**Memoria – elaboración del EDA – 2022.03.14, Gretel De Cuyper**

1. Selección del tema

* Idea inicial: análisis de la evolución de los préstamos de ebooks en las bibliotecas belgas (en función también de la evolución en oferta de lenguas)
  + Datos por obtener a través de APIs (deseo para poder seguir aprendiendo a trabajar con ellos)
  + Problema principal:
    - Devolución del resultado en XML (no he podido averiguar si al final esto hubiera planteado un problema)
    - Solicitar permiso para obtener una key privada (permiso no obtenido todavía 2022.03.12)
* Decisión por otro tema en función de una experiencia personal: en el momento de ir a vivir a otra zona, mi agente de seguros me informó que la **tarifa del seguro de coche** **varía en función de dónde tienes tu domicilio.**
  + Una primera exploración confirma la presencia de datos accesibles abiertamente => tema establecido

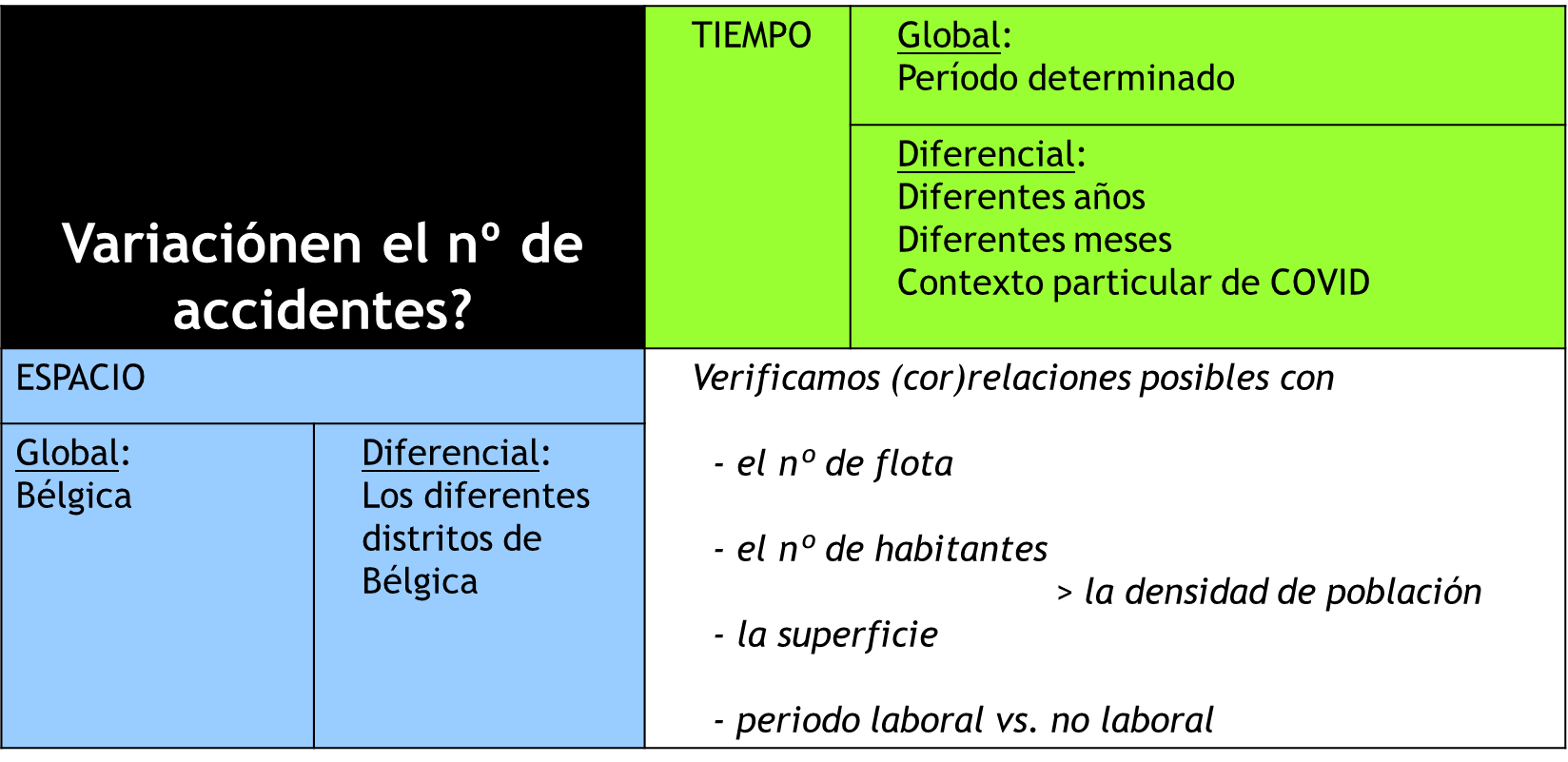
1. Formulación de la hipótesis

Hipótesis principal por investigar a través de datasets:

Se observa ‘alguna’ diferencia en el número de accidentes

* + En el espacio:
    - * diferencias entre distritos?
  + En el tiempo:
    - * diferencias a lo largo de los años?
      * Diferencias dentro de los años:
        + Días laborables vs. Fin de semana
        + Meses laborables vs. Meses de verano

Combinaremos el parámetro espacial constante con uno temporal variable, y *viceversa*.



Este esquema ha proporcionado el hilo y la estructura para el análisis (aunque parcial) llevado a cabo.

Quedan por averiguar si algunos parámetros van de par con las eventuales distinciones por encontrar en el número de accidentes.

**A nivel nacional**

Diferencias temporales (macro-anual):

* Van de par con la evolución de la población – de la densidad de la población?
* Van de par con la evolución de la flota?
* Suceso particular con posible Influencia: crisis COVID a partir de 2020/03

Diferencias temporales (internas al año):

* Van de par con la distinción entre días laborables vs. Días de fin de semana?
* Van de par con la distinción entre meses laborables vs. Meses de verano?
* **A nivel subnacional**

Diferencias en el espacio:

* Van de par con la densidad de la población?

1. Obtención de los datos

Nuestra fuente principal ha sido el Instituto Nacional de la Estadística (Bélgica).

<https://statbel.fgov.be/nl>

Dispone de un website con datos sobre varios temas, de acceso público desde el 2015.

<https://statbel.fgov.be/nl/open-data>

Los sets de datos que hemos necesitado para nuestro análisis han sido:

1. Sets (varios años) de datos sobre los **accidentes** – nivel Bélgica-provincias-distritos
2. Sets (varios años) de datos sobre la **estructura de la población** – nivel Bélgica-provincias-distritos

Estos sets están disponibles directamente como ‘open data’:

Ej. 2020:

<https://statbel.fgov.be/nl/open-data/verkeersongevallen-2020>

<https://statbel.fgov.be/nl/open-data/bevolking-naar-woonplaats-nationaliteit-burgerlijke-staat-leeftijd-en-geslacht-12>

Debido al gran volumen de los sets de tipo 2: descargable solo versión ZIP (.txt)

1. Un set de datos sobre la **superficie** – nivel Bélgica-provincias-distritos

Adaptando los datos a nuestras necesidades en la web, habíamos querido integrarlos en VSCode via webscraping. Gracias a una tutoría con Juan, me enteré que ciertos casos son menos aptos a tal efecto. Datos conseguidos descargando un formato csv.

<https://bestat.statbel.fgov.be/bestat/crosstable.xhtml?view=a9d38c56-1d35-4d18-b1cc-00a43b17dfe7>

1. Un set de datos (incluyendo ya varios años) sobre la **flota de vehículos** – nivel Bélgica

<https://statbel.fgov.be/nl/themas/mobiliteit/verkeer/voertuigenpark#panel-12> (descargar archivo.xls via enlace debajo de la página)

Tal como se puede observar, la frecuencia anual de los datos empezó solo a partir de hace unos 15 años. Cuando los datos no son todavía anuales, tampoco están disponibles siempre los mismos años para cada tipo de dato requerido. Esto ha hecho que hayamos adaptado nuestra selección original de escala temporal para el análisis de 2006,2011,2016-2020 a 2008, 2012, 2016-2020.

La calidad de un buen análisis depende no solo de la calidad del trabajo realizado por el analista, sino también de la fiabilidad de los datos en sí mismo. Para los accidentes, el instituto menciona una fiabilidad de 90% para el registro de accidentes con lesiones mortales, un 50% si hay heridos graves, y tan solo un 20% para los accidentes con heridos leves. La fiabilidad de los datos es un hecho que siempre debería tenerse en cuenta en los resultados finales.

En general, cuanto más información se tenga sobre cómo se reunieron los datos, mejor para el entendimiento profundo del análisis. El Instituto Nacional de Estadística ofrece mucha, sobre todo para los dos primeros datasets (cómo se han reunido los datos, disponibilidad de los datos, …)

1. Limipieza de los datos

La limpieza ha consistido sobre todo en:

* Limpieza de contenido
  + Quitar columnas/filas no relevantes o duplicados para nuestro propósito
  + Cálculos entre columnas/filas para obtener resultados deseados, con o sin operaciones previas de ‘groupby’, aplicar mascarillas
  + Detectar valores NaN y tomar decisión acerca de qué hacer con ellos
* Limpieza de layout
  + Renombrar columnas
  + Redondear cifras obtenidas
  + Cambiar formato ‘ciéntifico’ de valores (no conseguido)

En muchas ocasiones ha sido un proceso reiterativo: después de haber limpiado, convenía otra limpieza debido a la limpieza anterior (ej. renombrar columnas).

Entre los retos encontrados/’lessons learnt’:

* + - * Importación de datos de un notebook a otro: a priori no funciona

Solucionado mediante info en la web: instalación librería import\_ipynb

* + Reto: al principio, solucionado, después resultó igualmente inestable
    - Solucionado gracias a tutoría con Juan: escritura a formato .csv de los dataframes útiles, reunir en carpeta Datos, para que sea posible su lectura en el Notebook nuevo

*Lesson learnt para próximas ocasiones: trabajar en formato .py evitará este tipo de problemas…*

- El saber qué datos más vale conservar, y qué datos se pueden echar. Un dato no relevante al inicio del análisis podría resultar serlo a un estadio más avanzado…

*Lesson learnt: cuánto más claro el objetivo deseado del análisis desde el principio, menor se volverá este tipo de problema*

- Alguna pérdida de trabajo en VSCode por motivo desconocido

*Buena práctica general… guardar de manera ‘activa’ cuanto más posible*

- Diversas operaciones para limpiar: buscando soluciones en internet

*Generalmente se han podido resolver los problemas, o encontrar alternativas para conseguir el objetivo*

* Modificaciones que no se actualizan

*Lesson learnt: cerrar y volver a abrir fichero (o VSCode) a menudo conlleva la solución*

* Al juntar el número de accidentes con el número de la flota, nos encontramos con una operación ‘merge’ que – a pesar de tener rellenados todos los datos los 2 dataframes – borró 2 filas en el resultado final. En otra ocasión, nos conservó solo una fila de tantas. Las causas han quedado desconocidas. Para no perder tiempo en profundizar más, recorrimos a la alternativa de concatenar, que no ha planteado más problemas.

*Lesson learnt: el merge todavía necesita otra mirada… + no siempre visible a ojo: la clave necesita siempre ser del mismo datatype*

* El haberse quedado con la sensación que el trabajo habría podido ganar en encapscular más operaciones sobre los datasets en definiciones (ahora hemos recorrido a unas pocas tan solo, alguna basado en la web, alguna de elaboración ‘propia’). El uso de clases tal vez hubiera podido ser relevante también, pero aquí también: falta de entendimiento profundo para saber si y cómo poner en práctica.

*Lesson learnt: más práctica debería aportar más fluidez en el encapsular*

* Preparar gráficos comparativos con escalas muy distintas: uso de Minmax scaling. Medio logro, pero no implementado

*Lesson learnt: existe una herramienta, estudiar más para implementar efectivamente en gráfico*

* Alguna mezcla de lengua todavía (inglés – español – neerlandés)

*Lesson learnt: decidir desde el principio en qué lengua se van a presentar los resultados – si no es para uso fuera de los analistas, no tiene tanta importancia*

* La función RUN ALL de Jupyter Notebook, aparentemente no lo ejecuta todo. Una ejecución manual de todos los códigos se ha impuesto más de una vez… debido al usuario, o a Jupyter Notebook?

1. El propio análisis para evaluar las hipótesis

Al haber limpiado los sets para nuestro propósito en diversos Notebooks, hemos conseguido 4 tipos de datos directamente listos para usar en nuestro análisis:

1. El número de accidentes – cada año | promedio por año *(en el periodo preestablecido)*
2. El número de habitantes – cada año | promedio por año *(en el periodo preestablecido)*
3. La densidad de la población (derivado de 2 y 3)
4. La flota

Los análisis han sido llevados a cabo en Notebooks ‘MAIN’, uno para el nivel Bélgica, otro para el nivel de los distritos belgas. Han podido ofrecer una respuesta a todas las hipótesis planteadas.

6. Visualización de los datos

La parte técnica de la visualización, a veces con alguna lucha por todavía no estar preparadas como hace falta los dataframes (ej. 2 variables en un eje x (valores de habitantes y accidentes + valores por año aplicado a estas dos variables)). En este caso, vuelta a elaborar los dataframes para formatear después en un gráfico.