

Introducción a Python

(para estudiantes de psicología)

Juan C. Correa, Ph.D.
<https://correajc.com>

j.correa.n@gmail.com

Tercera Jornada de Metodología Cuantitativa en Psicología



Objetivo de esta Charla

En psicología, no es muy frecuente la enseñanza de lenguajes de programación como Python o R para complementar cursos de contenido estadístico o metodológico. Con esta charla, se pretende ilustrar algunos pasos estratégicos para facilitar la enseñanza de Python para estudiantes de psicología.



- 1 Nota de Precaución
 - 2 Python: Primeros Pasos
 - 3 Python con Anaconda
 - Jupyter Notebook
 - Spyder
 - 4 Recursos Bibliográficos
 - 5 Regresión Simple
 - 6 Regresión Múltiple en Python
 - 7 Sugerencias
- Referencias



Python es muy versátil y útil a la psicología y disciplinas afines (e.g., [Psychopy](#), [Expyriment](#), [OpenSesame](#), [Pygaze](#), [PsyUtils](#), [NIPY](#), [PyMC3](#), [Kabuki](#)). Es imposible brindar una aproximación exhaustiva a esta versatilidad en una sola charla.

Esta charla presenta una aproximación a Python desde el punto de vista de la estadística que es más frecuentemente enseñada en pregrados de psicología para el análisis de datos.



Python: Primeros Pasos





<https://youtu.be/OmmklYlRGzo>

Decidir cuál editor de códigos usar.



Entender la diferencia conceptual entre librerías

Librerías Básicas
(Ruta Larga)



Librerías Especializadas
(Ruta Corta)



<https://www.marsja.se/best-python-libraries-psychology/>



Ejemplo: Definición de función Alfa de Cronbach con librerías básicas

```
In [2]: import pandas as pd  
import numpy as np
```

```
In [3]: # Primero, es necesario transformar la base de datos a una matriz de correlación  
# El número de variables es igual al número de columnas en la base de datos  
# Para esto, recorreremos las columnas y agregaremos  
# cada correlación relevante a una matriz llamada "r_s".  
# Luego, calcularemos la media de "r_s"
```

```
In [4]: def cronbach_alpha(df):  
  
    df_corr = df.corr()  
  
    N = df.shape[1]  
  
    rs = np.array([])  
    for i, col in enumerate(df_corr.columns):  
        sum_ = df_corr[col][i+1:].values  
        rs = np.append(sum_, rs)  
    mean_r = np.mean(rs)  
  
    cronbach_alpha = (N * mean_r) / (1 + (N - 1) * mean_r)  
    return cronbach_alpha
```

[https://towardsdatascience.com/
cronbachs-alpha-theory-and-application-in-python-d2915dd63586](https://towardsdatascience.com/cronbachs-alpha-theory-and-application-in-python-d2915dd63586)



Ejemplo: Aplicación de Alfa de Cronbach con librerías especializada (pingouin)

```
>>> import pingouin as pg
>>> data = pg.read_dataset('cronbach_wide_missing')
>>> # In R: psych:alpha(data, use="pairwise")
>>> pg.cronbach_alpha(data=data)
(0.732660835214447, array([0.435, 0.909]))
```

[https://pingouin-stats.org/generated/
pingouin.cronbach_alpha.html#pingouin.cronbach_alpha](https://pingouin-stats.org/generated/pingouin.cronbach_alpha.html#pingouin.cronbach_alpha)



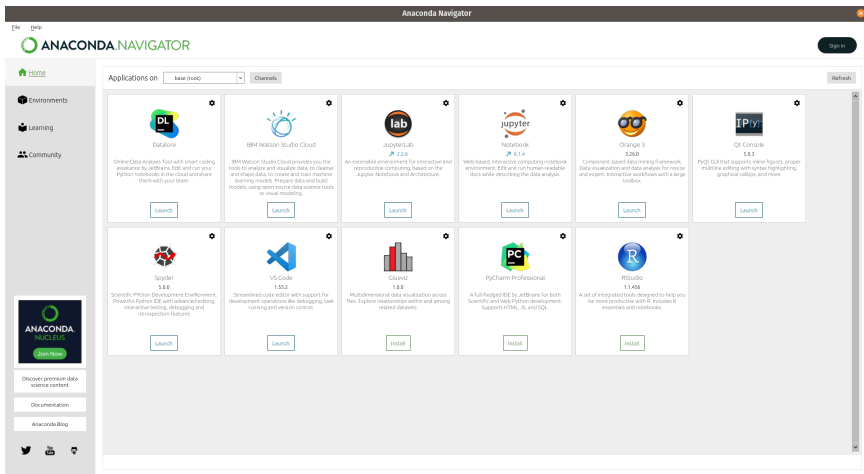
Python con Anaconda





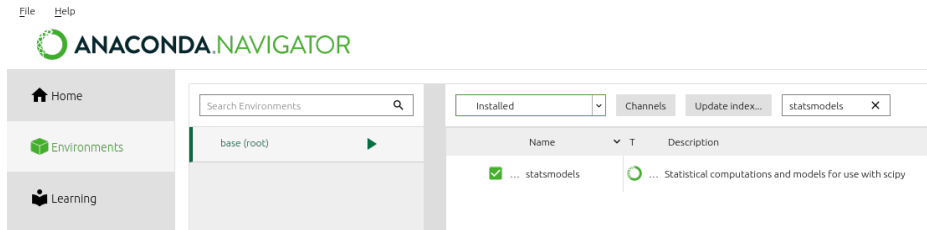
Anaconda-Navigator es un sistema de gestión de paquetes, basado en conda, y permite desarrollar proyectos basados en Python o R, además de incluir otras herramientas útiles para ciencia de datos.





Interfaz gráfica de Anaconda





Para instalar nuevas librerías dentro de Anaconda, basta con hacer clic en “Environments” y luego seleccionar el menú **Not Installed** para escribir a mano derecha el nombre específico de la librería que se quiere instalar.



Python con Anaconda Jupyter Notebook



Select items to perform actions on them.

Upload

New ▾


☐ 0 ▾

/ Documents

Name ▾

vor e



..

☐

Folder beamer

☐

Folder Beamer

☐

Folder Books

Notebook:

Python 3

Other:

Text File

Folder

Terminal

vor einem Tag

Clic en **New** y luego en **Python 3**





In []:

De manera predeterminada, los archivos creados en jupyter notebook tienen el nombre **Untitled**. Debemos cambiarle el nombre haciendo clic sobre Untitled.



Rename Notebook

×

Enter a new notebook name:

RegMult

Cancel

Rename

Acá vamos a poner el nombre a nuestro primer Jupyter Notebook como **RegMult**.



In [1]: `import pandas as pd`

In [2]: `datos = pd.read_csv("/home/juan/cars.csv")`

In [3]: `datos.head()`

Out[3]:

	Car	Model	Volume	Weight	CO2
0	Toyoty	Aygo	1000	790	99
1	Mitsubishi	Space Star	1200	1160	95
2	Skoda	Citigo	1000	929	95
3	Fiat	500	900	865	90
4	Mini	Cooper	1500	1140	105

In [4]: `X = datos[['Weight', 'Volume']]`
`y = datos['CO2']`

In [5]: `import statsmodels.api as sm`

In [6]: `model = sm.OLS(y, X).fit()`

In [7]: `model.summary()`

Out[7]:

OLS Regression Results

Dep. Variable:	CO2	R-squared (uncentered):	0.977
Model:	OLS	Adj. R-squared (uncentered):	0.975
Method:	Least Squares	F-statistic:	714.9
Date:	Mon, 03 May 2021	Prob (F-statistic):	1.67e-28



File Edit View Insert Cell Kernel Widgets Help

New Notebook ▶
Open...

Make a Copy...
Save as...
Rename...
Save and Checkpoint Ctrl-S
Revert to Checkpoint ▶

Print Preview...
Download as ▶
Trusted Notebook
Close and Halt

Run ■ ↻ ▶▶ Code ▼

Model	Volume	Weight	CO2
Aygo	1000	790	99
Space Star	1200	1160	95
Citigo	1000	929	95
500	900	865	90
Cooper	1500	1140	105

models.api as sm

AsciiDoc (.asciidoc)
HTML (.html)
LaTeX (.tex)
Markdown (.md)
Notebook (.ipynb)
PDF via LaTeX (.pdf)
Python (.py)
PDF via pypeteer (.html)

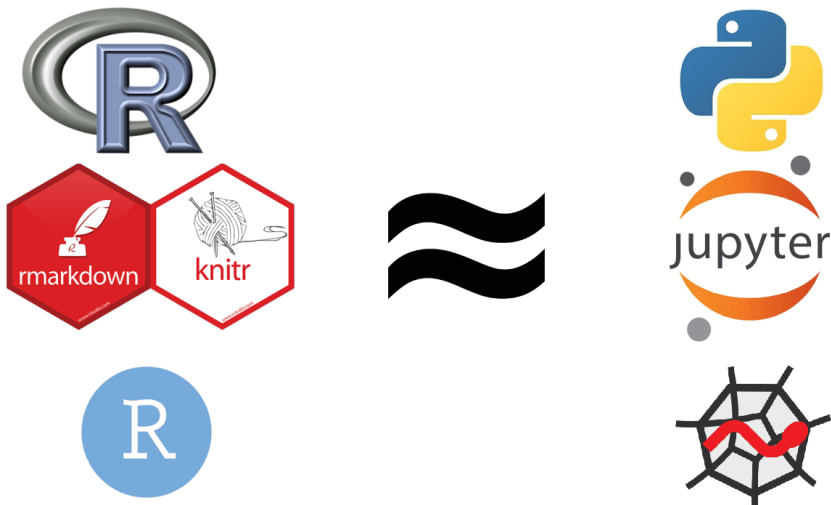
results.sum
sm.graphics
plt.show()

TypeError
<ipython-in
----> 1 fig
2 res

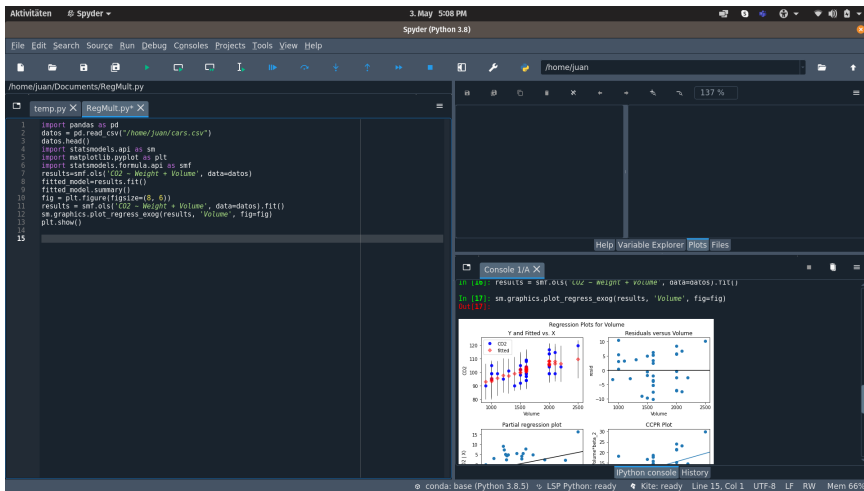


Python con Spyder





La apariencia de Spyder es muy similar a la de RStudio



Trabajar en Spyder es como elaborar un script en R (archivo .R)



Hay varias semejanzas entre Python y R, desde el punto de vista de las instrucciones (sintaxis) que debemos escribir para que el software realice lo que deseemos.

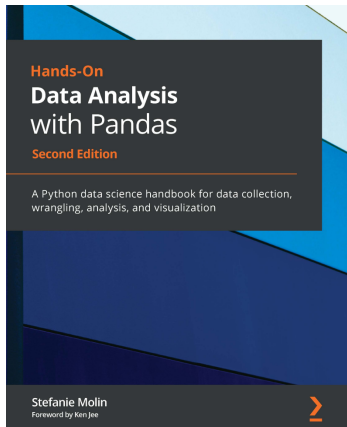
```
import pandas as pd  
library(readr)
```

```
datos = pd.read_csv(/home/juan/cars.csv")  
readr::datos <- read_csv(/home/juan/cars.csv")
```



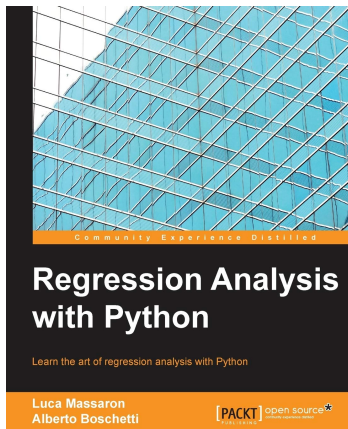
Recursos Bibliográficos





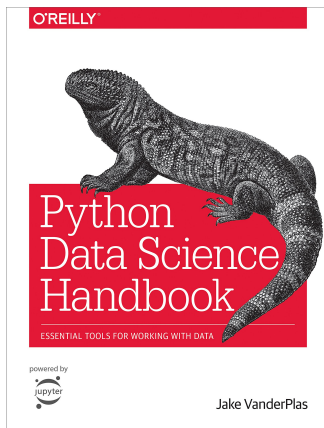
El libro de Molin (2021) es un buen recurso para aprender fundamentos de análisis de datos. Pero su aproximación dista mucho del tipo de estadística aplicada en psicología.





El libro de Massaron y Boschetti (2016) es un buen recurso para aprender a implementar análisis de regresiones. Esta aproximación es más orientada a machine learning, pero no hace una cobertura adecuada al problema del chequeo de supuestos en regresión.





El libro de VanderPlas (2017) aborda data manipulation, data visualization, y machine learning. Su cobertura deja por fuera un montón de estadística estándar en psicología u otras ciencias sociales (e.g., psicometría, modelos de ecuaciones estructurales, redes).





El paper de Seabold y Perktold (2010) es probablemente, a la fecha, el recurso más orientado a estadística que puede encontrarse en Python.

Su documentación online está disponible en <https://www.statsmodels.org/>



Regresión Simple



Regresión con Mínimos Cuadrados Ordinarios

```
import numpy as np
import statsmodels.api as sm
spector_data = sm.datasets.spector.load(as_pandas=False)
spector_data.exog = sm.add_constant(spector_data.exog, prepend=False)
mod = sm.OLS(spector_data.endog, spector_data.exog)
res = mod.fit()
print(res.summary())
```



Regresión con Cuadrados Ordinarios Ponderados

```
import numpy as np
import statsmodels.api as sm
spector_data = sm.datasets.spector.load(as_pandas=False)
spector_data.exog = sm.add_constant(spector_data.exog, prepend=False)
mod = sm.WLS(spector_data.endog, spector_data.exog)
res = mod.fit()
print(res.summary())
```




```
In [11]: print(res.summary())
```

WLS Regression Results

```
=====
Dep. Variable:          y      R-squared:          0.416
Model:                  WLS    Adj. R-squared:       0.353
Method:                 Least Squares    F-statistic:      6.646
Date:                  Mon, 03 May 2021    Prob (F-statistic): 0.00157
Time:                  19:15:39    Log-Likelihood:    -12.978
No. Observations:      32    AIC:              33.96
Df Residuals:          28    BIC:              39.82
Df Model:               3
Covariance Type:        nonrobust
=====
```

	coef	std err	t	P> t	[0.025	0.975]
x1	0.4639	0.162	2.864	0.008	0.132	0.796
x2	0.0105	0.019	0.539	0.594	-0.029	0.050
x3	0.3786	0.139	2.720	0.011	0.093	0.664
const	-1.4980	0.524	-2.859	0.008	-2.571	-0.425

```
=====
Omnibus:                0.176    Durbin-Watson:          2.346
Prob(Omnibus):           0.916    Jarque-Bera (JB):        0.167
Skew:                    0.141    Prob(JB):                0.920
Kurtosis:                2.786    Cond. No.                 176.
=====
```



Regresión Múltiple



```
import pandas as pd
datos = pd.read_csv(/home/juan/cars.csv)
import statsmodels.api as sm
import matplotlib.pyplot as plt
import statsmodels.formula.api as smf
results=smf.ols('CO2 ~ Weight + Volume', data=datos)
fitted_model=results.fit()
fitted_model.summary()
```



OLS Regression Results

```

=====
Dep. Variable:          C02      R-squared:                0.377
Model:                  OLS      Adj. R-squared:         0.339
Method:                 Least Squares      F-statistic:         9.966
Date:                   Mon, 03 May 2021    Prob (F-statistic):   0.000411
Time:                   19:19:18           Log-Likelihood:      -114.39
No. Observations:      36             AIC:                234.8
Df Residuals:          33             BIC:                239.5
Df Model:               2
Covariance Type:       nonrobust
=====

```

```

=====
               coef      std err          t      P>|t|      [0.025      0.975]
-----
Intercept      79.6947      5.564      14.322      0.000      68.374      91.016
Weight          0.0076      0.006       1.173      0.249      -0.006      0.021
Volume          0.0078      0.004       1.948      0.060      -0.000      0.016
=====

```

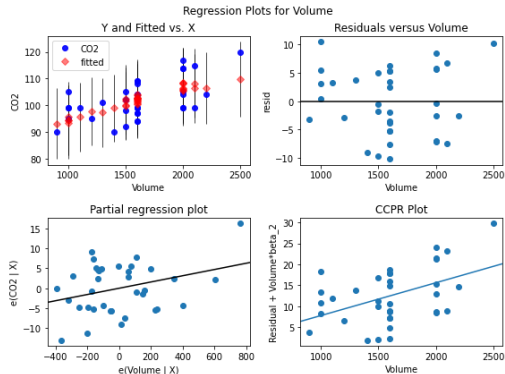
```

=====
Omnibus:          4.957      Durbin-Watson:         0.944
Prob(Omnibus):    0.084      Jarque-Bera (JB):      1.836
Skew:             -0.025     Prob(JB):              0.399
Kurtosis:         1.895     Cond. No.              1.16e+04
=====

```



```
fig = plt.figure(figsize=(8, 6))
results = smf.ols('CO2 ~ Weight + Volume', data=datos).fit()
sm.graphics.plot_regress_exog(results, 'Volume', fig=fig)
plt.show()
```



Sugerencias



- Aprenda intentando reproducir ejemplos concretos ya disponibles.
- La instalación de librerías por fuera de Anaconda puede ser algo problemática. Persevere con la solución.
- Pandas, Matplotlib, y statsmodels ya ofrecen un conjunto básico de funciones que le permitirán analizar datos usando técnicas estadísticas de uso común en psicología.
- Evite los materiales dedicados a Machine Learning sin tener unas sólidas bases en estadística.



- Massaron, L., y Boschetti, A. (2016). *Regression Analysis with Python*. New York, USA: Pakt Publisher.
- Molin, S. (2021). *Data Analysis with Pandas* (2nd ed.). New York, USA: Pakt Publisher.
- Seabold, S., y Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with python. En *Proceedings of the 9th python in science conference* (Vol. 57, p. 61).
- VanderPlas, J. (2017). *Python Data Science Handbook Essential Tools for Working with Data*. New York, USA: O'Reily.

