Apuntes

201_{200}¡Hola! Como tu profesor experto en **Análisis de Datos Masivos**, he analizado exhaustivamente todas las transcripciones de las sesiones. He destilado el contenido eliminando el ruido administrativo para entregarte esta **Guía Maestra de Estudio**.

Esta guía no sigue el orden cronológico de las clases, sino el **flujo lógico de un proyecto de datos**, priorizando lo que será evaluado en el examen (interpretación y programación en R).

📘 GUÍA MAESTRA DE ESTUDIO: Análisis de Datos con R

1. PREPARACIÓN Y LIMPIEZA DE DATOS (El cimiento)

Antes de modelar, debes entender y limpiar los datos. Si los datos son basura, el modelo será basura ("Garbage in, Garbage out").

Conceptos Clave

- **Tipos de Variables:**
  - **Numéricas (`int`, `num`):** Para cálculos matemáticos (medias, correlaciones).
    - **Factor (`factor`):** Crucial para variables categóricas (texto con opciones limitadas, ej: "Sí/No", "Rojo/Verde"). R asigna un código interno a cada categoría.
    - **Valores Nulos (`NA`):** Deben tratarse antes de modelar. Opciones: Eliminar la fila (si son pocos datos) o imputar (reemplazar por la media o cero).

Código Recurrente

```r
# Carga de datos
library(readxl)
datos <- read_excel("ruta/archivo.xlsx") # Para Excel
```

```r
datos <- read.csv("ruta/archivo.csv")    # Para CSV (más
        estándar)

# Exploración inicial (¡OBLIGATORIO!)
dim(datos)      # Dimensiones (filas, columnas)
str(datos)      # Estructura y tipos de datos
summary(datos)  # Resumen estadístico (Media, Mediana,
        Cuantiles)

# Tratamiento de Nulos
colSums(is.na(datos))       # Cuenta nulos por columna
datos_limpios <- na.omit(datos) # Elimina filas con nulos
        (Recomendado para examen si hay pocos)

# Conversión a Factor (Vital para Clasificación)
datos$columna_texto <- as.factor(datos$columna_texto)
```

---

2. ANÁLISIS DESCRIPTIVO Y RELACIONAL

Métricas Clave

- **Media vs. Mediana:** Si son muy diferentes, indica presencia de **Outliers** (valores anómalos). La mediana es más robusta a outliers.
- **Cuantiles:**
  - *1er Cuantil:* El 25% de los datos es menor o igual a este valor.
    - *3er Cuantil:* El 75% de los datos es menor o igual a este valor.
    - **Covarianza:** Indica el **signo** de la relación (positiva o negativa), pero no la fuerza.
    - **Correlación (Pearson):** Indica **fuerza** y signo. Va de -1 a 1.
      - `0`: Independientes.
        - `0 a 0.5`: Débil.
          - `0.5 a 1`: Fuerte.
            - *Nota:* **Correlación no implica causalidad**.

Código Recurrente

```r
# Solo con variables numéricas
cor(datos_num) # Matriz de correlación
cov(datos_num) # Matriz de covarianza

# Selección de numéricas (con dplyr)
library(dplyr)
datos_num <- select_if(datos,
             is.numeric)
```

---

3. APRENDIZAJE SUPERVISADO (Predicción y Clasificación)

Concepto: Tienes una columna "objetivo" (solución) y datos históricos. Buscas predecir ese valor. Requiere dividir en **Entrenamiento (Training)** y **Testeo (Test)**.

A. Regresión Lineal (`lm`)

- **Objetivo:** Predecir un número continuo (ej. Precio, Ventas).
- **Interpretación:**
  - **P-valor (Pr(>|t|)):** Si es **< 0.05**, la variable es **significativa** (importante). Si es mayor, se puede despreciar.
    - **R-cuadrado (R^2):** Qué porcentaje de la variabilidad explica el modelo (cerca de 1 es bueno).
      - **Coeficiente:** Cuánto aumenta la variable objetivo si aumenta en 1 la variable explicativa.

B. Clasificación (Logística y Árboles)

- **Objetivo:** Predecir una categoría (Sí/No, 0/1).
- **Regresión Logística (`glm`):** Solo para 2 categorías (Binomial). Da una probabilidad. Corte habitual en 0.5.
- **Árboles de Decisión (`rpart`):** Reglas visuales. Sirve para más de 2 categorías.
- **Validación (Matriz de Confusión):**
  - **Accuracy:** % de aciertos totales.
    - **Sensibilidad:** Capacidad de detectar Positivos (ej. enfermos).
      - **Especificidad:** Capacidad de detectar Negativos (ej. sanos).

Código Recurrente

```r
# División Train/
        Test
        (Crucial)
library(caret)
set.seed(123) # Para
        reproducibilidad
indice <-
        createDataPartition(datos$V
        p=0.8, list=FALSE)
train <-
        datos[indice, ]
test <- datos[-indice, ]

# Regresión Lineal
modelo_lm <- lm(Y
        ~ ., data
        = train)
summary(modelo_lm)
        # Para ver
        p-valores y
        R2

# Regresión
        Logística
modelo_glm <-
        glm(Y ~ .,
        data =
        train,
        family =
        "binomial")
prediccion_prob <-
        predict(modelo_glm,
        newdata = test,
        type = "response")
prediccion_clase <-
        ifelse(prediccion_prob
        > 0.5, 1, 0) #
        Convertir a 0/1

# Árbol de Decisión
library(rpart);
        library(rpart.plot)
modelo_arbol <-
        rpart(Y
        ~ ., data
        = train,
        method =
        "class")
rpart.plot(modelo_arbol)
        # Dibujo del
        árbol
```

```
# Matriz de
        Confusión
confusionMatrix(as.factor(prediccio
        as.factor(test$Y))
```

---

4. APRENDIZAJE NO SUPERVISADO (Agrupación)

Concepto: No hay columna objetivo (no hay solución previa). Buscas patrones o grupos. **NO** se divide en Train/Test.

Clustering (K-Means)

- **Objetivo:** Agrupar datos por similitud.
- **Elección de K (Número de grupos):**
  - **Método del Codo:** Donde la curva dobla.
    - **Función `nbclust`:** Regla de la mayoría (Recomendada).
  - **Interpretación:** Usar un Árbol de Decisión sobre el resultado del clúster para entender las características de cada

grupo (el “truco”).

Código Recurrente

```r
library(factoextra); library(NbClust)
# Calcular número óptimo
NbClust(datos_num, min.nc=2, max.nc=8, method="kmeans")

# Ejecutar K-Means
modelo_km <- kmeans(datos_num, centers = 2) # Si elegimos 2
datos$cluster <- modelo_km$cluster # Guardar resultado
```

---

5. SERIES TEMPORALES

Concepto: Datos ordenados cronológicamente. Importante definir la **frecuencia**.

Componentes y Predicción

- **Componentes:** Tendencia, Ciclo, Estacionalidad, Ruido.

- **Modelo:** Usamos `auto.arima` para que ajuste automáticamente.
- **Predicción:** Usamos `forecast`.

Código Recurrente

```
library(forecast)
# Crear objeto serie temporal (ej. mensual frecuencia 12)
serie <- ts(datos$valor, start=c(2000,1), frequency=12)

# Modelo y Predicción
modelo_arima <- auto.arima(serie)
prediccion <- forecast(modelo_arima, h=12) # Predecir 12 periodos
plot(prediccion)
```

LIBRERÍAS ESENCIALES

Librería	Función Principal
readxl	read_excel
readr	read_csv

Librería	Función Principal
dplyr	select, filter, %>
caret	createDataPartit confusionMatrix
rpart	rpart
rpart.plot	rpart.plot
factoextra	fviz_nbclust
NbClust	NbClust
forecast	auto.arima, forec
ggplot2	ggplot

🔥

PREGUNTAS DE EXAMEN PROBABLES

(Basadas en el énfasis repetitivo del profesor durante las sesiones)

1. **Interpretación de Regresión:**
   - *"Dado el siguiente `summary`, ¿es significativa la variable precio?"*
     - **Respuesta:** Mirar el **p-valor**. Si es < 0.05, SÍ es significativa. Si tiene asteriscos

(***), es muy significativa.

1. **Diferencia Supervisado vs. No Supervisado:**
   - *“¿Qué algoritmos usarías para segmen*

tarclientesinconocergruposprevios? ¿Yparapredecirsiunc

lienteabandonará (Sí/No)?"

- **Respuesta:** Segmentar=Clusterin

g(Nosupervisado).Abandono=Clasificación/Logística(Supe

r v i s a d o) .

1. **M a t r i z d e C o n f u s i ó n :**

- *“ I n t e r p r e t a l a S e n s i b i l i d a d o c*

alcula el Accuracy.”

- **Respuesta:** Accuracy = (Aciertos / Total).

Sensibilidad = Capacidad de detectar positivos (recordar reje

plotTestCOVID).

1. **Elección de Ken Clustering:**
   - *"Según la salidad de*

NbClust, formar?"