

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Ingeniería en Ciencias y Sistemas
Seminario de Sistemas 2

Practica 1

Esdras Jonatan Noj Larios
201513699

Problema 1

En el problema 1 improtamos los datos de la siguiente manera:

```
ventas <- read.csv("/Users/esdras/Desktop/Practica 1 Seminario/Ventas/VENTAS.csv")
```

Procedemos a crear un data frame de la siguiente manera:

```
data_frame_ventas <- data.frame(ventas)
```

Filtramos los datos de los países que nos interesan, esto lo hacemos de la siguiente manera:

```
países <- data_frame_ventas[data_frame_ventas$Country == "Guatemala" |  
data_frame_ventas$Country == "El Salvador" |  
data_frame_ventas$Country == "Honduras" | data_frame_ventas$Country ==  
"Nicaragua" |  
data_frame_ventas$Country == "Costa Rica" | data_frame_ventas$Country ==  
"Panama" |  
data_frame_ventas$Country == "Belice",]
```

Obtenemos la frecuencia absoluta

```
fr_abs <- table(países$Country)
```

Frecuencia Acumulada

```
fr_acum <- cumsum(fr_abs)
```

Frecuencia Relativa

```
fr_rlva <- prop.table(fr_abs)
```

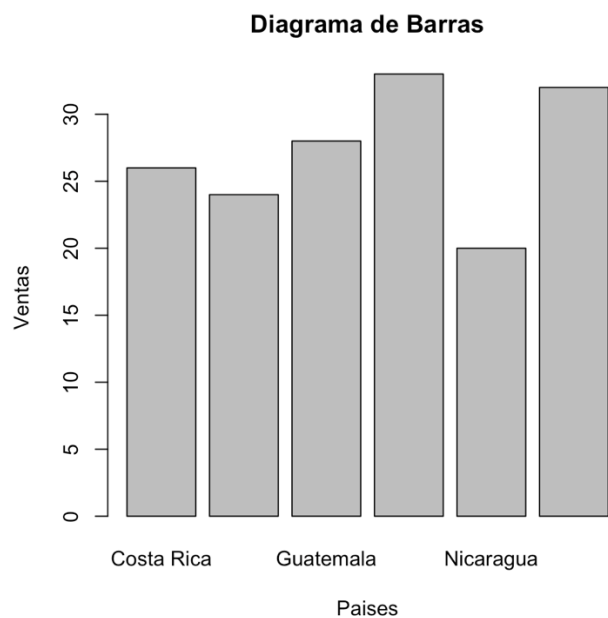
Frecuencia Relativa Acumulada

```
fr_rlva_acum <- cumsum(fr_rlva)
```

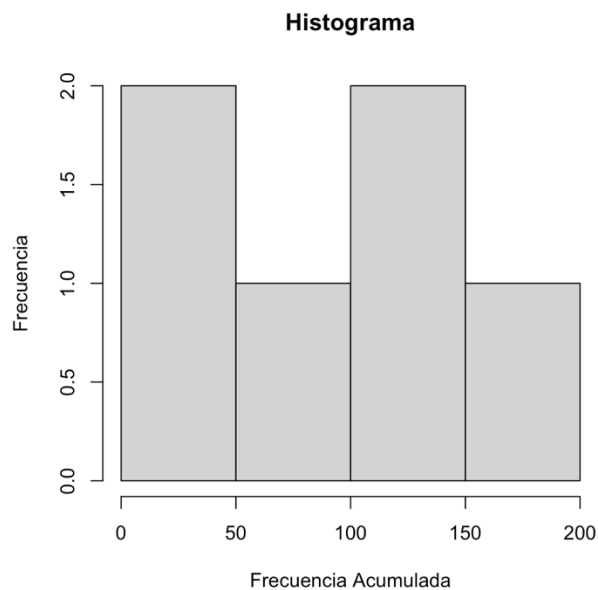
Generamos una tabla de frecuencias y la asignamos a un data frame

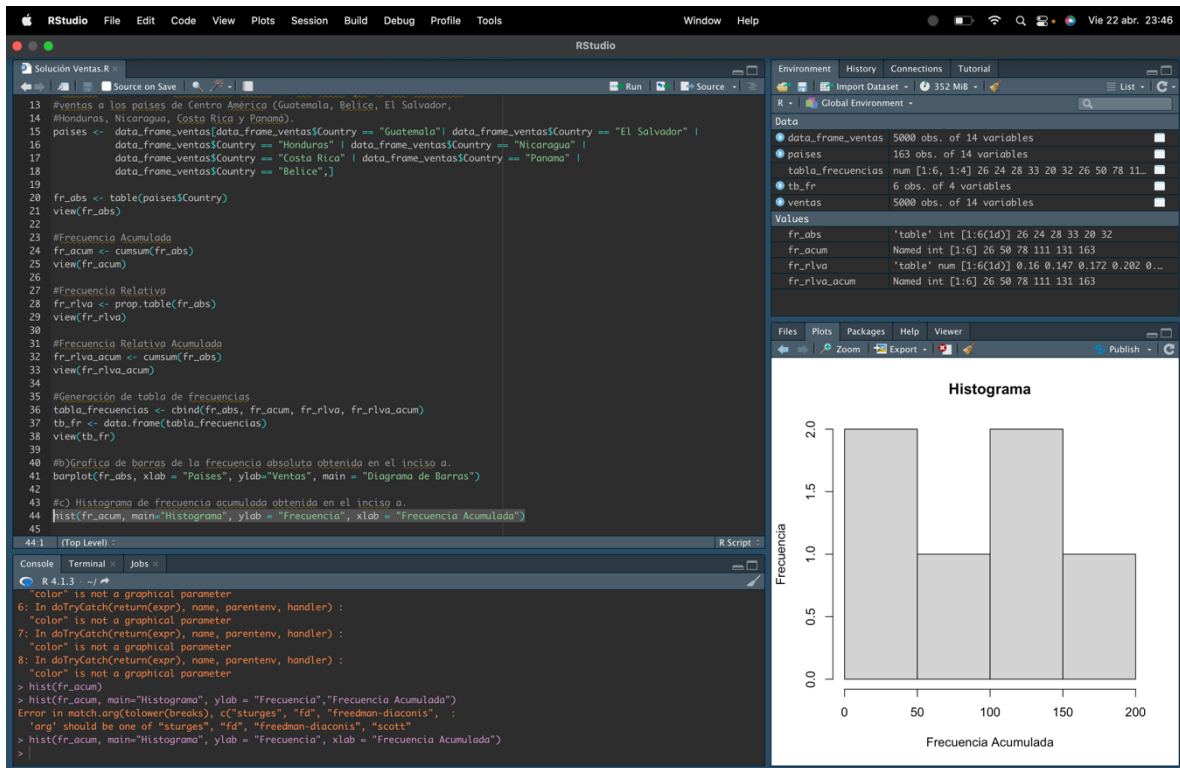
```
tabla_frecuencias <- cbind(fr_abs, fr_acum, fr_rlva, fr_rlva_acum)  
tb_fr <- data.frame(tabla_frecuencias)
```

Con el data frame podríamos generar una gráfica de barras, con el siguiente comando:
`barplot(fr_abs, xlab = "Países", ylab="Ventas", main = "Diagrama de Barras")`



podríamos generar un gráfico de histograma con el siguiente comando:
`hist(fr_acum, main="Histograma", ylab = "Frecuencia", xlab = "Frecuencia Acumulada")`





Problema 2

Importamos los datos con el siguiente comando:

```
data_cpu <- read.csv("/Users/esdras/Desktop/Practica 1
Seminario/CPU/DesempenioCPU.csv")
```

Una vez importados los datos procedemos a filtrar por la marca que nos interesa, esto lo hacemos con el comando siguiente:

```
datos_col <- data_cpu[data_cpu$Compania == 'amdahl',]
```

Obtenemos el promedio con el comando siguiente:

```
promedio <- mean(datos_col$PRP)
```

Obtenemos el máximo con el comando siguiente:

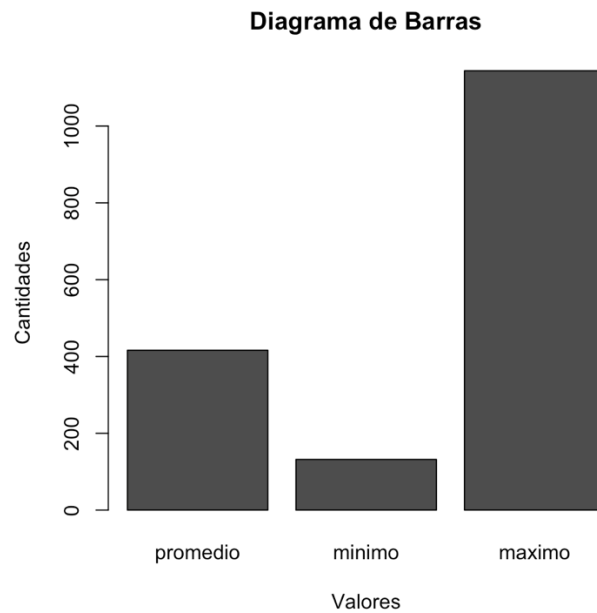
```
maximo <- max(datos_col$PRP)
```

Obtenemos el mínimo con el comando siguiente:

```
minimo <- min(datos_col$PRP)
```

Creamos una tabla con el comando siguiente, para agrupar los datos:
`data_solution <- cbind(promedio, minimo, maximo)`

Para graficar los datos generados lo hacemos de la siguiente manera:
`barplot (data_solution, main="Diagrama de Barras", xlab= "Valores", ylab="Cantidades")`



Problema 3

Importamos los datos

```
data_decesos <- read.csv("/Users/esdras/Desktop/Practica 1  
Seminario/DecesoSexoEdad/MuertesSexoEdad.csv")
```

Creamos el data frame

```
df_decesos <- data.frame(data_decesos)
```

Filtramos los datos para que solo queden los que nos interesen, usamos las siguientes condiciones:

```
datos_limpios <- df_decesos[df_decesos$Sex == 'Male' | df_decesos$Sex == 'Female',]  
datos_limpios <- df_decesos[df_decesos$Age != 'Total',]
```

La frecuencia absoluta se obtiene de la siguiente manera:

```
fr_abs <- table(datos_limpios$Age)
```

La frecuencia acumulada se obtiene de la siguiente manera:

```
fr_acum <- cumsum(fr_abs)
```

La frecuencia relativa se obtiene de la siguiente manera:

```
fr_rlva <- prop.table(fr_abs)
```

La frecuencia relativa acumulada se obtiene de la siguiente manera:

```
fr_rlva_acum <- cumsum(fr_rlva)
```

Creamos una tabla de frecuencias

```
tabla_frecuencias <- cbind(fr_abs,fr_acum,fr_rlva,fr_rlva_acum)
```

La tabla creada la asignamos a un data.frame, para poder visualizar de mejor manera los datos.

```
data_table_f <- data.frame(tabla_frecuencias)
```

Podremos ver la tabla de frecuencias con el comando siguiente:

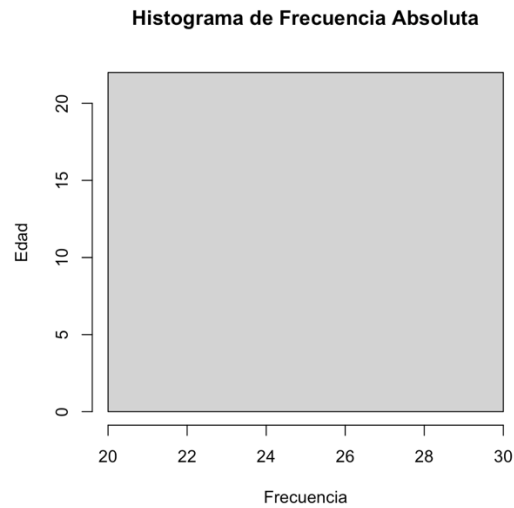
```
view(data_table_f)
```

	fr_abs	fr_acum	fr_rlva	fr_rlva_acum
1<96>4	30	30	0.04545455	0.04545455
10<96>14	30	60	0.04545455	0.09090909
100 and over	30	90	0.04545455	0.13636364
15<96>19	30	120	0.04545455	0.18181818
20<96>24	30	150	0.04545455	0.22727273
25<96>29	30	180	0.04545455	0.27272727
30<96>34	30	210	0.04545455	0.31818182
35<96>39	30	240	0.04545455	0.36363636
40<96>44	30	270	0.04545455	0.40909091
45<96>49	30	300	0.04545455	0.45454545
5<96>9	30	330	0.04545455	0.50000000
50<96>54	30	360	0.04545455	0.54545455
55<96>59	30	390	0.04545455	0.59090909
60<96>64	30	420	0.04545455	0.63636364
65<96>69	30	450	0.04545455	0.68181818
70<96>74	30	480	0.04545455	0.72727273
75<96>79	30	510	0.04545455	0.77272727
80<96>84	30	540	0.04545455	0.81818182
85<96>89	30	570	0.04545455	0.86363636
90<96>94	30	600	0.04545455	0.90909091
95<96>99	30	630	0.04545455	0.95454545
Infant	30	660	0.04545455	1.00000000

Histograma de frecuencia absoluta

Se grafica con el comando siguiente:

```
hist(fr_abs, main="Histograma de Frecuencia Absoluta",ylab="Edad", xlab="Frecuencia")
```



Pólígono de frecuencia absoluta

poligo<-graph.freq(tabla_frecuencias,main = "Freccuencia Absoluta", xlab="Edad", ylab = "Decesos")

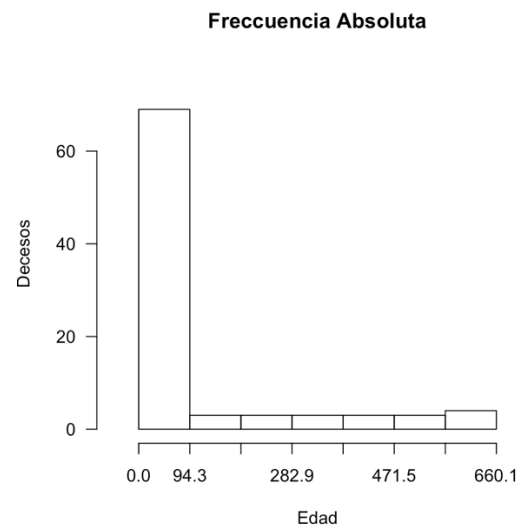
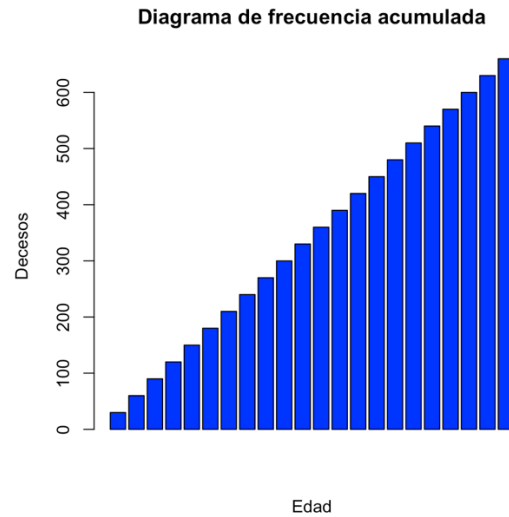


Diagrama de frecuencia acumulada

Se grafica con el comando siguiente:

barplot(fr_acum, main = "Diagrama de frecuencia acumulada", xlab= "Edad", ylab = "Decesos",col="blue")



Problema 4

Comando para importar datos

```
data_presion <- read.csv("/Users/esdras/Desktop/Practica 1
Seminario/Presion/PresionSanguineaEdad.csv")
```

Se agregan los datos a un data frame

```
df_presion <- data.frame(data_presion)
```

Se inicia el análisis para comprobar la relación entre la presión sanguínea y la edad

```
cor(df_presion)
```

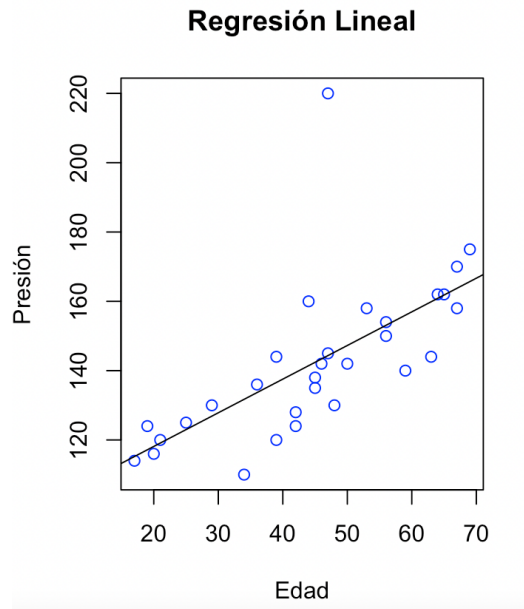
	Index	Uno	Edad	Systolic.Blood.Pressure
Index	1.0000000	NA	-0.2742924	-0.3104962
Uno	NA	1	NA	NA
Edad	-0.2742924	NA	1.0000000	0.6575673
Systolic.Blood.Pressure	-0.3104962	NA	0.6575673	1.0000000

Si existe una relación entre edad y presión, siendo esta de 0.65, por lo cual se procede a la regresión lineal.

```
data_regresion <- lm(Systolic.Blood.Pressure~Edad, data = df_presion)
```



```
plot(df_presion$Edad, df_presion$Systolic.Blood.Pressure,
     main="Regresión Lineal", ylab="Presión", xlab="Edad", col="blue")
abline(data_regresion, col="black")
```



Problema 5

Pobreza - Desempleo

El análisis es de 0.8154184 por lo que si se puede hacer correlación lineal entre pobreza y desempleo

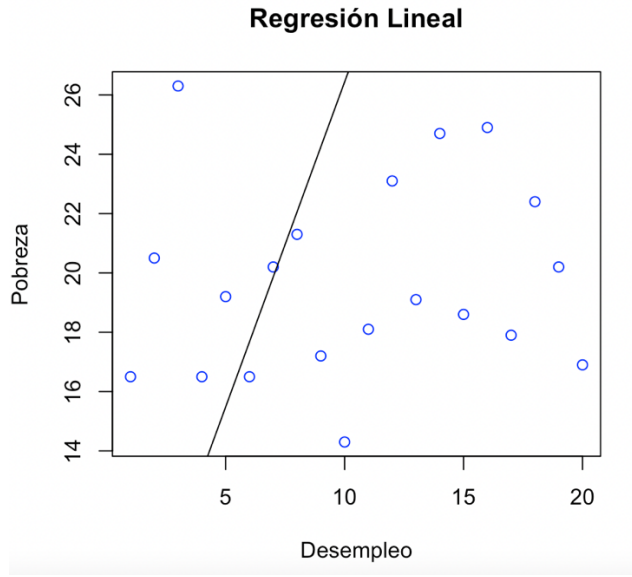
	Indice	Habitantes	Porcentaje.con.ingresos.debajo.de.5000	Porcentaje.desempleado	Asesinatos.por.1000000.habitantes
Indice	1.0000000	0.1485210	0.1177125	0.2988504	0.2152649
Habitantes	0.1485210	1.0000000	-0.1637798	-0.2334055	-0.0670984
Porcentaje.con.ingresos.debajo.de.5000	0.1177125	-0.1637798	1.0000000	0.8154184	0.8397782
Porcentaje.desempleado	0.2988504	-0.2334055	0.8154184	1.0000000	0.8648415
Asesinatos.por.1000000.habitantes	0.2152649	-0.0670984	0.8397782	0.8648415	1.0000000

Para en análisis de regresión lineal usamos el siguiente comando:

```
data_regresion1 <- lm(Porcentaje.con.ingresos.debajo.de.5000~Porcentaje.desempleado,
data = df_pobreza)
```

Con base a los datos obtenidos anteriormente en la regresión lineal generamos la gráfica de la siguiente manera:

```
plot(df_pobreza$Porcentaje.con.ingresos.debajo.de.5000, df_presion$Porcentaje.desempleado,
     main="Regresión Lineal", ylab="Pobreza", xlab="Desempleo", col="blue")
abline(data_regresion, col="black")
```



Pobreza-Cantidad de asesinatos

	Indice	Habitantes	Porcentaje.con.ingresos.debajo.de.5000	Porcentaje.desempleado	Asesinatos.por.1000000.habitantes
Indice	1.0000000	0.1485210	0.1177125	0.2988504	0.2152649
Habitantes	0.1485210	1.0000000	-0.1637798	-0.2334055	-0.0670984
Porcentaje.con.ingresos.debajo.de.5000	0.1177125	-0.1637798	1.0000000	0.8154184	0.8397782
Porcentaje.desempleado	0.2988504	-0.2334055	0.8154184	1.0000000	0.8648415
Asesinatos.por.1000000.habitantes	0.2152649	-0.0670984	0.8397782	0.8648415	1.0000000

El valor es de: 0.8397782, por lo que sí se puede hacer un análisis de regresión

Con el comando vamos a generar el análisis de regresión lineal:

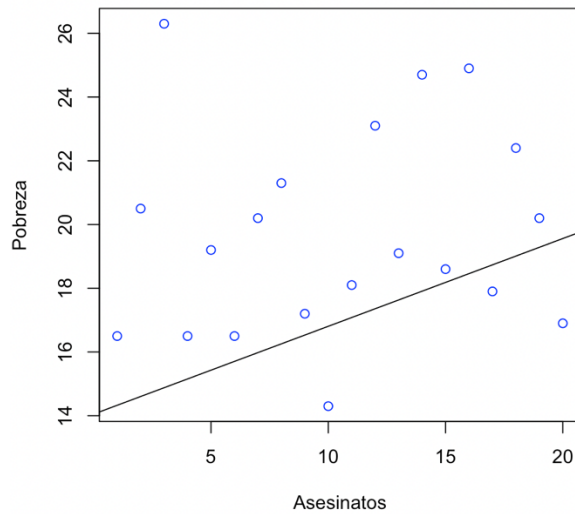
```
data_regresion1 <-
```

```
lm(Porcentaje.con.ingresos.debajo.de.5000~Asesinatos.por.1000000.habitantes, data =  
df_pobreza)
```

El comando para graficar la regresión lineal es el siguiente:

```
plot(df_pobreza$Porcentaje.con.ingresos.debajo.de.5000,  
df_pobreza$Asesinatos.por.1000000.habitantes,  
main="Regresión Lineal", ylab="Pobreza", xlab="Asesinatos", col="blue")  
abline(data_regresion1, col="black")
```

Regresión Lineal



```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Window Help
RStudio
Solución Ventas.R Solución.R
Source on Save Run Source
1 library("readr")
2 library("tidyverse")
3
4 # Importar los datos del csv
5 data_pobreza <- read_csv("/Users/esdras/Desktop/Practica 1 Seminario/Pobreza/PobrezaDesempleoAsesinatos.csv")
6
7 df_pobreza <- data.frame(data_pobreza)
8
9 cor(df_pobreza)
10 ##0.8154184 si se puede hacer correlación lineal entre pobreza y desempleo
11
12 data_regresion1 <- lm(Porcentaje.con.ingresos.debajo.de.5000~Porcentaje.desempleado, data = df_pobreza)
13
14 # Grafica de regresión lineal
15 plot(df_pobreza$Porcentaje.con.ingresos.debajo.de.5000, df_presion$Porcentaje.desempleado,
16      main="Regresión Lineal", ylab="Pobreza", xlab="Desempleo", col="blue")
17 abline(data_regresion1, col="black")
18
19 ###Pobreza y cantidad de asesinatos
20
21 ####Regresión para pobreza-asesinatos
22 data_regresion1 <- lm(Porcentaje.con.ingresos.debajo.de.5000~Asesinatos.por.1000000.habitantes, data = df_pobreza)
23
24
25 plot(df_pobreza$Porcentaje.con.ingresos.debajo.de.5000, df_presion$Asesinatos.por.1000000.habitantes,
26      main="Regresión Lineal", ylab="Pobreza", xlab="Asesinatos", col="blue")
27 abline(data_regresion1, col="black")
15.1 (Top Level)
Console Terminal Jobs
R 4.1.3 ~ /
Error in xy.coords(x, y, xlabel, ylabel, log) :
  object 'df_presion' not found
> abline(data_regresion1, col="black")
Error in int.abline(a = a, b = b, h = h, v = v, untf = untf, ...) :
  plot.new has not been called yet
> plot(df_pobreza$Porcentaje.con.ingresos.debajo.de.5000, df_presion$Porcentaje.desempleado,
+      main="Regresión Lineal", ylab="Pobreza", xlab="Desempleo", col="blue")
Error in xy.coords(x, y, xlabel, ylabel, log) :
  object 'df_presion' not found
> abline(data_regresion1, col="black")
Error in int.abline(a = a, b = b, h = h, v = v, untf = untf, ...) :
  plot.new has not been called yet
>
```