Coding with ChatGPT

[Xijin.Ge@SDSTATE.edu](mailto:Xijin.Ge@SDSTATE.edu) 2024 Data Science Symposium workshop

# Activity 1. Get Ready

1. Download files for this workshop by going to <https://github.com/gexijin/workshop24/>, click on Code and download a Zip file. Unzip to a folder. We will use this folder for this workshop. Alternatively, you can clone the repository using GitHub Desktop.
2. Install **Visual Studio Code**. Google “VS code download” and follow the instructions to install.

# Activity 2. HTML, CSS, JavaScript

Survey: Know HTML, CSS, JavaScript?

Prerequisite:

* Login to ChatGPT.
* Use VS Code or another text editor such as NotePad, NotePad++ (highly recommended) for Windows; TextEdit, or Sublime Text for MacOS.

Learning outcome:

* Use ChatGPT to learn.
* Progressively improve code.

|  |  |  |
| --- | --- | --- |
|  | Prompts | Note |
| 1 | Act as a professor of computer science. I am learning HTML. Give me the Hello World example. | Copy the code. Paste to a Text Editor. Save as **hello.html**. Click on it to open. |
| 2 | Add a short paragraph to this page about the history of HTML. | **Save** to the same file. **Refresh** browser. |
| 3 | Add another paragraph to this page with links to the Wikipedia page of HTML: <https://en.wikipedia.org/wiki/HTML> |  |
| 4 | Help me understand CSS by changing the font and background color. |  |
| 5 | Show me how JavaScript can be used to add interactivity. Add a button to this page, which shows a message when clicked. |  |

Note: GPT-4 is much better than GPT-3.5 in understanding and responding, even with ambiguous prompts. Microsoft Copilot does not follow instructions. Google Bard splits the CSS as separate files. To solve this problem, add “Combine everything into one file.” to the prompt.

# Activity 3. To-do list

Prerequisite:

* Login to ChatGPT.
* A text editor

Learning outcomes:

* Resubmit prompts to get different code.
* Revise previous prompts.
* Refine iteratively.

|  |  |  |
| --- | --- | --- |
|  | Prompts | Note |
| 1 | Write code for a to-do list that runs in a web browser. | Resubmit a few times. See the difference in response. |
| 2 | Write code for a to-do list that runs in a web browser. Tasks can be entered into a box. Once I hit enter, the task appears below the box, along with a checkbox. | Revise the first prompt.  Resubmit to get a better working version. |
| 3 | Once a task is checked, change the font background to green. |  |
| 4 | When a task is checked, show an encouraging message like "Great job!" in large size that disappears on its own. |  |
| 5 | Add today’s date and a timer showing current time. |  |
| 6 | Customize by yourself. |  |

# Activity 4. Shooting game.

Prerequisite:

* Login to ChatGPT.
* A text editor

Learning outcomes:

* Resubmit prompts to get different code.
* Revise previous prompts.
* Refine iteratively
* GPT-4 is far better than any other models

|  |  |  |
| --- | --- | --- |
|  | Prompts |  |
| 1 | Write code for a simple shooting game that runs in the browser. Put everything in one file. | Resubmit a few times. See the difference in response. |
| 2 | Write code for a shooting game that run in the browser. Put everything in one file. This game will feature a player at the bottom of a 400x400 playing area. The player can be moved left or right using the arrow keys. When the space key is pressed the player shoots a projectile. Targets appear randomly on top of the screen and move down. Targets disappear after being hit by projectiles. | Revise the first prompt.  Resubmit to get a better working version. |
| 3 | Change background color to dark grey. |  |
| 4 | Instead of using the keyboard, the player moves horizontally using the mouse and shoots when I left click. |  |
| 5 | Add a score counter. |  |
| 6 | If the player is hit by the targets, the game is over. User can restart. |  |

Note: GPT-4 can do this with ease. GPT-3.5 struggles. Needs to try many times to get it work. Microsoft Copilot also struggles. Google Bard gives half of the code.

# Uploaded imageActivity 5: GPT-4 Vision

Prerequisite:

GPT-4 subscription.

ChatGPT-4 app on the phone.

Follow up prompt:

When time is up, beep for 20 seconds. Then start a 3-minute timer automatically, so that I can stand up and stretch. Beep for 10 seconds at the end of that to remind me to return to my laptop.

# Activity 6. Analyzing data using ChatGPT: EDA

Prerequisite:

* Login to ChatGPT.
* Navigate to the **workshop24-main** folder. Find the **R** subfolder. Click on the **workshop.Rproj** to start RStudio.
* Start a Quarto document. Paste all you prompts, each linked with its code chunk.

Learning outcomes:

* Prompt engineering for data science
* Include data description in prompts
* Run code using Quarto in RStudio

|  |  |  |
| --- | --- | --- |
|  | Prompts | Note |
| 1 | Act as an experienced data scientist. Write R code that I can use to analyze data on my laptop. Read the file heartatk4R.csv in the current folder and print a small part of it. | Change to Python for Python |
| 2 | Create summary statistics of the data. Show the data structure. This file contains information about heart attack patients. These columns are numeric variables: 'charges', 'length', and 'age'. These columns are categorical variables: 'diagnosis', 'sex', 'outcome', and 'died’. The length column indicates the length of the hospital stay. | Revise the first prompt.  Resubmit to get a better working version. |
| 3 | Remove rows with missing values. Print the number of rows removed. |  |
| 4 | Plot the distribution of all numeric variables. |  |
| 5 | Create a density plot of age, separated by sex. |  |
|  | Split that by died. |  |
| 6 | Plot charges by length. Color code by sex. |  |
| 7 | Create a regression model of charges vs. all other variables. Show the summary of the model. | GPT3.5 starts to forget about variables. Due to limited context length. Only uses age and sex and length. |
| 8 | Include all variables in the file. |  |
| 9 | Give me the typical diagnostic plots. |  |

Note: GPT-4 can do this with ease. GPT-3.5 struggles. Needs to try many times to get it work. Microsoft Copilot also struggles. Google Bard gives half of the code.

|  |  |  |
| --- | --- | --- |
|  | Prompts | Note |
| 1 | Act as an experienced data scientist. Write R code that I can use to analyze data on my laptop. Read the file heartatk4R.csv in the current folder and print a small part of it. The first few rows of the data is:  diagnosis sex outcome died charges length age  41041 F 122 0 4752 10 79  41041 F 122 0 3941 6 34  41091 F 122 0 3657 5 76  41081 F 122 0 1481 2 80 |  |
| 2 | Create summary statistics of the data. Show the data structure. This file contains information about heart attack patients. | Revise the first prompt.  Resubmit to get a better working version. |
| 4 | Plot the distribution of all numeric variables. |  |
| 5 | Create a density plot of age, separated by sex. |  |
|  | Split that by died. |  |

# Activity 7: Use ChatGPT to develop strategies. And write prompts.

|  |  |  |
| --- | --- | --- |
|  | Prompts | Note |
|  | Act as an experienced data scientist. Give me detailed steps for exploratory analysis for a file containing information about heart attack patients. These columns are numeric variables: 'charges', 'length', and 'age'. These columns are categorical variables: 'diagnosis', 'sex', 'outcome', and 'died’. The length column indicates the length of the hospital stay. | Use in a separate chat --- an agent! |
| 1 | Act as an experienced data scientist. Give me detailed steps to build and test a model to predict patient survival based on file containing information about heart attack patients. These columns are numeric variables: 'charges', 'length', and 'age'. These columns are categorical variables: 'diagnosis', 'sex', 'outcome', and 'died’. The length column indicates the length of the hospital stay. |  |

# Activity 8: Create a shiny app

Prerequisite:

* ChatGPT
* RStudio

Show the distribution of charges by patient characteristics.

|  |  |  |
| --- | --- | --- |
|  | Prompts | Note |
| 1 | Act as an experienced data scientist. Write R code that I can use to create a Shiny app using data on my laptop. Read the file heartatk4R.csv in the current folder. This file contains information about heart attack patients. These columns are numeric variables: 'charges', 'length', and 'age'. These columns are categorical variables: 'diagnosis', 'sex', 'outcome', and 'died’. The length column indicates the length of the hospital stay. Remove rows with missing values. In the side bar, show a checkbox for sex, either F for female or M for male. In the main panel, show a density plot of charges in patients who are either female or male. | Change to Python for Python |
| 2 | When the app starts, it shows the distribution of males by default. The check box is mutually exclusive. When a gender is selected, the distribution for that gender is shown. |  |
| 3 | To further specify a sub-population of patients, enable the user to select one of the many diagnosis codes. And show the distribution of charges. |  |
| 4 | Similarly, enable users to select died and outcome status. |  |
| 5 | Add a similar plot of length of stay for the selected population. |  |
| 6 | For the selected population, calculate and show the average and median charges, along side with 95% confidence interval. |  |

# Activity 9: RTutor

Go to <https://RTutor.ai>

Upload the heartatk.csv file.

Go to EDA, generate report.

# Show me the distribution of charges.

# Plot the distribution of age by sex and outcome.

# Create a density plot of age by sex. Split into panels by outcome

# Show me the relationship between charges and length.

# remove missing values and calculate the correlation between length and charges.

* create a regression model to predict charges and show me the diagnostic plots.

[Video](https://youtu.be/a-bZW26nK9k)

# Activity 10: Chatlize.ai

Upload the heartatk.csv file.

In addition to the default prompt, add this sentence: “**Conduct exploratory data analysis**”. Check the box next to **Plan**. And then **Submit**.

# Activity 11: Setup GitHub Copilot in RStudio

From RStudio: Tools > Global Options > Copilot Tab

[Video](https://youtu.be/YZdp6To__1g?si=HIXafhP_4sG4SUsX&t=35)

# Activity 12: Setup GitHub Copilot in VS Code

 Setting up **GitHub Copilot** in **Visual Studio Code (VS Code)** is a straightforward process. Let’s get you started:

1. Install VS Code.
2. **Install the GitHub Copilot Extension**:
   * Open **Visual Studio Code**.
   * Go to the **Extensions** view by clicking on the square icon on the sidebar.
   * Search for **“GitHub Copilot”** in the search bar.
   * Click the **“Install”** button to add the extension.
3. **Sign in to GitHub**:
   * If you haven’t previously authorized VS Code with your GitHub account, you’ll be prompted to sign in.
   * In your browser, GitHub will request the necessary permissions for GitHub Copilot.
   * Approve these permissions by selecting **“Authorize Visual Studio Code”**.
4. **Activate Your Free Trial**:
   * If you haven’t already activated your free trial for Copilot, the extension will notify you in VS Code.
   * Select **“Signup for GitHub Copilot”** to activate your trial.
   * You can learn more about billing for Copilot in the GitHub Copilot documentation.
5. Install the **GitHub Copilot Chat Extension**

More info on [GitHub](https://docs.github.com/en/copilot/quickstart), [VS Code](https://code.visualstudio.com/docs/editor/github-copilot), [YouTube](https://www.youtube.com/watch?v=7hN7m9AgDm0).

# Activity 13: Using Copilot in VS Code

Tips:

1. Comment to request code
2. Q&A comment blocks. # q:
3. Copilot Chat Windows. Ctrl + Shift + I
4. Multiple solutions. Hover, switch between. Ctrl + Enter
5. Highlight, right click, Copilot: Explain, Fix, Document, Test
6. Agents @workspace @vscode

<https://youtu.be/2nPoiUJpDaU> <https://youtu.be/a2DDYMEPwbE>

# Activity 14: Documenting, debugging, testing

Prerequisite:

* RStudio or VS Code;
* GitHub copilot or prompting

Learning outcomes:

* How to add documentation, test, and debug
* Inline coding with GitHub Copilot

1. # Write a function to return the number of days between two dates.
2. Add detailed documentation.
3. Fix the error.