Ejercicios N°2: Trabajando con data frames

Introducción al Análisis de Datos Geoespaciales CBIT202-18

La guía consta de **7 ejercicios** de conceptos básicos de R. Esta será entregada en formato PDF y deberá ser recreada en RMarkdwon, poniendo como nombre del alchivo "Ejer2_apellido_nombre.Rmd". Recuerde reemplazar todas las partes con '#???' y '???' con su código, y/o trabaje en la zona de 'desarrollo'.

Realice el desarrollo de estos ejercicios al interior del chunk correspondiente. Si necesita escribir hágalo ocupando #, esto es recomendable para aclarar qué hace su código. Puede ayudarse de inteligencia artificial para investigar sobre el funcionamiento de funciones, pero resuelva las preguntas en base a su propio conocimiento e investigación. Recuerde que puede ocupar ? para saber más información sobre las funciones, paquetes instalados y datos contenidos en R base.

```
# Por ejemplo:

# Funcion
?mean()
# Paquete
?stats()
# Datos
?iris()
```

Para acceder a los atajos del teclado puede presionar Alt + Shift + K (en Windows/Linux), Option + Shift + K (en macOS) o Help > Keyboard Shortcuts Help. No olvide que puede retroceder en cualquier cambio que realice con Ctrl + Z.

Ejercicios (18pts):

Indique su nombre:

```
# Nombre alumno:
```

Los ejercicios de esta guía requieren del paquete "datos", el cual contiene conjuntos de datos de distintos paquetes, incluyendo R base, traducidos al español. Por favor, corra el siguiente código para instalarlo y cargarlo antes de continuar con la guía:

```
install.packages("datos") # instalar (solo se hace una vez, si lo tiene instalado comente esta linea)

library(datos) # llamar
```

1) Reescriba el siguiente código con el uso de la pipe (%>% o |>). **(2pts)**

```
# resumen de un data frame
glimpse(diamantes)

# Con %>% :

# la desviación estándar redondeada de la raíz cuadrada del logaritmo de los números del 0 al 60 de 5 e
round(sd(sqrt(log(seq(0, 60, by = 5)))), 3)

# Con %>% :
```

```r
# contar los diamantes por categoria
count(diamantes, corte)

# Con %>% :



# filtrar los diamantes pequeños
filter(diamantes, quilate < 3)

# Con %>% :



# seleccionar las variables x, y, z de pequeños y transformar su clase a numérico.
mutate(select(pequeños, x, y, z), across(c(x, y, z), as.numeric))

# Con %>% :



# seleccionar la variables precio de pequeños, transformar la columma en un vector (con unlist()) y cal
mean(unlist(select(pequeños, precio)))

# Con %>% :



# calcular los promedios por corte y ver el resultado como una tabla con view().
view(summarise(group_by(diamantes, corte), media_precio = mean(precio, na.rm = TRUE)))

# Con %>% :
```

2) Instale, si es necesario, y cargue el paquete "tidyverse". ¿Qué significa el aviso que entrega el paquete tidyverse sobre conflictos? ¿Cómo puedo asegurarme de estar llamando a la función filter() del paquete tidyverse sabiendo que el paquete stats (R base) contiene una función con el mismo nombre? ¿Y cómo se utilizaría filter() del paquete stats (no es necesario usarla, solo escribirla)?

Para el data frame 'trees', utilice la función filter() de tidyverse para saber cuántos árboles poseen una altura mayor a 80. (**2 pts**)

```r
# Desarrollo:
```

3) El siguiente código crea y guarda un archivo separado por comas (CSV) en su computadora, por favor NO modifique lo que está antes de la parte de desarrollo. Cargue este archivo desde su computadora a R, para esto ayúdese de la función getwd() para saber la localización de su directorio de trabajo, ya que en este se habrá guardado su archivo, y la función read_csv() o read_table().

Explique c[omo se crean las variables del data frame 'df_creado', que hace cada función en la creación de 'df_creado'. **(2pts)**

```r
# Creación de un df
df_creado <- data.frame(ID = c(1:1000),
                arboles = sample(c('roble', 'rauli', 'coigue'), size = 1000, replace = TRUE),
                DAP = rbeta(1000, shape1 = 2, shape = 5),
                estado_sanitario = factor(rbinom(1000, size = 1, prob = 0.2), levels = c(0, 1), labels

# Guardado del df en su computadora con el nombre df_ejercicio_7
write.csv(df, 'df_ejercicio_7.csv')

# ver working directory
getwd()

# lea el data frame
df_leido <- #???
print(df_leido)

# Desarrollo:
```

4) Mencione las 3 características de los datos ordenados y explique por qué el data frame table2 no se encuentra ordenado, en comparación con table1, que sí lo esta. **(2pts)**

```r
# Data frame ordenado:
print(table1)

# Data frame NO ordenado:
print(table2)

# Desarrollo:
```

5) Filtre (filter()), seleccione columnas (select()), ordene (arrange()) y agrupe (group_by() y summarise()), de tidyverse, en el siguiente data frame con datos de una Encuesta Social General de EE.UU., de tal manera que responda las siguientes preguntas solo con la información solicitada. Tenga en cuenta que las preguntas pueden ser respondidas usando distintos enfoques. Aproveche además otras funciones como length(), first(), mean(), unique(), colnames(), count(), etc. **(4 pts)**

Tome en cuenta que pueden haber datos faltantes (Na's) en vuelos a la hora de generar calculos como mean() y otros

a- ¿En qué años nacieron todas las personas cristianas encuestadas? b- ¿Cuál es el valor más alto de horas viendo televisión en la encuesta? c- ¿Hay más personas casadas, divorciadas, viudas o que nunca se han casado? d- ¿Cuál es la edad media de los encuestados según su religión? ¿Los integrantes de qué religión, en promedio, son más jóvenes? e- ¿Cómo puedo quitar la variable "partido" del data frame? f- ¿Cómo puedo saber los valores únicos de la columna "anio" (año de la encuesta)? g- ¿La edad promedio es la misma entre encuestas de distintos años? h- ¿Cuál es la edad de las 10 personas más jóvenes encuestadas? ¿Y cuál es su partido? i- ¿Ve algún patrón entre los rangos de ingresos y la edad (puede usar la media, mediana o desviación estándar para responder)? j- Formule una pregunta sobre los datos (escrita explícitamente) y respóndala con código.

```r
encuesta %>% glimpse()
encuesta?

a <- #???
print(a)
```

```r
b <- #???
print(b)

c <- #???
print(c)

d <- #???
print(d)

e <- #???
print(e)

f <- #???
print(f)

g <- #???
print(g)

h <- #???
print(h)

i <- #???
print(i)

# Pregunta propia:
j <- #???
print(j)


# Correr para borrar los objetos en memoria (ahorar memoria)
rm(list = ls())
```

6) Basado en el data frame 'vuelos', correspondiente a datos de todos los vuelos que despegaron de Nueva York durante el año 2013, genere las siguientes nuevas variables con mutate() o trasmute(), segun corresponda (Tome en cuenta que transmute genera nuevas variables, al igual que mutate, solo que no las integra en el data frame original, solo entrega un data frame con las nuevas variables. Para más información escriba: transmute()?): (**4**)

No es necesario que guarde cada nuevo data frame como un objeto, solo imprimalo en consola.

- A- Primero, agregue al "vuelos", las variables ganancia (en terminos de la diferencia entre el atraso de salida y el atraso de llegada, es decir, cuantos minutos "recupera", o pierde, durante el vuelo), ganacia_por_hora (la ganancia por hora de vuelo) y velocidad (siendo la velocidad media del avión durante el vuelo).

- B- Genere las mismas nuevas variables que B, pero en un data frame que solo contenga estas nuevas variables.

- C- Genere un data frame que contenga la variable horario_salida y, en base a esta variable, dos nuevas variables que separen esta variable en la hora dentro de la que sale el avion y los minutos dentro de

esa hora en las que sale el avion (pista: use %/% y %%). Antes de responder pregúntese: ¿En que formato de datos esta la variable horario_salida?

- D- Las variables horario_salida y salida_programada tienen un formato conveniente para leer, pero es difícil realizar cualquier cálculo con ellas porque no son realmente números continuos. Apoyandoce en el data frame del punto C, transforme estas variables, o genere dos nuevas variables que si sean continuas, como el número de minutos desde la medianoche. Y corra el histograma que agrupa la salida de vuelos por hora ¿Ve algún patrón en el grafico? (**NO modifique el código del grafico**)

```
# A:
vuelos %>% mutate(
  ganancia = #???
  ganancia_hora = #???
  velocidad = #???
)
```

```
# B:
# Desarrollo:
```

```
# C:
transmute(vuelos,
  horario_salida,
  hora = #???
  minuto = #???
)
```

```
vuelos_salida <- transmute(vuelos,

  horario_salida,
  hora_salida = #???
  minuto_salida = #???
  minutos_continuos_salida = #???

  salida_programada,
  hora_programada = #???
  minuto_programada = #???
  minutos_continuos_programada = #???
)

# Corra el plot una vez listo el data frame vuelos_salida
ggplot(vuelos_salida, aes(x = minutos_continuos_salida)) +
  geom_histogram(binwidth = 60, fill = "lightblue", color = "black", boundary = 0) +
  scale_x_continuous(breaks = seq(0, 1440, by = 180)) +
  labs(title = "Vuelos agrupados por hora", x = "Minutos Continuos", y = "Frecuencia") +
  theme_minimal()

# ¿Ve algun patron en el grafico?
# Desrrollo>:
```

Corra el siguiente chunk para borrar los objetos en memoria, así ahorrará memoria.

```
rm(list = ls())
```

7) Utilizando las funciones group_by() y summarise(), genere los siguientes data frames:

- df1: Con el data frame "vuelos", genere un data frame con el número de vuelos, distancia promedio y atraso promedio, por destino desde Nueva York. Para calcular el número de vuelos, explore la función específica n(). Al calcular las medias, considere que puede haber datos faltantes (NA's) en los datos de vuelos; vea los argumentos de mean() para manejar estos casos.
- df2: Con el data frame "clima", genere un data frame con los días con mayor temperatura media diaria por mes y origen. (2 pts)

```
# df1
# Desarrollo:
datos::vuelos %>% glimpse()


# df 2
# Desarrollo:
datos::clima %>% glimpse()
```

Bonus

Investige las funciones pivot_longer y pivot_wider, y ocupe una de estás para transformar la tabla2 en datos ordenados

```
# Desarrollo:
# data frame no ordenado
table2

# data frame ordenado
table1

# ordenar el data frame table2
tabla_ordenada <- #???

print(tabla_ordenada)
```