



UNIVERSIDAD
DE SANTIAGO
DE CHILE

DIPLOMADO EN

➤ **PRODUCT MANAGER**

ÁREA DE MARKETING



MÓDULO II:

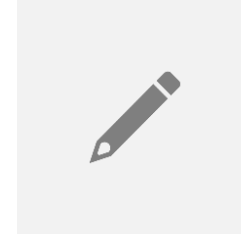
DATA SCIENCE PARA
EL PRODUCT
MANAGER

**FACULTAD DE
ADMINISTRACIÓN Y ECONOMÍA**
DEPARTAMENTO DE ADMINISTRACIÓN



Agenda

Sesión #2



Introducción a Python

-
1. **Stack** herramientas de análisis
 2. **Python**
 3. **¿Por qué Python?**
 4. **Librerías**
 5. **Github**

Recordatorio primera clase



Stack herramientas de análisis

All-in-one

BI

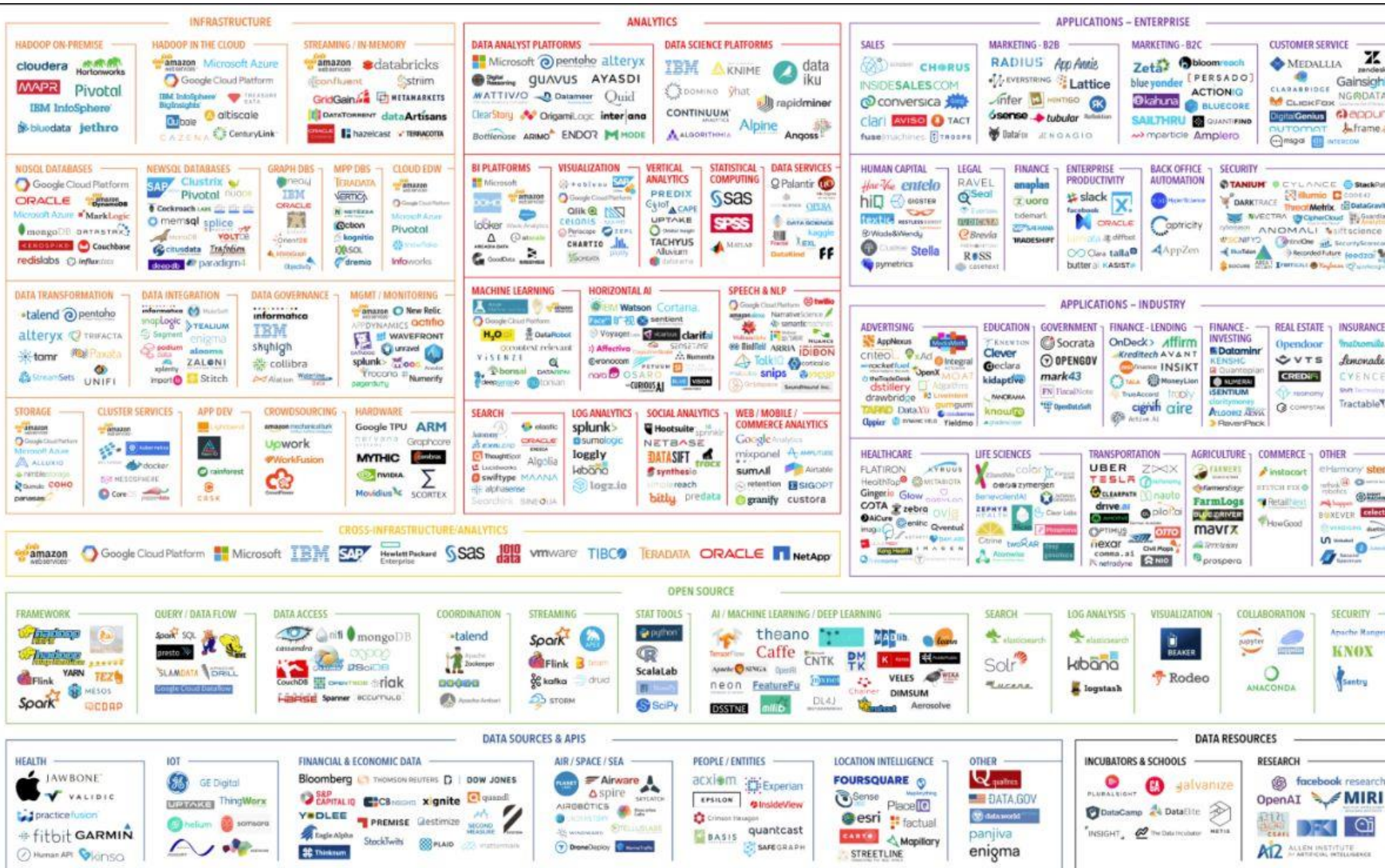
Open source



Stack herramientas de análisis

PRODUCT MANAGER

ÁREA DE MARKETING



Stack herramientas de análisis



All-in-one



BI

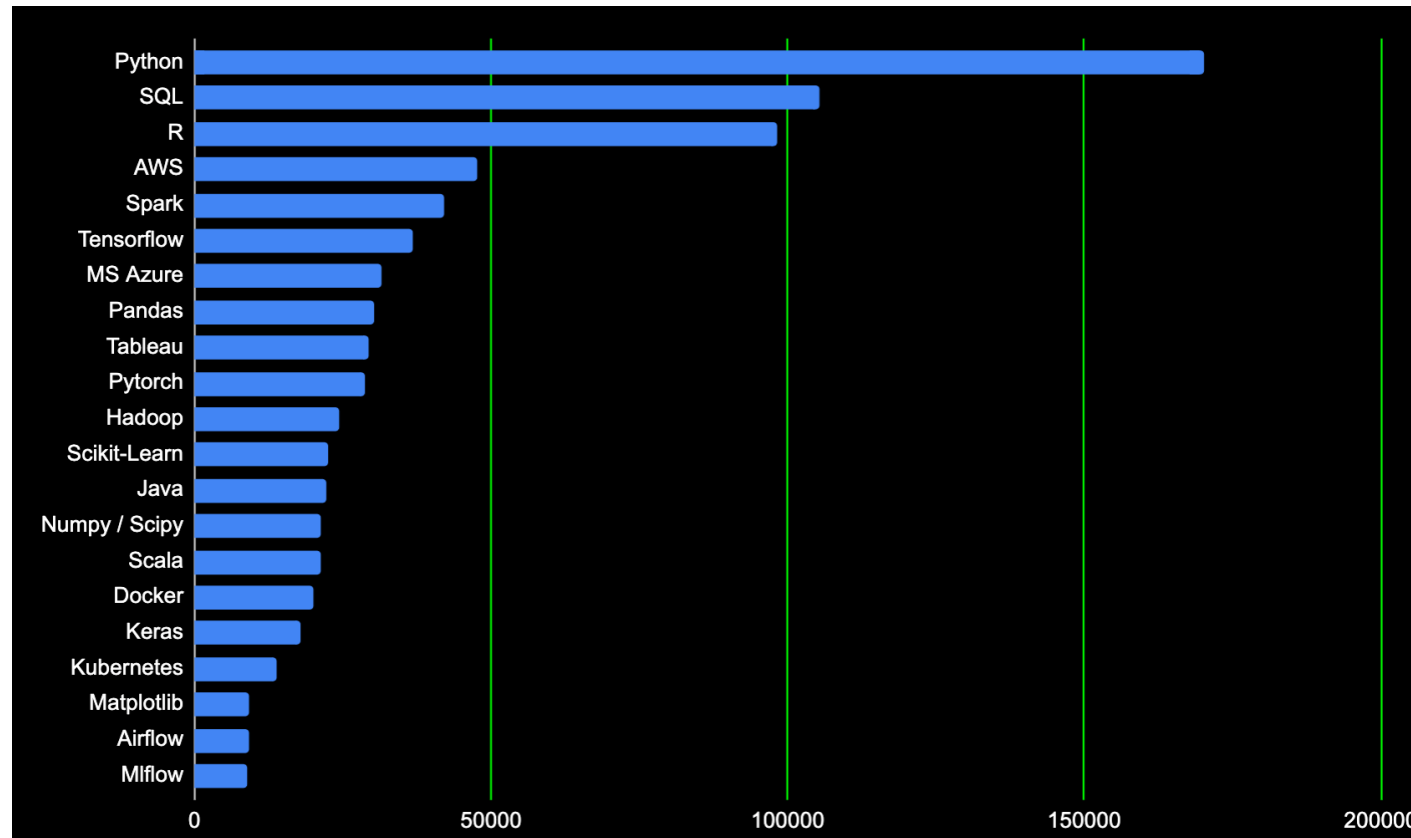


SAP Analytics Cloud

Open source



Ranking Herramientas más solicitadas para un Data Scientist en 2023



Python Posibilidades

Estadísticas

Permite estimar modelos y hacer pruebas estadísticas

Visualización

Posibilita la generación de gráficos a partir de datos contenidos en listas o arreglos

Data wrangling

Facilita vectores y matrices con distintas funciones matemáticas de alto nivel, lo que facilita la manipulación de datos

Data scraping

Entrega las herramientas para hacer scraping y crawling en distintas páginas web

Machine learning y analítica avanzada

Disponibiliza algoritmos básicos para análisis de clasificación, regresión y clustering, además de algoritmos avanzados de deep learning y otros

DIPLOMADO EN

 **PRODUCT
MANAGER**

ÁREA DE MARKETING

Python

¿Qué es y cómo funciona?



- Es un **lenguaje** de programación
- Opera a través de **implementaciones** y **distribuciones** (programas)
- Se puede manipular a través de distintas **interfaces/entornos**
- Provee de **librerías** específicas que permiten lograr distintos objetivos

Python

¿Qué es y cómo funciona?



- Es un **lenguaje** de programación
- Opera a través de **implementaciones** y **distribuciones** (programas)
- Se puede manipular a través de distintas **interfaces/entornos**
- Provee de **librerías** específicas que permiten lograr distintos objetivos



Python

Algunas librerías

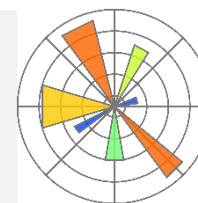
Estadísticas

Permite estimar modelos y hacer pruebas estadísticas



Visualización

Posibilita la generación de gráficos a partir de datos contenidos en listas o arreglos (independiente de limitantes de un programa!)



Data wrangling

Facilita vectores y matrices con distintas funciones matemáticas de alto nivel, lo que facilita la manipulación de datos



Data scraping

Entrega las herramientas para hacer scraping y crawling en distintas páginas web



Machine learning y analítica avanzada

Disponibiliza algoritmos básicos para análisis de clasificación, regresión y clustering, además de algoritmos avanzados de deep learning y otros



DIPLOMADO EN

**PRODUCT
MANAGER**

ÁREA DE MARKETING

Python

Algunas librerías

DIPLOMADO EN

**PRODUCT
MANAGER**

ÁREA DE MARKETING

Data Wrangling

with pandas Cheat Sheet

<http://pandas.pydata.org>

[Pandas API Reference](#) [Pandas User Guide](#)

Creating DataFrames

	a	b	c
1	4	7	10
2	5	8	11
3	6	9	12

```
df = pd.DataFrame(
    {"a": [4, 5, 6],
     "b": [7, 8, 9],
     "c": [10, 11, 12]},
    index = [1, 2, 3])
```

Specify values for each column.

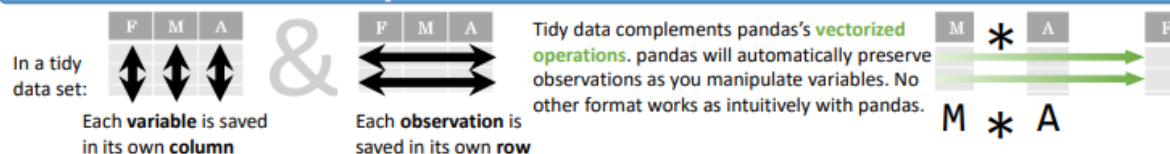
```
df = pd.DataFrame(
    [[4, 7, 10],
     [5, 8, 11],
     [6, 9, 12]],
    index=[1, 2, 3],
    columns=['a', 'b', 'c'])
```

Specify values for each row.

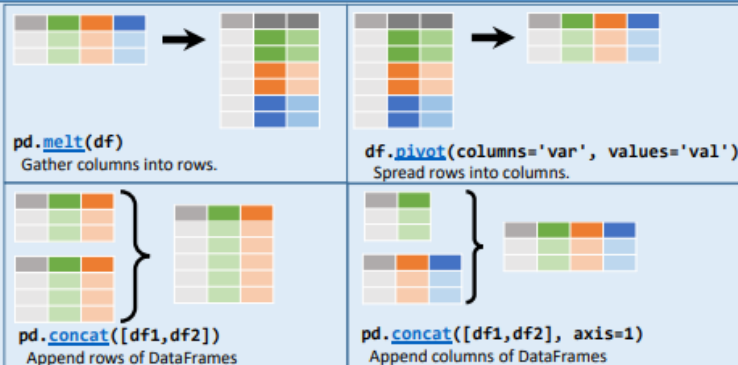
	a	b	c
1	4	7	10
2	5	8	11
3	6	9	12

```
df = pd.DataFrame(
    {"a": [4, 5, 6],
     "b": [7, 8, 9],
     "c": [10, 11, 12]},
    index = pd.MultiIndex.from_tuples(
        [('d', 1), ('d', 2)]
```

Tidy Data – A foundation for wrangling in pandas



Reshaping Data – Change layout, sorting, reindexing, renaming



```
df.sort_values('mpg')
Order rows by values of a column (low to high).

df.sort_values('mpg', ascending=False)
Order rows by values of a column (high to low).

df.rename(columns = {'y': 'year'})
Rename the columns of a DataFrame

df.sort_index()
Sort the Index of a DataFrame

df.reset_index()
Reset index of DataFrame to row numbers, moving index to columns.

df.drop(columns=['Length', 'Height'])
Drop columns from DataFrame
```

Subset Observations - rows

```
df[df.Length > 7]
Extract rows that meet logical criteria.

df.drop_duplicates()
Remove duplicate rows (only considers columns).

df.sample(frac=0.5)
Randomly select fraction of rows.

df.sample(n=10)
Randomly select n rows.

df.nlargest(n, 'value')
Select and order top n entries.
```

Subset Variables - columns

```
df[['width', 'length', 'species']]
Select multiple columns with specific names.

df['width'] or df.width
Select single column with specific name.

df.filter(regex='regex')
Select columns whose name matches regular expression regex.
```

Subsets - rows and columns

```
Use df.loc[] and df.iloc[] to select only rows, only columns or both.
Use df.at[] and df.iat[] to access a single value by row and column.
First index selects rows, second index columns.

df.iloc[10:20]
Select rows 10-20.

df.iloc[:, [1, 2, 5]]
Select columns in positions 1, 2 and 5 (first column is 0).

df.loc[:, 'x2': 'x4']
```

Using query

Python

Funciones básicas

DIPLOMADO EN

 **PRODUCT
MANAGER**

ÁREA DE MARKETING

<code>abs()</code>	<code>divmod()</code>	<code>input()</code>	<code>open()</code>	<code>staticmethod()</code>
<code>all()</code>	<code>enumerate()</code>	<code>int()</code>	<code>ord()</code>	<code>str()</code>
<code>any()</code>	<code>eval()</code>	<code>isinstance()</code>	<code>pow()</code>	<code>sum()</code>
<code>basestring()</code>	<code>execfile()</code>	<code>issubclass()</code>	<code>print()</code>	<code>super()</code>
<code>bin()</code>	<code>file()</code>	<code>iter()</code>	<code>property()</code>	<code>tuple()</code>
<code>bool()</code>	<code>filter()</code>	<code>len()</code>	<code>range()</code>	<code>type()</code>
<code>bytearray()</code>	<code>float()</code>	<code>list()</code>	<code>raw_input()</code>	<code>unichr()</code>
<code>callable()</code>	<code>format()</code>	<code>locals()</code>	<code>reduce()</code>	<code>unicode()</code>
<code>chr()</code>	<code>frozenset()</code>	<code>long()</code>	<code>reload()</code>	<code>vars()</code>
<code>classmethod()</code>	<code>getattr()</code>	<code>map()</code>	<code>repr()</code>	<code>xrange()</code>
<code>cmp()</code>	<code>globals()</code>	<code>max()</code>	<code>reversed()</code>	<code>zip()</code>
<code>compile()</code>	<code>hasattr()</code>	<code>memoryview()</code>	<code>round()</code>	<code>__import__()</code>
<code>complex()</code>	<code>hash()</code>	<code>min()</code>	<code>set()</code>	
<code>delattr()</code>	<code>help()</code>	<code>next()</code>	<code>setattr()</code>	
<code>dict()</code>	<code>hex()</code>	<code>object()</code>	<code>slice()</code>	
<code>dir()</code>	<code>id()</code>	<code>oct()</code>	<code>sorted()</code>	

Python

Algunos cursos recomendados



- Udemy - Python for Data Science and Machine Learning Bootcamp
- Coursera - Crash Course on Python (Google)
- Coursera - Programming for Everybody (Getting Started with Python) (University of Michigan)

Veámoslo en acción...



Gracias

👤 Cristian León Meza

📞 +569 5696 2718

✉ cristian.leon.me@Usach.cl

🌐 <https://www.linkedin.com/in/cristian-león>