

Examen 1 - Ciencia de Datos para Negocios

2025-04-21

Introducción

Este examen evalúa tus habilidades para analizar un conjunto de datos relacionados con alquileres temporales de propiedades. Vas a trabajar con cinco tablas interrelacionadas y tenés que realizar una serie de análisis para extraer información valiosa para el negocio. No te preocupes -por ahora- es una evaluación de código, no de analizar los resultados, pero si lo hacés, los podemos conversar a ver qué hiciste.

- **IMPORTANTE:** Si tu legajo termina en un número par, solo debes hacer los ejercicios pares. Si termina en un número impar, solo debes hacer los ejercicios impares
- **IMPORTANTE 2:** los ejercicios 12, 15, 16 y 17 tienen un grado de dificultad mayor a lo visto en clase. No tengan miedo de equivocarse y no duden en usar **TODAS** las herramientas que conocen para resolverlo.

Si tienen dudas, pueden escribirme, pero que sea el último recurso, parte de la materia es aprender a lidiar con los problemas de código.

FECHA DE ENTREGA: tienen tiempo para entregarlo hasta el martes 29/04 a las 19hs. Se deberá enviar a nsidicaro.fce@gmail.com (leer el final de las consignas donde indica el formato de entrega).

Descripción de los datos

El conjunto de datos consiste en cinco tablas principales que representan un sistema de alquiler temporal de propiedades, similar a plataformas como Airbnb:

usuarios_alquilan

Contiene información sobre los usuarios que alquilan propiedades (huéspedes):

Variable	Descripción
id_alquila	Identificador único del usuario
sexo	Género del usuario (Masculino/Femenino)
edad	Edad del usuario
provincia_nacimiento	Provincia de origen del usuario
cantidad_alquileres	Número de alquileres registrados hasta la fecha
calificacion_media	Puntuación media recibida como huésped (1-5)
fecha_registro	Fecha en que el usuario se registró en la plataforma
nivel_verificacion	Nivel de verificación de identidad (Básico/Intermedio/Completo)
idiomas_hablados	Idiomas que habla el usuario (separados por comas)

Variable	Descripción
preferencia_ubicacion	Provincia de preferencia para alquileres

usuarios_arrendatarios

Contiene información sobre los usuarios que ofrecen propiedades en alquiler (anfitriones):

Variable	Descripción
id_arrendatario	Identificador único del arrendatario
sexo	Género del arrendatario
edad	Edad del arrendatario
provincia_nacimiento	Provincia de origen del arrendatario
cantidad_viviendas_publicadas	Número de propiedades listadas por el arrendatario
calificacion_media	Puntuación media recibida como anfitrión (1-5)
fecha_registro	Fecha en que el arrendatario se registró en la plataforma
nivel_verificacion	Nivel de verificación de identidad
es_superanfitrión	Indica si tiene estatus de superanfitrión (TRUE/FALSE)
tiempo_respuesta_hrs	Tiempo promedio de respuesta en horas
anos_experiencia	Años de experiencia como anfitrión
politica_cancelacion	Tipo de política de cancelación ofrecida

casas

Contiene información sobre las propiedades disponibles para alquiler:

Variable	Descripción
id_casa	Identificador único de la propiedad
id_arrendatario	ID del arrendatario propietario
tipo_vivienda	Tipo de alojamiento (Casa/Departamento/Cabaña/etc.)
provincia_ubicacion	Provincia donde se encuentra la propiedad
capacidad_maxima	Número máximo de huéspedes
numero_habitaciones	Cantidad de habitaciones
numero_banos	Cantidad de baños
precio_base_noche	Precio base por noche en pesos
categoria	Categoría de la propiedad (Económica/Estándar/Premium/etc.)
distancia_centro_km	Distancia al centro en kilómetros
anio_construccion	Año de construcción del inmueble
amenities	Servicios ofrecidos (separados por comas)
calificacion_limpieza	Puntuación de limpieza (1-5)
calificacion_ubicacion	Puntuación de ubicación (1-5)
calificacion_general	Puntuación general (1-5)
mascotas_permitidas	Indica si se permiten mascotas (TRUE/FALSE)
fumadores_permitidos	Indica si se permite fumar (TRUE/FALSE)
eventos_permitidos	Indica si se permiten eventos (TRUE/FALSE)

transacciones

Registra las transacciones de alquiler realizadas:

Variable	Descripción
id_transaccion	Identificador único de la transacción
id_alquila	ID del usuario que alquila
id_arrendatario	ID del arrendatario de la propiedad
id_casa	ID de la propiedad alquilada
fecha_transaccion	Fecha en que se realizó la transacción
origen_reserva	Canal por el que se realizó la reserva

detalles_transacciones

Contiene detalles específicos de cada transacción:

Variable	Descripción
id_transaccion	ID de la transacción (vincula con la tabla transacciones)
cantidad_noches	Número de noches de estadía
precio_noche	Precio final por noche (puede diferir del precio base)
cantidad_personas	Cantidad de personas alojadas
fecha_checkin	Fecha de inicio de la estadía
fecha_checkout	Fecha de fin de la estadía
metodo_pago	Método utilizado para el pago
estado_transaccion	Estado actual (Completada/Cancelada/Pendiente/En curso)
descuento_aplicado	Porcentaje de descuento aplicado (0-30)
costo_limpieza	Cargo adicional por limpieza
costo_servicio	Cargo por servicio de la plataforma
impuestos	Monto de impuestos aplicados
calificacion_huesped	Puntuación dada al huésped (1-5)
calificacion_anfitrión	Puntuación dada al anfitrión (1-5)

Preparación inicial

Antes de comenzar con los ejercicios, configurá tu entorno de trabajo ejecutando el siguiente código:

```
# === INFORMACIÓN DEL ESTUDIANTE ===
# Nombre:
# Apellido:
legajo <- 12345 # REEMPLAZÁ CON TU NÚMERO DE LEGAJO
# Email:

# === CONFIGURACIÓN INICIAL ===
# Establecer semilla para reproducibilidad
set.seed(legajo)

# Cargar librerías necesarias
library(tidyverse)
library(lubridate) # Para manejo de fechas
library(scales)    # Para formateo de valores

# Crear directorio para resultados
```

```

if(!dir.exists("resultados")) {
  dir.create("resultados")
}

# Cargar los datos de examen
source("generar_datos_examen.R")
datos <- generar_datos_examen(legajo)
guardar_datos_examen(datos)

# Rutas archivos
instub <- 'datos_examen'
outstub <- 'resultados'

# Cargar datos
transacciones <- read.csv(file.path(instub,"transacciones.csv"))
usuarios_alquilan <- read.csv(file.path(instub,"usuarios_alquilan.csv"))
usuarios_arrendatarios <- read.csv(file.path(instub,"usuarios_arrendatarios.csv"))
casas <- read.csv(file.path(instub,"casas.csv"))
detalles_transacciones <- read.csv(file.path(instub,"detalles_transacciones.csv"))

# Convertir columnas de fecha a tipo Date
transacciones$fecha_transaccion <- ymd(transacciones$fecha_transaccion)
usuarios_alquilan$fecha_registro <- ymd(usuarios_alquilan$fecha_registro)
usuarios_arrendatarios$fecha_registro <- ymd(usuarios_arrendatarios$fecha_registro)
detalles_transacciones$fecha_checkin <- ymd(detalles_transacciones$fecha_checkin)
detalles_transacciones$fecha_checkout <- ymd(detalles_transacciones$fecha_checkout)

```

Instrucciones generales

- Es fundamental que las tablas resultantes tengan **exactamente los mismos nombres de columnas** y estructura que se especifican en cada consigna.
- Se va a evaluar la coincidencia en nombres de columnas, cantidad de filas y columnas, y los valores de los datos.
- Para cada ejercicio, tenés que guardar el resultado usando `file.path(outstub, nombre_archivo)` con el nombre de archivo especificado.
- Todos los ejercicios están diseñados para ser resueltos con las herramientas que vimos en clase, principalmente usando el ecosistema tidyverse.
- **IMPORTANTE:** El archivo .R que entregues debe estar guardado con codificación UTF-8.
- **IMPORTANTE 2:** Si tu legajo termina en un número par, solo debes hacer los ejercicios pares. Si termina en un número impar, solo debes hacer los ejercicios impares

Ejercicios

Ejercicio 1: Análisis temporal de ingresos

Objetivo: Hacé un análisis de los ingresos mensuales generados por la plataforma de alquiler temporal.

Pistas:

1. Empezá extrayendo el año y mes de cada transacción usando `floor_date` con la unidad “month”

2. Uní la tabla de transacciones con detalles_transacciones usando `left_join`
3. Calculá el ingreso por transacción sumando `precio_noche * cantidad_noches + costos adicionales`
4. Agrupá por mes con `group_by` y calculá métricas con `summarise`: `total_ingresos`, `cantidad_transacciones`, `ingreso_promedio`
5. Calculá `noches_totales` y la tasa de ocupación mensual (considerá los días del mes y el número de propiedades)
6. Ordená cronológicamente con `arrange`

Columnas requeridas:

- `anio_mes` (fecha del primer día del mes)
- `total_ingresos`
- `cantidad_transacciones`
- `ingreso_promedio`
- `noches_totales`
- `tasa_ocupacion` (porcentaje)

Guardar como: “`ingresos_mensuales.csv`”

Ejercicio 2: Segmentación de propiedades por rentabilidad

Objetivo: Segmentá las propiedades por su nivel de rentabilidad para identificar las más eficientes.

Pistas:

1. Uní las tablas de transacciones con detalles_transacciones
2. Agrupá por `id_casa` para calcular el ingreso total y días de ocupación
3. Calculá el `ingreso_por_dia` dividiendo `ingreso_total` entre `dias_ocupacion`
4. Usá la función `ntile()` para crear cuartiles de rentabilidad con la etiqueta “Q1”, “Q2”, etc.
5. Creá un campo `alta_rentabilidad` que sea `TRUE` para el cuartil más alto (Q4)
6. Agregá información descriptiva sobre la propiedad usando la tabla `casas` con `left_join`
7. Ordená por rentabilidad descendente

Columnas requeridas:

- `id_casa`
- `ingreso_total`
- `dias_ocupacion`
- `ingreso_por_dia`
- `cuartil_rentabilidad` (Q1, Q2, Q3, Q4)
- `alta_rentabilidad` (TRUE/FALSE para Q4)
- `tipo_vivienda`
- `provincia_ubicacion`
- `categoria`

Guardar como: “`propiedades_por_rentabilidad.csv`”

Ejercicio 3: Análisis de efectividad de descuentos

Objetivo: Evaluá cómo los diferentes niveles de descuento afectan las métricas de reserva según la temporada.

Pistas:

1. Creá grupos de descuento usando `case_when`: “0%”, “1-10%”, “11-20%”, “>20%”
2. Uní con la tabla de transacciones para obtener las fechas
3. Determiná la temporada según el mes:
 - Verano: meses 12, 1, 2
 - Otoño: meses 3, 4, 5
 - Invierno: meses 6, 7, 8
 - Primavera: meses 9, 10, 11
4. Agrupá por `grupo_descuento` y `temporada`
5. Calculá: número de transacciones, duración promedio, ingreso promedio por día y tasa de cancelación

Columnas requeridas:

- `grupo_descuento`
- `temporada`
- `numero_transacciones`
- `duracion_promedio`
- `ingreso_promedio_dia`
- `tasa_cancelacion`

Guardar como: “efectividad_descuentos.csv”

Ejercicio 4: Perfil de usuarios premium

Objetivo: Identificá y analizá las características del 10% de usuarios con mayor gasto en la plataforma.

Pistas:

1. Calculá el gasto total por usuario uniendo transacciones con `detalles_transacciones`
2. Agrupá por `id_alquila` y sumá los montos con `summarise`
3. Ordená por gasto total descendente con `arrange(desc())`
4. Identificá el top 10% de usuarios usando `head()` con `ceiling(nrow(df) * 0.1)`
5. Uní con la tabla de `usuarios_alquilan` para obtener datos demográficos
6. Analizá las transacciones de estos usuarios para identificar su categoría y provincia preferida (la más frecuente usando `table()` y `sort()`)
7. Agregá estas preferencias al perfil con otro `left_join`

Columnas requeridas:

- `id_alquila`
- `gasto_total`
- `num_transacciones`
- `edad`
- `sexo`
- `provincia_nacimiento`
- `cantidad_alquileres`
- `calificacion_media`
- `categoria_preferida`
- `provincia_preferida`

Guardar como: “perfil_usuarios_premium.csv”

Ejercicio 5: Análisis geográfico de rentabilidad

Objetivo: Compará la rentabilidad de alojamientos entre diferentes provincias.

Pistas:

1. Uní las tablas casas, transacciones y detalles_transacciones
2. Agrupá por provincia_ubicacion
3. Calculá con `summarise`:
 - num_propiedades usando `n_distinct(id_casa)`
 - precio_promedio con `mean()`
 - tasa_ocupacion: días ocupados dividido por días disponibles totales
 - ingreso_promedio por propiedad
 - calificacion_promedio
4. Creá un ratio_precio_calificacion dividiendo precio_promedio entre calificacion_promedio
5. Calculá un ranking_rentabilidad con `rank()` basado en ingreso ajustado por calificación

Columnas requeridas:

- provincia_ubicacion
- num_propiedades
- precio_promedio
- tasa_ocupacion
- ingreso_promedio
- calificacion_promedio
- ratio_precio_calificacion
- ranking_rentabilidad

Guardar como: “rentabilidad_geografica.csv”

Ejercicio 6: Estacionalidad y factor temporal

Objetivo: Analizá cómo varía el mercado a lo largo del año.

Pistas:

1. Uní transacciones con detalles_transacciones
2. Extraé el mes de cada transacción con `month()` y agregá el nombre del mes con `month(fecha, label=TRUE, abbr=FALSE)`
3. Agrupá por mes y calculá:
 - precio_promedio
 - ocupacion_promedio
 - duracion_promedio
 - metodo_pago_frecuente usando `which.max(table(metodo_pago))`
4. Calculá los promedios anuales para todo el dataset
5. Creá índices relativos: indice_precio y indice_ocupacion (valor mensual / promedio anual * 100)
6. Ordená por mes

Columnas requeridas:

- mes
- nombre_mes
- indice_precio
- indice_ocupacion
- duracion_promedio
- metodo_pago_frecuente

Guardar como: “estacionalidad_mercado.csv”

Ejercicio 7: Análisis de relación precio-calificación

Objetivo: Evaluá cómo se relaciona el precio con las calificaciones recibidas.

Pistas:

1. Creá categorías de precio usando `case_when`:

- Económico: < 10000
- Moderado: $10000-19999$
- Alto: $20000-34999$
- Premium: ≥ 35000

2. Convertí a factor ordenado con `factor(levels=c("Económico", "Moderado", "Alto", "Premium"))`

3. Agrupá por categoría_precio y calculá estadísticas

4. Analizá también por tipo de vivienda y categoría de precio

5. Pivoteá los datos para formato más legible con `pivot_wider`

6. Identificá “buenas inversiones”: propiedades económicas/moderadas con calificación ≥ 4.5

7. Identificá “malas inversiones”: propiedades caras con calificación ≤ 3.5

Columnas requeridas para resultado principal:

- categoria_precio
- num_propiedades
- calificacion_promedio
- calificacion_limpieza_promedio
- calificacion_ubicacion_promedio
- precio_promedio
- relacion_calidad_precio

Guardar como:

- “relacion_precio_calificacion.csv”
- “relacion_precio_tipo_pivoteada.csv”
- “buenas_inversiones.csv”
- “malas_inversiones.csv”

Ejercicio 8: Segmentación de arrendatarios

Objetivo: Creá una clasificación de arrendatarios según diferentes criterios.

Pistas:

1. Uní usuarios_arrendatarios con datos de sus propiedades en casas
2. Calculá la antigüedad en días con `as.numeric(Sys.Date() - fecha_registro)`
3. Creá segmentos según:
 - Volumen: número de propiedades (“Pequeño”, “Mediano”, “Grande”, “Muy grande”)
 - Precio: nivel de precio promedio (“Económico”, “Estándar”, “Premium”, “Lujo”)
 - Experiencia: años de antigüedad (“Novato”, “Experimentado”, “Veterano”)
 - Calidad: calificación media (“Básico”, “Bueno”, “Muy bueno”, “Excelente”)
4. Creá una etiqueta compuesta combinando segmentos
5. Analizá las características de cada segmento con agrupaciones

Columnas requeridas:

- id_arrendatario
- segmento_volumen
- segmento_precio
- segmento_experiencia
- segmento_calidad
- segmento_compuesto
- num_propiedades
- precio_promedio
- calificacion_media
- es_superanfitrion
- antigüedad_anos
- tiempo_respuesta_hrs

Guardar como:

- “segmentos_arrendatarios.csv”
- “analisis_segmento_volumen.csv”
- “analisis_segmento_precio.csv”

Ejercicio 9: Análisis de ratio precio/capacidad

Objetivo: Analizá el precio por persona en diferentes tipos de alojamiento.

Pistas:

1. Calculá el precio por persona dividiendo precio_base_noche entre capacidad_maxima
2. Clasificá las propiedades según su ratio en categorías: “Bajo”, “Medio”, “Alto”, “Premium”
3. Convertí a factor ordenado
4. Analizá el ratio por tipo de vivienda y provincia con agrupación
5. Creá una tabla pivoteada usando `pivot_wider`
6. Identificá las mejores oportunidades creando un índice de calidad-precio (calificación/precio)

Columnas requeridas para oportunidades:

- id_casa
- tipo_vivienda
- provincia_ubicacion
- precio_base_noche
- capacidad_maxima
- precio_por_persona
- calificacion_general
- indice_calidad_precio

Guardar como:

- "ratio_precio_capacidad.csv"
- "distribucion_ratio_por_tipo.csv"
- "mejores_oportunidades_precio_capacidad.csv"

Ejercicio 10: Red de transacciones entre provincias

Objetivo: Analizá los flujos de viajeros entre provincias.

Pistas:

1. Uní transacciones con usuarios_alquilan para obtener la provincia_origen
2. Uní con casas para obtener la provincia_destino
3. Contá transacciones por par origen-destino con `group_by` y `summarise`
4. Calculá el flujo neto por provincia:
 - Agrupá por provincia de origen para contar salidas
 - Agrupá por provincia de destino para contar entradas
 - Uní ambos conjuntos y calculá `flujo_net = entradas - salidas`
5. Identificá las top 5 rutas más comunes con `arrange` y `head`
6. Creá una columna ruta que combine origen y destino

Columnas requeridas para top rutas:

- provincia_origen
- provincia_destino
- num_transacciones
- ruta

Guardar como: "top_rutas.csv"

Ejercicio 11: Pivoteo de amenities por tipo de vivienda

Objetivo: Analizá qué amenities son más comunes en cada tipo de vivienda.

Pistas:

1. Usá `separate_rows(amenities, sep = ",")` para dividir el campo de texto en filas individuales

2. Limpiá los datos con `trimws()` y `filter(amenities != "")`
3. Calculá el total de viviendas por tipo
4. Contá ocurrencias de cada amenity por tipo y calculá el porcentaje
5. Pivoteá a formato ancho con `pivot_wider` usando los porcentajes como valores

Columnas requeridas:

- tipo_vivienda
- Columnas para cada amenity (con el porcentaje como valor)

Guardar como: “amenities_por_tipo.csv”

Ejercicio 12: Análisis comparativo de métodos de pago

Objetivo: Compará cómo diferentes métodos de pago se relacionan con comportamientos de reserva.

Pistas:

1. Uní detalles_transacciones con transacciones y casas
 2. Calculá la anticipación de reserva: `as.numeric(fecha_checkin - fecha_transaccion)`
 3. Agrupá por método de pago y calculá estadísticas básicas
 4. Realizá un análisis cruzado:
- Agrupá por metodo_pago y categoria
 - Calculá cantidad y monto_promedio
5. Pivoteá para crear una tabla comparativa con `pivot_wider` y `names_glue`

Columnas requeridas:

- metodo_pago
- cantidad_transacciones
- monto_promedio
- duracion_promedio
- anticipacion_promedio
- tasa_cancelacion
- Columnas cruzadas por categoría (e.g., Económica_cantidad, Económica_monto_promedio)

Guardar como: “analisis_metodos_pago.csv”

Ejercicio 13: Comparativa de calificaciones

Objetivo: Analizá cómo se relacionan las calificaciones entre huéspedes y anfitriones.

Pistas:

1. Filtrá transacciones con ambas calificaciones usando `filter(!is.na(calificacion_huesped) & !is.na(calificacion_anfitrion))`
2. Uní con datos de usuarios y arrendatarios para obtener demografía
3. Calculá la diferencia entre calificaciones y su dirección
4. Creá rangos de edad: “18-25”, “26-35”, “36-50”, “51+”

5. Agrupá por sexo y rango de edad para análisis demográfico

Columnas requeridas:

- sexo_huesped
- rango_edad_huesped
- num_transacciones
- calificacion_media_recibida
- calificacion_media_otorgada
- diferencia_media

Guardar como: “analisis_demografico_calificaciones.csv”

Ejercicio 14: Análisis de antigüedad de usuarios y comportamiento

Objetivo: Evaluá cómo cambia el comportamiento de los usuarios a medida que acumulan más tiempo en la plataforma.

Pistas:

1. Calculá la antigüedad en días para cada usuario
2. Creá categorías de antigüedad: “0-6 meses”, “7-12 meses”, “1-2 años”, “Más de 2 años”
3. Convertí a factor ordenado para mantener el orden lógico
4. Uní con datos de transacciones para calcular actividad
5. Determiná categoría y provincia frecuente para cada usuario
6. Analizá preferencias por categoría de antigüedad y pivoteá los resultados

Columnas requeridas:

- categoria_antigüedad
- categoría_frecuente (columnas pivoteadas)
- porcentajes por categoría

Guardar como: “preferencias_por_antigüedad.csv”

Ejercicio 15: Matriz de transición entre categorías de vivienda

Objetivo: Analizá cómo los usuarios cambian entre categorías de alojamiento en reservas consecutivas.

Pistas:

1. Identificá usuarios con múltiples reservas: `group_by(id_alquila)` y `filter(n() > 1)`
2. Ordená las reservas por usuario y fecha: `arrange(id_alquila, fecha_transaccion)`
3. Enumerá reservas por usuario con `row_number()`
4. Creá pares de reservas consecutivas usando `inner_join` y filtrando `num_reserva_siguiente == num_reserva + 1`
5. Calculá frecuencias de transición y porcentajes
6. Pivoteá para crear una matriz de transición

Columnas requeridas:

- categoria (actual)
- Columnas para cada categoría siguiente con porcentajes

Guardar como: “matriz_transicion_pivoteada.csv”

Ejercicio 16: Análisis de precios por metro cuadrado implícito

Objetivo: Estimá y analizá el precio por metro cuadrado según características de la propiedad.

Pistas:

1. Estimá el tamaño usando la fórmula: $20\text{m}^2 \text{ base} + (15\text{m}^2 * \text{habitaciones}) + (5\text{m}^2 * \text{baños})$
2. Calculá el precio por metro cuadrado: $\text{precio_base_noche} / \text{tamaño_estimado_m2}$
3. Agrupá por provincia y tipo de vivienda
4. Calculá promedios con **summarise**
5. Pivoteá por tipo de vivienda para comparar fácilmente

Columnas requeridas:

- provincia_ubicacion
- Columnas para cada tipo de vivienda con precio_m2_promedio

Guardar como: “precio_m2_pivoteado.csv”

Ejercicio 17: Calendario de ocupación mensual

Objetivo: Analizá patrones estacionales de ocupación por provincia.

Pistas:

1. Filtrá transacciones con fechas válidas
2. Expandí cada estancia a fechas individuales:
 - Usá **rowwise()** y **list(seq(fecha_checkin, fecha_checkout - 1, by = "day"))**
 - Luego **unnest()** para convertir las listas en filas
3. Extraé año y mes de cada fecha
4. Calculá ocupación mensual por propiedad considerando días del mes
5. Uní con datos de casas y agrupá por provincia y mes
6. Calculá índices de estacionalidad y variabilidad

Columnas requeridas:

- provincia_ubicacion
- mes_numero
- mes_nombre
- ocupacion_promedio
- ocupacion_anual_promedio
- indice_estacionalidad
- variabilidad

Guardar como: “estacionalidad_provincia.csv”

Ejercicio 18: Análisis de distribución de edades

Objetivo: Compará las distribuciones de edad entre huéspedes y anfitriones.

Pistas:

1. Combiná datos de usuarios_alquilan y usuarios_arrendatarios con `bind_rows`
2. Creá rangos de edad: “18-25”, “26-35”, “36-45”, “46-55”, “56-65”, “66+”
3. Calculá frecuencias por rango y tipo de usuario
4. Pivoteá para crear tabla comparativa con `pivot_wider`
5. Calculá diferencias entre porcentajes
6. Generá estadísticas descriptivas: media, mediana, moda, desviación estándar

Columnas requeridas para comparativa:

- rango_edad
- Huésped (porcentaje)
- Anfitrión (porcentaje)
- diferencia

Guardar como:

- “distribucion_edades.csv”
- “estadisticas_edad.csv”

Criterios de evaluación

Para cada ejercicio se va a evaluar:

- Correcta estructura y nombres de columnas
- Coincidencia en cantidad de filas y columnas
- Precisión en los valores calculados
- Correcta implementación de las técnicas solicitadas
- Guardado adecuado de los archivos resultantes

Recordá que es necesario cumplir con los nombres exactos de las columnas y archivos especificados para facilitar la evaluación automática.

Entrega

Una vez que hayas completado todos los ejercicios, asegurate de:

1. Guardar todos los archivos resultantes en la carpeta “resultados”
2. Verificar que incluiste tu información personal en la sección de preparación
3. Comprobar que el código se ejecuta sin errores
4. **Guardar tu archivo .R con codificación UTF-8**
5. Enviar el archivo .R a nsidicaro.fce@gmail.com