatter plot showing a positive correlation between GrLivArea and SalePrice, with a small cluster of points circled in black. To the left of the notebook is a chat sidebar with messages from Alice and Bob. A green box highlights a message from Bob: 'Let me check their values.' Below it, another message from Bob is selected: 'They were both sold in 2008 cell'. A green box also highlights the 'Select two messages' text. At the bottom right, a 'Navigate to diff view' button is visible, and a 'Diff' button is highlighted in the chat sidebar.

# Navigating Messages and Notebook Content

## From messages to notebook content -- "what changes were made"

## Design of Callisto

jupyter Alice&Bob\_Assignment Last Checkpoint: 08/23/2019 (autosaved) Logout Control Panel

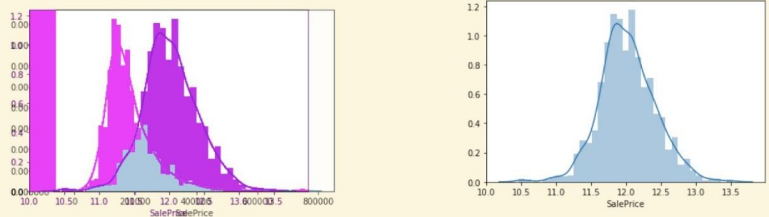
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code Share Filter bob-participant littleprifool


### Diff View

```
In [7]: 00 -1,3 +1,3 00
1 ##### TODO ##### 1 ##### TODO #####
2 # histogram 2 # histogram
3 - sns.distplot(df_train['SalePrice']) 3 + sns.distplot(np.log(df_train['SalePrice']))
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f756228deb8> <matplotlib.axes.\_subplots.AxesSubplot at 0x7f7561cfe438>



```
In [ ]: 1
In [ ]: 1 ## 3. Multivariable study
In [ ]: 1 ### 3.1 Example Analysis
In [ ]: 1 ### 3.1.1 Example Numerical Variable Analysis with `GrLivArea` and `SalePrice`
2
3 To begin with, let's take a look at how `GrLivArea` relates to `SalePrice` by implementing a scatter plot.
In [7]: 1 # scatter plot grlivarea/saleprice
2 label = 'GrLivArea'
3 data = pd.concat([df_train['SalePrice'], df_train[label]], axis=1)
4 data.plot.scatter(x=label, y='SalePrice', ylim=(0,800000))
```



alice-participant 2:00PM Execute a modified cell

```
00 -1,7 +1,7 00
##### TODO #####
# list 5 features
- # 1.
+ # 1. GrLivArea
- # 2.
+ # 2. YearRemodAdd
- # 3.
+ # 3. Utilities
- # 4.
+ # 4. YrSold
- # 5.
+ # 5. OverallQual
```

alice-participant 2:09PM Execute a modified cell

```
00 -1,2 +1,3 00
##### TODO #####
# histogram
+ sns.distplot(df_train['SalePrice'])
```

alice-participant 2:19PM Execute a modified cell

```
00 -1,3 +1,3 00
##### TODO #####
# histogram
- sns.distplot(df_train['SalePrice'])
+ sns.distplot(np.log(df_train['SalePrice']))
```

bob-participant 2:11PM Execute a modified cell

bob-participant 2:26PM Execute a modified cell

```
00 -1 +1 00
-
+ print(np.unique(df_train['OverallQual']))
```

bob-participant 2:31PM Execute a modified cell

Chat

bob-participant 1:54PM hi, i'm bob, i'm excited to solve our data science problem together.

alice-participant 1:55PM hi, i'm Alice!

bob-participant 1:56PM so I think we need to solve these 6 tasks

bob-participant 1:56PM it looks like it's related to trying to predict house prices from a number of variables

alice-participant 1:56PM write your message

# Navigating Messages and Notebook Content

From notebook content to messages -- “why changes were made”

The screenshot displays the Jupyter web interface. At the top, the Jupyter logo and name are visible, along with a search bar and navigation buttons for 'Share', 'Filter', 'Bob', and 'Alice'. The main content area is titled 'Current Notebook' and shows a notebook with the following content:

**House Price Prediction**  
Is there a linear relationship between GrLivArea and SalePrice?

```
In [8]: # scatter plot grlivarea/saleprice
data.plot.scatter(x='GrLivArea', y='SalePrice');
```

The notebook output shows a scatter plot of blue data points. A yellow arrow points from the notebook content to a chat window on the right. The chat window, titled 'Chat', has a search bar and a list of messages:

- Alice 3:30pm: What about these outliers? marker
- Bob 3:31pm: Let me check their values.
- Bob 3:33pm: They were both sold in 2008 cell
- Alice 3:34pm: Financial crash?

At the bottom of the chat window is a text input field labeled 'Write your message' and a send button. On the left side of the interface, a sidebar shows a message history for Alice and Bob, with timestamps and descriptions of their actions (e.g., 'Executed a modified cell').

# Navigating Messages and Notebook Content

From notebook content to messages -- “why changes were made”

Design of Callisto


The screenshot displays the Jupyter web interface. At the top, the Jupyter logo and name are visible, along with a search bar and a 'Filter' button. The main area is divided into three sections:

- Left Panel:** A list of recent activity. Alice executed a modified cell at 3:30pm, and Bob executed a modified cell at 3:31pm. Alice's code cell is expanded, showing the following code:

```
@@ -1,7 +1,7 @@  
# scatter plot grlivarea/s  
+ data.plot.scatter(x='GrLiv
```
- Center Panel:** The current notebook content. The title is "House Price Prediction" with the question "Is there a linear relationship between GrLivArea and SalePrice?". Below the title is a code cell labeled "In [8]:" containing the code:

```
# scatter plot grlivarea/saleprice  
data.plot.scatter(x='GrLivArea', y='SalePrice');
```

The code cell is followed by a scatter plot showing a positive correlation between GrLivArea and SalePrice. A black oval highlights a cluster of outliers in the bottom right corner of the plot. The plot has edit and delete icons to its right.
- Right Panel:** A chat window titled "Chat" with a search bar. The chat history shows the following messages:
  - Alice (3:30pm): "What about these outliers? marker"
  - Bob (3:31pm): "Let me check their values."
  - Bob (3:33pm): "They were both sold in 2008 cell"
  - Alice (3:34pm): "Financial crash?"At the bottom of the chat window is a text input field labeled "Write your message" and a send icon.



**House Prices: Advanced Regression Techniques**

Predict sales prices and practice feature engineering, RFs, and gradient boosting

5,001 teams · Ongoing

[Overview](#) [Data](#) [Notebooks](#) [Discussion](#) [Leaderboard](#) [Rules](#) [Join Competition](#)

- Stage 1: the real-time collaboration study
  - Participants working in pairs on a data science task in real time (N = 8 + 4)
  - 90 minute lab session
- Stage 2: the follow-up study
  - A third individual joining the shared project using Callisto or a lite version with no contextual links (N = 20)

## Stage 1: The Real-time Collaboration Study

## Evaluation

- Manual references (7.5/101 messages per group)
  - Mostly use cell pointers
  - *I can know my collaborator's cursor so it is easy to know what she is talking about. So we didn't use much references, only a few cell links. (P3, expert)*
- Automatically inferred references
  - 92% are connected to the correct context

## Stage 2: Following up with the Collaboration Process

## Evaluation

- Comparing how a new collaborator followed up with an ongoing collaborative project
  - Explore the notebook and answer five questions related to prior analysis
  - Use the tool in depth to follow up on their work
- Merging and modifying the collaboration assets (the notebook history, chat messages, and their connections) produced in Stage 1

## Stage 2: Following up with the Collaboration Process

## Evaluation

- Questionnaire score -- significantly improved

Control Condition	Experiment Condition
The need to check chat messages	
<ul style="list-style-type: none"><li>• Difficult to follow the chat messages</li></ul>	<ul style="list-style-type: none"><li>• Keep the filtering mode enabled</li><li>• Go back and forth to check context of messages</li><li>• Better understand how a code change resulted in an output change</li></ul>

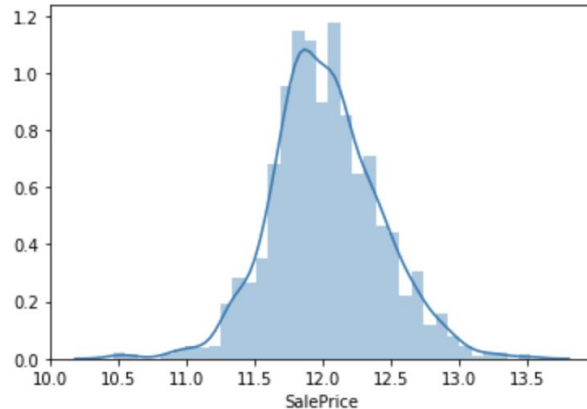


## Stage 2: Following up with the Collaboration Process

## Evaluation

```
In [6]: ##### TODO #####  
# histogram  
# looks like there are some outliers on the right (before log)  
sns.distplot(np.log(df_train['SalePrice']))
```

```
Out[6]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb575c4c828>
```



The result looks much better!

## Stage 2: Following up with the Collaboration Process

## Evaluation

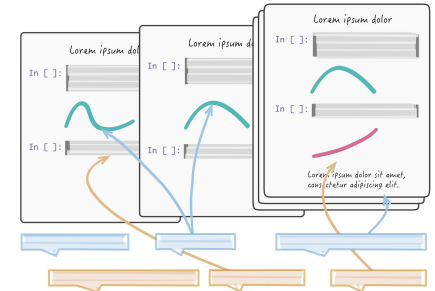


- **Reducing the Burden of Communication**
  - Hesitant to make accurate and polished references, or to create references
  - Remote collaborators co-design a shared artifact that changes over time
- **Improving the Accuracy of Contextual Links**
  - E.g., if a message describes a future action, the relevant cell may not exist when the message is sent
- **Towards Generating Meta-Narratives**
  - Not only need to understand the computational narrative itself but also how that narrative evolved—the meta-narrative behind the narrative

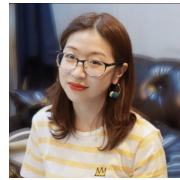
# Callisto: Capturing the “Why” by Connecting Conversations with Computational Narratives



- empirical evidence of the challenges that data scientists encounter when catching up with an ongoing group project
- the design of Callisto with a set of features to make chat messages more useful for understanding the past exploration process in the notebook
- empirical insights into how users engage with and perceive these features
- evidence that creating mappings between messages, notebook elements, and versions helps data scientists understand and follow up on the exploration pipeline



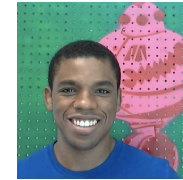
**April Yi Wang**  
Ph.D. Student  
Michigan SI



**Zihan Wu**  
Joining Ph.D. Student  
Michigan SI



**Chris Brooks**  
Assistant Professor  
Michigan SI



**Steve Oney**  
Assistant Professor  
Michigan SI