Formulario Evaluación Final

Principio de la Multiplicación

Si un experimento está compuesto de k experimentos con tamaños muestrales $n_1, \ldots, n_k$, entonces

$$\# S = n_1 \times n_2 \times \cdots \times n_k.$$

Permutación

Muestras de tamaño r sin reemplazo e importando el orden de ingreso:

$$n \times (n-1) \times (n-2) \times \cdots \times (n-r+1) = \frac{n!}{(n-r)!} \quad \text{(Botón } \textbf{nPr} \text{ de la caluladora)}$$

Combinación

Muestras de tamaño r sin reemplazo y sin importar el orden de ingreso: $\binom{n}{r} = \dfrac{n!}{r! \times (n-r)!}$ (Botón **nCr**)

Ordenamiento Multinomial

Queremos asignar n objetos a k grupos distintos de tamaños $n_1, \ldots, n_k$. El número de grupos distintos con las características dadas son

$$\binom{n}{n_1\, n_2\, \cdots\, n_k} = \frac{n!}{n_1! \times \cdots \times n_k!} \quad \text{con} \sum_{i=1}^{k} n_i = n$$

Igualdades

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^k\, b^{n-k}; \qquad \sum_{k=x}^{\infty} \phi^k = \frac{\phi^x}{1-\phi} \quad \text{si } |\phi| < 1;$$

$$\sum_{k=0}^{\infty} \frac{\lambda^k}{k!} = \exp(\lambda); \qquad \sum_{x=0}^{\infty} \binom{x+k-1}{k-1} \phi^x = \frac{1}{(1-\phi)^k} \quad \text{si } 0 < \phi < 1 \text{ y } k \in \mathbb{N}$$

Propiedades función $\Gamma(\cdot)$ y $B(\cdot, \cdot)$

$$(1) \quad \Gamma(k) = \int_0^{\infty} u^{k-1}\, e^{-u}\, du; \quad (2) \quad \Gamma(a+1) = a\,\Gamma(a); \quad (3) \quad \Gamma(n+1) = n!, \quad \text{si } n \in \mathbb{N}_0;$$

$$(4) \quad \Gamma(1/2) = \sqrt{\pi}; \quad (5) \quad B(q,\, r) = \int_0^1 x^{q-1}\,(1-x)^{r-1}\, dx; \quad (6) \quad B(q,\, r) = \frac{\Gamma(q)\,\Gamma(r)}{\Gamma(q+r)}$$

Distribución Gamma

$$(1) \quad \text{Si } T \sim \text{Gamma}(k,\, \nu), \text{ con } k \in \mathbb{N} \longrightarrow F_T(t) = 1 - \sum_{x=0}^{k-1} \frac{(\nu\, t)^x\, e^{-\nu\, t}}{x!}$$

$$(2) \quad \text{Gamma}(1,\, \nu) = \text{Exp}(\nu) \qquad (3) \quad \text{Gamma}(\eta/2,\, 1/2) = \chi^2(\eta)$$

Medidas descriptivas

$$\mu_X = \text{E}(X), \quad \sigma_X^2 = \text{E}\left[(X-\mu_X)^2\right], \quad \delta_X = \frac{\sigma_X}{\mu_X}, \quad \theta_X = \frac{\text{E}\left[(X-\mu_X)^3\right]}{\sigma_X^3}, \quad K_X = \frac{\text{E}\left[(X-\mu_X)^4\right]}{\sigma_X^4} - 3$$

$$M_X(t) = \text{E}\left(e^{t\,X}\right), \quad \text{E}[g(X)] = \begin{cases} \displaystyle\sum_{x \in \Theta_X} g(x) \cdot p_X(x) \\ \displaystyle\int_{-\infty}^{\infty} g(x) \cdot f_X(x)\, dx \end{cases}, \quad \text{Rango} = \text{máx} - \text{mín}, \quad \text{IQR} = x_{75\,\%} - x_{25\,\%}$$

$$\text{Cov}(X,\, Y) = \text{E}[(X-\mu_X)\cdot(Y-\mu_Y)] = \text{E}(X\cdot Y) - \text{E}(X)\cdot\text{E}(Y) \qquad , \qquad \rho = \frac{\text{Cov}(X, Y)}{\sigma_X \cdot \sigma_Y}$$

Teorema de Probabilidades Totales

$$p_Y(y) = \sum_{x \in \Theta_X} p_{X,Y}(x,y); \qquad f_X(x) = \int_{-\infty}^{+\infty} f_{X,Y}(x,y)\,dy$$

$$p_X(x) = \int_{-\infty}^{+\infty} p_{X\,|\,Y=y}(x) \cdot f_Y(y)\,dy; \qquad f_Y(y) = \sum_{x \in \Theta_X} f_{Y\,|\,X=x}(y) \cdot p_X(x)$$

$$\mathrm{E}(X) = \int_{-\infty}^{+\infty} \mathrm{E}(X \,|\, Y = y) \cdot f_Y(y)\,dy \qquad \mathrm{E}(Y) = \sum_{x \in \Theta_X} \mathrm{E}(X \,|\, X = x) \cdot p_Y(y)$$

Esperanza y Varianza Condicional

$$\mathrm{E}(Y) = \mathrm{E}[\mathrm{E}(Y \,|\, X)] \quad \text{y} \quad \mathrm{Var}(Y) = \mathrm{Var}[\mathrm{E}(Y \,|\, X)] + \mathrm{E}[\mathrm{Var}(Y \,|\, X)]$$

Transformación

Sea $Y = g(X)$ una función cualquiera, con k raíces:

$$f_Y(y) = \sum_{i=1}^{k} f_X\left(g_i^{-1}(y)\right) \cdot \left| \frac{d}{dy}\, g_i^{-1}(y) \right| \quad \text{o} \quad p_Y(y) = \sum_{i=1}^{k} p_X\left(g_i^{-1}(y)\right)$$

Sea $Z = g(X, Y)$ una función cualquiera:

$$p_Z(z) = \sum_{g(x,y)=z} p_{X,Y}(x,y)$$

Sea $Z = g(X, Y)$ una función invertible para X o Y fijo:

$$f_Z(z) = \int_{-\infty}^{\infty} f_{X,Y}(g^{-1}, y) \left| \frac{\partial}{\partial z} g^{-1} \right| dy = \int_{-\infty}^{\infty} f_{X,Y}(x, g^{-1}) \left| \frac{\partial}{\partial z} g^{-1} \right| dx$$

Distribución Normal Bivariada

$$f_{X,Y}(x,y) = \frac{1}{2\,\pi\,\sigma_X\,\sigma_Y\,\sqrt{1-\rho^2}} \times \exp\left\{ -\frac{1}{2(1-\rho^2)} \left[\left(\frac{x-\mu_X}{\sigma_X}\right)^2 + \left(\frac{y-\mu_Y}{\sigma_Y}\right)^2 - 2\,\rho \left(\frac{x-\mu_X}{\sigma_X}\right) \left(\frac{y-\mu_Y}{\sigma_Y}\right) \right] \right\}$$

$$Y \,|\, X = x \sim \text{Normal}\left(\mu_Y + \frac{\rho\,\sigma_Y}{\sigma_X}\,(x - \mu_X),\, \sigma_Y\,\sqrt{(1-\rho^2)} \right)$$

$$X \sim \text{Normal}(\mu_X,\, \sigma_X) \qquad \text{e} \qquad Y \sim \text{Normal}(\mu_Y,\, \sigma_Y)$$

Teorema del Límite Central

Sean $X_1, \ldots, X_n$ variables aleatorias independientes e idénticamente distribuidas, entonces

$$Z_n = \frac{\displaystyle\sum_{i=1}^{n} X_i - n \cdot \mu}{\sqrt{n}\,\sigma} = \frac{\overline{X}_n - \mu}{\sigma/\sqrt{n}} \longrightarrow Z \sim \text{Normal}(0,1),$$

cuando $n \to \infty$, $\mathrm{E}(X_i) = \mu$ y $\mathrm{Var}(X_i) = \sigma^2$.

Distribución	Densidad de Probabilidad	Θ_X	Parámetros	Esperanza y Varianza
Binomial	$\binom{n}{x}\, p^x\,(1-p)^{n-x}$	$x = 0, \ldots, n$	$n,\, p$	$\mu_X = n\,p$ $\sigma_X^2 = n\,p\,(1-p)$ $M(t) = [p\,e^t + (1-p)]^n, \quad t \in \mathbb{R}$
Geométrica	$p\,(1-p)^{x-1}$	$x = 1, 2, \ldots$	p	$\mu_X = 1/p$ $\sigma_X^2 = (1-p)/p^2$ $M(t) = p\,e^t/[1-(1-p)\,e^t], \quad t < -\ln(1-p)$
Binomial-Negativa	$\binom{x-1}{r-1}\, p^r\,(1-p)^{x-r}$	$x = r, r+1, \ldots$	$r,\, p$	$\mu_X = r/p$ $\sigma_X^2 = r\,(1-p)/p^2$ $M(t) = \left\{p\,e^t/[1-(1-p)\,e^t]\right\}^r, \quad t < -\ln(1-p)$
Poisson	$\dfrac{(\nu\,t)^x\,e^{-\nu\,t}}{x!}$	$x = 0, 1, \ldots$	ν	$\mu_X = \nu\,t$ $\sigma_X^2 = \nu\,t$ $M(t) = \exp\left[\lambda\left(e^t - 1\right)\right], \quad t \in \mathbb{R}$
Exponencial	$\nu\,e^{-\nu\,x}$	$x \geq 0$	ν	$\mu_X = 1/\nu$ $\sigma_X^2 = 1/\nu^2$ $M(t) = \nu/(\nu - t), \quad t < \nu$
Gamma	$\dfrac{\nu^k}{\Gamma(k)}\, x^{k-1}\, e^{-\nu\,x}$	$x \geq 0$	$k,\, \nu$	$\mu_X = k/\nu$ $\sigma_X^2 = k/\nu^2$ $M(t) = [\nu/(\nu - t)]^k, \quad t < \nu$
Normal	$\dfrac{1}{\sqrt{2\,\pi}\,\sigma}\, \exp\left[-\dfrac{1}{2}\left(\dfrac{x-\mu}{\sigma}\right)^2\right]$	$-\infty < x < \infty$	$\mu,\, \sigma$	$\mu_X = \mu$ $\sigma_X^2 = \sigma^2$ $M(t) = \exp(\mu\,t + \sigma^2\,t^2/2), \quad t \in \mathbb{R}$
Log-Normal	$\dfrac{1}{\sqrt{2\,\pi}\,(\zeta\,x)}\, \exp\left[-\dfrac{1}{2}\left(\dfrac{\ln\,x-\lambda}{\zeta}\right)^2\right]$	$x \geq 0$	$\lambda,\, \zeta$	$\mu_X = \exp\left(\lambda + \dfrac{1}{2}\,\zeta^2\right)$ $\sigma_X^2 = \mu_X^2\left(e^{\zeta^2} - 1\right)$ $E(X^r) = e^{r\,\lambda}\,M_Z(r\,\zeta)$, con $Z \sim$Normal(0,1)
Uniforme	$\dfrac{1}{(b-a)}$	$a \leq x \leq b$	$a,\, b$	$\mu_X = (a+b)/2$ $\sigma_X^2 = (b-a)^2/12$ $M(t) = [e^{t\,b} - e^{t\,a}]/[t\,(b-a)], \quad t \in \mathbb{R}$
Beta	$\dfrac{1}{B(q,\,r)}\,\dfrac{(x-a)^{q-1}\,(b-x)^{r-1}}{(b-a)^{q+r-1}}$	$a \leq x \leq b$	$q,\, r$	$\mu_X = a + \frac{q}{q+r}\,(b-a)$ $\sigma_X^2 = \frac{q\,r\,(b-a)^2}{(q+r)^2\,(q+r+1)}$
Hipergeométrica	$\dfrac{\binom{m}{x}\binom{N-m}{n-x}}{\binom{N}{n}}$	$\text{máx}\{0, n+m-N\} \leq x \leq \text{mín}\{n, m\}$	$N,\, m,\, n$	$\mu_X = n\,\frac{m}{N}$ $\sigma_X^2 = \left(\frac{N-n}{N-1}\right)\, n\,\frac{m}{N}\,\left(1 - \frac{m}{N}\right)$

Otras distribuciones

- Si $T \sim$ Weibull(η, β), se tiene que

$$F_T(t) = 1 - \exp\left[-\left(\frac{t}{\eta}\right)^{\beta}\right] \quad f_T(t) = \frac{\beta}{\eta}\left(\frac{t}{\eta}\right)^{\beta-1}\exp\left[-\left(\frac{t}{\eta}\right)^{\beta}\right], \quad t > 0$$

  Con $\beta > 0$, es un parámetro de forma y $\eta > 0$, es un parámetro de escala. Si t_p es el percentil $p \times 100\,\%$, entonces

$$\ln(t_p) = \ln(\eta) + \frac{1}{\beta}\cdot\Phi^{-1}_{\text{Weibull}}(p), \quad \Phi^{-1}_{\text{Weibull}}(p) = \ln[-\ln(1-p)]$$

  Mientras que su m-ésimo momento está dado por

$$E(T^m) = \eta^m\,\Gamma(1 + m/\beta)$$

$$\mu_T = \eta\,\Gamma\left(1 + \frac{1}{\beta}\right), \quad \sigma^2_T = \eta^2\left[\Gamma\left(1 + \frac{2}{\beta}\right) - \Gamma^2\left(1 + \frac{1}{\beta}\right)\right]$$

- Si $Y \sim$ Logística(μ, σ), se tiene que

$$F_Y(y) = \Phi_{\text{Logística}}\left(\frac{y-\mu}{\sigma}\right); \qquad f_Y(y) = \frac{1}{\sigma}\,\phi_{\text{Logística}}\left(\frac{y-\mu}{\sigma}\right), \quad -\infty < y < \infty$$

  donde

$$\Phi_{\text{Logística}}(z) = \frac{\exp(z)}{[1+\exp(z)]} \quad \text{y} \quad \phi_{\text{Logística}}(z) = \frac{\exp(z)}{[1+\exp(z)]^2}$$

  son la función de probabilidad y de densidad de una Logística Estándar. $\mu \in \mathbb{R}$, es un parámetro de localización y $\sigma > 0$, es un parámetro de escala. Si y_p es el percentil $p \times 100\,\%$, entonces

$$y_p = \mu + \sigma\,\Phi^{-1}_{\text{Logística}}(p) \quad \text{con} \quad \Phi^{-1}_{\text{Logística}}(p) = \log\left(\frac{p}{1-p}\right)$$

  Su esperanza y varianza están dadas por: $\mu_Y = \mu$ y $\sigma^2_Y = \frac{\sigma^2\,\pi^2}{3}$.

- Si $T \sim$ Log-Logística(μ, σ), se tiene que

$$F_T(t) = \Phi_{\text{Logística}}\left(\frac{\ln(t)-\mu}{\sigma}\right); \quad f_T(t) = \frac{1}{\sigma\,t}\,\phi_{\text{Logística}}\left(\frac{\ln(t)-\mu}{\sigma}\right) \quad t > 0$$

  Donde $\exp(\mu)$, es un parámetro de escala y $\sigma > 0$, es un parámetro de forma. Si t_p es el percentil $p \times 100\,\%$, entonces

$$\ln(t_p) = \mu + \sigma\,\Phi^{-1}_{\text{Logística}}(p)$$

  Para un entero $m > 0$ se tiene que

$$E(T^m) = \exp(m\,\mu)\,\Gamma(1 + m\,\sigma)\,\Gamma(1 - m\,\sigma)$$

  El m-ésimo momento no es finito si $m\,\sigma \geq 1$.

  Para $\sigma < 1$: $\mu_T = \exp(\mu)\,\Gamma(1+\sigma)\,\Gamma(1-\sigma)$

  y para $\sigma < 1/2$: $\sigma^2_T = \exp(2\,\mu)\,\left[\Gamma(1+2\,\sigma)\,\Gamma(1-2\,\sigma) - \Gamma^2(1+\sigma)\,\Gamma^2(1-\sigma)\right]$

- Un variable aleatoria T tiene distribución t-student si su función de densidad está dada por:

$$f_T(t) = \frac{\Gamma[(\nu+1)/2]}{\sqrt{\pi\,\nu}\,\Gamma(\nu/2)}\left(1 + \frac{t^2}{\nu}\right)^{-(\nu+1)/2}, \quad -\infty < t < \infty$$

  - $\mu_T = 0$, para $\nu > 1$.
  - $\sigma^2_T = \frac{\nu}{\nu-2}$, para $\mu > 2$.

- Si $T \sim$ Fisher(η, ν), se tiene que

$$f_T(t) = \frac{\Gamma(\frac{\eta+\nu}{2})}{\Gamma(\eta/2)\Gamma(\nu/2)}\left(\frac{\eta}{\nu}\right)^{\frac{\eta}{2}}\frac{t^{\frac{\eta}{2}-1}}{\left(\frac{\eta}{\nu}\,t + 1\right)^{\frac{\eta+\nu}{2}}}, \quad t > 0$$

  - $\mu_T = \frac{\nu}{\nu-2}$, para $\nu > 2$.
  - $\sigma^2_T = \frac{2\,\nu^2\,(\eta+\nu-2)}{\eta\,(\nu-2)^2\,(\nu-4)}$, para $\nu > 4$.

Formulario R

Funciones en R:

- Pedir ayuda e información sobre una función: `?NombreDeLaFuncion, help(NombreDeLaFuncion)`
- Raíz cuadrada: `sqrt()`
- Logaritmos: `log(), log2(), log10()`
- Exponencial: `exp()`
- Valor absoluto: `abs()`
- Signo: `sign()`
- Funciones trigonométricas: `sin(), cos(), tan()`
- Funciones trigonométircas inversas: `asin(), acos(), atan()`
- Resto de una división: `%%`
- Factorial: `factorial()`
- Logaritmo factorial: `lfactorial()`

Objetos:

- Para ver objetos creados en la sesión de trabajo: `ls(), objects()`
- Eliminar un objeto en especial: `rm(objeto)`
- Borrar todos los objetos del espacio de trabajo: `rm(list=ls())`
- Guardar todos los objetos del espacio de trabajo: `save.image(file="nombre archivo")`
- Guardar objeto `x` del espacio de trabajo: `save(x,file="nombre archivo)`

Otras funciones:

- Para crear secuencias: `seq(from=a, to=b, by=c)`
- Para repetir un elemento o vector cierta cantidad de veces: `rep(x, each=a,...)`
- Para obtener el tamaño de un vector: `length(vector)`
- Para llamar al elemento en la posición `i` de un vector: `vector[i]`

Operadores lógicos:

- Menor: `<`
- Menor o igual: `<=`
- Mayor: `>`
- Mayor o igual: `>=`
- Igual: `==`
- No es igual: `!=`
- Y: `&`
- O: `|`
- : No: `!`

Matrices:

- Definir una matriz del vector con datos `data` con `n` filas y `k` columnas, rellenando la matriz por columnas: `matrix(data, nrow=n, ncol=k, byrow=FALSE)`
- Obtener la fila `i` de la matriz `A`: `A[i,]`
- Obtener la columna `i` de la matriz `A`: `A[,i]`
- Obtener el elemento de la fila `i` y columna `j` de la matriz `A`: `A[i,j]`
- Diagonal de una matriz: `diag()`
- Producto elemento a elemento: `*`
- Producto matricial: `%*%`
- Dimensiones de una matriz: `dim(), nrow(), ncol()`
- Transpuesta de una matriz: `t()`
- Determinante de una matriz: `det()`
- Inversa de una matriz: `solve()`
- Concatenar filas: `rbind()`
- Concatenar columnas: `cbind()`
- Saber si un objeto es una matriz: `is.matrix()`
- Definir un objeto como una matriz: `as.matrix()`

Paquetes:

- Instalar un paquete: `install.packages("Nombre Package")`
- Cargar paquete: `library(Nombre Package)`

Cargar bases de datos:

- TXT, DAT y CSV: `read.table(file=, header=,dec=,...)`
- CSV: `read.csv()`
- XLS y XLSX del paquete `readxl`: `readXL()`
- Vector de datos: `scan()`
- DTA del paquete `foreign`: `read.dta()`
- Datos de cualquier tipo del paquete `rio`: `import()`
- Seleccionar directamente un archivo de la unidad de trabajo: `file.choose()`
- Vectorizar las columnas de la base de datos para poder trabajar con sus variables: `attach()`
- Obtener nombres de las variables de la base de datos: `names()`
- Obtener directorio de trabajo: `getwd()`
- Definir directorio de trabajo: `setwd("Dirección nuevo directorio")`
- Obtener clase de una columna: `class()`
- Convertir una variable en categórica: `as.factor()`
- Convertir una variable en numérica: `as.numeric()`

Estadística descriptiva:

- Media de una muestra: `mean()`
- Varianza: `var()`
- Desviación estándar: `sd()`
- Resumen de vector numérico: `summary()`

- Cuantiles: `quantile()`
- Mínimo: `min()`
- Máximo: `max()`
- Rango: `range(X)[2]-range(X)[1]`
- Rango intercuartil: `IQR(X)=quantile(X, 0.75)-quantile(X, 0.25)`
- Mediana: `median()`
- Coeficiente de variación: δ`=sd(X, na.rm=TRUE)/mean(X, na.rm=TRUE)`
- Coeficiente de asimetría: `Skewness = function(x){`
  `n = length(x)`
  `sum((x - mean(x))^3/n)/sd(x)^3`
  `}`
- Coeficiente de kurtosis: `Kurtosis = function(x){`
  `n = length(x)`
  `sum((x - mean(x))^4/n)/sd(x)^4-3`
  `}`
- Covarianza entre `X` e `Y`: `cov(X,Y)`
- Correlación entre `X` e `Y`: `cor(X,Y)`

Gráficos:

- Graficar un vector `x` versus un vector `y`: `plot(x,y)`
- Agregar un punto (x_1,y_1): `points(`x_1,y_1`)`
- Agregar línea entre dos puntos (x_1,y_1) y (x_2,y_2): `lines(c(`x_1,x_2`), c(`y_1,y_2`))`
- Agregar líneas horizontales: `abline(h=)`
- Agregar líneas verticales: `abline(v=)`
- Agregar línea con intercepto a_1 y pendiente b_1: `abline(a=`a_1`, b=`b_1`)`
- Dibujar gráfico en blanco: `plot(x, y, type="n")`
- En un `plot()` se definen los límites `c(a,b)` del eje `x` y `c(d,e)` del eje `y` con: `plot(x, y, xlim=c(a,b)), ylim=c(d,e))`
- Se etiquetan los ejes `x`, `y` con los argumentos: `plot(x, y, xlab=, ylab=)`
- Se da un título al gráfico con el argumento: `plot(x, y, main=)`
- Modificar color en elementos de un gráfico con el argumento: `col="..."`
- Modificar grosor de línea de un gráfico con el argumento: `lwd=`
- Agregar una etiqueta en un punto `(x,y)`: `x,y,label=etiqueta`
- Graficar un boxplot de la variable `x`: `boxplot(x=, main=, xlab=, ylab=, horizontal=, col=)`
- Graficar un boxplot de la variable `x` con respecto a un factor `y`: `boxplot(x~y, main=, xlab=, ylab=, horizontal=, col=)`
- Graficar un histograma de la variable `x`: `hist(x=, main= , break=, freq=NULL, xlab=, ylab=, col=)`
- Graficar un gráfico de barras de una variable categórica `x`: `barplot(table(x))`

Modelos de probabilidad:

- En general para cierta distribución `DISTR` existen las siguientes funciones:
  - `qDISTR(x,...)` entrega $P(X = x)$
  - `pDISTR(p,...)` entrega $P(X \leq x)$
  - `qDISTR(q,...)` entrega el valor de x tal que $P(X \leq x) = q$
  - `rDISTR(n,...)` genera una muestra proveniente de un modelo de distribución
- Binomial: `_binom(,size=`n`,prob=`p`)`
- Geométrica: `_geom(,prob=`p`)`
- Binomial-Negativa: `_nbinom(x=`$x - r$`, size=`r`, prob=`p`)`
- Poisson: `_pois(,lambda=`λ`)`
- Uniforme: `_unif(,min=`a`,max=`b`)`
- Normal: `_norm(,mean=`μ`,sd=`σ`)`
- Log-Normal: `_norm(,meanlog=`μ`,sdlog=`σ`)`
- Exponencial: `_exp(,rate=`λ`)`
- Gamma: `_gamma(,shape=`ν`, scale=`λ`)`
- Chi Cuadrado: `_chisq(,df=`n`)`
- t-Student: `_t(,df=`n`)`
- Fisher: `_f(,df1=`n_1`,df2=`n_2`)`
- Hipergeométrica: `_hyper(, m=`m`, n=`$N-m$`, k=`n`)`
- Weibull:
- Logística: `_logis(, location=`μ`, scale=`σ`)`
- Graficar función de densidad: `curve(dDISTR(x,...), from=, to=,...)`
- Graficar función de distribución: `curve(pDISTR(x,...),from=, to=, ...)`

Programación básica:

- if: `if(expresión lógica){`
```
expresión 1
}
else{
expresión 2
}
```
- Otra forma de utilizar `if` y `else`. Si se cumple `condición1` entonces se ejecuta `expresión1`, si no `expresión2`: `ifelse(condición1, expresión1, expresión2)`
- for: `for(x in vector){`
```
expresión 1
...
}
```
- while: `while(condición lógica){`
```
expresión 1
}
```
- Crear una función para retornar un objeto utilizando `return()` o para devolver una lista de objetos utilizando `list()`: `function(argumento1, argumento2, ...){`
```
expresión1
...
return() ó list() }
```

Otras funciones:

- Calcular integrales de una funcion en términos de x: `integrate(funcion, lower=, upper=)`
- Para imprimir algún valor o texto: `print()`
- Para concatenar textos y números: `paste(objeto1, objeto2, ...)`
- Calcular el mínimo de cada vector x_i: `pmin(`x_1`, ..., `x_n`, na.rm=FALSE)`
- Calcular el máximo de cada vector x_i: `pmax(`x_1`, ..., `x_n`, na.rm=FALSE)`
- Calcular para cada elemento del vector `X` la función `FUN`: `sapply(X,FUN,argumentosFUN)`
- Calcular para cada fila o columna de la matriz `X` la función `FUN`, si se quiere aplicar a las filas `MARGIN=1` y para columnas `MARGIN=2`: `apply(X,FUN,MARGIN=,argumentosFUN)`
- Calcular la función `FUN` al vector `X` dependiendo de los valores del argumento `INDEX` que se asume categórico y del mismo largo que `X`: `tapply(X, INDEX, FUN, argumentosFUN)`
- Probabilidad conjunta a partir de una `tabla` de frecuencias: `prop.table(tabla)`

Análisis de Regresión:

- Gráfico de correlaciones de matriz X: `pairs(X)`
- Para agregar correlaciones en triángulo superior del gráfico pairs: `panel.cor=function(x,y){`
```
par(usr=c(0,1,0,1))
txt=as.character(format(cor(x,y), digits=2))
text(0.5, 0.5, txt, cex=6*abs(cor(x,y)))
}
pairs(X, upper.panel=panel.cor)
```
- Para un análisis de regresión simple donde queremos estimar la ecuación: $y = \alpha + \beta x$. Estimación de parámetros α y β:

$$\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}$$
$$\hat{\beta} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{j=1}^{n}(x_j - \bar{x})^2}$$

- Estimador insesgado para la varianza:

$$s^2_{Y|x} = \frac{1}{n-2}\left[\sum_{i=1}^{n}(y_i - \bar{y})^2 - \hat{\beta}^2\sum_{i=1}^{n}(x_i - \bar{x})^2\right]$$

- Coeficiente de determinación ajustado: $r^2 = 1 - \frac{s^2_{Y|x}}{s^2_Y}$
- Función para generar un modelo de regresión lineal simple para explicar `Y` en términos de `X`: `lm(Y ~ X)`
- Para obtener un resumen del modelo creado se puede aplicar: `summary(lm(Y ~ X))`

Gráficos de probabilidad (qqplot):

- Para construir el gráfico para un vector `Y` y saber si se ajusta a una distribución normal y obtener los estimadores para la expresión $x_{(q)} = \mu + \sigma\Phi^{-1}(p_q)$ el código es el siguiente:
```
Y=sort(Y)
N=length(Y)
m=1:N
x=m/(N+1)
p=(1:N)/(N+1)
plot(qnorm(x), Y, xaxt="n",
ylab="Valores de Y",
xlab="Probabilidad Acumulada", bty="n",
lwd=2, las=1,
xlim=c(-3.5,3.5))
axis(1,at=qnorm(p),label=p,las=1)
abline(lm(Y qnorm(x)), lwd=2 ,col=2)
fit=lm(Y qnorm(x))$coef
hat.mu=fit[1]
hat.sigma=fit[2]
hat.mu;hat.sigma
```
- Para construir el gráfico para un vector `Y` y saber si se ajusta a una distribución log-normal y obtener los estimadores para la expresión $\ln x_{(q)} = \lambda + \zeta\Phi^{-1}(p_q)$ el código es el siguiente:
```
Y=sort(Y)
N=length(Y)
m=1:N
x=m/(N+1)
p=(1:N)/(N+1)
plot(qnorm(x), log(Y), xaxt="n", ylab="Valores
de Y",
xlab="Probabilidad Acumulada", bty="n",
lwd=2, las=1,
```

```
xlim=c(-3.5,3.5))
axis(1,at=qnorm(p),label=p,las=1)
abline(lm(log(Y) qnorm(x)), lwd=2 ,col=2)
fit=lm(Y qnorm(x))$coef
hat.lambda=fit[1]
hat.zeta=fit[2]
hat.lambda;hat.zeta
```

- La función vista en clases para obtener las estimaciones para una distribución Weibull de parámetros β (forma) y ν (escala) es la siguiente:

```
QQ.Weibull = function(y){
x=sort(y)
N=length(y)
p=(1:N)/(N+1)
plot(log(x) log(-log(1-p)), pch=20, col="darkblue",
bty="n", las=1, main=expression("QQ-Weibull"),
ylab=expression(log(x[p])),
xlab=expression(log(-log(1-p))))
abline(lm(log(x)  log(-log(1-p))), lwd=3,
col="darkorange")
aux=lm(log(x)  log(-log(1-p)))
aux }
eta= as.numeric(exp(QQ.Weibull(X)$coef[1]))
beta= as.numeric(1/QQ.Weibull(X)$coef[2])
```