

# IDL Agent クイックスタート

## 1. このガイドでできるようになること

このクイックスタートを完了すると、次のことができるようになります。

- IDL Agent を使って、IDL コード/IDL Notebook を作成、実行し、結果の説明や改善提案を得る流れを理解できます。
- VS Code の GitHub Copilot Chat を Agent mode で使い、ツール実行/承認 (Approve/Allow) の流れを理解できます。
- 2 つの演習を通して、実務的な IDL Agent 機能のワークフローを体験できます。

## 2. IDL Agent とは

- IDL Agent は、IDL コードを自律的に「書く・実行する・説明する・トラブルシューティングする」ための支援機能です。
- IDL Agent は GitHub Copilot Chat および IDL for VSCode 拡張機能を活用して動作します。IDL for VSCode 拡張機能が MCP (Model Context Protocol) サーバとして動作することにより、IDL に最適化された支援が可能になります。
- IDL Agent は IDL の標準ライセンスで動作し、追加のオプションライセンスを認証する必要はありません。

## 3. 事前準備とセットアップ

### 3.1. 必要なコンポーネント

以下のインストーラおよび動作環境を用意します。

- IDL 9.2 以上 (または ENVI 6.2 / IDL9.2 以上)
- [Visual Studio Code](#) (最新版推奨)
- VSCode の拡張機能: [IDL for VSCode](#) (バージョン 6.0 以降)
- GitHub Copilot の有償サブスクリプション (GitHub Copilot Chat から高機能な LLM を選択する際には有償契約が必要となります: [公式案内](#))
- インターネット接続 ([GitHub Copilot 機能はオフラインで利用できません](#))

### 3.2. インストール順序

- ① IDL 9.2 以降または ENVI6.2/IDL9.2 をインストール・セットアップします。
- ② Visual Studio Code をインストール／更新します。
- ③ IDL for VSCode 拡張機能をインストール／更新します。

インターフェース左側の拡張機能アイコンをクリックし、IDL for VSCode 拡張機能がインストールされていることを確認してください。インストールされていない場合には、「IDL」と検索して NV5 Geospatial Software によって作成・公開されている拡張機能を選択しインストールするか、[マイクロソフトの公式マーケットプレイスから入手](#)してください。

- 参考: [IDL for VSCode 拡張機能の公式ドキュメントページ](#)

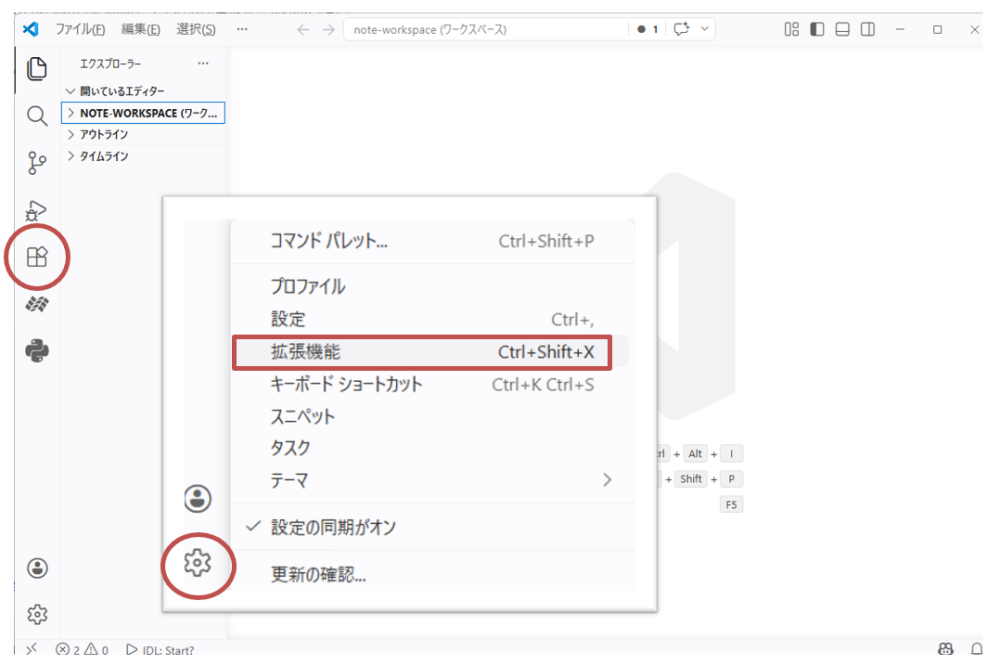


図 1: 拡張機能メニューを開く

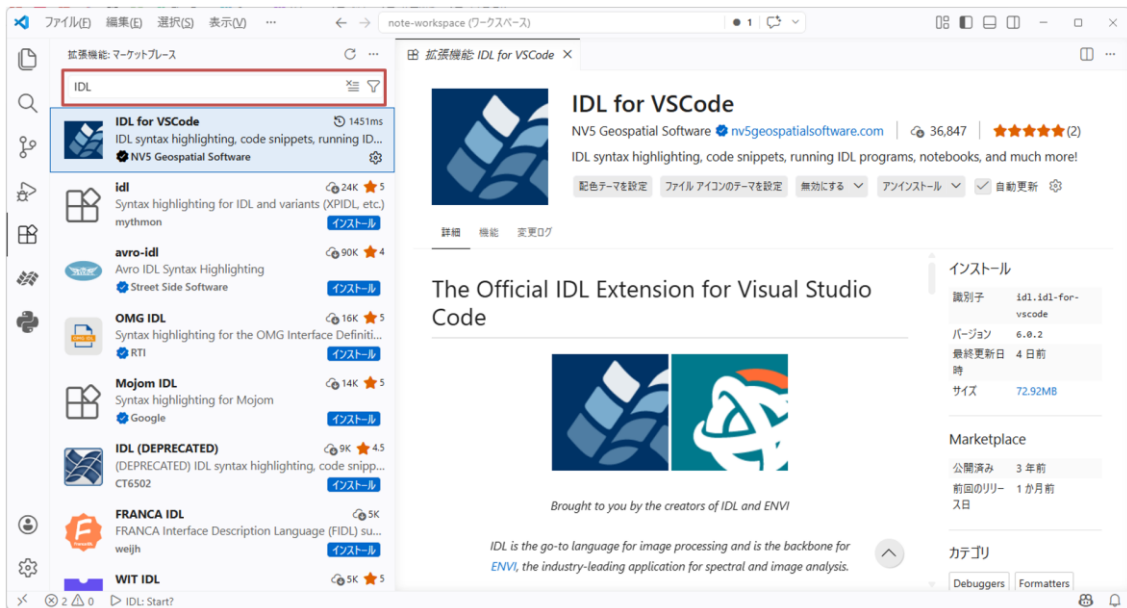


図 2: IDL for VSCode 拡張機能の入手

- ④ GitHub Copilot をセットアップします。有償の GitHub Copilot サブスクリプションのある GitHub アカウントで VSCode にサインインしてください。

VS Code インターフェースの上部にある「Sign in to use AI Features (AI 機能を使用するにはサインインしてください)」ボタン（検索ウィンドウの横にあります）をクリックします。パネルが表示されます。

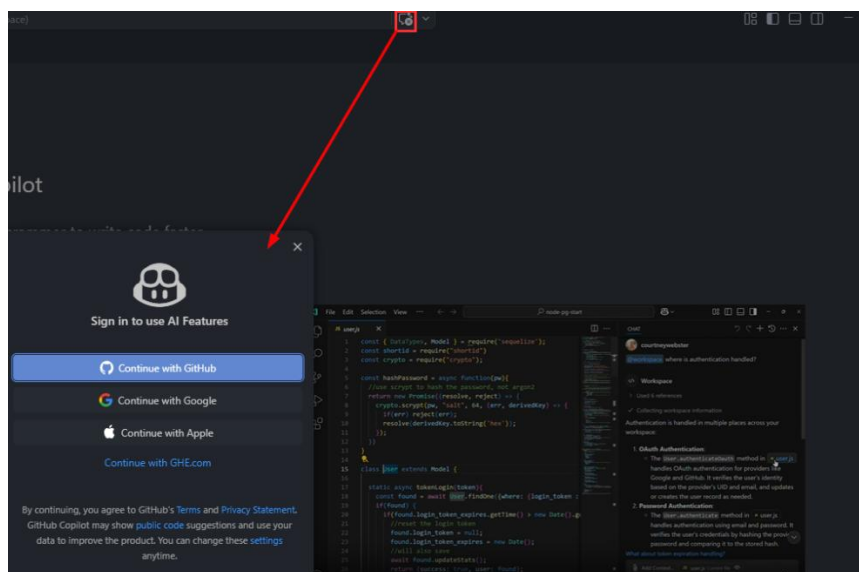


図 3: Sign in to use AI Features ボタン

Continue with GitHub ボタンを選択し、指示される手順にしたがって VS Code での使用を承認してください。また、VS Code インターフェースの左下隅にあるアカウントアイコンをクリックし、GitHub にログインしていることを確認してください。

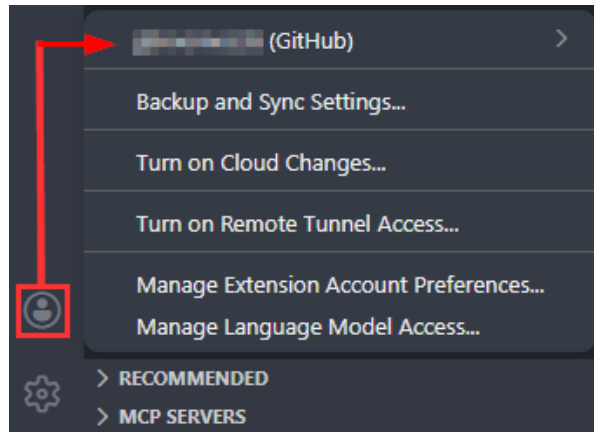


図 4: アカウントアイコンとログイン情報

## 4. 起動と設定の確認

### 4.1. 拡張機能の確認

3.2 章の手順に従って、以下の状態になっていることを確認してください。

- VS Code に IDL for VSCode 拡張機能がインストールされている
- GitHub アカウントでサインインが完了している

### 4.2. GitHub Copilot Chat を Agent mode で使う

- ① VSCode 上部のクイックアクセス右側にあるチャットボタンから、新規 Copilot Chat を開きます。



図 5:チャットボタンと新規 Copilot Chat 選択メニュー

- ② 新規チャットの文字入力箇所の下部にある、エージェントの設定ボタンが Agent mode (六角形アイコン) であることを確認します。エージェントの設定ボタンでは、Ask, Plan, Agent の 3 つのモードが選択できます。IDL Agent を使用する場合は、Agent mode を設定します。また、このモードは Copilot がツールを使ってアクションしようとする際に、実行/承認 (Approve/Allow) が必要です。

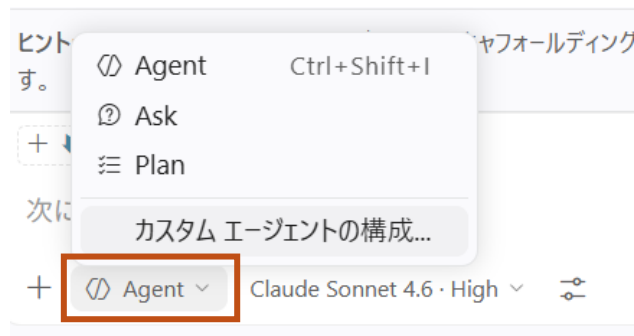


図 6: エージェントの設定ボタン

- ③ LLM の設定がデフォルトの Auto から Claude Sonnet 4.x のようなモデルとなっていることを確認します<sup>1</sup>。IDL Agent は Claude Sonnet で検証・テストされているため、この LLM の使用が推奨されています。



図 7: 利用する大規模言語モデル(LLM)の選択

### 4.3. ワークスペース（作業フォルダ）を開く

本演習で生成する IDL ソースファイル(.pro)や IDL ノートブックファイル(.idlInb)の出力先となる、任意の作業フォルダを VS Code の Workspace として開きます。

- VSCode の[メニューバー]>[ファイル]>[フォルダを開く]から、適当な場所に配置した作業用フォルダを選択します（本演習では例としてデスクトップ配下の work フォルダを使用しています）。

<sup>1</sup> 選択可能な大規模言語モデル (LLM) は、GitHub のアカウント管理メニューから明示的に有効化 (Enable) する必要があります。また、使用する言語モデルおよび推論時間の組み合わせに応じて、ユーザーごとに毎月割り当てられた AI クレジットが消費されます。クレジットの上限を超過した場合、契約プランや設定内容によっては、追加の従量課金が発生する場合があります。詳細は [GitHub の公式ドキュメント](#) をご確認ください。

- 選択した作業用フォルダがワークスペースに表示されていることを確認します。

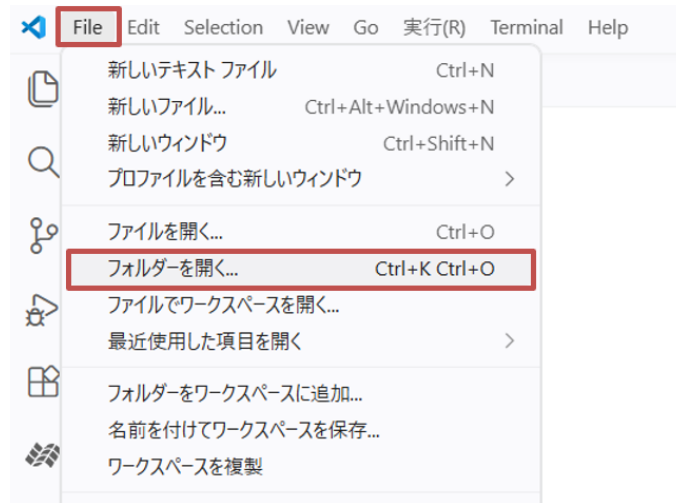


図 8: VSCode で任意のフォルダを開く

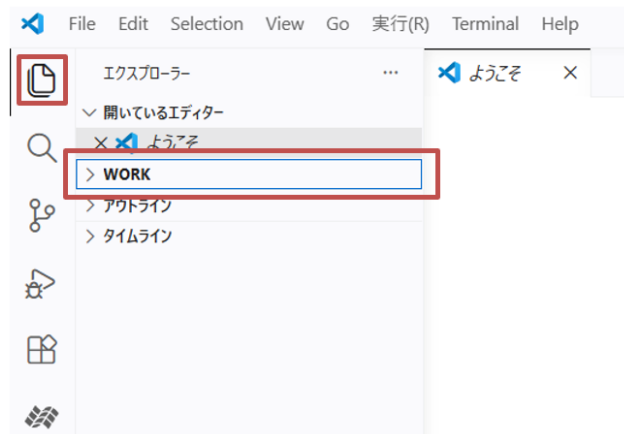


図 9: 追加した作業用フォルダの表示を確認

## 5. プロンプト作成のコツ

演習をはじめる前に、いくつかのプロンプト作成のコツを記載します。

### 5.1. 「IDL」「IDL Notebook」を明示する

- 依頼文に「IDL」を含めると、IDL の実行や IDL のコンテキスト(文脈)に寄せやすくなります。
- Notebook を作成したい場合は「IDL Notebook」を明記し、Python/Jupyter の Notebook ではなく VS Code ネイティブの IDL Notebook に誘導します。

### 5.2. コンテキスト（文脈）を渡す

Copilot Chat のクリップ（paperclip）から、コードやフォルダ、複数ファイルなどをコンテキストとして追加できます。

例：

- この \*.pro を添付するので、処理内容を要約して改善してください
- このフォルダ（workspace）内のファイルを参照して Notebook を作ってください
- この選択した箇所について説明してください。

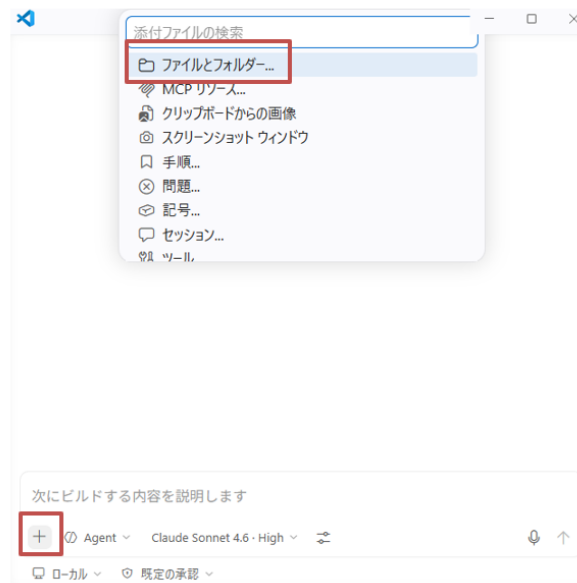


図 10: チャットでの明示的なコンテキストの追加

エディタからファイルを開いている場合、そのファイルや選択箇所をコンテキストとしてチャットに渡すことが可能です。選択箇所については、チャットにファイル名の右側に行番号が表示されています。

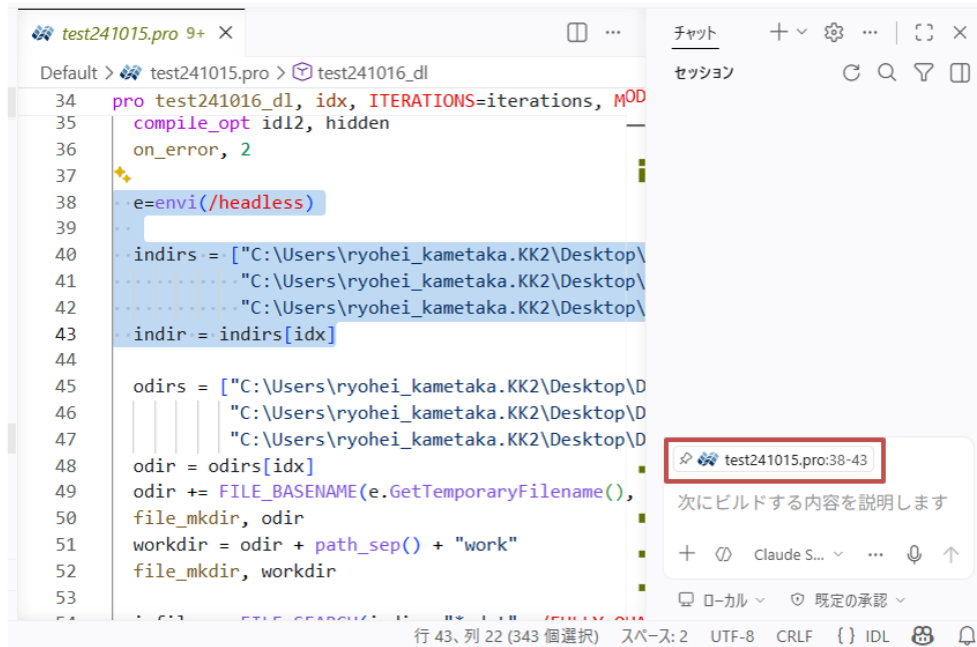


図 11:コード選択箇所とチャットへの反映

### 5.3. 迷走したら止める・分割する

- 意図と違う方向に進んだ場合は Stop して、要求を短く明確に言い直します。
- 話題が変わる場合は新規チャットに分けると、コンテキストの混乱を避けられます。

## 6. 演習 1 : HelloWorld plot (IDL Notebook)

### 6.1. 本演習の概要

IDL Agent の公式ドキュメントにおいて例題として示されている“hello-world”の IDL Notebook を作成し、簡単なプロット（正弦波）を実行して、基本となる操作手順を確認します。

### 6.2. 期待するアウトプット

- IDL Notebook が生成され、正弦波のプロットが表示されること
- Notebook 内にコードの簡単な説明が含まれること（生成時に依頼します）。

### 6.3. 手順

- ① Copilot Chat を開き、Agent mode であることを確認します。
- ② 下の「プロンプト例」を入力します<sup>2</sup>。

IDL Notebook で“hello-world”の例を作ってください。  
 サイン波 (sinusoidal curve) を生成してプロットし、各セルに短い説明コメントも入れてください。  
 ノートブックの説明は日本語で作成してください。プロットする図のタイトルやラベルについては英語で記載してください。

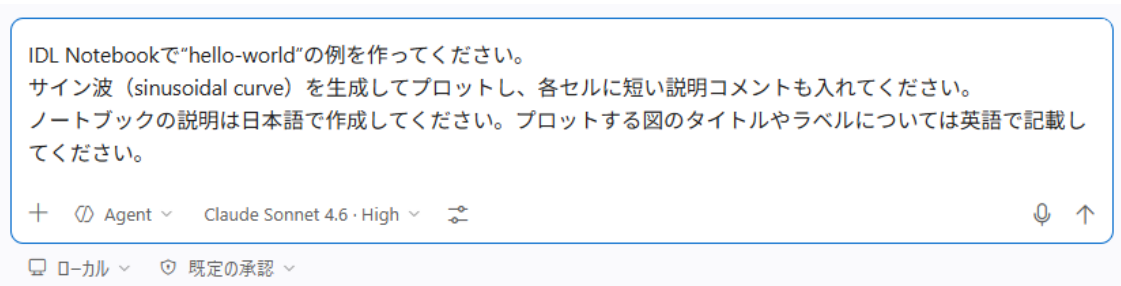


図 12: GitHub Copilot Chat へのプロンプトの入力

<sup>2</sup> Notebook を確実に IDL で作らせたい場合は、必ず「IDL Notebook」と明記します。

- ③ 実行途中で Copilot がツール実行を求めたら実行/承認 (Approve/Allow) します。具体的には以下の図にあるように、チャットからのレスポンス/IDL for VSCode (MCP サーバ)が IDL Notebook を実行しようとしているため、「このセッションで許可する」ボタンから実行許可を与えます。



図 13:実行許可を与える

- ④ 生成された IDL Notebook ファイル(.idlnb)を開き、内容を確認してください。

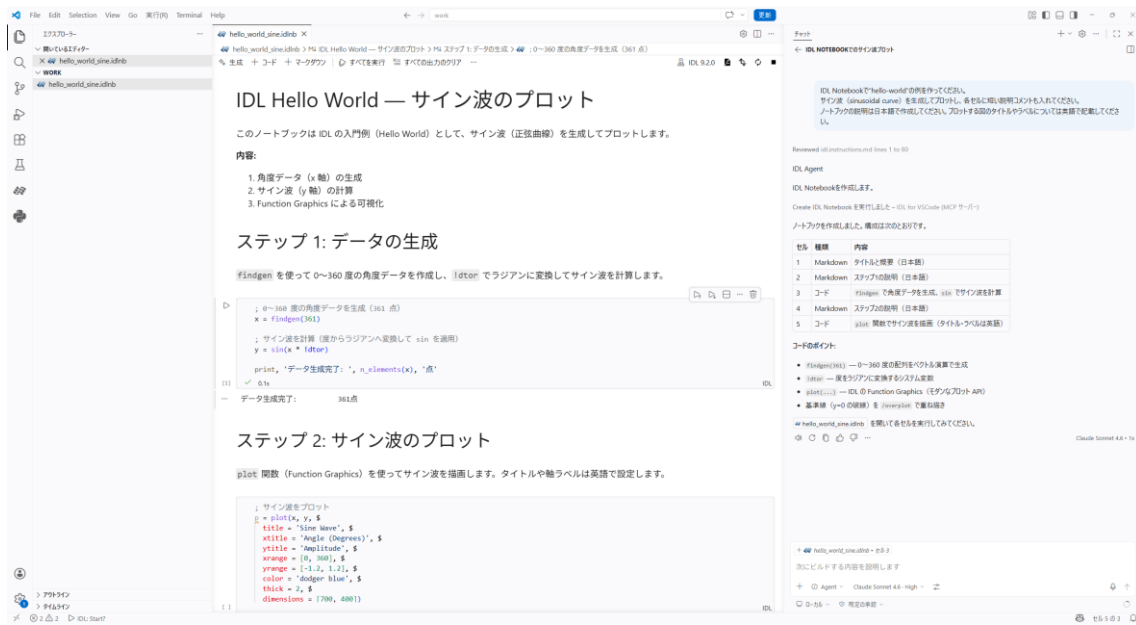


図 14: 出力された IDL ノートブックを開いた状態

- ⑤ 開いたファイルのステップ 2 のセルまでを実行すると、プロットが描画されます。まず、IDL のソースコードとして認識されていることを確認します。セルの右下に「IDL」と表示されていることを確認し、セルのコード左部にある実行ボタンからセルの処理を実行してください。描画結果がセルの下に表示されます。

## ステップ 2: サイン波のプロット

`plot` 関数 (Function Graphics) を使ってサイン波を描画します。タイトルや軸ラベルは英語で設定します。

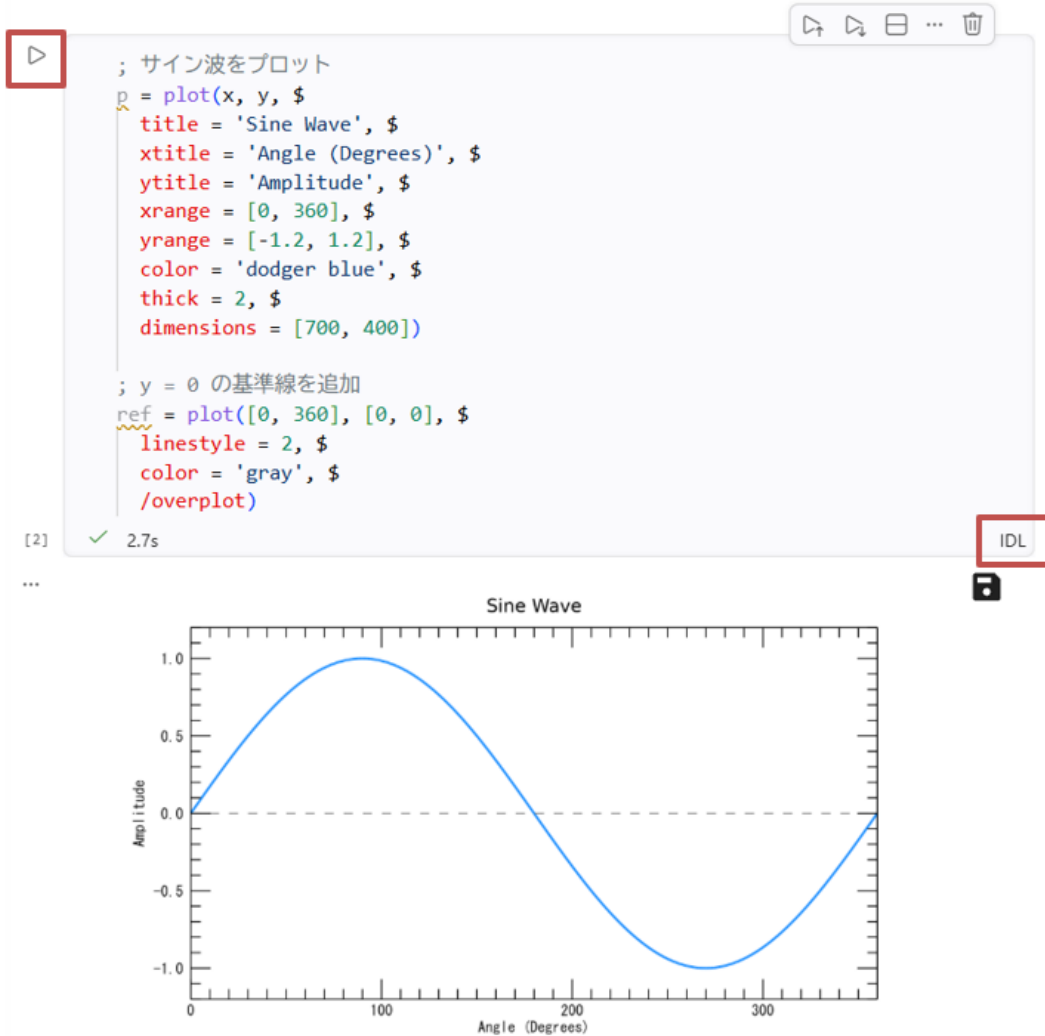


図 15: IDL Notebook にてセルの実行と結果の描画

## 7. 演習 2 : CSV ファイルの読み込みとプロット

### 7.1. 本演習の概要

IDL のインストールディレクトリにある、金星の衝突クレーターに関する基本的な情報を記録したサンプルデータ VenusCraterData.csv を READ\_CSV 関数で読み込み、直径でフィルタして散布図と回帰直線を作成します。

[Windows] C:\Program Files\NV5\IDL92\examples\data\VenusCraterData.csv  
 [macOS] /Applications/NV5/IDL92/examples/data/VenusCraterData.csv  
 [Linux] /usr/local/NV5/IDL92/examples/data/VenusCraterData.csv

	A	B	C	D	E	F	G	H
1	Name	Lat	Lon	Diameter	dnd	EjectaRadius	CentralStructure	
2	Mead	12.5	57	268.7	19	110.2	193.6	
3	Isabella	-29.9	204.2	176	12	162	141.1	
4	Meitner	-55.6	321.6	150.7	15	64.9	114.2	
5	Klenova	78.2	104.7	141.9	15	78.7	100.1	
6	Stanton	-23.2	199.3	107	32	96.3	77.6	
7	Cleopatra	65.9	7	105	-52	0	60	
8	Baker	62.5	40.4	105	-2	43	59.6	
9	Rosa Bonheur	9.7	288.8	102.2	12	54.8	59.7	
10	Joliot-Curie	-1.7	62.4	100.9	16	76.3	57.9	
11	Cochran	51.9	143.4	98.1	15	76.4	53	
12	Maria Celeste	23.4	140.4	96.6	3	65.3	61.7	
13	Sayers	-67.5	229.6	93.6	-1	87.6	48	
14	Greenaway	22.9	145.1	92.3	51	61.1	50	
15	Potanina	31.7	53	89.6	2	63.2	54	

図 16: VenusCraterData.csv

表 1: VenusCraterData.csv のフィールド情報

カラム名	内容
名称	クレーターの国際天文学連合 (IAU) による正式名称。
緯度	クレーター中心の緯度。誤差は±0.05°です。
経度	クレーター中心の経度。誤差は±0.05°です。
リム間直径	クレーターの平均直径 (km)。
DND	クレーターの底 (dni) と周辺の地形上の地点 (dno) における後方散乱係数の差。 (1 DN = 0.2 dB のレーダー後方散乱に相当)
噴出物半径	クレーターの噴出物エプロンの平均半径 (km)。
中心構造物直径	中心構造物を囲む面積と同等の面積を持つ円の直径 (km)。

## 7.2. 期待するアウトプット

- CSV ファイルを読み込み、条件に当てはまるレコードのみを抽出する
- リム間直径と噴出物半径の散布図を描画する（SCATTERPLOT 関数）
- LINFIT 関数で算出した回帰直線を描画結果に重ね描きする
- 各ステップを対話的に、ソースファイルを更新する形で進め、最終的に IDL Notebook として出力する

## 7.3. 手順

- ① 演習 1 で利用していたチャットは閉じ、新しいチャットを立ち上げてください。  
まず、以下のプロンプトを実行し、ソースファイルを作成させます。

現在のワークスペースに testScatter.pro というファイル名の IDL ソースファイルを作成し、testScatter というプロシージャを定義してください。  
プロシージャの内容は何も記載せず、そのままにします。

- ② 出力されたソースファイル testScatter.pro を VSCode で開きます。このソースファイルでは、IDL Agent によって自動的にプロシージャ testScatter が定義されているので、[保持]ボタンからこの内容を受け入れます。

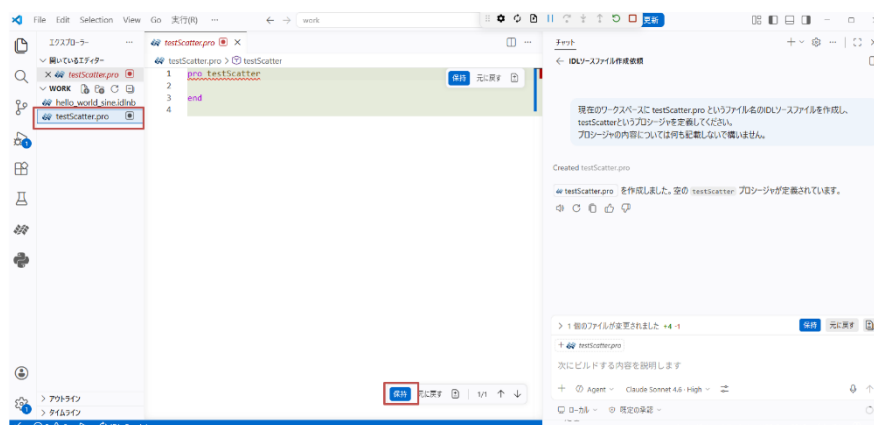


図 17: 自動で定義された testScatter プロシージャ

- ③ 以下のプロンプトをチャットへ入力し、実行して処理を追加します。その際、コンテキストとしてソースファイルが指定されていることを確認してください。



ます。

読み込んだファイルにはどのようなカラムが含まれているかを教えてください。



図 20: カラムの種類の確認

「クレーター直径」と「放出物の半径」について散布図を作成する処理を追加してください。条件として、クレーター直径が 10km 以上のもののみを対象としてください。

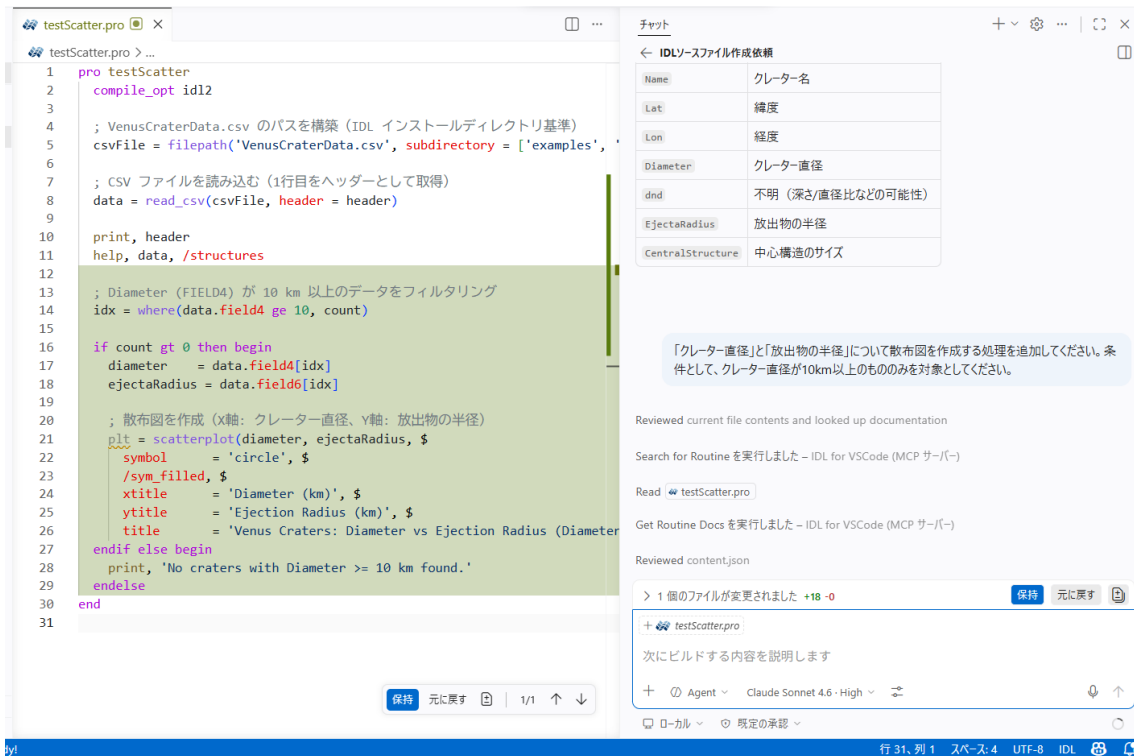


図 21: 条件に当てはまるデータの抽出と散布図のプロット処理の追加

回帰係数を求め、結果に回帰線を重ねて表示してください。

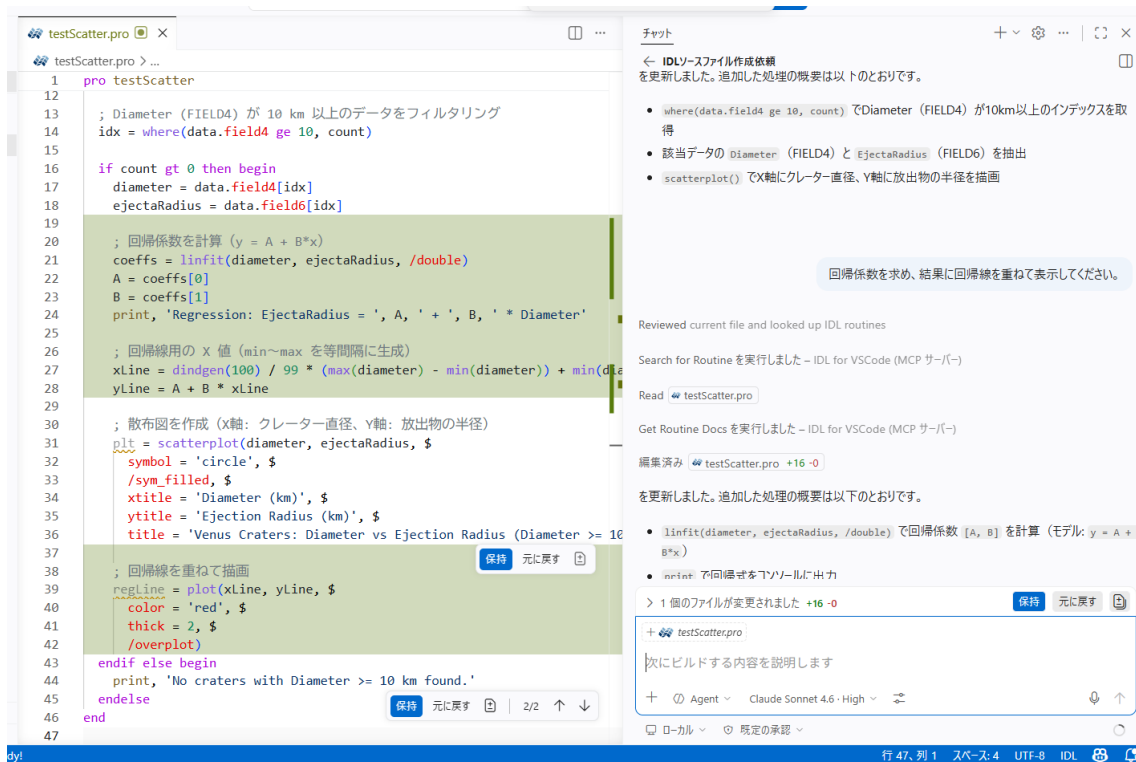


図 22: データから回帰係数を算出し、散布図に重ねる処理の追加

- ⑥ ここまでで完成したソースファイルから IDL Notebook を作成します。以下のプロンプトをチャットに入力して実行し、結果を確認します。  
各セルを実行し、正しく表示されることを確認してください<sup>3</sup>。

この完成したソースファイルの内容を IDL ノートブックとしてまとめてください。各セルには、短い説明コメントも入れてください。

<sup>3</sup> 実行時にエラーとなった場合には、チャットでエラーが出力されている箇所とエラーメッセージを入力して、正しいものに修正するよう依頼します。

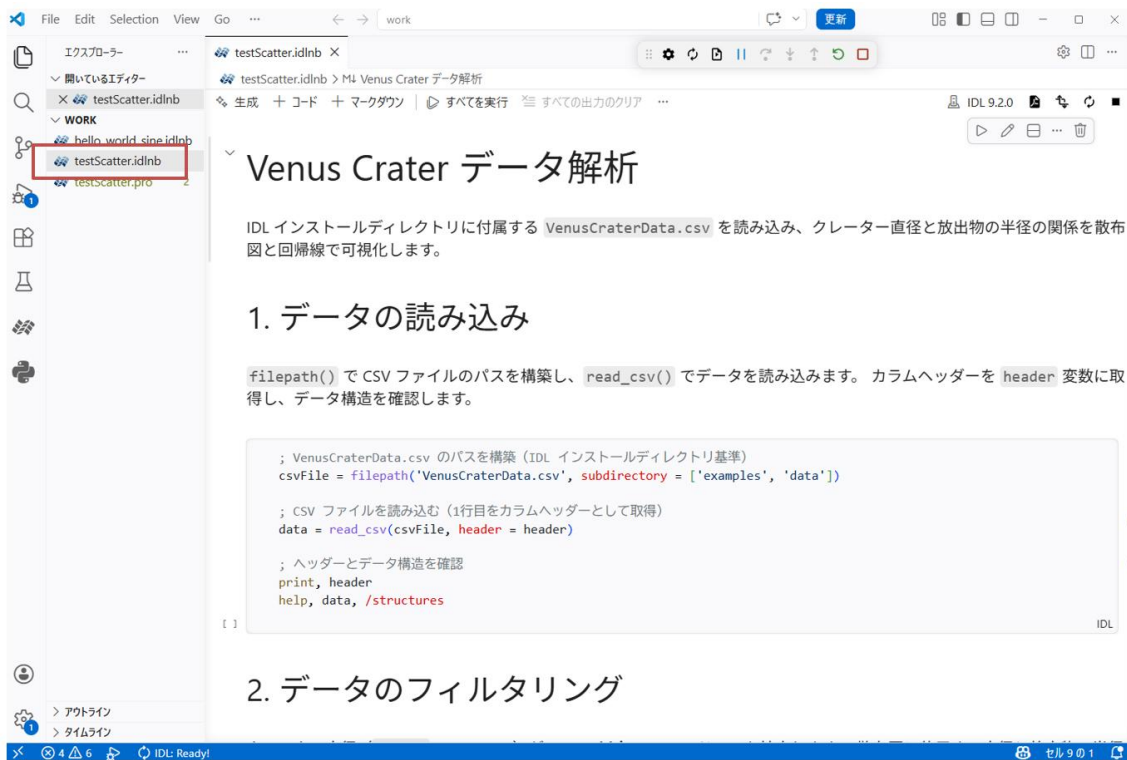


図 23: 出力された IDL Notebook の確認

- ⑦ 各セルを実行し、最後の散布図と回帰直線の描画が正しく表示されるか確認してください<sup>4</sup>。

<sup>4</sup> 実行時にエラーとなった場合には、チャットでエラーが出力されている箇所とエラーメッセージを入力して、正しいものに修正するよう依頼します。

## 4. 散布図と回帰線の描画

散布図に `scatterplot()` を使い、`plot()` の `/overplot` キーワードで回帰線を重ねて表示します。

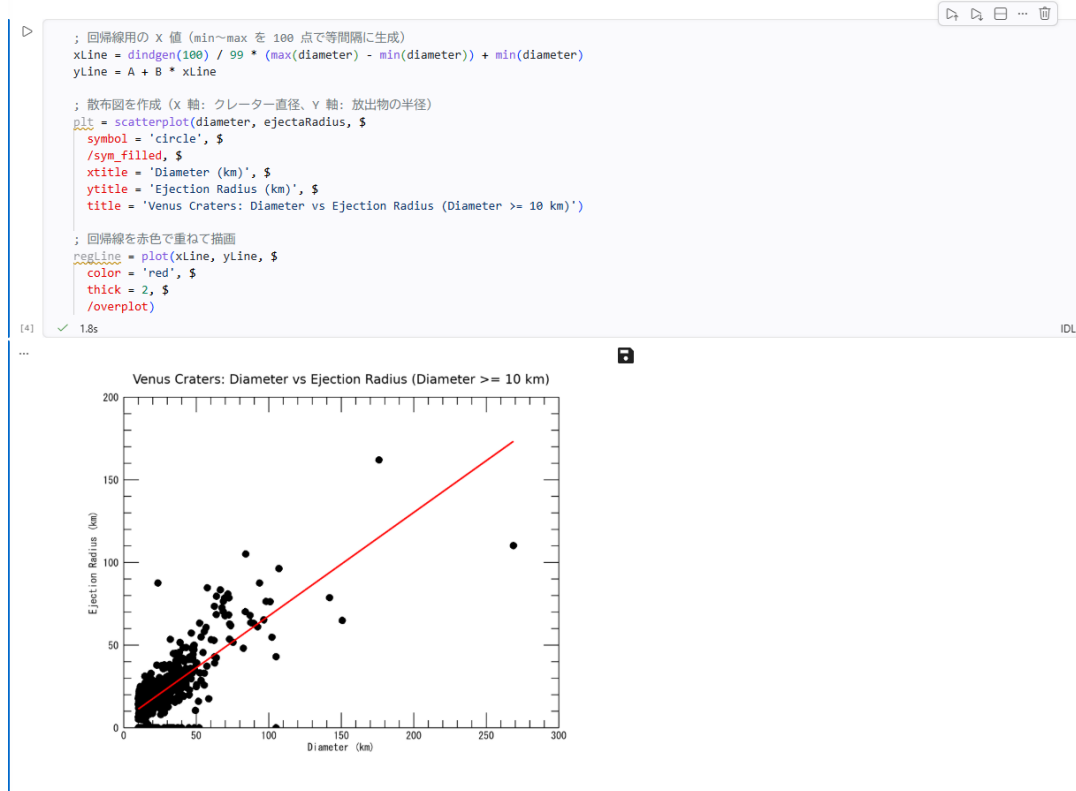


図 24: 各セルの実行と散布図と回帰線の描画

## 7.4. 演習 2 の出力コード例

出力されるサンプルコードの例を示します。これは利用環境や、使用している大規模言語モデルなどによって異なる可能性があります。

```

pro testScatter
  compile_opt idl2

; VenusCraterData.csv のパスを構築 (IDL インストールディレクトリ基準)
csvFile = filepath('VenusCraterData.csv', subdirectory = ['examples',
'data'])

; CSV ファイルを読み込む (1 行目をヘッダーとして取得)
    
```

```

data = read_csv(csvFile, header = header)

print, header
help, data, /structures

; Diameter (FIELD4) が 10 km 以上のデータをフィルタリング
idx = where(data.field4 ge 10, count)

if count gt 0 then begin
  diameter = data.field4[idx]
  ejectaRadius = data.field6[idx]

  ; 回帰係数を計算(y = A + B*x)
  coeffs = linfit(diameter, ejectaRadius, /double)
  A = coeffs[0]
  B = coeffs[1]
  print, 'Regression: EjectaRadius = ', A, ' + ', B, ' * Diameter'

  ; 回帰線用の X 値(min~max を等間隔に生成)
  xLine = dindgen(100) / 99 * (max(diameter) - min(diameter)) +
min(diameter)
  yLine = A + B * xLine

  ; 散布図を作成(X 軸: クレーター直径、Y 軸: 放出物の半径)
  plt = scatterplot(diameter, ejectaRadius, $
    symbol = 'circle', $
    /sym_filled, $
    xtitle = 'Diameter (km)', $
    ytitle = 'Ejection Radius (km)', $
    title = 'Venus Craters: Diameter vs Ejection Radius (Diameter >=
10 km)')

  ; 回帰線を重ねて描画
  regLine = plot(xLine, yLine, $
    color = 'red', $
    thick = 2, $

```

```
    /overplot)  
endif else begin  
    print, 'No craters with Diameter >= 10 km found.'  
endelse  
end
```

## 8. 付録：トラブルシューティング

ここでは、既知の不具合や回避策について記載します。

### 8.1. IDL が起動しない／IDL の場所が検出されない

まずは IDL9.2 自体が起動できることを確認してください。Windows であれば [すべてのアプリ] > [ENVI6.2]または[IDL9.2] > [IDL 9.2 Command Line]アイコンを選択することで、macOS または Linux であればターミナルから idl コマンドを実行することで、それぞれコマンドラインモードにて IDL が起動できます。起動できない場合には、IDL のライセンスが認証されているか、(フローティングライセンスの場合)ライセンスサーバが停止していないかといった点をご確認ください。

IDL 自体の起動に問題ない場合、IDL for VSCode が正しく IDL 実行ファイルパスを認識できていない可能性があります。IDL for VSCode では、利用する IDL の実行ファイルの配置場所が指定できます。通常であれば最後に端末にインストールされたバージョンの IDL 実行ファイルパスが自動で取得されますが、IDL 自動検出に失敗した場合は手動指定が必要になることがあります。

拡張機能 IDL for VSCode アイコン > Quick Access > Specify IDL Directory から IDL の実行ファイルのあるパスを指定

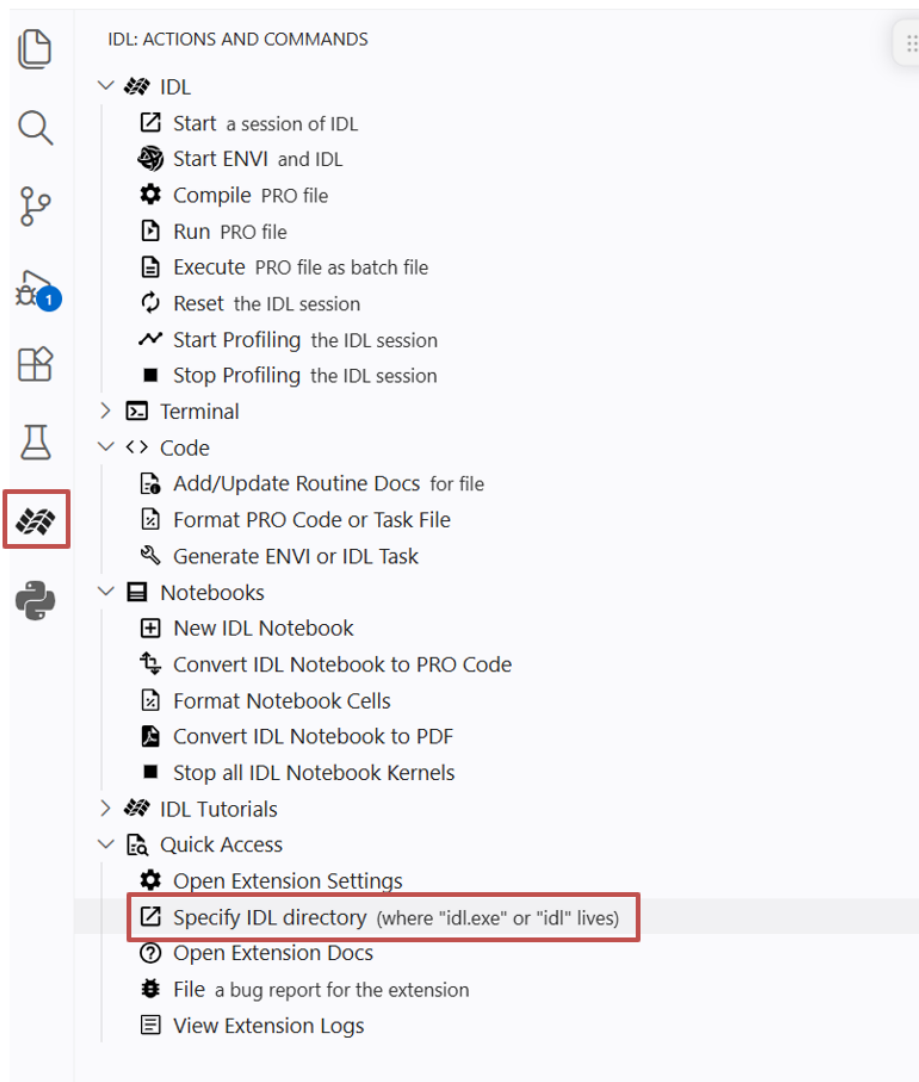


図 25: IDL for VSCode の IDL 実行ファイルパス指定箇所

## 8.2. エラーログを表示するにはどうすればよいですか？

ほとんどのエラーはダイアログでログへのリンクを示しますが、手動でアクセスする必要がある場合は、8.1 章の記載を参考に、拡張機能 IDL for VSCode アイコン > Quick Access > View Extension Logs を実行してください。

## 9. 参考リンク

- [IDL Agent 本社製品ページ](#)
- [IDL and ENVI Agent FAQ](#) :
  - 要件、ライセンス、トラブルシュートなどについて記載しています。
- [VSCode 公式ページ](#)
- IDL for VSCode
  - [マーケットプレイス](#)
  - [公式 HELP ドキュメント](#)
- [GitHub Copilot 公式ページ](#)
- [ENVI Agent](#)
  - ENVI Agent は、リモートセンシング解析ソフト「ENVI」での処理を、言語で対話しながら計画・実行できる AI 支援機能です。解析の目的を伝えるだけで、適切なワークフロー提案や前処理・実行までを支援し、作業を効率化します。
- [GeoAgent](#)
  - GeoAgent は、NV5 が提供する地理空間分野向けのエージェント型 AI で、ユーザーの目的を言葉で理解し、データ探索から解析、成果物生成までのエンドツーエンドの地理空間ワークフローを自動構築・実行します。GIS やリモートセンシングの専門知識がなくても、高度な解析結果を迅速に得られるのが特徴です。

## 10. 問い合わせ先

ご不明点・ご質問がある場合は、弊社の製品技術サポートにお問い合わせください。

NV5 Geospatial 株式会社 製品技術サポート  
support\_jp@NV5.com