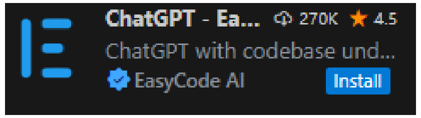
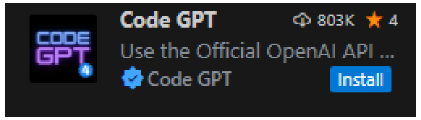
Applied Economics I

David Strömberg, Department of Economics, SU

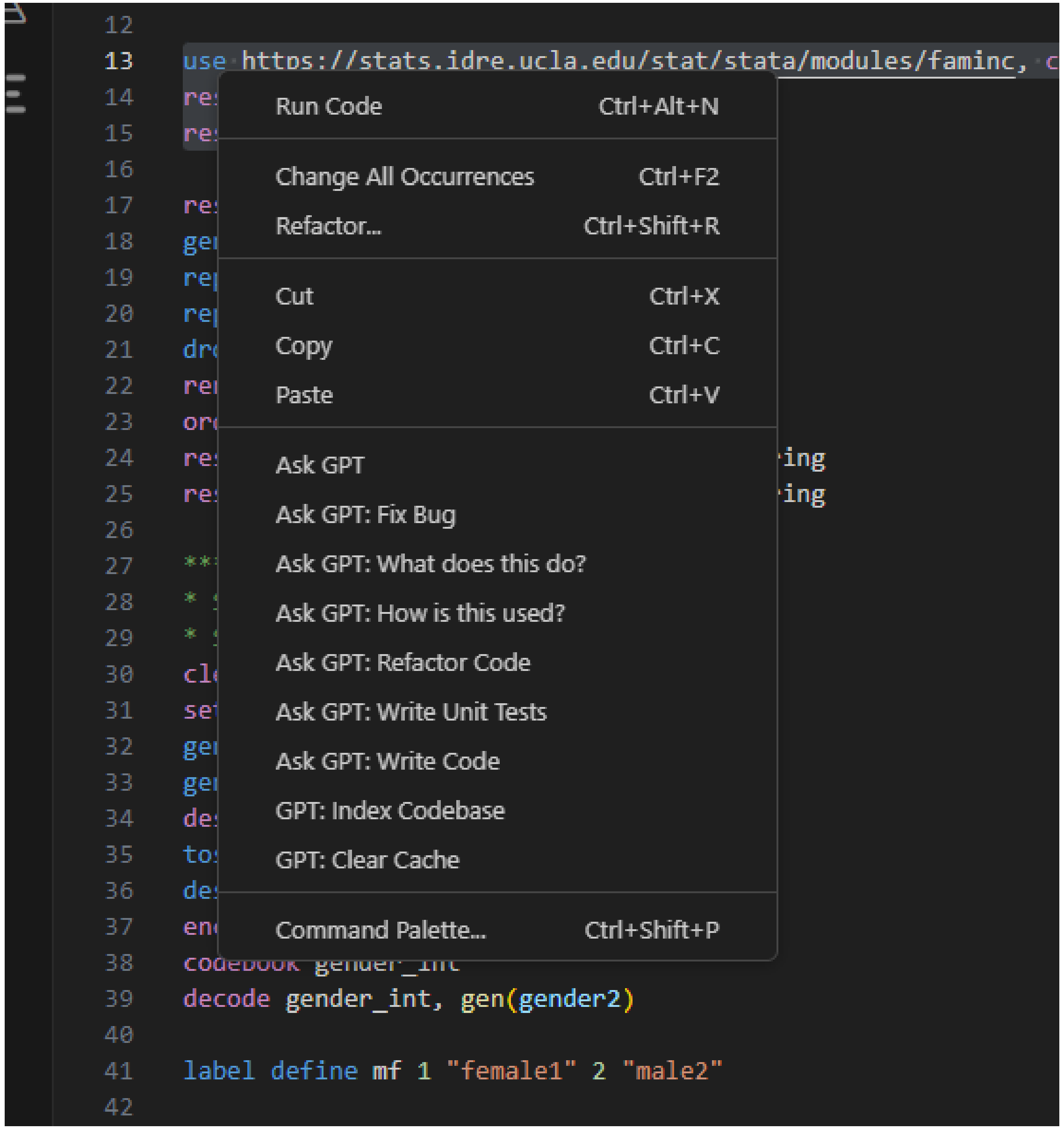
AI Coding Support

- Rapidly evolving: below a glimpse.
- VSC many extensions,
  - 

    ChatGPT – EasyCode, supports GPT-4 and GPT-3.5, has a free tier, and doesn't require an API key. (https://www.makeuseof.com/ways-you-can-use-chatgpt-vscode-together/)
  - 

    CodeGPT. Can use API key (for lower cost access to GPT4.

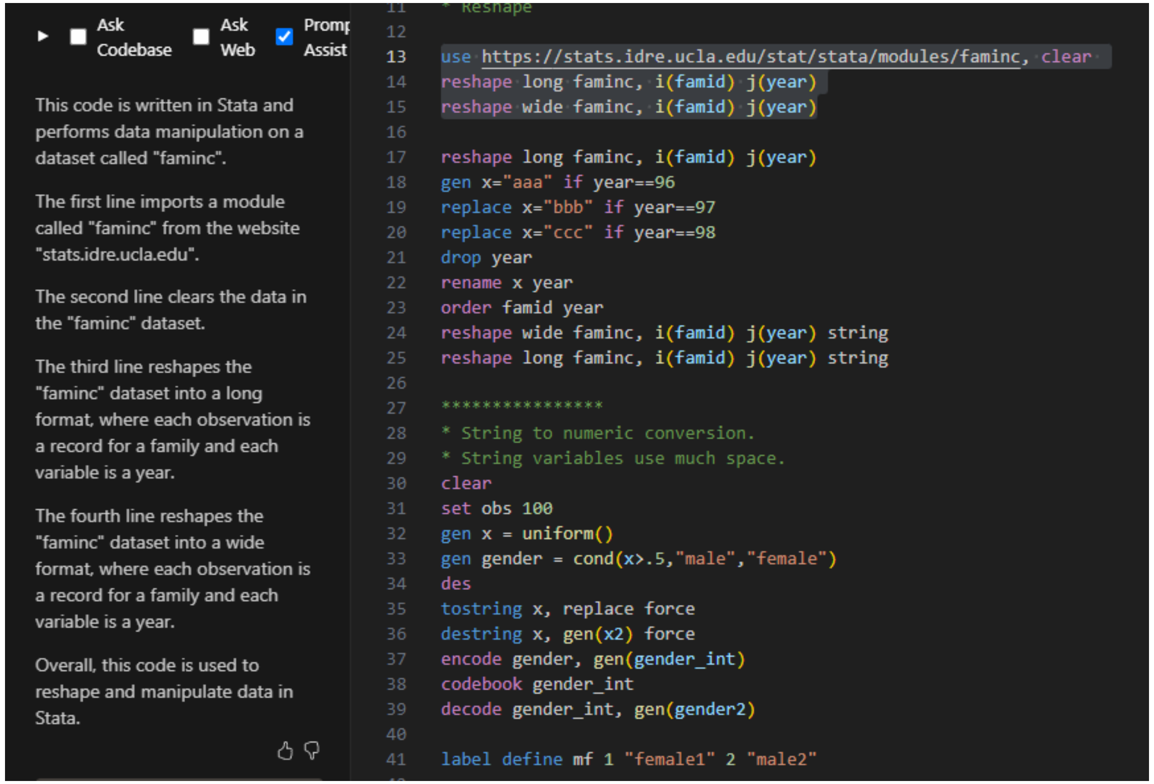
- GitHub copilot similar.



```
use https://stats.idre.ucla.edu/stat/stata/modules/faminc, c

decode gender_int, gen(gender2)

label define mf 1 "female1" 2 "male2"
```

Ask: What does this code do?

- Understanding of Stata is worse than bigger languages (GPT4 better than GPT-3.5)



▸ ☐ Ask Codebase ☐ Ask Web ☑ Promp Assist

This code is written in Stata and performs data manipulation on a dataset called "faminc".

The first line imports a module called "faminc" from the website "stats.idre.ucla.edu".

The second line clears the data in the "faminc" dataset.

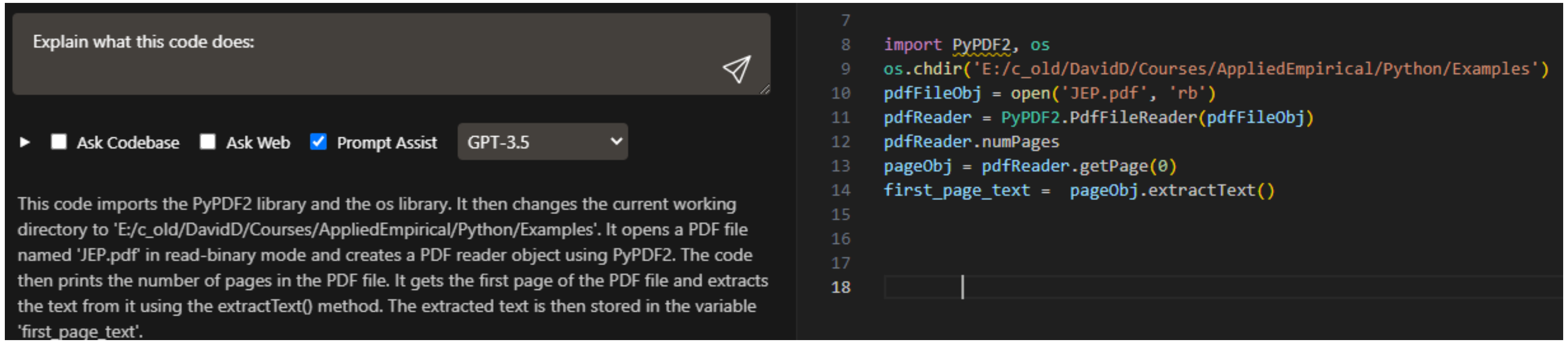
The third line reshapes the "faminc" dataset into a long format, where each observation is a record for a family and each variable is a year.

The fourth line reshapes the "faminc" dataset into a wide format, where each observation is a record for a family and each variable is a year.

Overall, this code is used to reshape and manipulate data in Stata.

```
* Reshape

use https://stats.idre.ucla.edu/stat/stata/modules/faminc, clear
reshape long faminc, i(famid) j(year)
reshape wide faminc, i(famid) j(year)

reshape long faminc, i(famid) j(year)
gen x="aaa" if year==96
replace x="bbb" if year==97
replace x="ccc" if year==98
drop year
rename x year
order famid year
reshape wide faminc, i(famid) j(year) string
reshape long faminc, i(famid) j(year) string

****************
* String to numeric conversion.
* String variables use much space.
clear
set obs 100
gen x = uniform()
gen gender = cond(x>.5,"male","female")
des
tostring x, replace force
destring x, gen(x2) force
encode gender, gen(gender_int)
codebook gender_int
decode gender_int, gen(gender2)

label define mf 1 "female1" 2 "male2"
```

- Python (R)



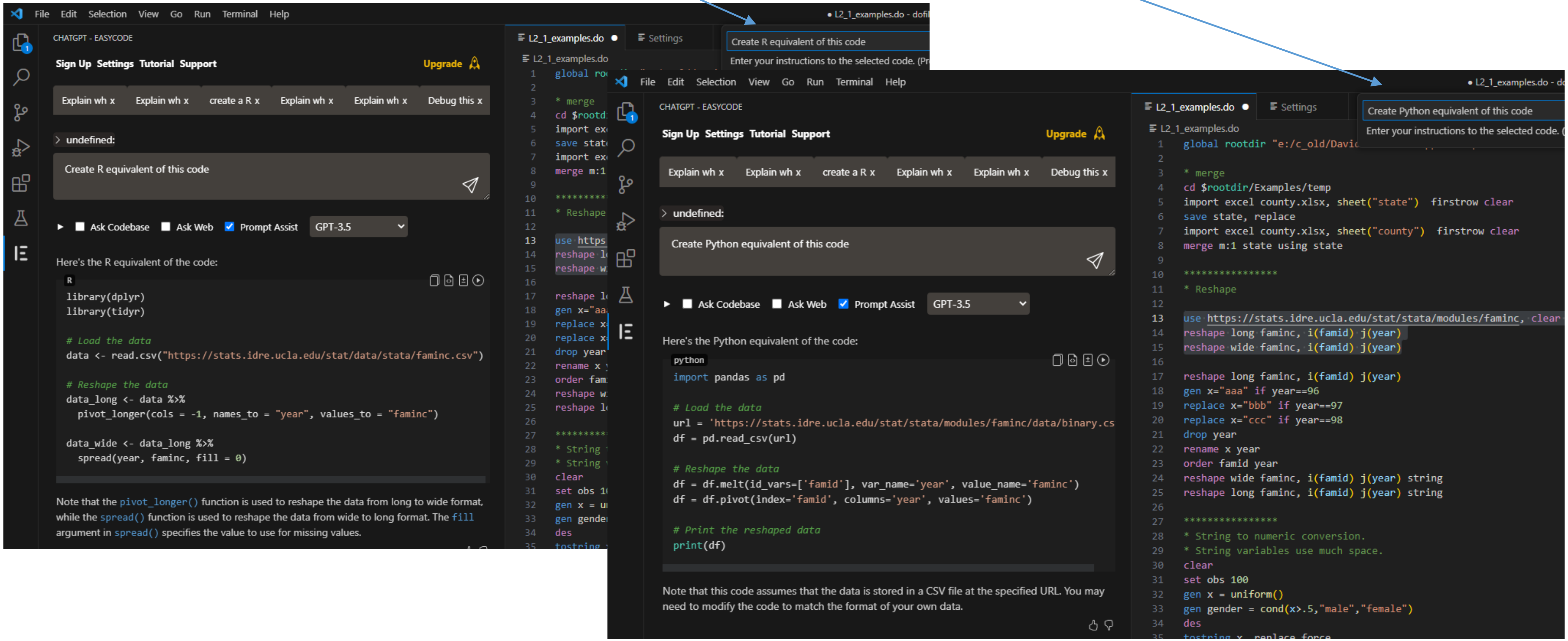
Explain what this code does:

▸ ☐ Ask Codebase ☐ Ask Web ☑ Prompt Assist GPT-3.5

This code imports the PyPDF2 library and the os library. It then changes the current working directory to 'E:/c_old/DavidD/Courses/AppliedEmpirical/Python/Examples'. It opens a PDF file named 'JEP.pdf' in read-binary mode and creates a PDF reader object using PyPDF2. The code then prints the number of pages in the PDF file. It gets the first page of the PDF file and extracts the text from it using the extractText() method. The extracted text is then stored in the variable 'first_page_text'.

```python

import PyPDF2, os
os.chdir('E:/c_old/DavidD/Courses/AppliedEmpirical/Python/Examples')
pdfFileObj = open('JEP.pdf', 'rb')
pdfReader = PyPDF2.PdfFileReader(pdfFileObj)
pdfReader.numPages
pageObj = pdfReader.getPage(0)
first_page_text =  pageObj.extractText()




```

Translate to R and Python (through Ask GPT)



Create R equivalent of this code

▶ ☐ Ask Codebase ☐ Ask Web ☑ Prompt Assist GPT-3.5

Here's the R equivalent of the code:

```r
library(dplyr)
library(tidyr)

# Load the data
data <- read.csv("https://stats.idre.ucla.edu/stat/data/stata/faminc.csv")

# Reshape the data
data_long <- data %>%
  pivot_longer(cols = -1, names_to = "year", values_to = "faminc")

data_wide <- data_long %>%
  spread(year, faminc, fill = 0)
```

Note that the `pivot_longer()` function is used to reshape the data from long to wide format, while the `spread()` function is used to reshape the data from wide to long format. The `fill` argument in `spread()` specifies the value to use for missing values.

Create R equivalent of this code

Enter your instructions to the selected code. (Pr

Create Python equivalent of this code

▶ ☐ Ask Codebase ☐ Ask Web ☑ Prompt Assist GPT-3.5

Here's the Python equivalent of the code:

```python
import pandas as pd

# Load the data
url = 'https://stats.idre.ucla.edu/stat/stata/modules/faminc/data/binary.cs
df = pd.read_csv(url)

# Reshape the data
df = df.melt(id_vars=['famid'], var_name='year', value_name='faminc')
df = df.pivot(index='famid', columns='year', values='faminc')

# Print the reshaped data
print(df)
```

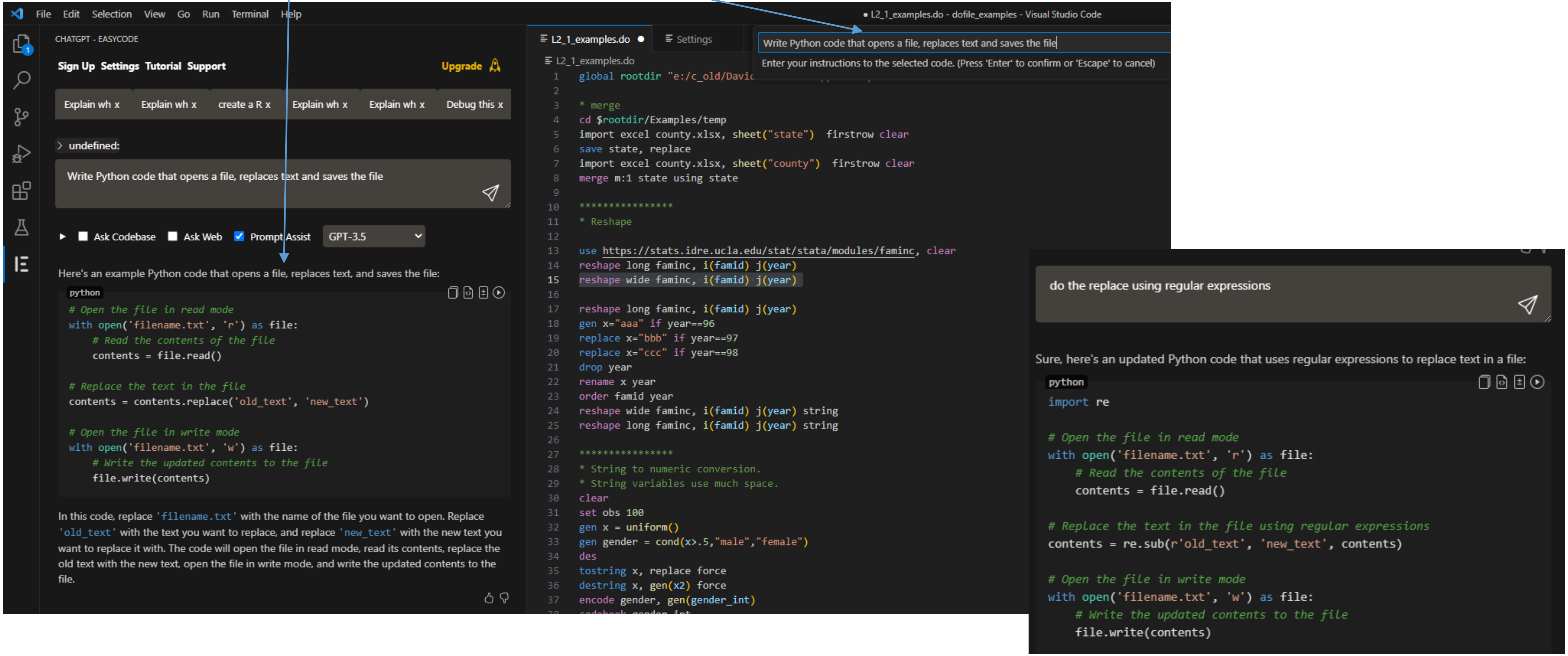
Note that this code assumes that the data is stored in a CSV file at the specified URL. You may need to modify the code to match the format of your own data.

Create Python equivalent of this code

Enter your instructions to the selected code. (

```stata
global rootdir "e:/c_old/Davic

* merge
cd $rootdir/Examples/temp
import excel county.xlsx, sheet("state")  firstrow clear
save state, replace
import excel county.xlsx, sheet("county")  firstrow clear
merge m:1 state using state

****************
* Reshape

use https://stats.idre.ucla.edu/stat/stata/modules/faminc, clear
reshape long faminc, i(famid) j(year)
reshape wide faminc, i(famid) j(year)

reshape long faminc, i(famid) j(year)
gen x="aaa" if year==96
replace x="bbb" if year==97
replace x="ccc" if year==98
drop year
rename x year
order famid year
reshape wide faminc, i(famid) j(year) string
reshape long faminc, i(famid) j(year) string

****************
* String to numeric conversion.
* String variables use much space.
clear
set obs 100
gen x = uniform()
gen gender = cond(x>.5,"male","female")
des
tostring x, replace force
```

Ask GPT to write code to do stuff



Write Python code that opens a file, replaces text and saves the file

Enter your instructions to the selected code. (Press 'Enter' to confirm or 'Escape' to cancel)

Write Python code that opens a file, replaces text and saves the file

Here's an example Python code that opens a file, replaces text, and saves the file:

```python
# Open the file in read mode
with open('filename.txt', 'r') as file:
    # Read the contents of the file
    contents = file.read()

# Replace the text in the file
contents = contents.replace('old_text', 'new_text')

# Open the file in write mode
with open('filename.txt', 'w') as file:
    # Write the updated contents to the file
    file.write(contents)
```

In this code, replace `'filename.txt'` with the name of the file you want to open. Replace `'old_text'` with the text you want to replace, and replace `'new_text'` with the new text you want to replace it with. The code will open the file in read mode, read its contents, replace the old text with the new text, open the file in write mode, and write the updated contents to the file.

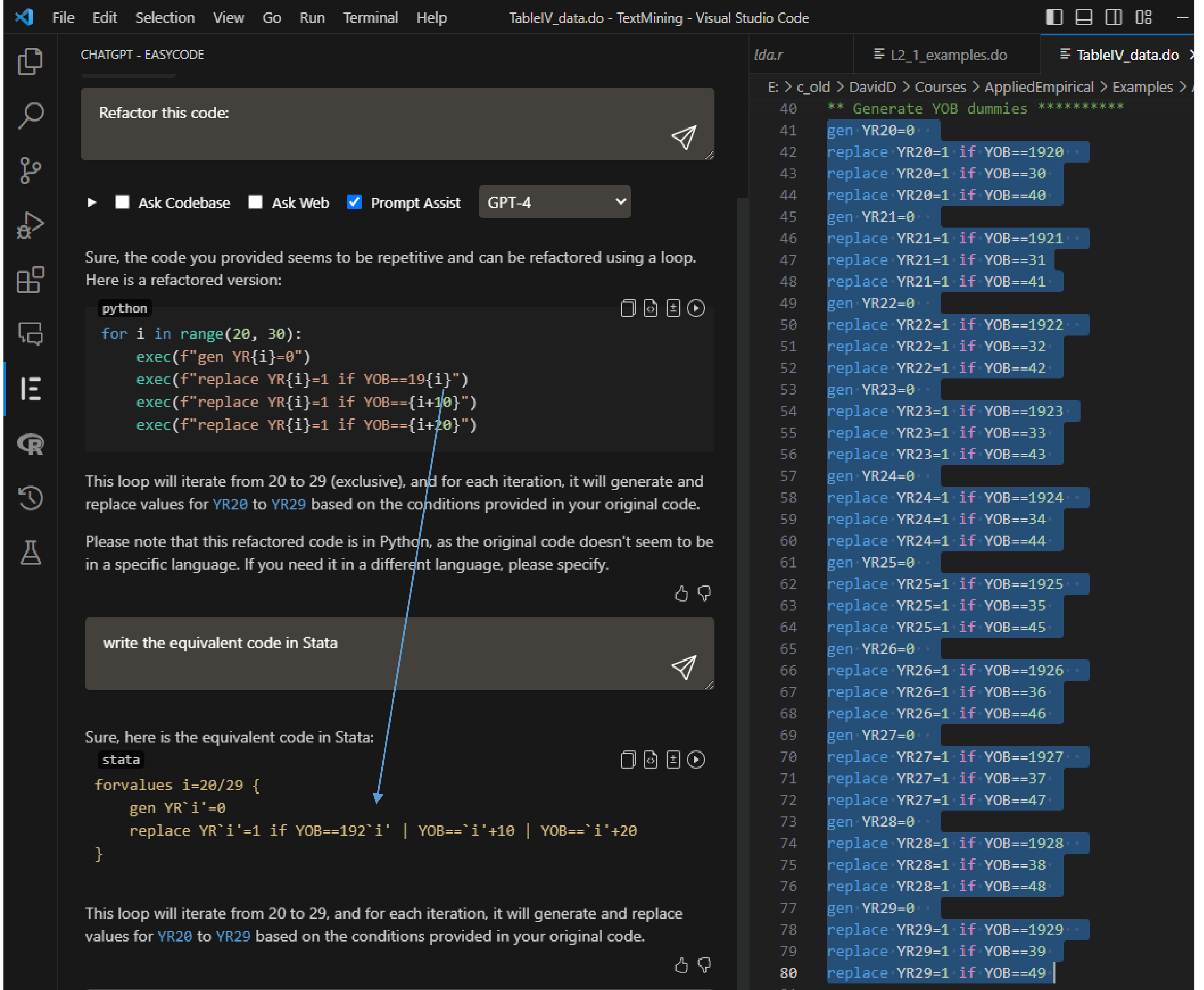
```
global rootdir "e:/c_old/Davi

* merge
cd $rootdir/Examples/temp
import excel county.xlsx, sheet("state")  firstrow clear
save state, replace
import excel county.xlsx, sheet("county")  firstrow clear
merge m:1 state using state

****************
* Reshape

use https://stats.idre.ucla.edu/stat/stata/modules/faminc, clear
reshape long faminc, i(famid) j(year)
reshape wide faminc, i(famid) j(year)

reshape long faminc, i(famid) j(year)
gen x="aaa" if year==96
replace x="bbb" if year==97
replace x="ccc" if year==98
drop year
rename x year
order famid year
reshape wide faminc, i(famid) j(year) string
reshape long faminc, i(famid) j(year) string

****************
* String to numeric conversion.
* String variables use much space.
clear
set obs 100
gen x = uniform()
gen gender = cond(x>.5,"male","female")
des
tostring x, replace force
destring x, gen(x2) force
encode gender, gen(gender_int)
```

do the replace using regular expressions

Sure, here's an updated Python code that uses regular expressions to replace text in a file:

```python
import re

# Open the file in read mode
with open('filename.txt', 'r') as file:
    # Read the contents of the file
    contents = file.read()

# Replace the text in the file using regular expressions
contents = re.sub(r'old_text', 'new_text', contents)

# Open the file in write mode
with open('filename.txt', 'w') as file:
    # Write the updated contents to the file
    file.write(contents)
```

Refactor code: produce equivalent code (note error in stata code)



Refactor this code:

Sure, the code you provided seems to be repetitive and can be refactored using a loop. Here is a refactored version:

```python
for i in range(20, 30):
    exec(f"gen YR{i}=0")
    exec(f"replace YR{i}=1 if YOB==19{i}")
    exec(f"replace YR{i}=1 if YOB=={i+10}")
    exec(f"replace YR{i}=1 if YOB=={i+20}")
```

This loop will iterate from 20 to 29 (exclusive), and for each iteration, it will generate and replace values for YR20 to YR29 based on the conditions provided in your original code.

Please note that this refactored code is in Python, as the original code doesn't seem to be in a specific language. If you need it in a different language, please specify.

write the equivalent code in Stata

Sure, here is the equivalent code in Stata:

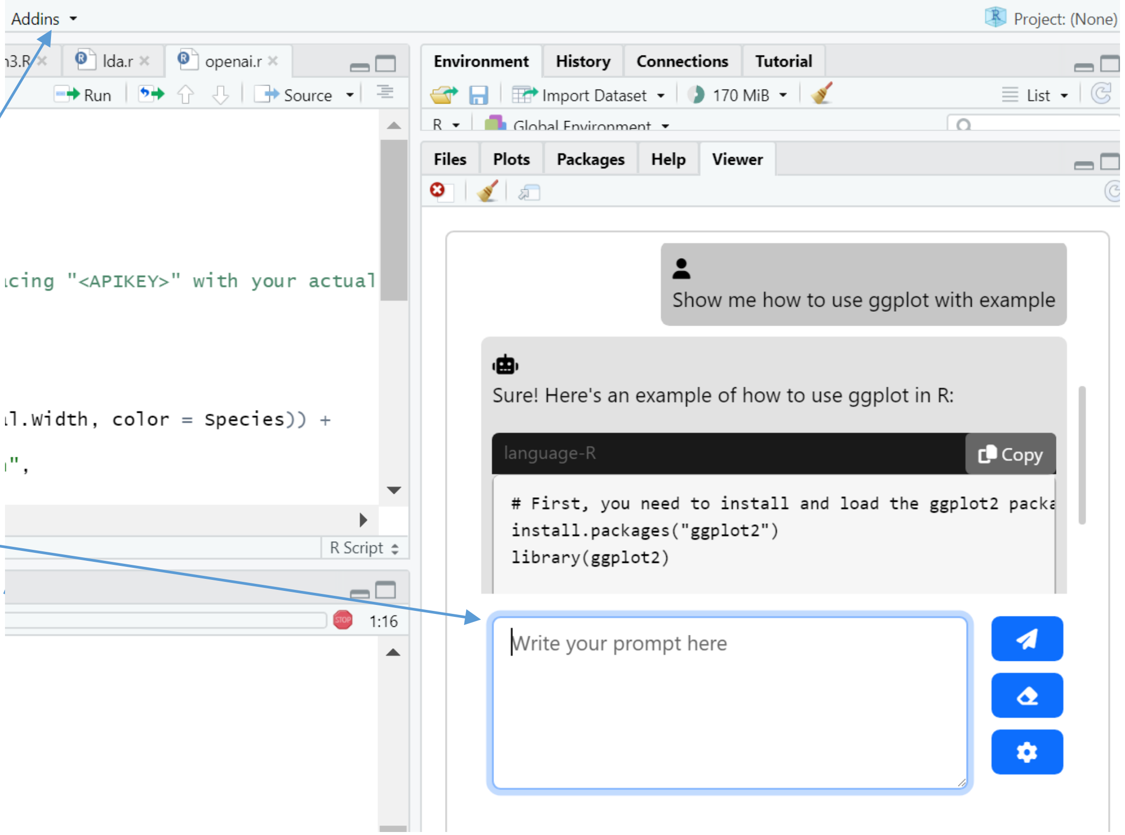
```stata
forvalues i=20/29 {
    gen YR`i'=0
    replace YR`i'=1 if YOB==192`i' | YOB==`i'+10 | YOB==`i'+20
}
```

This loop will iterate from 20 to 29, and for each iteration, it will generate and replace values for YR20 to YR29 based on the conditions provided in your original code.

```stata
** Generate YOB dummies **********
gen YR20=0
replace YR20=1 if YOB==1920
replace YR20=1 if YOB==30
replace YR20=1 if YOB==40
gen YR21=0
replace YR21=1 if YOB==1921
replace YR21=1 if YOB==31
replace YR21=1 if YOB==41
gen YR22=0
replace YR22=1 if YOB==1922
replace YR22=1 if YOB==32
replace YR22=1 if YOB==42
gen YR23=0
replace YR23=1 if YOB==1923
replace YR23=1 if YOB==33
replace YR23=1 if YOB==43
gen YR24=0
replace YR24=1 if YOB==1924
replace YR24=1 if YOB==34
replace YR24=1 if YOB==44
gen YR25=0
replace YR25=1 if YOB==1925
replace YR25=1 if YOB==35
replace YR25=1 if YOB==45
gen YR26=0
replace YR26=1 if YOB==1926
replace YR26=1 if YOB==36
replace YR26=1 if YOB==46
gen YR27=0
replace YR27=1 if YOB==1927
replace YR27=1 if YOB==37
replace YR27=1 if YOB==47
gen YR28=0
replace YR28=1 if YOB==1928
replace YR28=1 if YOB==38
replace YR28=1 if YOB==48
gen YR29=0
replace YR29=1 if YOB==1929
replace YR29=1 if YOB==39
replace YR29=1 if YOB==49
```

ChatGPT and RStudio

- Example: gptstudio (https://github.com/MichelNivard/gptstudio)
- Addins > gptstudio > ChatGPT
    - Type your question.
    - Click “Send” button
    - Copy and try code



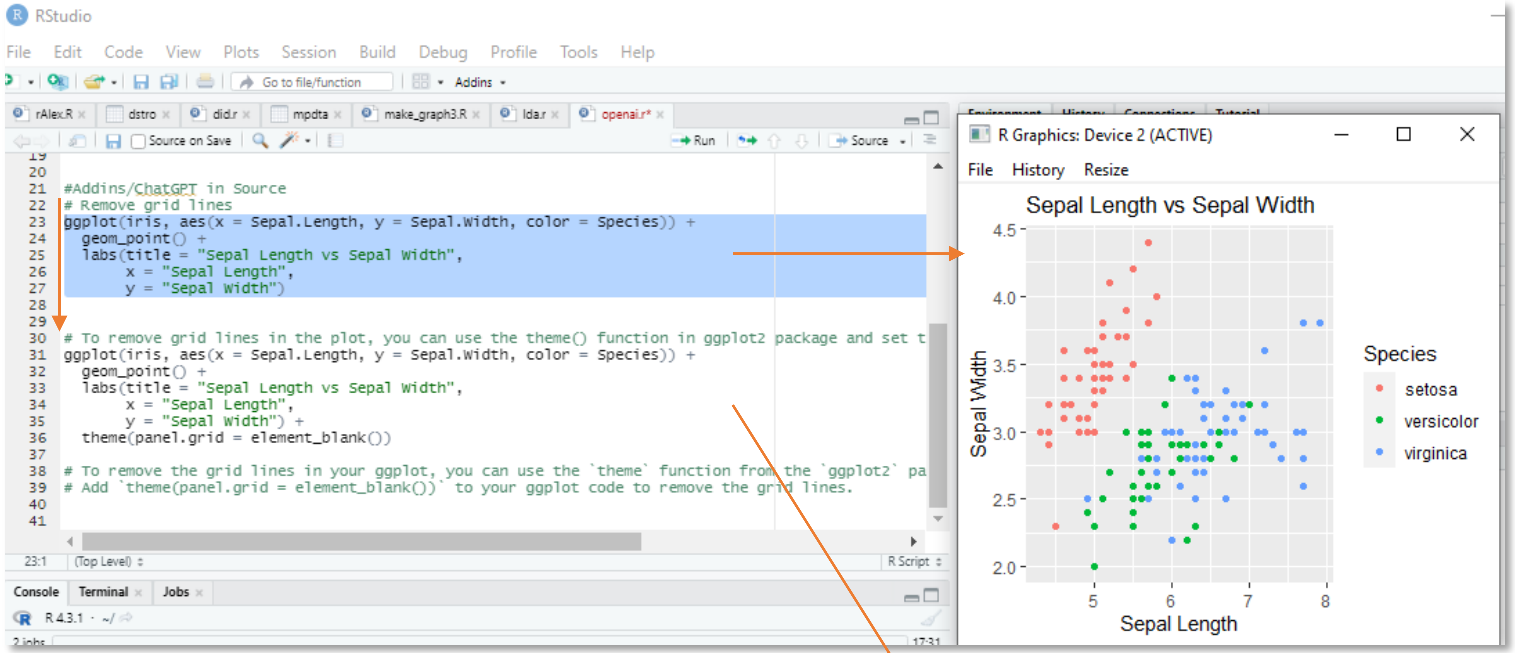
```
cing "<APIKEY>" with your actual
l.width, color = Species)) +
",
```

Show me how to use ggplot with example

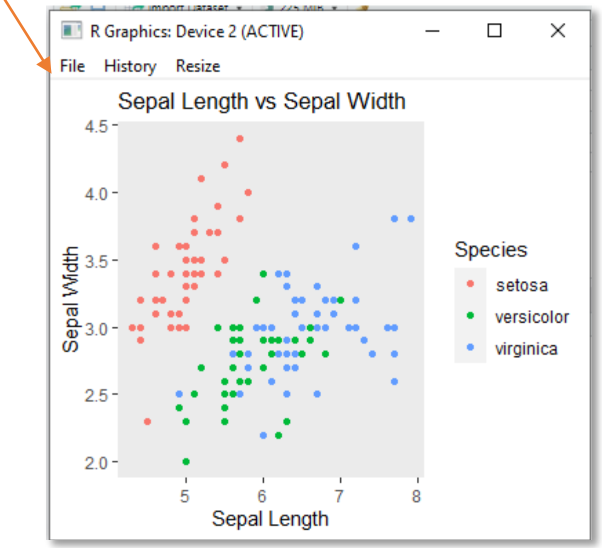
Sure! Here's an example of how to use ggplot in R:

```
# First, you need to install and load the ggplot2 packa
install.packages("ggplot2")
library(ggplot2)
```

- Addins > GPTSTUDIO > ChatGPT in Source

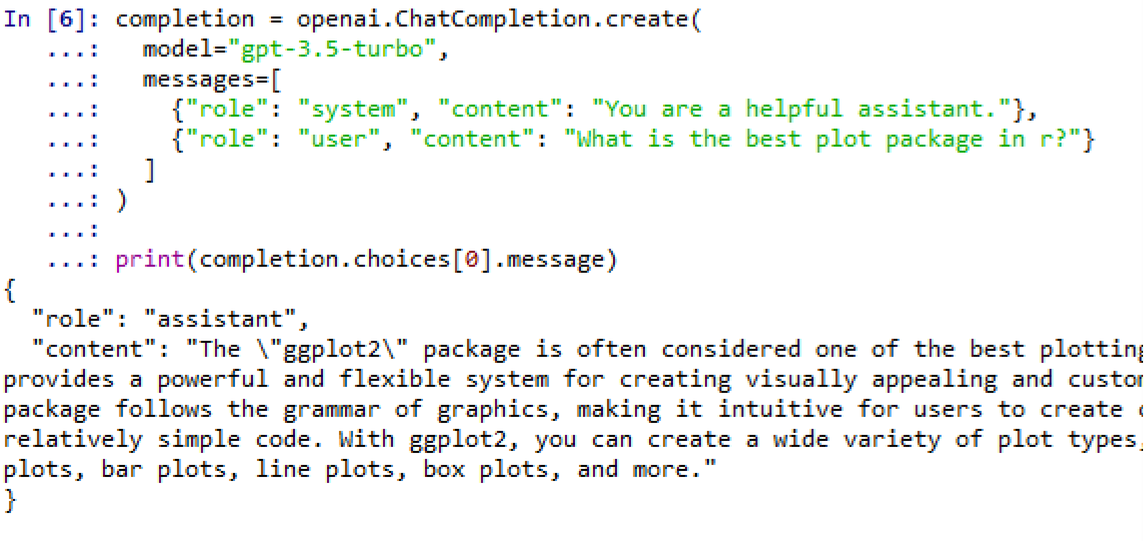


```r
#Addins/ChatGPT in Source
# Remove grid lines
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species)) +
  geom_point() +
  labs(title = "Sepal Length vs Sepal Width",
       x = "Sepal Length",
       y = "Sepal Width")


# To remove grid lines in the plot, you can use the theme() function in ggplot2 package and set t
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species)) +
  geom_point() +
  labs(title = "Sepal Length vs Sepal Width",
       x = "Sepal Length",
       y = "Sepal Width") +
  theme(panel.grid = element_blank())

# To remove the grid lines in your ggplot, you can use the `theme` function from the `ggplot2` pa
# Add `theme(panel.grid = element_blank())` to your ggplot code to remove the grid lines.
```

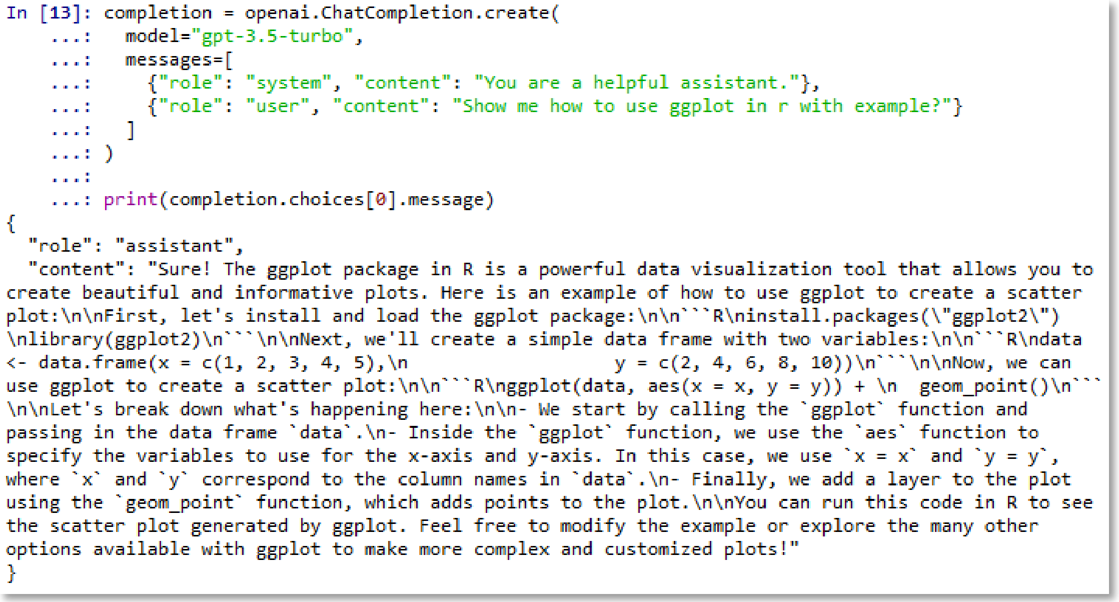


ChatGPT API

- The above extensions are just filters to the ChatGPT API. You can also access the API directly, for example, using a Python program.
- https://platform.openai.com/docs/api-reference/making-requests

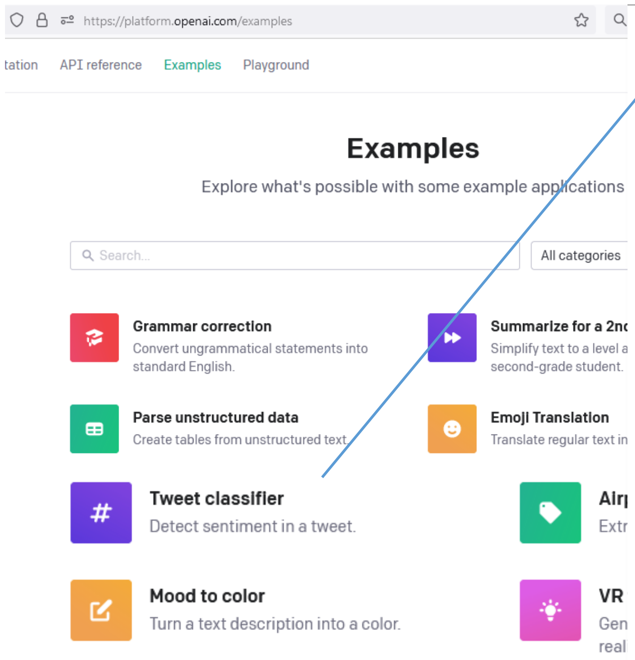


```
In [6]: completion = openai.ChatCompletion.create(
   ...:   model="gpt-3.5-turbo",
   ...:   messages=[
   ...:     {"role": "system", "content": "You are a helpful assistant."},
   ...:     {"role": "user", "content": "What is the best plot package in r?"}
   ...:   ]
   ...: )
   ...:
   ...: print(completion.choices[0].message)
{
  "role": "assistant",
  "content": "The \"ggplot2\" package is often considered one of the best plotting
provides a powerful and flexible system for creating visually appealing and custo
package follows the grammar of graphics, making it intuitive for users to create
relatively simple code. With ggplot2, you can create a wide variety of plot types
plots, bar plots, line plots, box plots, and more."
}
```



```
In [13]: completion = openai.ChatCompletion.create(
    ...:   model="gpt-3.5-turbo",
    ...:   messages=[
    ...:     {"role": "system", "content": "You are a helpful assistant."},
    ...:     {"role": "user", "content": "Show me how to use ggplot in r with example?"}
    ...:   ]
    ...: )
    ...:
    ...: print(completion.choices[0].message)
{
  "role": "assistant",
  "content": "Sure! The ggplot package in R is a powerful data visualization tool that allows you to
create beautiful and informative plots. Here is an example of how to use ggplot to create a scatter
plot:\n\nFirst, let's install and load the ggplot package:\n\n```R\ninstall.packages(\"ggplot2\")
\nlibrary(ggplot2)\n```\n\nNext, we'll create a simple data frame with two variables:\n\n```R\ndata
<- data.frame(x = c(1, 2, 3, 4, 5),\n                   y = c(2, 4, 6, 8, 10))\n```\n\nNow, we can
use ggplot to create a scatter plot:\n\n```R\nggplot(data, aes(x = x, y = y)) + \n  geom_point()\n```
\n\nLet's break down what's happening here:\n\n- We start by calling the `ggplot` function and
passing in the data frame `data`.\n- Inside the `ggplot` function, we use the `aes` function to
specify the variables to use for the x-axis and y-axis. In this case, we use `x = x` and `y = y`,
where `x` and `y` correspond to the column names in `data`.\n- Finally, we add a layer to the plot
using the `geom_point` function, which adds points to the plot.\n\nYou can run this code in R to see
the scatter plot generated by ggplot. Feel free to modify the example or explore the many other
options available with ggplot to make more complex and customized plots!"
}
```

You can do many things using ChatGPT API



Examples

Explore what's possible with some example applications

Grammar correction
Convert ungrammatical statements into standard English.

Summarize for a 2n
Simplify text to a level a
second-grade student.

Parse unstructured data
Create tables from unstructured text.

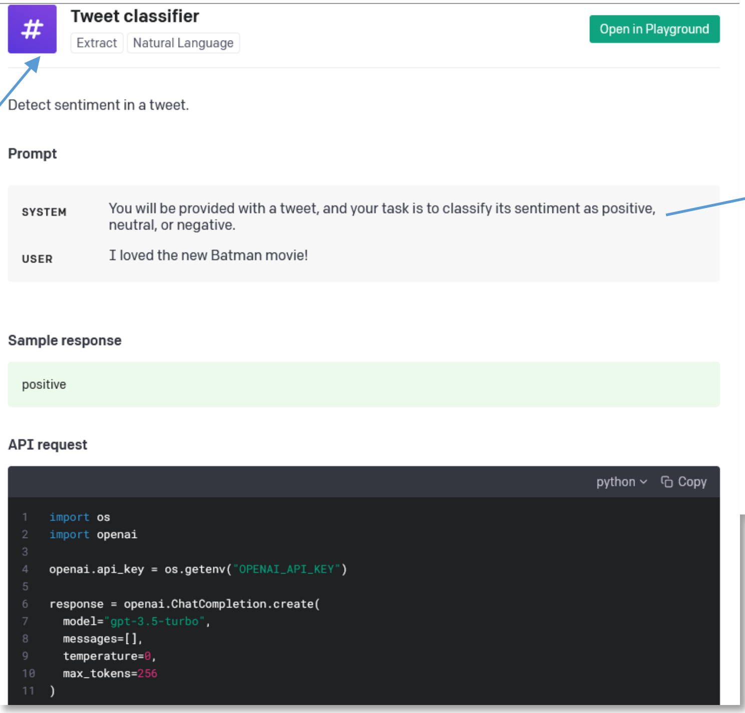
Emoji Translation
Translate regular text in

Tweet classifier
Detect sentiment in a tweet.

Air
Extr

Mood to color
Turn a text description into a color.

VR
Gen
real



Tweet classifier

Detect sentiment in a tweet.

Prompt

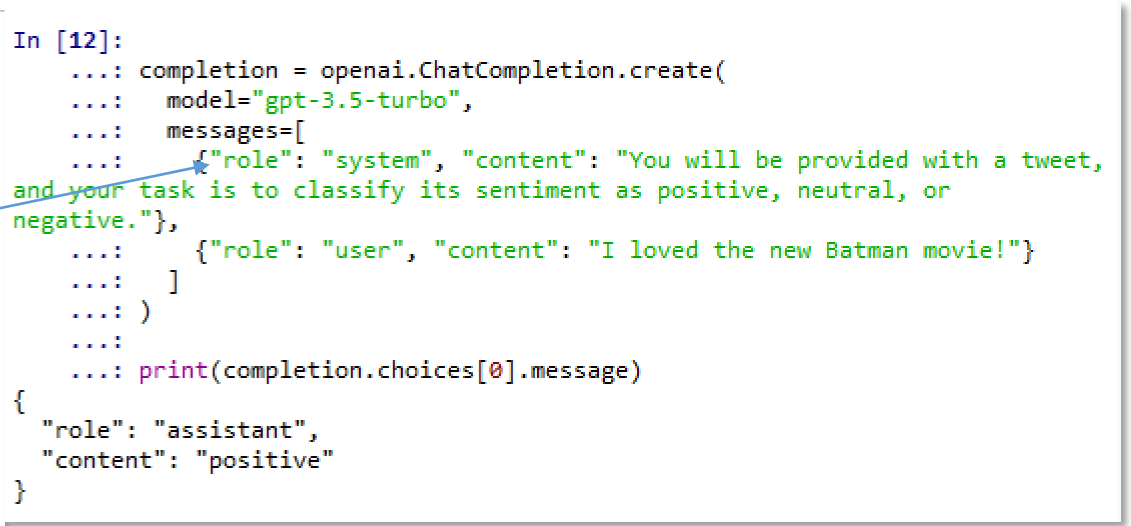
SYSTEM	You will be provided with a tweet, and your task is to classify its sentiment as positive, neutral, or negative.
USER	I loved the new Batman movie!

Sample response

positive

API request

```python
import os
import openai

openai.api_key = os.getenv("OPENAI_API_KEY")

response = openai.ChatCompletion.create(
  model="gpt-3.5-turbo",
  messages=[],
  temperature=0,
  max_tokens=256
)
```



```
In [12]:
    ...: completion = openai.ChatCompletion.create(
    ...:   model="gpt-3.5-turbo",
    ...:   messages=[
    ...:     {"role": "system", "content": "You will be provided with a tweet,
and your task is to classify its sentiment as positive, neutral, or
negative."},
    ...:     {"role": "user", "content": "I loved the new Batman movie!"}
    ...:   ]
    ...: )
    ...:
    ...: print(completion.choices[0].message)
{
  "role": "assistant",
  "content": "positive"
}
```