Procesos de validación de método analíticos

Cristóbal Honores

Resumen clase anterior

1. Importar bases de datos.
2. Lectura de bases de datos.
3. Selección de columnas y filtros.
4. dplyr.

Importar bases de datos

Importar cualquier base de datos:

import(….) Paquete: ”rio”

Importar desde Excel:

read_excel(…., sheet = …. **)** Paquete: “readxl”

Lectura de bases de datos

- **head(...):** Muestra las primeras filas de la base
- **tail(...):** Muetra las ultimas filas de la base
- **names(...):** Muestra los nombres de las columnas de la base
- **summary(...):** Realiza un resumen de la base por columna

Selección de columnas y filtros

Para seleccionar una columna en especifico de la base de dato se utiliza $ y luego el nombre de la columna.

Base$Columna

Bases[,]:

Fila: base_completa[numero de fila,]

Columna: base_completa[, numero de columna]

Dato especifico: base_completa[numero de fila,numero de columna]

Paquete: dplyr

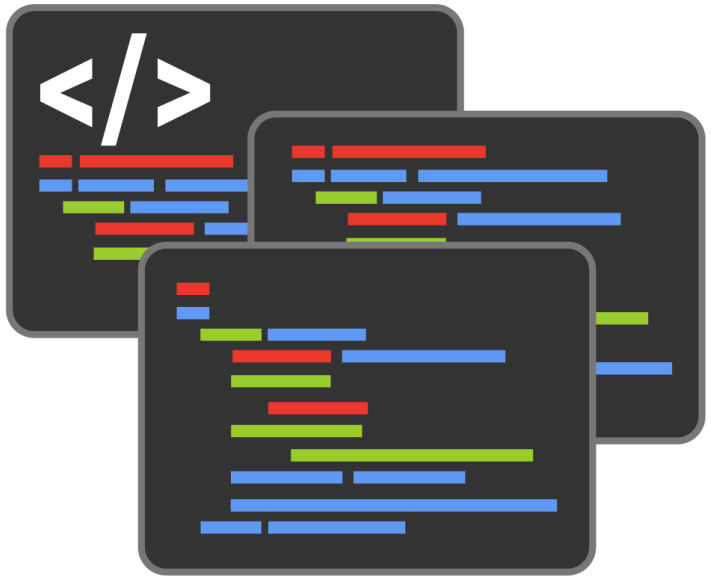
- full_join(): Une bases de datos por columnas.
- bind_rows(): Agrega filas a una base de datos.
- Select(): Selecciona columna(s) de una base de datos.
- Filter(): Realiza filtros de manera mas fácil.

Propósito de la clase

Aprender a utilizar herramientas que computacionales para un adecuado análisis de resultados y control de calidad.

Procesos de validación de método analíticos.

1. Estadísticos descriptivos.
2. Tabulación de resultados.
3. Control de calidad.
4. Test para detectar anomalías.



Estadísticos de centro

- Media:

```
mean(base$Variable)
```

- Mediana:

```
median(base$Variable)
```

- Moda:

```
library(modeest)
mfv(base$Variable)
```

Estadísticos de posición

- Cuartiles:

```
quantile(base$Variable)
```

- Percentiles:

```
quantile(base$Variable, prob=seq(0,1,length=101))
```

Estadísticos de variabilidad

- Rango:

```
range(base$varaible)
```

- Rango inter cuartil (RIC):

```
IQR(base$variable)
```

- Desviación estándar:

```
sd(base$variable)
```

Estadísticos de variabilidad

- Varianza:

```
var(base$variable)
```

- Error estándar :

```
sd(base$variable)/length(base$variable)
```

- Coeficiente de variación:

```
sd(base$variable)/mean(base$variable)
```

Tablas de frecuencia

La función **table()** sirve para construir tablas de frecuencia de una vía, a continuación la estructura de la función.

Tabla_1 <- **table**(base$..... , base$.....)

Genera la tabal de frecuencia de ..... y la guarda como objeto llamado “Tabla_1”.

Tablas simples

La función aggregate() permite crear una tabla de resumen de una variable y sus datos desdados.

aggregate(valores, varaible, data = ..., FUN = ...)

- data = base de datos (objeto).
- FUN = función que se desea aplicar.

Tablas de frecuencia

La función **addmargins()** se puede utilizar para agregar los totales por filas o por columnas a una tabla de frecuencia absoluta o relativa.

- addmargins(Tabla_1): Agrega las sumas totales por filas y columnas.
- addmargins(Tabla_1,margin = 1): Agrega las sumas totales por columnas.
- addmargins(Tabla_1,margin = 2): Agrega las sumas totales por filas.

Tablas de frecuencia

La función **prop.table()** se puede utilizar para tablas de frecuencia relativa a partir de tablas de frecuencia absoluta.

- prop.table(Tabla_1): Entrega las frecuencias globales.
- prop.table(Tabla_1,margin = 1): Agrega las proporciones por filas.
- prop.table(Tabla_1,margin = 2): Agrega las proporciones por columnas.

Tablas de frecuencia con fdth

La función fdt() del paquete fdth permite realizar una tabla de frecuencia con las frecuencias absolutas, relativas y acumuladas.

- **fdt**(variable,....)
  - start= Indica el inicio de mi primer intervalo.
  - end= Indica el final de mi ultimo intervalo.
  - h= Ancho de mis intervalos.
  - k= Numero de clases.

Tablas de frecuencia con fdth

Tabla de frecuencias de edad en 6 clases.

```
n <- fdt(Litiasis$EDAD,k=6)
```

OJO! Es necesario guardar esta tabla como un objeto.

```
print(n)
```

Control de calidad

1. Diagrama de Pareto.
2. Diagrama de causa-efecto.
3. Grafico de dispersión.
4. Carta de control.
5. Grafico de caja.

Diagrama de Pareto (qcc)

- El gráfico de Pareto es un gráfico de barras verticales en el que los valores se representan en orden decreciente de frecuencia relativa de izquierda a derecha.

- Ayuda a visualizar qué problemas necesitan atención primero al observar las barras más altas del gráfico

Diagrama de Pareto (qcc)

pareto.chart(x, ylab = ,ylab2 =, xlab, cumperc, ylim, main, col = plot = TRUE, ...)

x: Defectos

ylab: Nombre eje y izquierdo

ylab2: Nombre eje y derecho

xlab: Nombre eje X

cumerc: Divisiones porcentaje

ylim: Limites del eje Y derecho

main: Titulo de grafico

col: Color del grafico

plot: TRUE para grafiar

Diagrama de causa-efecto (qcc)

- Ayuda a identificar las posibles causas de un efecto, problema o resultado indeseable mientras las clasifica en categorías.

- El efecto analizado se coloca en la cabeza del pez mientras que las causas se colocan en las espinas de los peces según su categoría correspondiente.

Diagrama de causa-efecto (qcc)

cause.and.effect(cause, effect, title =, font = c(1, 3, 2))

- **cause:** Causas, utilizar siempre con una lista de la siguiente manera:

  list(Causa1=c("","",""),Causa2=c("","","")....).

- **effect:** Efecto.

- **title:** Titulo del diagrama.

- **font:** Fuente.

Grafico de dispersión

- Los gráficos de distribución ayudan a determinar si existe correlación entre los conjuntos de datos

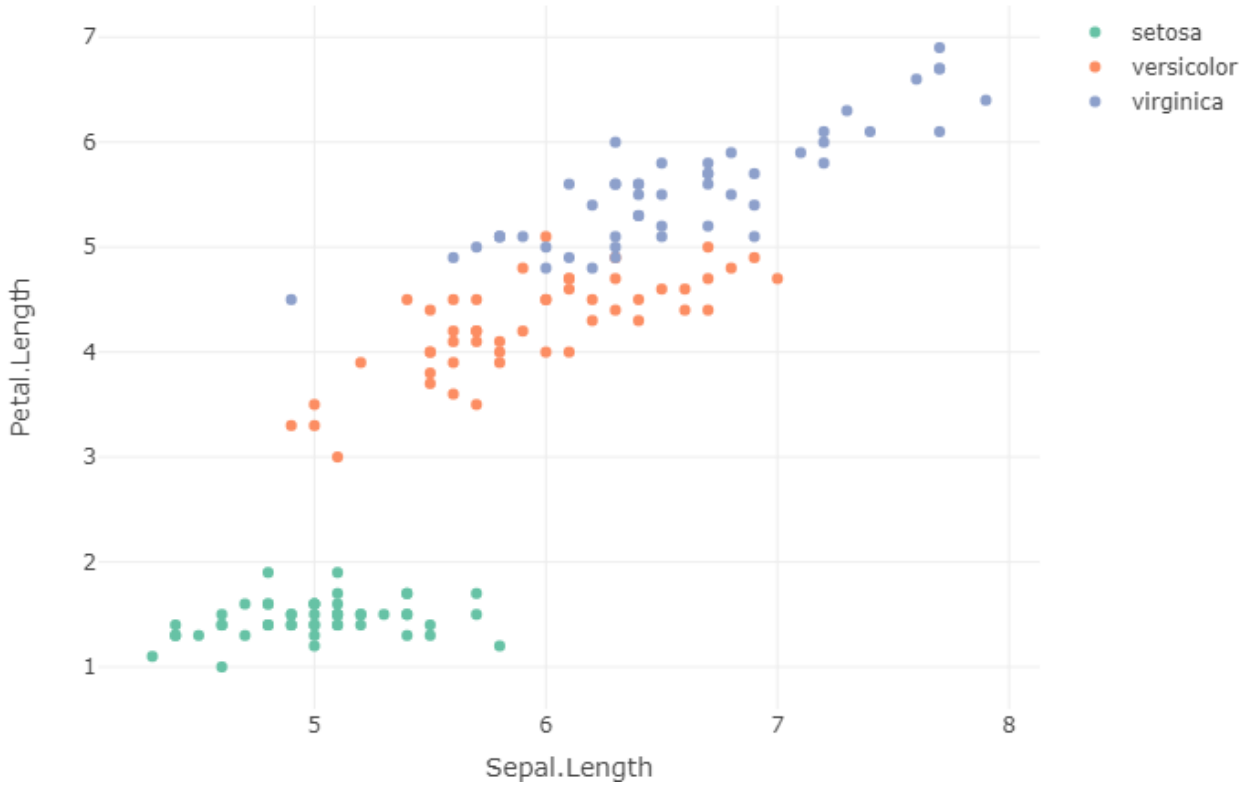


Gráfico de dispersión

```
plot(x, y, main, xlab, ylab, pch, col, )
```

```
ggplot(data, aes(x=..., y=...)) + geom_point()
```

```
plot_ly(data = ...., x = ~...., y = ~..., type = "scatter")
```

Matriz de dispersión

- **El grafico de Matriz de dispersión permite realizar todos los gráficos de dispersión posibles de una base de datos.**

pair(data, ~Variables)

ggpairs(data, columns=....,colour=.....) [paquete: GGally]

ggplotly(ggpairs(data, columns=....,colour=.....))
[paquete: ggplot & plotly & GGally]

Gráfico de caja

- Permite visualizar la distribución de los datos atreves de los cuartiles.
- Además muestra posibles outliers, definiendo limites inferiores y superiores.
- Cualquier dato sobre el limite superior o bajo el limite inferior es un posible dato anómalo (outlier).

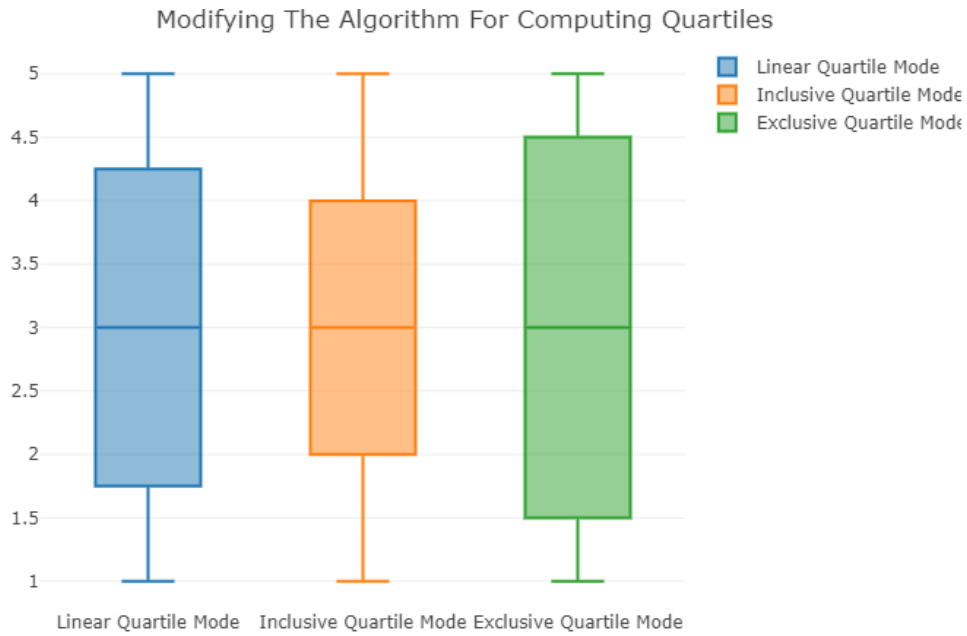


Gráfico de caja

- Limite inferior:

$$LI=Q1-(1.5*IQR)$$

- Limite superior:

$$LS=Q3+(1.5*IQR)$$

Gráfico de caja

boxplot(x, horiz=..., width=..., pch=..., ...)

ggplot(data, aes(x=..., y=...)) + geom_boxplot ()

[Paquete ggplot2]

plot_ly(data = ..., x = ~...., y = ~..., type = "box")

[Paquete Plotly]

Gráfico de violín

- Muestra la distribución de los datos atreves del grafico de cajas y del grafico de densidad.

Vioplot(x, ylim=..., xlim=...., col...., ...)

[Paquete vioplot]

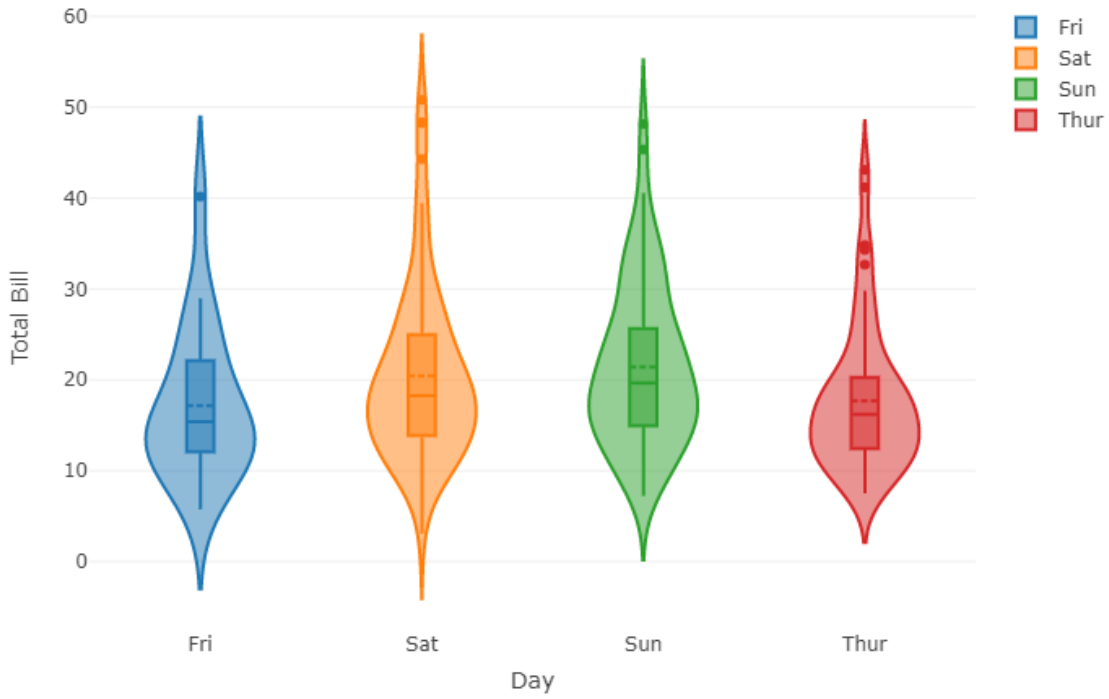


Gráfico de violín

ggplot(data, aes(x=..., y=...)) + geom_violin () + geom_boxplot ()

[Paquete ggplot2]

plot_ly(data = ..., x = ~...., y = ~..., type = "violin")

[Paquete Plotly]

Grafico de control (qcc)

- Muestran si las muestras de productos o procesos cumplen con las especificaciones previstas y, de no ser así, el grado en que varían de esas especificaciones.

- Permiten monitorear si un proceso está bajo control; ayuda a visualizar la variación; encontrar y corregir problemas cuando ocurren; predecir los rangos esperados si los resultados; y analizar patrones de variación de procesos por causas especiales.

Gráfico de control (qcc)

qcc(data, type, newdata, plot, ...)

- **data:** Datos a graficar. (calibrando)
- **type:** Especifica el tipo de grafico deseado
- **newdata:** Agrega nuevos datos a graficar. (calibrado)
- **plot:** TRUE entrega el grafico, FALSE no entrega el grafico.

Gráfico de control (qcc)

- **Nivel de confianza:** Especifica el nivel de confianza

- **nsigmas:** especifica el numero de sigmas, se invalida cuando se especifica el nivel de confianza.

- Tipos de gráficos:
  - **xbar:** Promedios
  - **R:** Rango
  - **S:** Desviación estándar

Agregar límites de advertencia en 2 desviaciones std.

1. Guardar el grafico como objeto.

```
q <- qcc(...)
```

2. Crear los limites de advertencia (a 2 sd).

```
(warn.limits <- limits.xbar(q$center, q$std.dev, q$sizes, 2))
```

3. Graficar los limites.

```
plot(q, restore.par = FALSE)
```
<- permite agregar líneas

```
abline(h = warn.limits, lty = 3, col = "chocolate")
```

Test Rosner (EnvStats)

- **Paquete: EnvStats**

- Realiza la prueba de Rosner para hasta k valores atípicos potenciales en un conjunto de datos, suponiendo que los datos sin valores atípicos provienen de una distribución normal (gaussiana).

```
rosnerTest(x, k = 3, alpha = 0.05, warn = TRUE)
```

Test Rosner (EnvStats)

rosnerTest(x, k = 3, alpha = 0.05, warn = TRUE)

x: Vector numérico de observaciones, puede tener valores NA, pero serán removidos.

k: Indica el numero de datos sospechosos de ser outlier.

alpha: Numero entre 0 y 1 que indica el error Tipo I.

warn: Indica si un valor es outlier o no, dependiendo del alpha.

Procesos de validación de método analíticos

Cristóbal Honores