Práctica - Herramientas de programación

CUNEF - Máster en Data Science

Septiembre 2022

Idea general

Tu tarea consiste en preparar un conjunto de datos para un proyecto de modelo de Marketing Mix. El objetivo del Marketing Mix Modelling (MMM) es cuantificar el efecto de la publicidad en las ventas de un producto. En esta práctica trabajarás con datos anonimizados de un producto desconocido.

Tarea

A continuación encontrarás varios enunciados que te guiarán en un análisis exploratorio de los datos. Tienes que realizar las tareas que te piden los enunciados con R. Escribe todos tus códigos, ordenados, en un script. Cuando lo tengas terminado, en RStudio haz click en `File -> Compile Report`. Si no hay errores, se creará un fichero con extensión `.html`. Sube ese fichero al espacio en Canvas habilitado para ello.

Si hay errores en tu código, no se generará ese fichero. Arregla los errores.

Aparte de que el código sea válido para lo que se pide, también se tendrá en cuenta su limpieza para la evaluación. Añade comentarios de todo lo que veas conveniente con `#'` (fíjate en el apóstrofo que hay después de la almohadilla). Usa comentarios también para responder a las preguntas.

Enunciados

1. Crea un data frame o tibble a partir de los datos del fichero `mktmix.csv`. Usa la función `clean_names()` del paquete `janitor` para cambiar los nombres de columnas. A partir de ahora, haré referencia a este data frame como `df_mmm`, aunque lo puedes llamar como quieras.

2. ¿Cuántas columnas tiene el data frame? ¿Y filas? ¿Cuáles son las clases de las columnas `base_price` y `discount`? Explica qué información crees que contienen. **En este pregunta no evalúo el código que uses sino las respuestas que des a las preguntas.

3. La clase de `newspaper_inserts` es `character`. Cambia sus valores para que sea numérica de la siguiente forma: todos los valores `NA` deben ser 0; los demás, 1. *Es verdad que pasarla a tipo `logical` haría que ocupara menos, pero como estás preparando datos para un modelo, es mejor que todo sea de tipo numérico.*

4. ¿Cuántos valores distintos (o únicos) hay en la columna `website_campaign` (`NA` no cuenta)? Crea nuevas columnas para cada una de estas categorías, definidas como 1 si `website_campaign` toma el valor de esa categoría y 0 en caso contrario. Por ejemplo, si una de las categorías de `website_campaign` es `"Google"`, crea una columna nueva llamada `google` que valga 1 en los registros en los que `website_campaign` valga `"Google"` y 0 en los demás. *Recomendación.* Los `NA` que hay en la columna original te pueden dar problemas a la hora de crear las nuevas columnas. Prueba a reemplazar esos `NA` por un valor auxiliar antes de crear las columnas.

5. Cada fila de la tabla representa una semana en el histórico. Calcula cuántas semanas ha habido campaña de Facebook y cuántas semanas ha habido campaña de Twitter (hazlo con las columnas que calculaste en el ejercicio anterior).

6. La columna `tv` indica la cantidad de inversión que se ha hecho en anuncios de televisión. La unidad es **grp**. ¿Cuántas semanas se ha realizado una inversión de menos de 50 grp?

7. Calcula la media de inversión en tv durante las semanas en las que hubo inversión en radio y en durante las que no hubo (aquellas en las que `radio` sea `NA`). ¿Qué media es mayor? **Se valorará positivamente que realices este ejercicio con `group_by()` y `summarise()`.

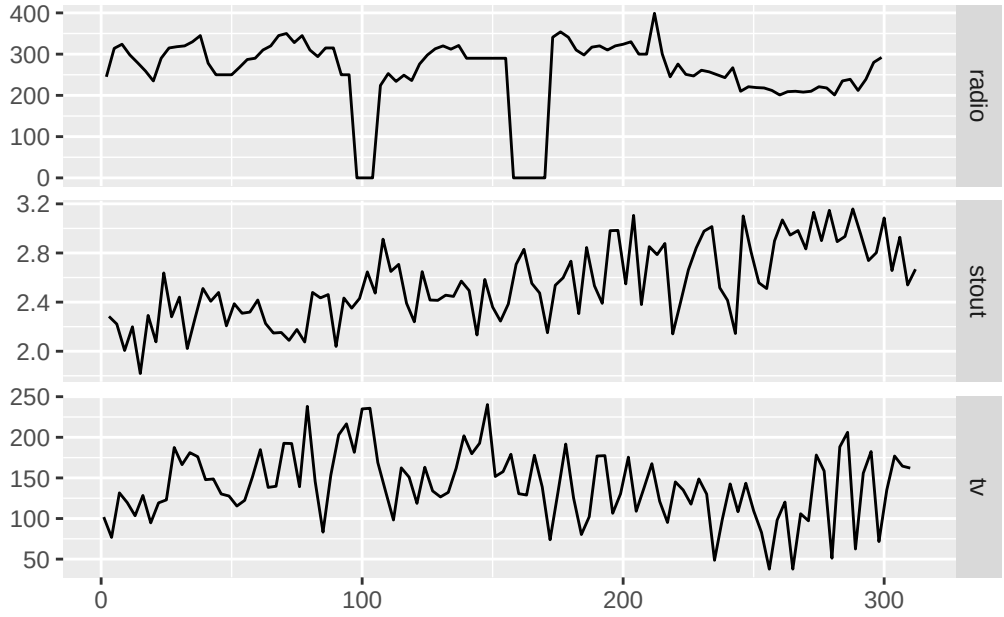
8. Crea un gráfico de líneas con ggplot2 que muestre la evolución de `new_vol_sales`, que es la columna con datos sobre las ventas del producto. Como no tienes datos de fechas, tendrás que inventarte el eje x: haz que sean valores 1, 2, 3, 4… hasta el número de filas. *Pista.* Puedes pasarle a ggplot directamente esos valores en el eje x, sin que sea una columna del data frame.

9. Crea un histograma y un boxplot de esa variable, `new_vol_sales`. A ojo, a partir de los gráficos, ¿cuál dirías que es la mediana de la variable? Y sin hacer cálculos, ¿crees que la media es mayor o menor?

10. Crea un data frame o tibble nuevo que tenga solo las columnas `tv`, `radio` y `stout`. Son las columnas que tienen datos de medios publicitarios: televisión, radio y exterior (carteles de esos que ves en la calle o en la carretera). Usa ese data frame para generar el gráfico siguiente. **Usa el siguiente código también como paso previo antes de generar el gráfico. Cambiará el formato de tu nuevo data frame y te será**

más directo construir el gráfico. Para usar el código, he asumido que el nombre que le das al data frame nuevo es `df_media` pero puedes cambiar el nombre si quieres. *Comentario.* He quitado las etiquetas de los ejes para no darte pistas de cómo creo el eje x. **Responde a lo siguiente:** ¿qué merece la pena comentar a raíz del gráfico?

```
library(tidyr)
df_media <- df_media %>%
  pivot_longer(everything())
```



11. La columna `in_store` mide el *stock* disponible que hay en las tiendas para vender el producto, de manera indexada. Crea un gráfico de dispersión con ggplot que compare `new_vol_sales` frente a `in_store`. Presta atención a qué columna pondrás en el eje x y cuál en el y: para ello, ten en cuenta que `new_vol_sales` será la variable objetivo del modelo, *i.e.*, el analista explicar esa variable en función de las demás. Además, añade una capa con `geom_smooth()`: los ejes x e y serán los mismos que pongas en el `geom` del gráfico de dispersión. **Comenta qué conclusiones sacas a la vista del gráfico** (piensa en qué relación hay entre el *stock* y las ventas).

12. Repite el gráfico anterior pero de dos formas diferentes (no pongas `geom_smooth` esta vez).

    - Colorea cada punto en función de la columna `newspaper_inserts`. Te recomiendo que uses `as.factor()` con esa columna en el gráfico.
    - Colorea cada punto de manera diferente en función de la columna `tv`.

13. Crea otra columna indicando si se ha aplicado descuento o no (es decr, si la columna `discount` es mayor que 0 o no). Puedes llamarla `discount_yesno`, por ejemplo, y puede ser numérica o `logical`. Luego agrega el data frame calculando la media de `base_price` para los casos en los que hay descuento y los casos en los que no. O sea, el resultado será un data frame de dos filas y dos columnas. Usa este data frame para crear un gráfico de columnas. *Observación.* Valoraré positivamente que no crees nuevos data frames (tampoco sobreescribas el origina porque lo necesitarás más adelante), sino que encadenes todas las operaciones con el operador *pipe* `%>%`, incluido el gráfico de ggplot.

14. Apóyate en el siguiente código para crear una función que ajuste un modelo de regresión en los datos. La función recibirá como entrada un vector `character` con las variables que se quieran usar como explicativas en el modelo. Con ese vector, dentro de la función crea un nuevo data frame que tenga esas variables y la variable explicativa, `new_vol_sales`. Este data frame lo usarás para ajustar el modelo siguiendo el código siguiente. Ese código calcula lo que se llama *R cuadrado ajustado*, una métrica qué indica cómo de bueno es el modelo: cuanto más alto, menor error hay. Finalmente, llama a la función usando como vector de entrada `c("tv", "radio")`, es decir, ajustarás un modelo que intente explicar las ventas en función de la publicidad realizada en televisión y en radio. *Observación.* Lo que en el código se llama `df_aux` hace referencia al data frame auxiliar que crearás dentro de la función.

```
# This code fits a linear regression model with all the variables
# in df_aux, using new_vol_sales as target variable
my_model <- lm(new_vol_sales ~ ., data = df_aux)

# Value to be returned by the function, the adjusted R squared.
summary(my_model)$adj.r.squared
```

15. Debajo tienes tres vectores con nombres de variables. Crea una lista con esos vectores, es decir, una lista de tres elementos (cada elemento será cada uno de los vectores). Ahora usa `map_dbl()` o `sapply()` para llamar a tu función para cada uno de los vectores. El resultado de tu código será un vector de tres números: cada número será el *R cuadradado ajustado* de los modelos que se ajustan con esos conjuntos de variables. **¿Qué modelo es el mejor, de acuerdo al R cuadrado ajustado?**

    - `c("base_price", "radio", "tv", "stout")`
    - `c("base_price", "in_store", "discount", "radio", "tv", "stout")`
    - `c("in_store", "discount")`
