

Python (4)

PYTHON

ETL



ETL

Extract, Transform and Load (Extraer, Transformar y Cargar, o ETL) es el proceso que permite a las organizaciones mover datos desde múltiples fuentes, reformatearlos y limpiarlos, y cargarlos en otro almacén de datos para ser analizados y apoyar un proceso de negocio.

Extraer

Consiste en extraer los datos desde los sistemas de origen, generalmente en forma de ficheros planos o bases de datos.

Transformar

Consiste en aplicar una serie de reglas sobre los datos extraídos con el fin de obtener los datos que finalmente serán cargados. Ej.: seleccionar únicamente determinadas columnas, traducir códigos, totalizar valores, combinar valores, dividir datos, etc.

Cargar

Consiste en cargar en el sistema de destino los datos transformados de la fase anterior.

ETL

¿QUÉ ES ETL?



PYTHON

EDA



EDA



PYTHON

CRISP-DM



CRISP-DM

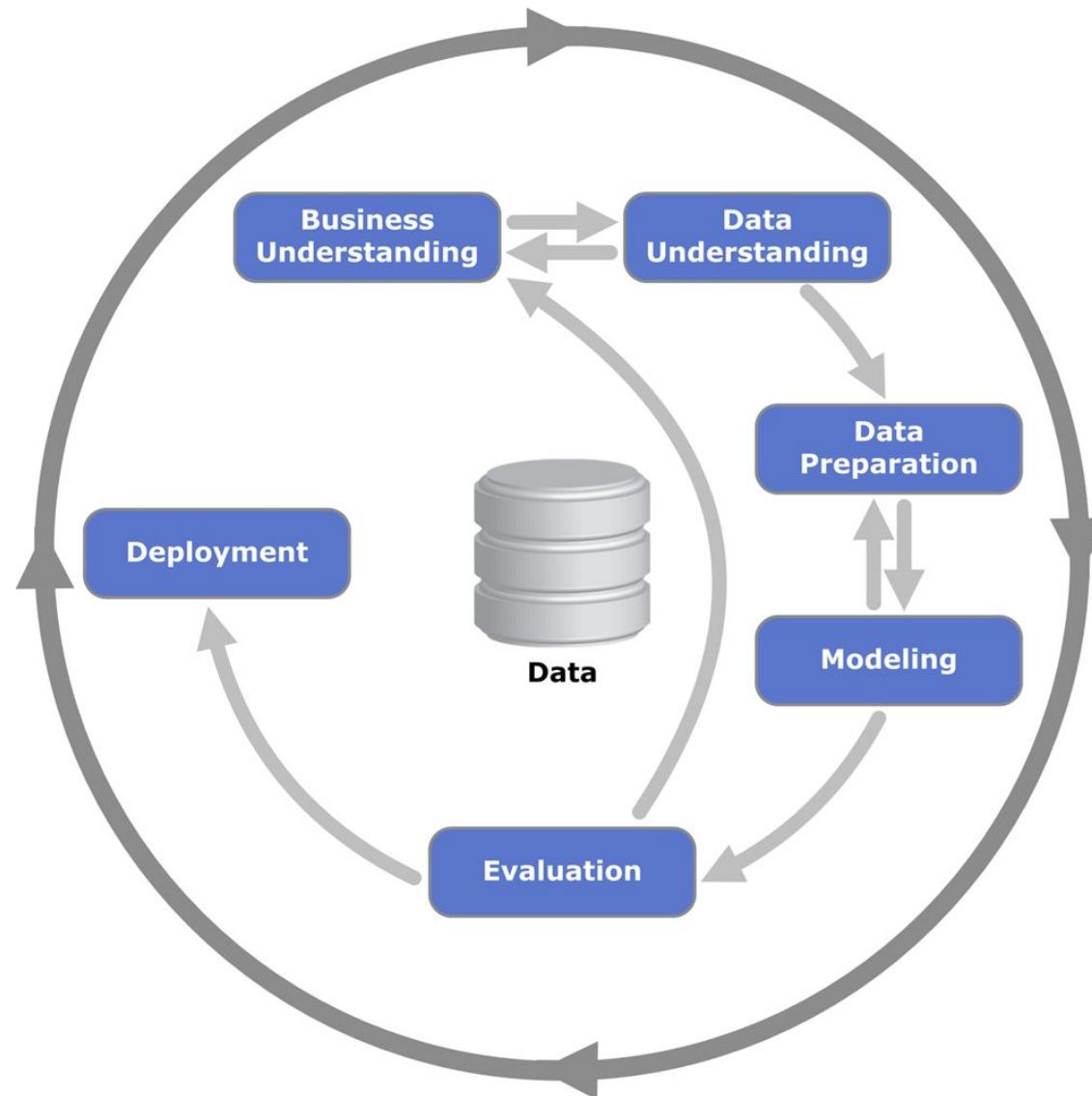
CRISP-DM (Cross-Industry Standard Process for Data Mining) es un estándar abierto que describe los enfoques comunes utilizados en el campo de la minería de datos.

El proceso consta de seis pasos o fases principales:

1. Comprensión del negocio (Business Understanding)
2. Comprensión de los datos (Data Understanding)
3. Preparación de los datos (Data Preparation)
4. Fase de modelado (Modeling)
5. Evaluación (Evaluation)
6. Despliegue (Deployment).

Generalmente estos pasos se llevan a cabo de forma secuencial, pero en ocasiones es necesario retroceder a pasos anteriores y repetir ciertas tareas.

CRISP-DM



Comprensión del negocio

Todo proyecto comienza con la comprensión del negocio.

No es posible construir un modelo elegante sin una comprensión profunda del negocio.

Esta fase se centra en:

- Entender el negocio.
- Comprender los objetivos y requisitos del proyecto desde una perspectiva comercial.
- Convertir este conocimiento en una definición de problema de análisis de datos.
- Diseñar un plan preliminar para lograr los objetivos.

Comprensión de los datos

Los datos del negocio pueden presentarse en formatos diferentes y almacenarse de múltiples formas.

Esta fase comprende:

- Tener en cuenta los requisitos de datos
- Llevar a cabo la recopilación inicial de los datos
- Familiarizarse con los datos
- Identificar problemas de calidad de los datos y descubrir las primeras ideas sobre los datos

Preparación de los datos

Incluso aunque los datos se presenten bien estructurados suelen encontrarse desordenados, por lo que es necesario prepararlos.

Esta fase comprende las actividades necesarias para, a partir de los datos sin procesar iniciales, construir el conjunto de datos final, que es el que se incorporará a las herramientas de modelado.

Esta fase comprende la selección de tablas, registros y atributos, así como la transformación y limpieza de datos.

Se trata de la fase que generalmente consume más tiempo, y representa aproximadamente las tres cuartas partes del trabajo de un analista de datos.

Las tareas que comprende esta fase son:

- Adquisición de datos (desde una base de datos, un archivo, etc.)
- Limpieza de los datos (identificar y corregir errores en los datos, tratar con los datos que faltan, etc.)
- Integrar datos (combinar diferentes conjuntos de datos)
- Transformar y enriquecer los datos (crear nuevas funciones a partir de funciones existentes)

Modelado

Durante la fase de modelado se seleccionan y aplican diferentes técnicas de modelado, y se calibran sus parámetros a valores óptimos.

En general existirán diversas técnicas para el mismo tipo de problema de minería de datos.

Algunas técnicas tienen requisitos específicos sobre la forma de los datos, de modo que en ocasiones es necesario retroceder a la fase de preparación de datos.

Las técnicas comunes de modelado incluyen:

- regresión
- clasificación
- agrupamiento

Evaluación

Una vez alcanzada esta etapa contamos con un modelo de datos de calidad suficiente desde el punto de vista del análisis de datos.

Antes de su implementación final es importante evaluar más a fondo el modelo y revisar los pasos ejecutados para construirlo, con el fin de asegurarse de que logra cumplir los objetivos comerciales.

Uno de los objetivos principales de esta fase es determinar si existe cualquier aspecto del negocio que no haya sido debidamente considerado.

Al finalizar esta fase debemos tomar una decisión acerca del uso de los resultados de la minería de datos.

Despliegue

En general la creación del modelo no es el final de proyecto.

Incluso si el propósito del modelo es aumentar el conocimiento de los datos, el conocimiento obtenido deberá organizarse y presentarse de una manera útil para el cliente.

En función de los requisitos del proyecto, la fase de despliegue puede ser tan simple como generar un informe o tan compleja como implementar una puntuación de datos repetible, o un proceso de extracción de datos.

En muchos casos será el cliente, no el analista de datos quien llevará a cabo los pasos de implementación.

Proceso iterativo

Es común que al comienzo de un proyecto no tengamos mucho conocimiento del dominio, o que existan problemas con los datos o el modelo no sea lo suficientemente valioso como para ponerlo en producción.

La principal ventaja del modelo CRISP-DM es que no es una ruta lineal desde el inicio del proyecto hasta su implementación, sino que permite retroceder pasos o saltar entre fases si es necesario.

PYTHON

Instalación de Anaconda



Anaconda

Anaconda es una distribución libre y abierta de Python y R utilizada en ciencia de datos que facilita la administración de paquetes mediante el gestor de paquetes conda.

Anaconda ofrece diversas herramientas utilizadas en Ciencia de Datos como Python, Jupyter Notebooks y pandas en una sola instalación.

Data science technology for a better world.

Anaconda offers the easiest way to perform Python/R data science and machine learning on a single machine. Start working with thousands of open-source packages and libraries today.

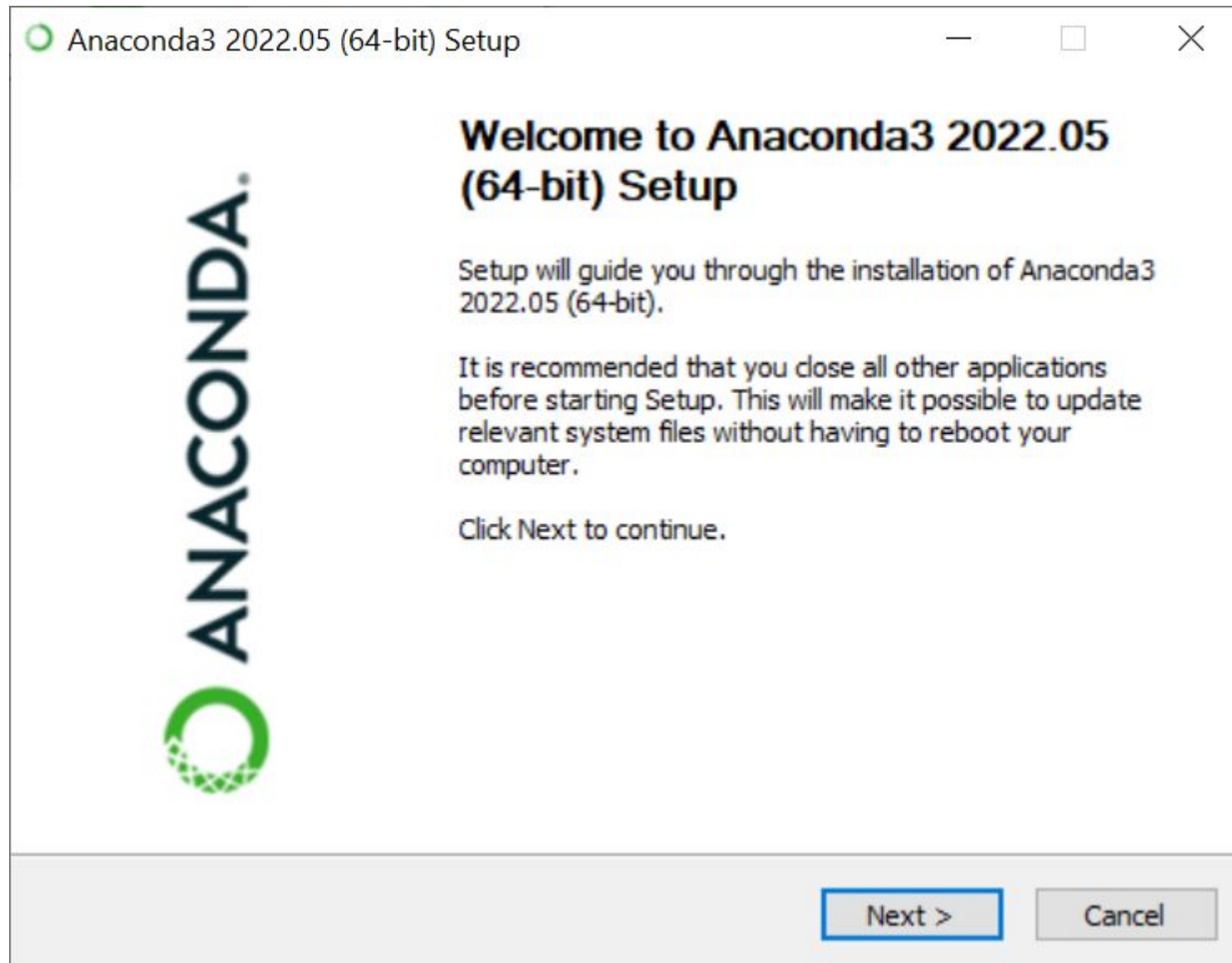
Download 

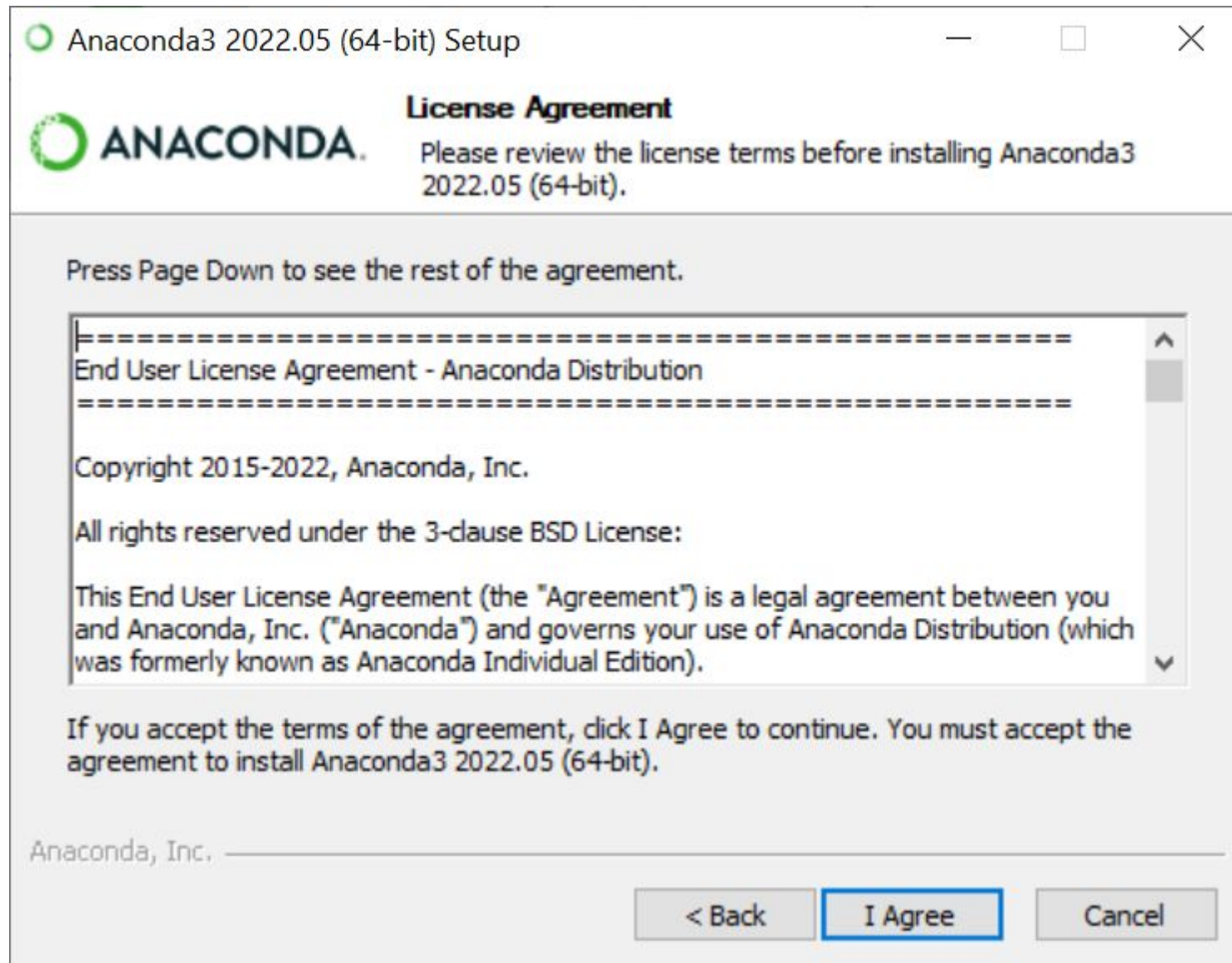
For Windows

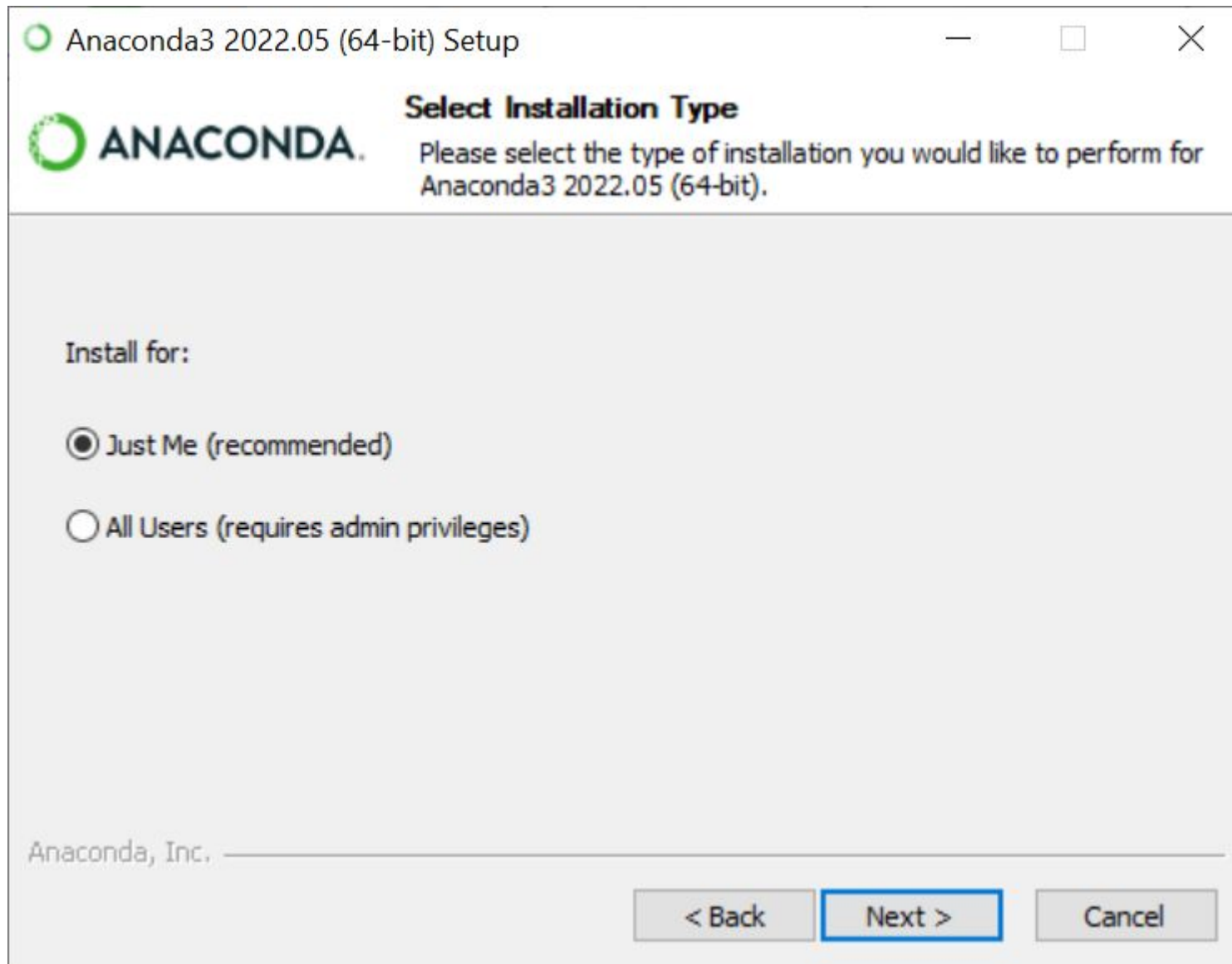
Python 3.9 • 64-Bit Graphical Installer • 594 MB

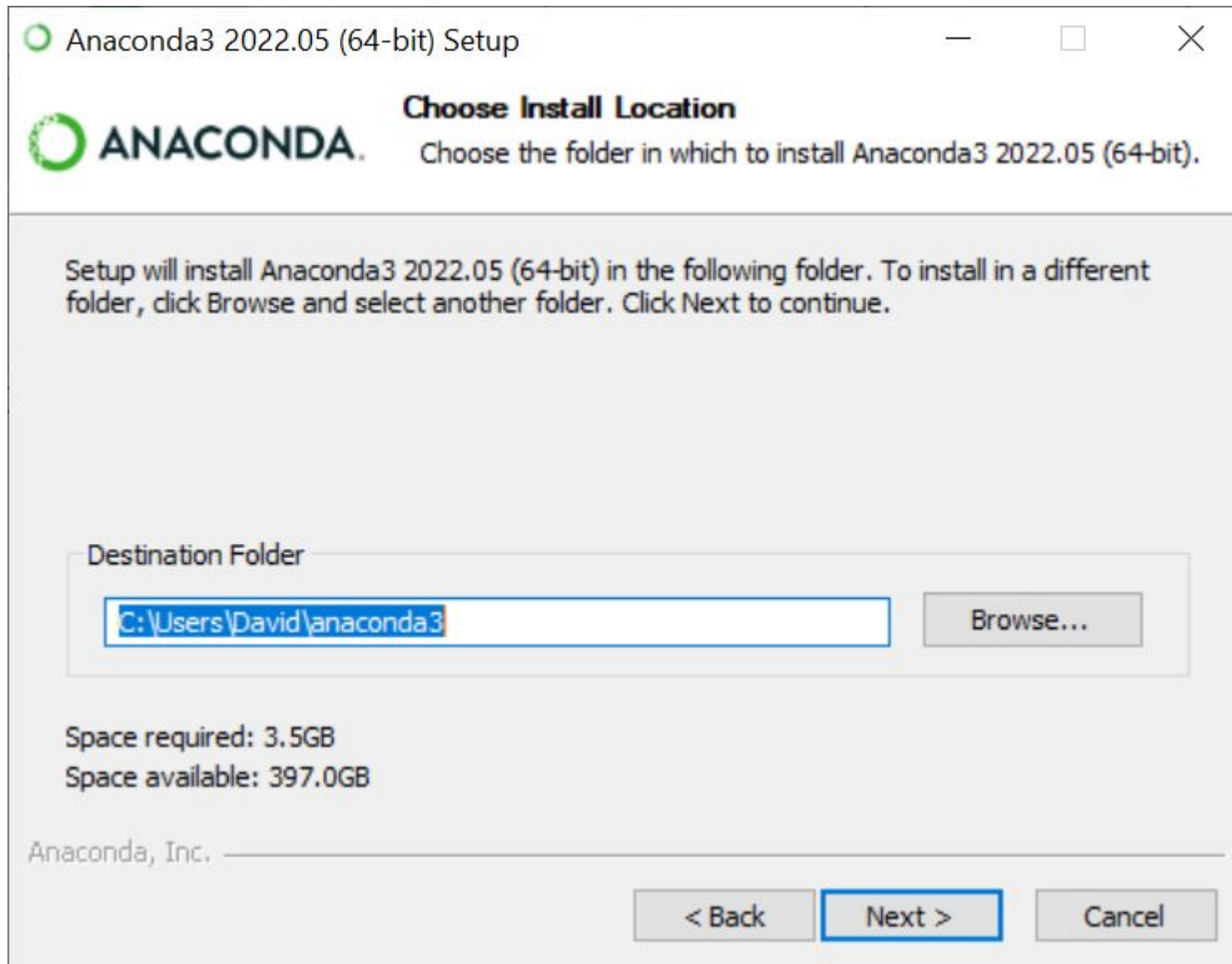
Get Additional Installers

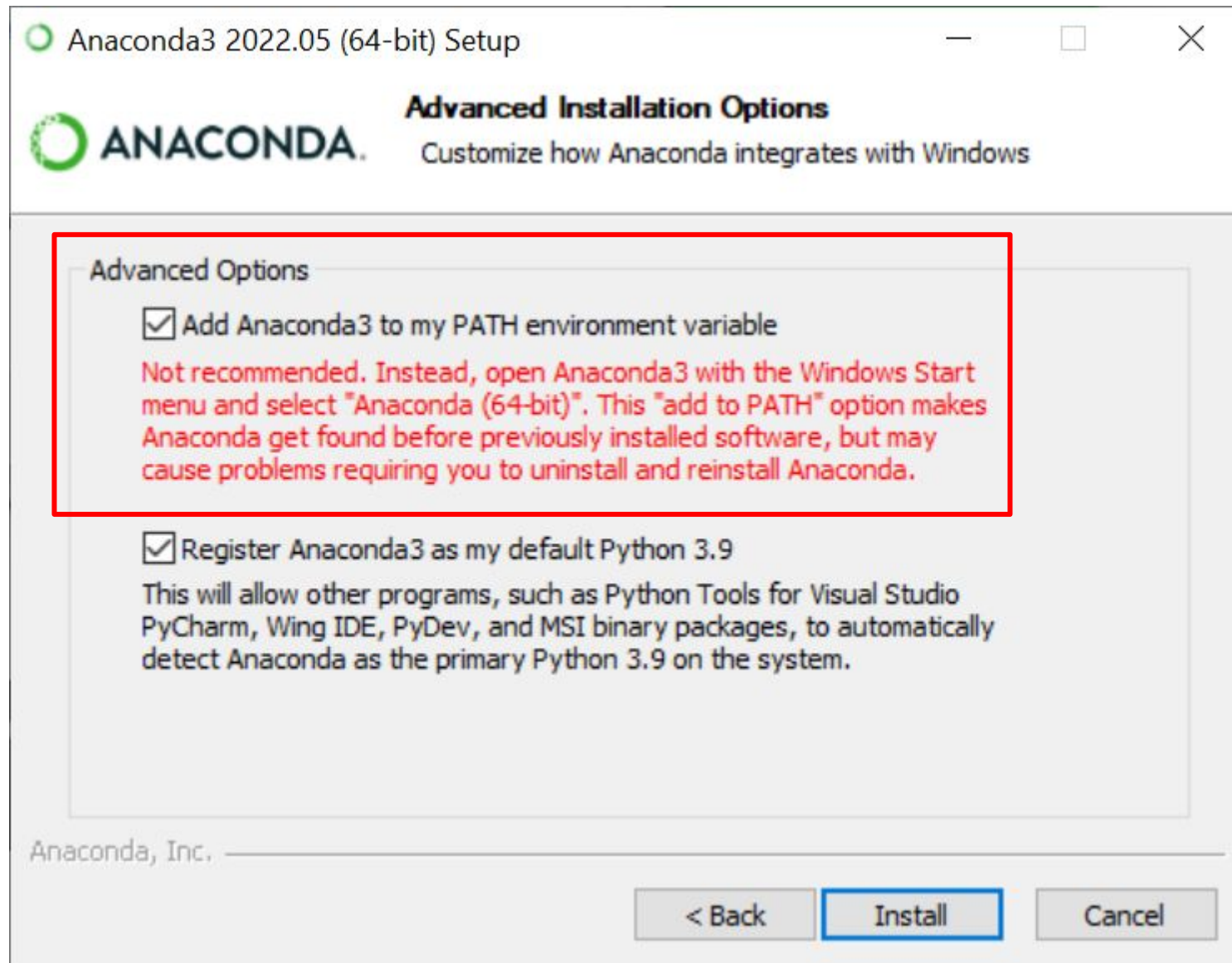


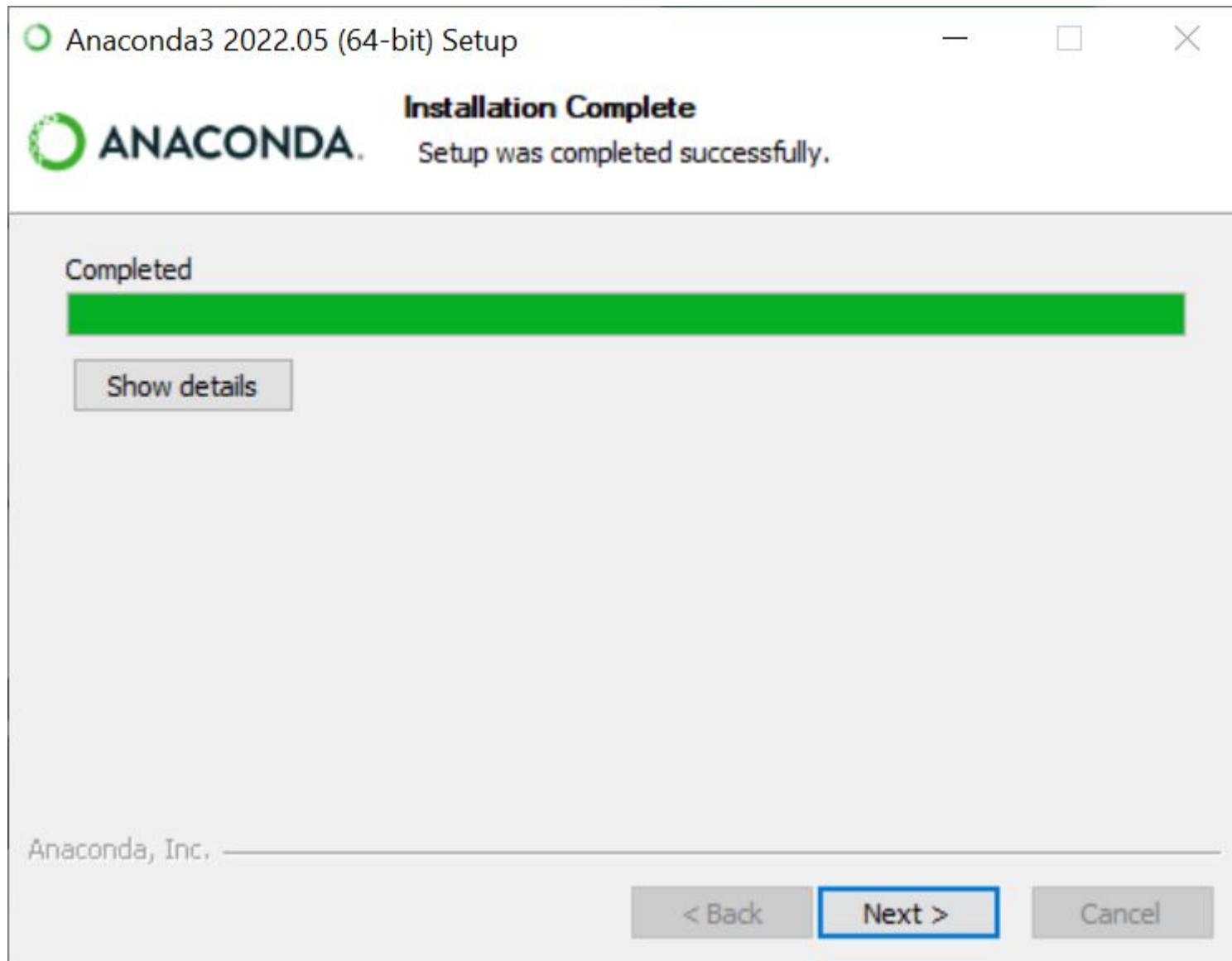


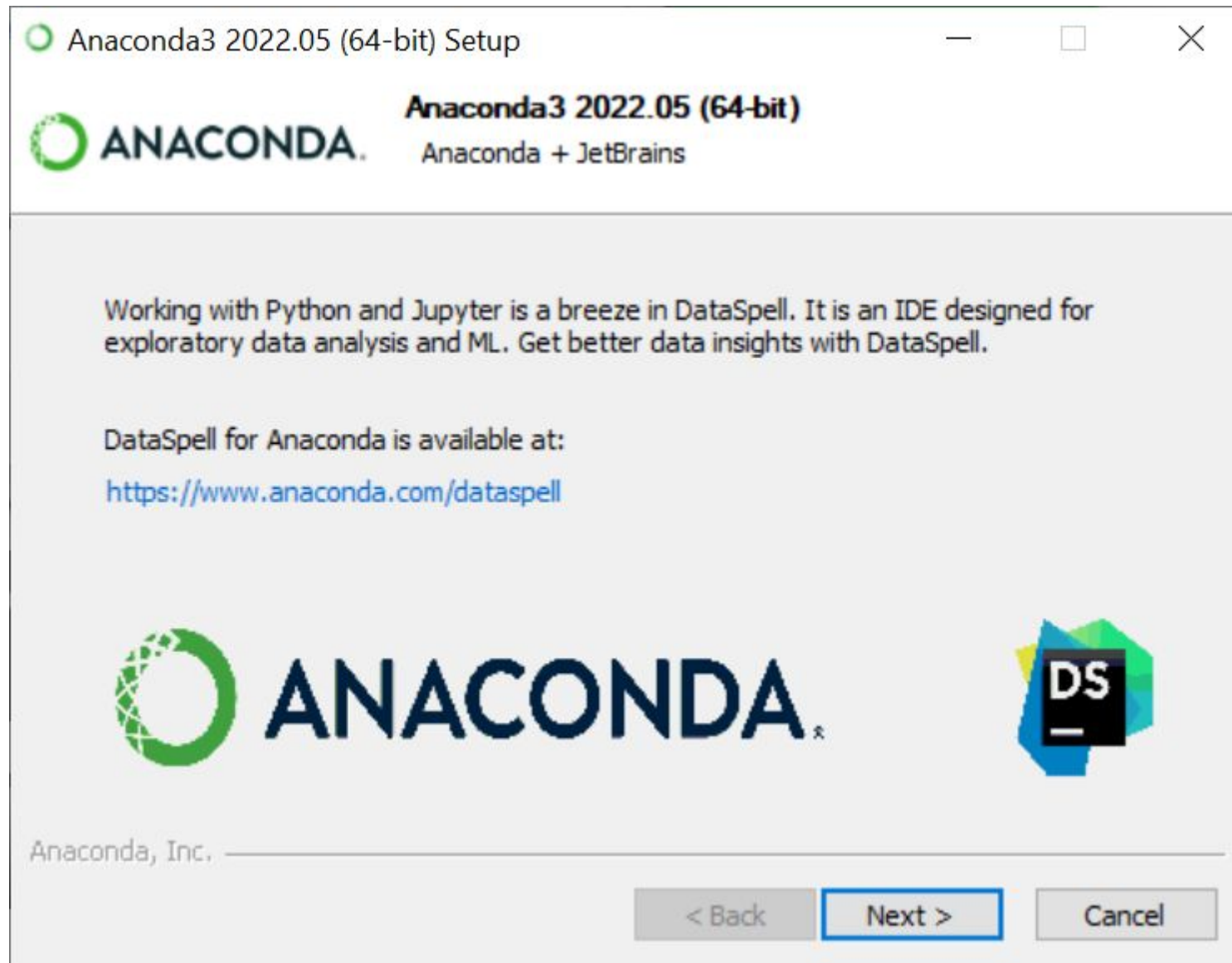


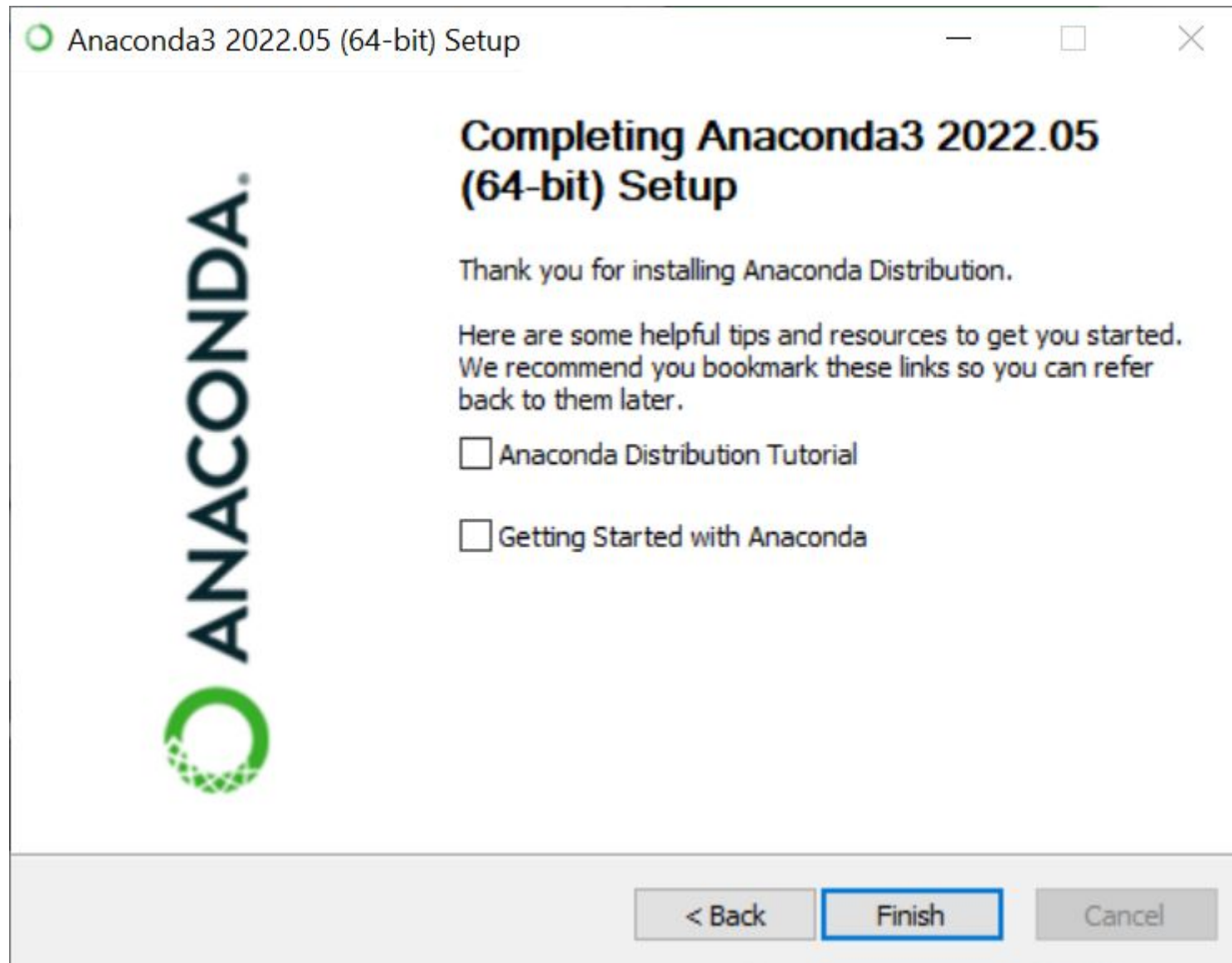












Anaconda Navigator

File Help

ANACONDA.NAVIGATOR Connect

Home

Environments

Learning

Community

ANACONDA.
Secure your software supply chain from the source
[Upgrade Now](#)

End-to-end package security, guaranteed

Documentation

Anaconda Blog

Twitter YouTube GitHub

Create Clone Import Backup Remove

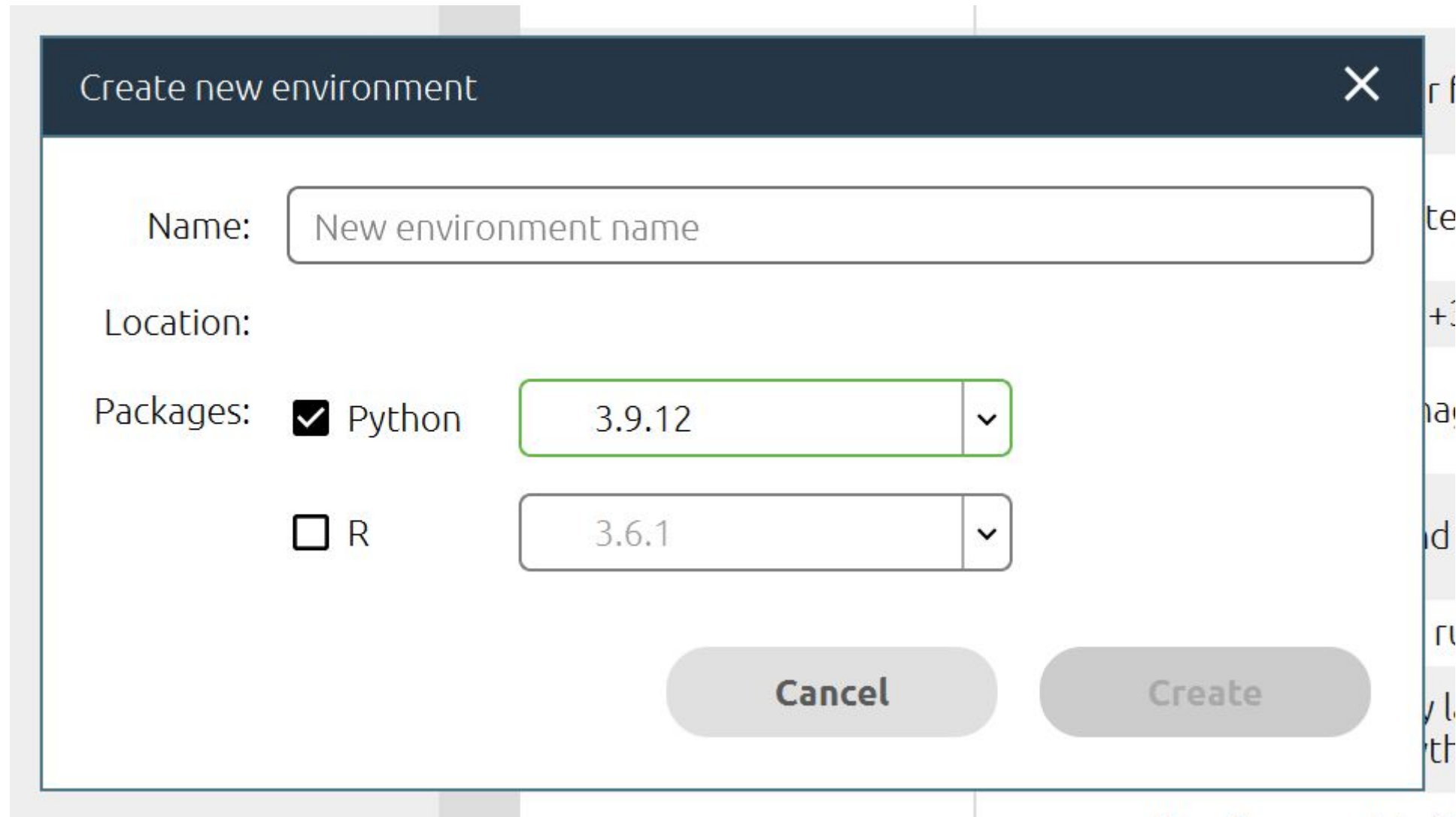
Search Environments

base (root)

Installed Channels Update index... Search Pack...

| Name | T | Description | Version |
|------------------------|---|---|-----------------------|
| ✓ _ipyw_jlab_nb_ext... | ○ | A configuration metapackage for enabling anaconda-bundled jupyter extensions | 0.1.0 |
| ✓ aiohttp | ○ | Async http client/server framework (asyncio) | 3.8.1 |
| ✓ aiosignal | ○ | Aiosignal: a list of registered asynchronous callbacks | 1.2.0 |
| ✓ alabaster | ○ | Configurable, python 2+3 compatible sphinx theme. | 0.7.12 |
| ✓ anaconda | ○ | Simplifies package management and deployment of anaconda | 2022.05 |
| ✓ anaconda-client | ○ | Anaconda.org command line client library | 1.9.0 |
| ✓ anaconda-project | ○ | Tool for encapsulating, running, and reproducing data science proje... | 0.10.2 |
| ✓ anyio | ○ | High level compatibility layer for multiple asynchronous event loop implementations on python | 3.5.0 |
| ✓ appdirs | ○ | A small python module for determining appropriate platform-specific dirs. | 1.4.4 |
| ✓ argon2-cffi | ○ | The secure argon2 password hashing algorithm. | 21.3.0 |
| ✓ argon2-cffi-bindings | ○ | Low-level python cffi bindings for argon2 | 21.2.0 |
| ✓ arrow | ○ | Better dates & times for python | 1.2.2 |
| ✓ astroid | ○ | A abstract syntax tree for python with inference support. | 2.6.6 |
| ✓ astropy | ○ | Community-developed python library for astronomy | 5.0.4 |

430 packages available



A dialog box titled "Create new environment" with a close button (X) in the top right corner. The dialog contains three main sections: "Name:", "Location:", and "Packages:". The "Name:" section has a text input field with the placeholder text "New environment name". The "Location:" section is currently empty. The "Packages:" section has two rows. The first row is for "Python", which is selected with a checked checkbox; it has a version dropdown menu showing "3.9.12". The second row is for "R", which is not selected with an unchecked checkbox; it has a version dropdown menu showing "3.6.1". At the bottom right of the dialog are two buttons: "Cancel" and "Create".

Create new environment

Name:

Location:

Packages:

☒ Python

☐ R

Cancel Create

PYTHON

Jupyter Notebooks




Jupyter Notebooks

Jupyter Notebooks es un entorno informático interactivo, de código abierto y basado en la web, que permite crear documentos con celdas que pueden contener código, texto, fórmulas matemáticas o gráficos.

Anaconda Navigator

File Help

 **ANACONDA.NAVIGATOR**


Connect ▾

Home

Environments

Learning




Community

 **ANACONDA.**
Secure your software supply chain from the **source**
[Upgrade Now](#)


End-to-end package security, guaranteed


Documentation


Anaconda Blog


  


Applications on base (root) Channels



IBM Watson Studio Cloud
IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.
[Launch](#)


JupyterLab
3.3.2
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
[Launch](#)


Notebook
6.4.8
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
[Launch](#)


Powershell Prompt
0.0.1
Run a Powershell terminal with your current environment from Navigator activated
[Launch](#)


Qt Console
5.3.0
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
[Launch](#)


Spyder
5.1.5
Scientific PYTHON Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
[Launch](#)



Quit

Logout

Files

Running

Clusters

Select items to perform actions on them.

Upload

New ▾



0



/ The Valley

Name ▾



..

ha

The notebook list is empty.

Notebook:


Python 3 (ipykernel)

Other:

Text File

Folder

Terminal

 jupyter Untitled (unsaved changes)



Logout

File

Edit

View

Insert

Cell

Kernel

Widgets

Help

Trusted



Python 3 (ipykernel) 



Run



Code



In []: |

Pandas

Características

Pandas nos permite:

- Limpiar
- Transformar
- Visualizar

los datos.

Mediante pandas podemos llevar a cabo tareas similares a las que realizaríamos en una hoja de cálculo, con algunas ventajas adicionales:

- Tamaño: pandas puede manejar hojas de datos mucho más grandes que suites de hojas de cálculo como Excel.
- Transformaciones complejas: pandas permite realizar transformaciones que serían inviables en una hoja de cálculo tradicional debido a su complejidad de cálculo y consumo de recursos.
- Automatización: la capacidad de automatización de una hoja de cálculo está limitada frente a las posibilidades que ofrece un lenguaje de programación como Python.

Pandas

Estructuras de datos

Las dos principales estructuras de datos en Pandas son:

- Las series
- Los Dataframes

Series

Las series son arrays unidimensionales.

Dataframes

Los Dataframes son arrays bidimensionales, y son el principal tipo de datos utilizado en pandas.

Al igual que una hoja de cálculo, un Dataframe tiene filas y columnas, las cuales se conocen también como Series.

Pandas

Dataframes

FILA



| | PROVINCIA | POBLACIÓN | CÓDIGO |
|---|-----------|-----------|--------|
| 0 | Madrid | 6736407 | M |
| 1 | Barcelona | 5629629 | B |
| 2 | Valencia | 2577506 | V |
| 3 | Sevilla | 1958922 | SE |
| 4 | Alicante | 1897848 | A |
| 5 | Málaga | 1700752 | MA |



ÍNDICE



SERIE

Pandas

Dataframes

CARACTERÍSTICAS



OBSERVACIONES



| | PROVINCIA | POBLACIÓN | CÓDIGO |
|---|-----------|-----------|--------|
| 0 | Madrid | 6736407 | M |
| 1 | Barcelona | 5629629 | B |
| 2 | Valencia | 2577506 | V |
| 3 | Sevilla | 1958922 | SE |
| 4 | Alicante | 1897848 | A |
| 5 | Málaga | 1700752 | MA |

Pandas

Terminología

| Excel | pandas |
|-----------------|--------------------|
| Hoja de cálculo | Dataframe |
| Columna | Serie, columna |
| Número de fila | Índice |
| Fila | Fila (observación) |
| Celda vacía | NaN (Not a Number) |

Pandas

Crear un Dataframe

Básicamente podemos crear un Dataframe de tres formas diferentes:

- A partir de arrays unidimensionales
- A partir de un diccionario
- A partir de un fichero de datos, por ejemplo un fichero CSV

Pandas

Crear un Dataframe a partir de arrays

Un data frame está formado por filas y columnas (Series), que son arrays unidimensionales.

Por tanto, podemos crear un Dataframe especificando las filas que lo componen, por ejemplo, en forma de lista anidada.:

Ej.:

```
data = [[1,4],[2,5],[3,6]]
```

| | col1 | col2 |
|---|------|------|
| 0 | 1 | 4 |
| 1 | 2 | 5 |
| 2 | 3 | 6 |

Pandas

Crear un Dataframe a partir de un diccionario

Otra forma de crear un Dataframe es a partir de un diccionario:

```
diccionario = { 'clave1' : valor1 , 'clave2' : valor2 }
```

Ej.:

```
diccionario = { 'col1' : [1, 2, 3] , 'col2' : [4,5,6] }
```

En este caso, cada elemento del diccionario vendría a representar una columna (Serie) del Dataframe.

| | col1 | col2 |
|---|------|------|
| 0 | 1 | 4 |
| 1 | 2 | 5 |
| 2 | 3 | 6 |

Pandas

Crear un Dataframe a partir de un fichero

Por último, es posible crear un Dataframe a partir de un fichero CSV.

Los archivos CSV pueden ser creados y abiertos desde aplicaciones de hoja de cálculo como Excel.

Generalmente, este tipo de archivos que contienen datos se conocen como Data Sets.

Para construir un Dataframe a partir de un fichero CSV simplemente tenemos que abrir el fichero CSV mediante pandas:

```
pd.read_csv('fichero.csv')
```

| | col1 | col2 |
|---|------|------|
| 0 | 1 | 4 |
| 1 | 2 | 5 |
| 2 | 3 | 6 |

Ejercicio individual

Construcción de un Dataframe

Consulta la siguiente URL:

https://es.wikipedia.org/wiki/Anexo:Comunidades_y_ciudades_aut%C3%B3nomas_de_Espa%C3%B1a

Crea un Jupyter Notebook que construya un Dataframe con las primeras 5 comunidades de la tabla mediante los tres métodos que hemos visto..

El Dataframe creado debe contener las siguientes Series o columnas.

- Índice numérico autogenerado
- Nombre de la comunidad
- Población

Intenta crear el Dataframe completo descargando la información necesaria del siguiente sitio web:

<https://www.ine.es/jaxiT3/Tabla.htm?t=2853&L=0>

¿Qué dificultades has encontrado?

Ejercicio individual



Ejercicio individual



PYTHON

Plataformas online







red.es



UNIÓN EUROPEA

"El FSE invierte en tu futuro"

Fondo Social Europeo

