# Actividad


- Regina Rodríguez Sánchez
- Emilio De Gyves García
- Emilio Rizo De la Mora


# Entregar

Archivo PDF de la actividad y la liga de la actividad en su repositorio.

# Nota:

Las tareas 1 a la 5 se califican como entregadas o no entregadas.

Al integrante que no participe en la actividad no se le tomará en cuenta para la calificación.

El límite para entregar las actividades es el viernes antes de las 23.59.

In [ ]:
```
# Si trabajemos en Google Colaboratory corremos las siguientes lineas de código
from google.colab import drive
drive.mount('/gdrive')
```

```
Mounted at /gdrive
```

In [ ]:
```
# No cambiamos a la carpeta donde tengamos el repositorio. Si es otra carpeta asegúrate de cambiar la ruta.
%cd /gdrive/MyDrive/SemanasTec/arte-analitica
```

```
/gdrive/MyDrive/SemanasTec/arte-analitica
```

# Insurance dataset

El dataset contiene información demográfica sobre los asegurado en una compañía de seguros:

- age: edad del asegurado principal
- sex: género del asegurado. female o male
- bmi: indice de masa corporal
- children: numero de niños que estan cubiertos con la poliza.
- smoke: si fuma el beneficiario. yes/no
- region: dónde vive el beneficiario. Estos datos son de Estados Unidos. Regiones disponibles: northeast, southeast, southwest, northwest
- charges: costo del seguro.

# Actividad

Carga el dataset **`data/insurance.csv`** y haz un análisis estadístico de las variables.

In [ ]:
```
# Carga las librerias necesarias
import numpy as np
import pandas as pd

# Importar los datos
credit = pd.read_csv('data/insurance.csv')
```

In [ ]:
```
# Carga los datos y muestra los primeros renglones
credit.head(5)
```

Out[ ]:

| | age | sex | bmi | children | smoker | region | charges |
|---|---|---|---|---|---|---|---|
| 0 | 19 | female | 27.900 | 0 | yes | southwest | 16884.92400 |
| 1 | 18 | male | 33.770 | 1 | no | southeast | 1725.55230 |
| 2 | 28 | male | 33.000 | 3 | no | southeast | 4449.46200 |
| 3 | 33 | male | 22.705 | 0 | no | northwest | 21984.47061 |
| 4 | 32 | male | 28.880 | 0 | no | northwest | 3866.85520 |

In [ ]:
```
# Crea una tabla resumen con los estadísticos generales de las variables numéricas
credit.describe()
```

Out[ ]:

| | age | bmi | children | charges |
|---|---|---|---|---|
| count | 1338.000000 | 1338.000000 | 1338.000000 | 1338.000000 |
| mean | 39.207025 | 30.663397 | 1.094918 | 13270.422265 |
| std | 14.049960 | 6.098187 | 1.205493 | 12110.011237 |
| min | 18.000000 | 15.960000 | 0.000000 | 1121.873900 |
| 25% | 27.000000 | 26.296250 | 0.000000 | 4740.287150 |
| 50% | 39.000000 | 30.400000 | 1.000000 | 9382.033000 |
| 75% | 51.000000 | 34.693750 | 2.000000 | 16639.912515 |
| max | 64.000000 | 53.130000 | 5.000000 | 63770.428010 |

In [ ]:
```
# ¿Cómo se correlacionan las variables numéricas entre sí?
# ¿Cuáles son las variables que tienen más correlación con los costos del seguro?
# R= la edad
credit.corr()
```

Out[ ]:

| | age | bmi | children | charges |
|---|---|---|---|---|
| age | 1.000000 | 0.109272 | 0.042469 | 0.299008 |
| bmi | 0.109272 | 1.000000 | 0.012759 | 0.198341 |
| children | 0.042469 | 0.012759 | 1.000000 | 0.067998 |
| charges | 0.299008 | 0.198341 | 0.067998 | 1.000000 |

In [ ]:
```
# ¿Cuántas asegurados son hombres y cuántas son mujeres?
# 676 hombres, 662 mujeres
credit['sex'].value_counts()
```

Out[ ]:
```
male    676
female    662
Name: sex, dtype: int64
```

In [10]:
```
# ¿Cuántos hombres y mujeres asegurados viven en cada región?
pd.crosstab(credit['region'],credit['sex'])
```

Out[10]:

| sex | female | male |
|---|---|---|
| region | | |
| northeast | 161 | 163 |
| northwest | 164 | 161 |
| southeast | 175 | 189 |
| southwest | 162 | 163 |

In [11]:
```
# En promedio, ¿quién paga más de cuota de seguro, los fumadores o no fumadores? Muéstralo con los datos
credit.groupby('smoker').mean()[['charges']]
```

Out[11]:

| | charges |
|---|---|
| smoker | |
| no | 8434.268298 |
| yes | 32050.231832 |

In [12]:
```
# De la pregunta anterior, quien tiene más variación (desviación estándar) en la cuota de seguro,
# ¿los fumadores o los no fumadores?
# R= los fumadores
credit.groupby('smoker').std()[['charges']]
```

Out[12]:

| | charges |
|---|---|
| smoker | |
| no | 5993.781819 |
| yes | 11541.547176 |

In [13]:
```
# ¿Cuáles son los cargos mínimos y máximos que las personas pagan dependiendo del género?
credit.groupby('sex').agg(['min', 'max'])['charges']
```

Out[13]:

| | min | max |
|---|---|---|
| sex | | |
| female | 1607.5101 | 63770.42801 |
| male | 1121.8739 | 62592.87309 |

# Guardar el resultado como pdf

- Escribe aquí abajo la liga de tu repositorio.
- (Haz doble clic en esta celda y copia la URL dentro del paréntesis)

Liga_al_repositorio_de_Regina

- Exporta el notebook a formato HTML.

In [17]:
```
!jupyter nbconvert --to HTML '/gdrive/MyDrive/SemanasTec/arte-analitica/4.2 - Actividad - Estadísticas de datos.ipynb'
```

```
[NbConvertApp] WARNING | pattern '/content/drive/MyDrive/SemanasTec/arte-analitica/4.2 - Actividad - Estadísticas de datos.ipynb' matched no files
This application is used to convert notebook files (*.ipynb)
        to various other formats.

        WARNING: THE COMMANDLINE INTERFACE MAY CHANGE IN FUTURE RELEASES.

Options
=======
The options below are convenience aliases to configurable class-options,
as listed in the "Equivalent to" description-line of the aliases.
To see all configurable class-options for some <cmd>, use:
    <cmd> --help-all

--debug
    set log level to logging.DEBUG (maximize logging output)
    Equivalent to: [--Application.log_level=10]
--show-config
    Show the application's configuration (human-readable format)
    Equivalent to: [--Application.show_config=True]
--show-config-json
    Show the application's configuration (json format)
    Equivalent to: [--Application.show_config_json=True]
--generate-config
    generate default config file
    Equivalent to: [--JupyterApp.generate_config=True]
-y
    Answer yes to any questions instead of prompting.
    Equivalent to: [--JupyterApp.answer_yes=True]
--execute
    Execute the notebook prior to export.
    Equivalent to: [--ExecutePreprocessor.enabled=True]
--allow-errors
    Continue notebook execution even if one of the cells throws an error and include the error message in the cell output (the default behaviour is to abort conversion). This flag is only relevant if '--execute' was specified, too.
    Equivalent to: [--ExecutePreprocessor.allow_errors=True]
--stdin
    read a single notebook file from stdin. Write the resulting notebook with default basename 'notebook.*'
    Equivalent to: [--NbConvertApp.from_stdin=True]
--stdout
    Write notebook output to stdout instead of files.
    Equivalent to: [--NbConvertApp.writer_class=StdoutWriter]
--inplace
    Run nbconvert in place, overwriting the existing notebook (only
            relevant when converting to notebook format)
    Equivalent to: [--NbConvertApp.use_output_suffix=False --NbConvertApp.export_format=notebook --FilesWriter.build_directory=]
--clear-output
    Clear output of current file and save in place,
            overwriting the existing notebook.
    Equivalent to: [--NbConvertApp.use_output_suffix=False --NbConvertApp.export_format=notebook --FilesWriter.build_directory= --ClearOutputPreprocessor.enabled=True]
--no-prompt
    Exclude input and output prompts from converted document.
    Equivalent to: [--TemplateExporter.exclude_input_prompt=True --TemplateExporter.exclude_output_prompt=True]
--no-input
    Exclude input cells and output prompts from converted document.
            This mode is ideal for generating code-free reports.
    Equivalent to: [--TemplateExporter.exclude_output_prompt=True --TemplateExporter.exclude_input=True]
--log-level=<Enum>
    Set the log level by value or name.
    Choices: any of [0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR', 'CRITICAL']
    Default: 30
    Equivalent to: [--Application.log_level]
--config=<Unicode>
    Full path of a config file.
    Default: ''
    Equivalent to: [--JupyterApp.config_file]
--to=<Unicode>
    The export format to be used, either one of the built-in formats
            ['asciidoc', 'custom', 'html', 'latex', 'markdown', 'notebook', 'pdf', 'python', 'rst', 'script', 'slides']
            or a dotted object name that represents the import path for an
            `Exporter` class
    Default: 'html'
    Equivalent to: [--NbConvertApp.export_format]
--template=<Unicode>
    Name of the template file to use
    Default: ''
    Equivalent to: [--TemplateExporter.template_file]
--writer=<DottedObjectName>
    Writer class used to write the
                                        results of the conversion
    Default: 'FilesWriter'
    Equivalent to: [--NbConvertApp.writer_class]
--post=<DottedOrNone>
    PostProcessor class used to write the
                                        results of the conversion
    Default: ''
    Equivalent to: [--NbConvertApp.postprocessor_class]
--output=<Unicode>
    overwrite base name use for output files.
                can only be used when converting one notebook at a time.
    Default: ''
    Equivalent to: [--NbConvertApp.output_base]
--output-dir=<Unicode>
    Directory to write output(s) to. Defaults
                                    to output to the directory of each notebook. To recover
                                    previous default behaviour (outputting to the current
                                    working directory) use . as the flag value.
    Default: ''
    Equivalent to: [--FilesWriter.build_directory]
--reveal-prefix=<Unicode>
    The URL prefix for reveal.js (version 3.x).
            This defaults to the reveal CDN, but can be any url pointing to a copy
            of reveal.js.
            For speaker notes to work, this must be a relative path to a local
            copy of reveal.js: e.g., "reveal.js".
            If a relative path is given, it must be a subdirectory of the
            current directory (from which the server is run).
            See the usage documentation
            (https://nbconvert.readthedocs.io/en/latest/usage.html#reveal-js-html-slideshow)
            for more details.
    Default: ''
    Equivalent to: [--SlidesExporter.reveal_url_prefix]
--nbformat=<Enum>
    The nbformat version to write.
            Use this to downgrade notebooks.
    Choices: any of [1, 2, 3, 4]
    Default: 4
    Equivalent to: [--NotebookExporter.nbformat_version]

Examples
--------

    The simplest way to use nbconvert is

            > jupyter nbconvert mynotebook.ipynb

            which will convert mynotebook.ipynb to the default format (probably HTML).

            You can specify the export format with `--to`.
            Options include ['asciidoc', 'custom', 'html', 'latex', 'markdown', 'notebook', 'pdf', 'python', 'rst', 'script', 'slides'].

            > jupyter nbconvert --to latex mynotebook.ipynb

            Both HTML and LaTeX support multiple output templates. LaTeX includes
            'base', 'article' and 'report'.  HTML includes 'basic' and 'full'. You
            can specify the flavor of the format used.

            > jupyter nbconvert --to html --template basic mynotebook.ipynb

            You can also pipe the output to stdout, rather than a file

            > jupyter nbconvert mynotebook.ipynb --stdout

            PDF is generated via latex

            > jupyter nbconvert mynotebook.ipynb --to pdf

            You can get (and serve) a Reveal.js-powered slideshow

            > jupyter nbconvert myslides.ipynb --to slides --post serve

            Multiple notebooks can be given at the command line in a couple of
            different ways:

            > jupyter nbconvert notebook*.ipynb
            > jupyter nbconvert notebook1.ipynb notebook2.ipynb

            or you can specify the notebooks list in a config file, containing::

                c.NbConvertApp.notebooks = ["my_notebook.ipynb"]

            > jupyter nbconvert --config mycfg.py

To see all available configurables, use `--help-all`.
```

- Haz doble clic en el archivo nuevo que se creó dentro de la carpeta `arte-analitica` y en la parte superior derecha dale clic en **imprimir**
- Imprime el archivo como PDF y súbelo a Canvas.