

## Introducción

En el presente trabajo se denotará las ventajas, desventajas, similitudes, y posibles problemas que se dio dando en el tramo de la codificación con los diferentes lenguajes seleccionados, en este caso R, Python, Ruby, Matlab y Lisp. Hablaremos sobre la experiencia obtenida en cada lenguaje y como esto está relacionado según el criterio de evaluación de lenguajes según Sebesta.

## Contenido

**Tema 2:** *Realice un documento en donde describa la experiencia con los lenguajes utilizados en este trabajo.*

1. *Describe el lenguaje seleccionado y el motivo de su elección. Describe su experiencia en cuanto a las implementaciones realizadas (con y sin el uso de librerías). Realice comentarios generales sobre las codificaciones. Analice las ventajas y desventajas de cada lenguaje utilizado.*

Primeramente, seleccioné MatLab como un tercer lenguaje, elegí este lenguaje que es muy utilizado para datascience en general y su fuerte motor para agrupamiento de datos, pero no pude lograr avanzar por la poca popularidad y documentación ya que en los foros no contemplaban muchas dudas para seguir avanzado y para un dataset del tamaño que manipulamos no era muy eficiente su manipulación, entonces me decidí por consiguiente por el lenguaje Ruby.

Ruby: La experiencia con Ruby fue similar a la de Python gracias a la librería Daru que se pudo simplificar todo, pero la manipulación de datos en Ruby fue mas compleja ya que la sintaxis en este lenguaje fue un poco más compleja, también se notó la poca legibilidad que tenía con el lenguaje. Una característica muy importante de este lenguaje es la multiplicidad ya que para hacer algún snippet simple podrías hacerlo de varias maneras.

Algunas desventajas en Ruby que podemos mencionar por ejemplo es la sobrecarga de los operadores, algunos operadores como ':' tenía diferente función según el contexto que se le daba, y otras como la legibilidad, no logre ver muchas dificultades ya que solo utilice las librerías y no pude corroborar posibles ineficiencias del lenguaje. La ventaja de Ruby sin duda es el alto tipado que tiene, además de una comunidad muy grande para ser un lenguaje no tan viejo como otros de la misma área.

El otro lenguaje que también se trato fue R que poco nada se le puede quitar merito a su capacidad de manipular el dataset, sus ventajas por mencionar algunas es la ortogonalidad y de fácil sintaxis, algunas desventajas de este lenguaje podría ser la alta curva de aprendizaje que tiene.

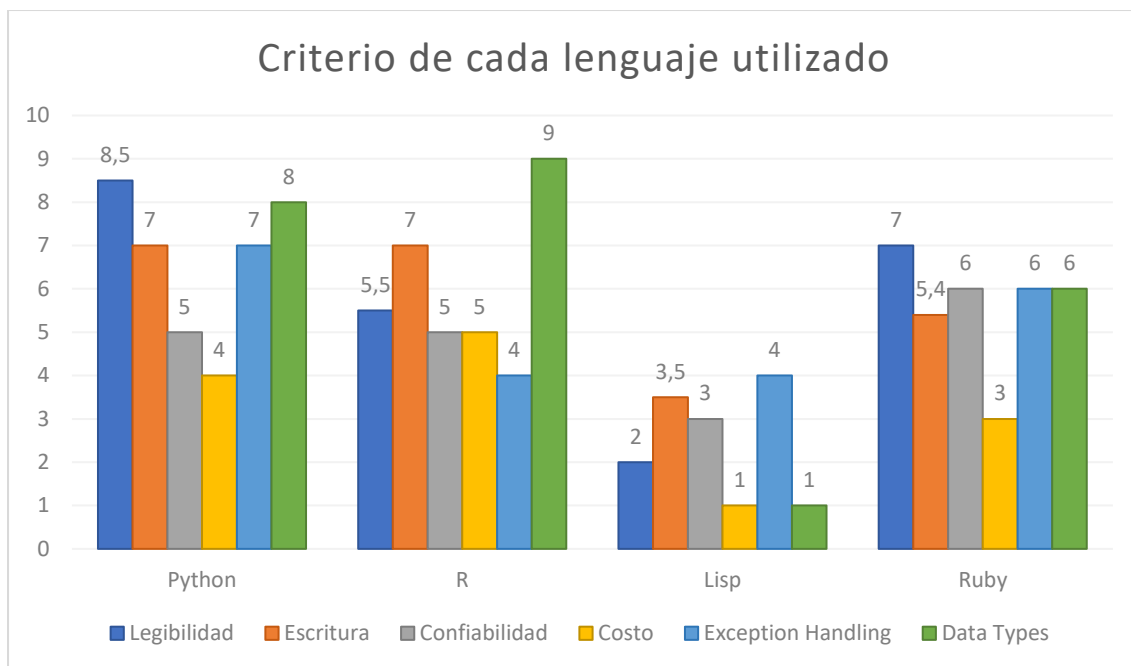
Y por último yo diría que lenguaje por excelencia para datascience, Python, para este lenguaje lo único que encontré fue ventajas (al menos para la curva de aprendizaje y manipulación de datos), lo malo de Python es la cantidad de memoria que utiliza, que fue mayor que por ejemplo comparar con R.

En general el uso de librerías en cada lenguaje utilizo ayudo bastante para la realización de el trabajo, la gran diferencia entre todos los lenguajes seleccionados es que R es un lenguaje que fue diseñado para este sector que estamos trabajando, junto a Python fueron muy superiores a otros lenguajes de programación en el mismo ámbito.

En Python hubo bastante diferencia frente al uso de librerías y al que no, fue también menos legible al no usar librerías, fue más difícil de leer escribir y depurar, para la obtención de correlaciones por ejemplo al utilizar método de Pearson manualmente fue mucho menos optimizado.

En Rasgos generales Ruby estuvo un poco por debajo de Python y R en todos los sentidos.

2. Realice una comparación de los lenguajes Python, R, LISP y el lenguaje seleccionado, para ello utilice los criterios de evaluación de lenguajes propuestos por Sebesta (2016) como la legibilidad, capacidad de escritura, confiabilidad y costo, y medirlo utilizando las características de simplicidad, tipo de datos, sintaxis, manejo de excepciones, expresividad, entre otros.



(\*) Como puntaje máximo tomamos el 10

(\*) El costo mientras mayor sea será mas eficiente.

Podemos apreciar que Python obtuvo un mayor puntaje acumulado en cuanto legibilidad, facilidad de escritura y manejo de excepciones.

En el lenguaje R lo que mejor se aprecio fue la facilidad de escritura y el bajo costo computacional que tenia

En tanto Ruby estuvo casi a la altura de Python, pero su popularidad y documentación eran menores a la de Python por lo que se concluyó esos puntajes.

Lisp lo considere un lenguaje abstracto, para lo que se utilizo se pudo observar que tenia un alto consumo computacional por las llamadas recursivas que esta tenia al ser un lenguaje funcional.

## Conclusión

Se pudo llegar a comprender las dificultades y las similitudes que tienen diferentes lenguajes de programación para el trato de un mismo ámbito, el tiempo de ejecución que estas emplean según el método utilizado. Se comprendió lo importante que es la legibilidad y la escritura de un lenguaje para que este tenga una curva de aprendizaje menor.

Se aplico criterios de evaluación de los lenguajes según Sebesta y se pudo observar que Python fue un lenguaje superior para la situación dada.

En R se denoto su motor para la manipulación de datasets, superior a Ruby y Python, pero esto no se vio muy bien reflejada por su poca legibilidad.

Con este trabajo practico se obtuvo experiencias nuevas en los cuatro lenguajes utilizados y sus grandes diferencias/similitudes entre las mismas.

Teniendo una base general de lenguajes conocidos se pudo adaptar fácilmente en los lenguajes seleccionados para la realización del trabajo practico.

## Bibliografía

- Capítulo 2 Sebesta (2016)
- [https://pandas.pydata.org/docs/reference/api/pandas.Series.to\\_frame.html](https://pandas.pydata.org/docs/reference/api/pandas.Series.to_frame.html)
- [https://pandas.pydata.org/docs/reference/api/pandas.Series.to\\_clipboard.html](https://pandas.pydata.org/docs/reference/api/pandas.Series.to_clipboard.html)
- [https://pandas.pydata.org/docs/reference/offset\\_frequency.html](https://pandas.pydata.org/docs/reference/offset_frequency.html)
- <https://pandas.pydata.org/docs/reference/arrays.html>
- <https://intellipaat.com/community/17913/import-multiple-csv-files-into-pandas-and-concatenate-into-one-dataframe>
- <https://www.emcsg.com/marketdata/priceinformation>
- <https://www.geeksforgeeks.org/get-the-specified-row-value-of-a-given-pandas-dataframe/>
- <https://www.datacamp.com/community/tutorials/sorting-in-r>
- <https://www.geeksforgeeks.org/change-column-name-of-a-given-dataframe-in-r/>
- <https://www.statmethods.net/management/subset.html>
- <https://statisticsglobe.com/diff-function-in-r>
- <https://discuss.analyticsvidhya.com/t/how-to-add-a-column-to-a-data-frame-in-r/3278>
- <https://stackoverflow.com/questions/20689650/how-to-append-rows-to-an-r-data-frame>
- <https://www.w3resource.com/pandas/dataframe/dataframe-transpose.php>
- [https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.sort\\_values.html](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.sort_values.html)
- <https://medium.com/dunder-data/select-a-single-column-of-a-pandas-dataframe-with-the-brackets-and-not-dot-notation-a5ec981cbae6>
- [How do I select a subset of a DataFrame? — pandas 1.2.4 documentation](#)
- [Indexing and selecting data — pandas 1.2.4 documentation](#)
- [python - Compute row average in pandas - Stack Overflow](#)
- [python - Divide multiple columns by another column in pandas - Stack Overflow](#)
- [pandas - Python: Divide each row of a DataFrame by another DataFrame vector - Stack Overflow](#)
- [pandas: Rename columns / index names \(labels\) of DataFrame | note.nkmmk.me](#)
- [How to Sort Pandas DataFrame \(with examples\) - Data to Fish](#)
- [Pandas Sort: Your Guide to Sorting Data in Python – Real Python](#)
- <https://www.rubyguides.com/2017/10/ruby-transpose-method/>
- <https://stackoverflow.com/questions/12911482/how-to-select-the-row-by-using-checkbox-in-ruby-on-rails/12912653>
- <https://stackoverflow.com/questions/12371930/ruby-watir-selecting-a-specific-row>
- <https://ruby-doc.org/stdlib-3.0.0/libdoc/csv/rdoc/CSV.html>