Resumen: Andrés Venegas

Actualizado el: 22/11/2022

Resumen PPT's

PPT 5: dplyr **#Si entró**

Tiene funciones para trabajar con bases de datos, como arrange, select, rename, distinct, summarise, sample_n (para hacer filas aleatorias) mutate.

Mutate:

Mutate se utiliza para agregarle columnas a una base de datos llamada $Data$, y se hace de la siguiente forma:

$$nueva\_data = mutate(Data, NOMBRE\_COLUMNA\_NUEVA = COL\_1 - COL\_2)$$

En el caso anterior, a la base de datos $Data$, se le agrego una nueva columna que tiene los valores de COL_1 restandole COL_2. Luego, si se quiere agregar otra columna cambiándole los valores a la que ya creamos recién, podemos hacer:

$$nueva\_data\_2 = mutate(nueva\_data, NOMBRE\_NUEVA\_COLUMNA = ifelse(condicion, SI, ifelse(\dots)))$$

Esto permite generar condiciones para ir agregando los valores. En condición se debe poner el nombre de alguna columna, que vendría siendo el valor que se quiere comparar.

Ejemplo:

$$datos.filtrados.mutate = mutate(copia\_mutada, GRUPO\_PODER\_NETO = ifelse(Poder\_neto < 10, 1, ifelse(Poder\_neto < 20, 2, ifelse(Poder\_neto < 30, 3, 4))))$$

En el código anterior, mutamos la base de datos copia_mutada, le agregamos la columna llamada GRUPO_PODER_NETO, y en base a el valor de la columna Poder_neto, fuimos colocándole valores al código del ifelse. La idea de el ifelse es poner una condición. Si esa condición se cumple, entonces se hace lo que viene después de la primera coma (","). En caso de que no se cumpla la condición, entonces ocurrirá lo que viene después de la segunda coma. Y la gracia es poner una cadena de ifelse dentro de los ifelse. Funciona de la siguiente manera:

$$ifelse(condicion, SI\ OCURRE, NO\ OCURRE)$$

Y la idea es colocar otro ifelse dentro del ifelse, justo en donde dice "NO OCURRE".

Obs: el pipe %>% se usa para concatenar y no tener que poner adentro del paréntesis el nombre de la variable a trabajar. Forma rápida → Ctr+shift+m

PPT 6: Funciones de distribución marginales, condicionales, conjunta. #Si entró

A modo general, considerar que se tiene una base de datos grande. De esa base de datos, trabajaremos con dos datos. Los X y los Y que los podemos asignar como:

$$Y = base\_datos\$Edad$$

$$X = Base\_datos\$Peso$$

Luego, se debe formar una tabla de datos para poder hacer el calculo de las probabilidades, haciendo:

$$Tabla = table(X, Y)$$

Ahora, veamos como obtener todos los tipos de probabilidades.

- **Probabilidad conjunta:**

Para hacer la tabla de frecuencias relativas debemos hacer prop.table así:

$$Prob\_conjunta = prop.table(Tabla)$$

- **Probabilidades marginales:**

Si se quiere saber la probabilidad de X solamente, si los valores de X están en las filas, solamente debemos sumar las probabilidades de las filas haciendo

$$f_x = rowSums(Prob\_conjunta)$$

Ahora bien, si la probabilidad de X está en las columnas, debemos hacer:

$$f_x = colSums(Prob\_conjunta)$$

- **Probabilidades condicionales:**

Si piden la probabilidad condicional de X dado Y, como esto significa X|Y, solamente debemos fijarnos, como nemotecnia, qué variable es la que "divide" en la condición. En este caso es Y. Luego, debemos ver si Y está en las filas o en las columnas. En el caso de que esté en las filas, debemos hacer (margin 1):

$$Prob\_X\_dado\_Y = prop.table(tabla, margin = 1)$$

Y en el caso de que esté en las columnas, debemos hacer (margin = 2):

$$Prob\_X\_dado\_Y = prop.table(tabla, margin = 2)$$

Aquí es importante notar que las probabilidades de la fila o columna en donde se encuentra Y, deben sumar 1.

- **Covarianza y Correlación**

Se calculan utilizando el comando:

$$Covarianza = cov(x, y)$$

$$Correlación = cor(x, y)$$

Donde X e Y deben ser un vector de datos.

- **Distribución normal Bivariada:**

No entra seguramente. Se debe usar la librería **mvtnorm,** tener un vector de las medias, una matriz de las desviaciones estándar y luego utilizar el siguiente comando para calcular la probabilidad de una conjunta entre datos continuos, por ejemplo $P(0 < X < 4, -1 < Y < 1)$,:

$$Pmvnorm(lower = c(0,-1), upper = c(4,1), mean = mu\_vector, sigma = sigma\_matriz)\ [1]$$

PPT 7: Método → Gráfico de probabilidades para estimar parámetros #Si entró

- **Función** $lm(Y \sim X)$

En este lab se aprendió a utilizar la función lm(Y ~ X). En estos ejercicios nos van a pedir estimar los parametros de una distribución a partir de los datos.Para ello, es importante hacer lo siguiente:

1. Identificar la distribución solicitada
2. Encontrar ecuación de la recta de la distribución en el ppt
3. Identificar los parametros
4. Tener los datos
5. Ordenarlos de menor a mayor
6. Generar probabilidades acumuladas
7. Identificar Y de la ecuación de la recta(es el dato)
8. Identificar X de la ecuación de la ecuación de la recta (son las probabilidades)
9. Utilizar función: modelo = lm(Y~X)
10. Encontrar intercepto y pendiente haciendo modelo$coefficients, que corresponden a los estimadores.
11. Recomendable asignar esos valores a variables, porque después piden calcular la probabilidad.

Todo lo anterior se ve reflejado en el siguiente código, donde datos son los datos a interés, como la altura de las personas.

```
Y = Datos
Y=sort(Y) #Ordenar
N=length(Y) #Saber el largo
m=1:N#generar enteros
x=m/(N+1) #generar probabilidad
modelo = lm(y~g(x))#g es la función de cada ecuación
fit=modelo$coef
estimador_mu =fit[1]
estimador_sigma=fit[2]
```

Ojo: los valores de la ecuación de la recta dependen de cada distribución! Es decir, se elige una ecuación distinta para normales, lognormales, Weibull, exponencial, exponencial trasladada, uniforme, etc. No son siempre iguales.

Tip: a veces hay que aplicar logaritmo, pero como ley, utilizar la formula tal cual como nos la dan! Y creerle… Solamente poner logaritmo cuando piden calcular la probabilidad de una loglogistica. (plogis(log(x)…))

PPT 8: Teorema del límite central y Estadísticos de Orden.

- **Teorema del limite central:** **#No entró**

Hay 3 formas de aproximar a una distribución normal. A continuación, entender que el primer dato es cómo distribuyen los X_i, el segundo dato es cómo distribuye la suma de los Xi y el tercer dato es cómo distribuye la aproximación al usar teorema del limite central.

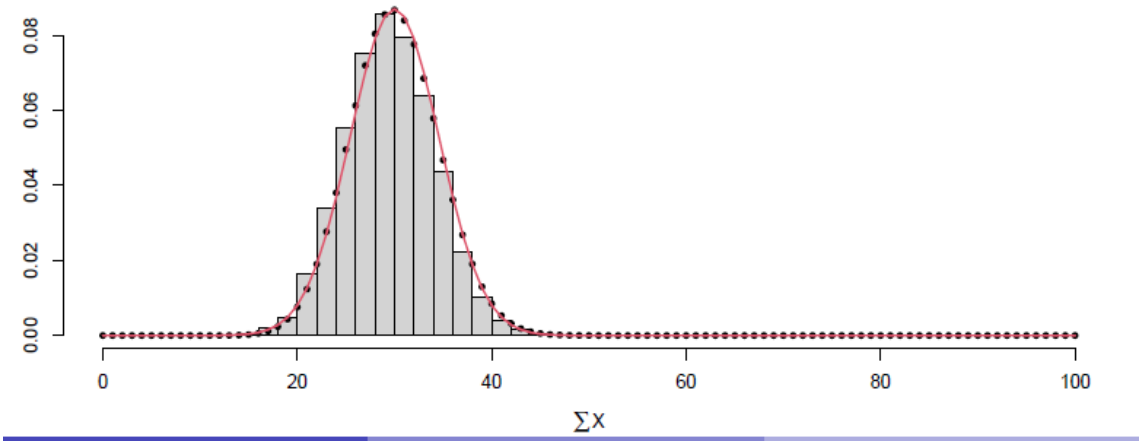
1. $Bernulli(p) \rightarrow Binomial(n,p) \rightarrow Normal(np, \sqrt{np(1-p)})$
2. $Exponencial(v) \rightarrow Gamma(n,v) \rightarrow Normal(\frac{n}{v}, \frac{\sqrt{n}}{v})$
3. $Poisson(\lambda) \rightarrow Poisson(n\lambda) \rightarrow Normal(n\lambda, \sqrt{n\lambda})$
4. $Uniforme(0,1) \rightarrow Normal(n\mu, \sigma\sqrt{n}) \rightarrow Normal(\frac{n}{2}, sqrt\left(\frac{n}{12}\right))$

Si pidieran hacer un ejercicio de esto, se tienen que simular muchas variables aleatorias usando "rDistr", luego se tiene que graficar, y se debe observar la curva obtenida y ver si se parece a una normal. (NUNCA HE VISTO EJERCICIOS DE ESTO)

```
N <- 10000 # Cantidad de sumas simuladas (variables aleatorias a crear)
n <- 100 # Tamaño de cada suma (parámetro de una binomial)
p <- 0.3 #Parametro de una binomial

# Suma de Bernoulli(p) es Binomial (distribución del segundo paso)
# Entonces hay que simular una binomial con los parámetros n y p, N veces.
sumaX <- rbinom(N, size = n, prob = p)

# Luego se grafica la simulación
hist(sumaX, freq = F, xlim = c(0,100), main = "", xlab = expression(sum(X)))
points(0:n, dbinom(0:n, size = n, prob = p), pch = 20)
#Generamos la curva aproximada como una normal.
curve(dnorm(x, mean = n*p, sd = sqrt(n*p*(1-p))),add = TRUE, lwd = 2, col = 2)
```



- **Estadísticos de orden** **#No entró**

Para estos ejercicios, hay que saber identificar si la variable aleatoria es un mínimo (Y_1) o un máximo (Y_n). Además, solamente hay que saber usar las siguientes fórmulas.

Primero, hay que considerar que:

$$Y_n = \max\{X_1, \dots, X_n\} \quad y\ que \quad Y_1 = \min\{X_1, \dots, X_n\}$$

Con ello, sabemos que las funciones de densidad y de probabilidad acumulada son:

Valor mínimo:

$$f_{Y_1}(y) = n \cdot [1 - F_x(y)]^{n-1} \cdot fx(y)$$

$$F_{Y_1}(y) = 1 - (1 - [F_x(y)])^n$$

Valor máximo:

$$f_{Y_n}(y) = n \cdot [F_x(y)]^{n-1} \cdot fx(y)$$

$$F_{Y_n}(y) = [F_x(y)]^n$$

En general, solo hay que reemplazar en la acumulada los valores que nos entregan. Por lo tanto, solamente hay que:

1. Identificar la distribución de los X_i
2. Ver el tamaño de la muestra n, (cantidad de X_i)
3. Usar la acumulada de la X_i para calcular $F_x(y)$
4. Reemplazar valores y calcular la probabilidad solicitada para el valor de y

Ejemplo: Si los $X_i \sim Binomial$, con $n = 22$, $p = 0{,}6$ y Tamaño muestra $N = 6$, entonces calcular la probabilidad de que el máximo sea menor a 20. Por lo tanto, vamos a usar la última formula.

$$Probabilidad\_pedida = pbinom(20, 22, 0.6)\text{^}6$$

PPT 9: Estimar parámetros **#Si entró**

Para estos ejercicios usaremos la forma más fácil que es usando el comando **fitdist** de la librería **fitdstrplus.** Para empezar, siempre se deben tener datos almacenados en alguna variable. Estos deben ser como un "vector" de datos. Es decir, solo una fila o solo una columna, por ejemplo, queremos estimar los parámetros de la variable aleatoria Altura, entonces solo necesitamos los valores de las alturas de la muestra.

- **Método de Momentos**

Para empezar, se necesita importar la librería:

$$library(fitdistrplus)$$

Luego se utiliza el siguiente comando:

$$EM <- fitdist(data = DATOS, distr = \text{"NOMBRE\_DISTR"}, \text{method} = \text{"}mme\text{"})$$

Luego, para acceder a los parámetros estimados hay que hacer:

$$EM\$estimate$$

- **Método de Máxima Verosimilitud**

Solamente hay que usar el siguiente comando que guarda los parámetros estimados en EMV

$$EMV <- fitdist(data = DATOS, distr = \text{"NOMBRE\_DISTR"}, \text{method} = \text{"}mle\text{"})$$

De forma general, $DATOS$ son los datos a analizar (como la altura de las personas), "$NOMBRE\_DISTR$" (va entre comillas) es el nombre de la distribución que se utilizará para estimar y la dan en el enunciado de la pregunta. Finalmente, $method$ sirve para indicar qué método vamos a utilizar, y también va entre comillas.

Obs: la forma alternativa de hacer todo esto, es usando las formulas de las clases, pero es más tedioso.

Tip: es importante saber qué parámetro es qué parámetro. Por ejemplo, en R, la variable *lambda* se llama *meanlog*, y chi o zeta se llama *sdlog* en una distribución lognormal.

PPT 10: Test de Hipotesis #Si entró

Para estos ejercicios, se podrán hacer tests para los siguientes casos:

- Test para la media con μ desconocido y σ conocido → usar **z.test**
- Test para la media con μ desconocido y σ desconocido → usar **t.test**
- Test para la desviación estándar, con μ desconocido y σ desconocido → usar **sigma.test**
- Test para comparar varianzas → usar **var.test**
- Test para comparar medias (requiere usar el test anterior también) → usar **t.test** de 2 variables.
- #**prop.test** dudo que entre

En estos ejercicios nos preguntan sobre un valor de un parámetro (μ, σ, etc), y nosotros debemos ver si ese valor es válido o no usando los test de hipótesis. Además, hay que validar o rechazar la hipótesis.

También, en los ejercicios de test de hipótesis nos van a pedir 3 cosas:

1. Estimador
2. Valor p
3. Rechazar o no alguna de las hipótesis (la que mencionan en el enunciado)

Los ejercicios se resuelven de la siguiente manera:

Y ojooo: no le tengan miedo, es casi que solo 1 línea de código :D Y salen todos los códigos en el ppt (es casi que copiar y pegar nada más).

Entonces, para resolverlos se debe:

1. Identificar el test a utilizar
2. Identificar el valor del parámetro "inicial". (μ_0, σ_0, etc)
3. Escribir el valor de α que se pide en el enunciado
4. Identificar hipotesis nula y alternativa (H0 y Ha)
5. En base a la hipotesis alternativa, elegir el código a utilizar. Hay 3 opciones para cada test, por ejemplo, para el z.test se tiene:

H_a	Código
$\mu \neq \mu_0$	`z.test(X, mu=mu_0, sd=sigma, alternative="two.sided", conf.level=1-`α`)`
$\mu > \mu_0$	`z.test(X, mu=mu_0, sd=sigma, alternative="greater", conf.level=1-`α`)`
$\mu < \mu_0$	`z.test(X, mu=mu_0, sd=sigma, alternative="less", conf.level=1-`α`)`

6. Ejecutar el código que sale en el ppt
7. Analizar el resultado y buscar las 3 cosas que piden, lo cual se hace así:

```r
alpha <- 0.05
mu0 <- 60
# H_0: mu=mu0 vs H_a: mu!=mu0
z.test(X, mu=mu0, sd=sigma, alternative = "two.sided", conf.level = 1-alpha)
```

```
##
##  One Sample z-test
##
## data:  X
## z = -13.2, n = 1000.00000, Std. Dev. = 11.00000
## Std. Dev. of the sample mean = 0.34785, p-value
## < 2.2e-16
## alternative hypothesis: true mean is not equal to 60
## 95 percent confidence interval:
##  54.72644 56.08999
## sample estimates:
## mean of X
##  55.40822
```

```r
# Como p-value < alpha rechazo H_0
# Concluyo que mi media es distinta a 60
```

Es decir, primero sabemos que el **estimador** es z, y su valor es lo encerrado en color azul. Luego, queremos el **valor p,** que en este caso está encerrado en rojo. Finalmente, hay que concluir diciendo que:

- Si $valor\ p < \alpha$, entonces se rechaza la hipotesis nula (y se afirma la hipotesis alternativa)
- Si $valor\ p > \alpha$, entonces, no se rechaza la hipotesis nula (es decir, se confirma la hipotesis nula y se rechaza la hipotesis alternativa)

Tip: en general, preguntan por la **hipotesis alternativa**, por lo tanto, cuando busquen qué código utilizar, guíense por la hipotesis alternativa que les dieron. Por ejemplo, si dice que $\mu < \mu_0$, entonces deberíamos utilizar la tercera opción.

H_a	Código
$\mu \neq \mu_0$	`z.test(X, mu=mu_0, sd=sigma, alternative="two.sided", conf.level=1-`α`)`
$\mu > \mu_0$	`z.test(X, mu=mu_0, sd=sigma, alternative="greater", conf.level=1-`α`)`
$\mu < \mu_0$	`z.test(X, mu=mu_0, sd=sigma, alternative="less", conf.level=1-`α`)`

También, mencionar que la variable **X** son los datos que se van a utilizar, pero se debe usar solamente la columna que piden. Por ejemplo, si piden trabajar con las edades de la base de datos, tienen que hacer:

$$X = base\_de\_datos\$edades$$

Lo único extra es que para realizar el test de comparación de medias, se debe realizar el test de varianzas antes, para saber si las varianzas son iguales. En caso de que lo sean, en el test de comparación de medias se debe poner **var.equal = TRUE**. En el caso de que sean distintas, se pone **var.equal = FALSE.** Tal como se ve aquí:

H_a	Código
$\mu_1 \neq \mu_2$	`t.test(x, y, var.equal = TRUE, alternative = "two.sided", mu=0)`
$\mu_1 > \mu_2$	`t.test(x, y, var.equal = TRUE, alternative = "greater", mu=0)`
$\mu_1 < \mu_2$	`t.test(x, y, var.equal = TRUE, alternative = "less", mu=0)`

Andrew Venegas

LIBRERIAS:

- **Mvtnorm** → para calcular probabilidad de normal bivariada. #No importa
- **Rio** → Para importar
- **Dplyr** → para filtrar y mutar
- **TeachingDemos** → Para hacer tests de hipótesis z, t, chi, sigma, var, t. Obtener valor p y obtener valor del estimador.
- **fitdistrplus** → Para estimar parámetros por método de momentos o por método de máxima verosimilitud usando función fitdist()
- función **lm()** → para estimar parámetros según sus ecuaciones de la recta (intercepto y pendiente)

¿Cómo filtrar?

Usar librería dplyr con la función filter o usar subset de R. Ambas funcionan así:

$$Filter(data, nombre\_columna == "valor")$$

$$Filter(data, nombre\_columna == c(valor\ 1, valor\ 2, \dots, valor\ n))$$

$$Subset(data, nombre\_columna == "valor")$$

$$subset(data, nombre\_columna == c(valor\ 1, valor\ 2, \dots, valor\ n))$$

O si se quiere solamente un dato:

$$Tabla[valor\_fila, valor\_columna]$$

$$Tabla[c(valores\_fila), c(valores\_columna)]$$

#Tip: Si piden la probabiliadad conjunta de varias cosas, se puede filtrar por vectores haciendo:

$$Prob\_conjunta[c(valor1, valor2, valor\_n),\ \ c(valor\_1, valor\_2, \dots, valor\_n)]$$

Tips extras:

- Tener todos los ppts abiertos
- Saber qué hay en cada ppt
- Llegar temprano
- Saber filtrar (SI O SI)
- Saber agregar columnas a una base de datos (SI o SI)
- Si piden estimar, es posible hacerlo con:
    - método de grafico de probabilidades (PPT 7)
    - método de momentos (PPT 9)
    - método de máxima verosimilitud (PPT 9)