Estadística descriptiva

Selección de actividades resueltas

© Jose Fco. Martínez Boscá, Arnau Mir Torres, Lluís M. Pla Aragonés &

Ángel A. Juan (Editor)

UOC Universitat Oberta de Catalunya

Introducción

Conocer cuántos mensajes “spam” pasan por un servidor de correo de una determinada empresa puede realizarse de dos maneras. En primer lugar, se puede intentar adivinar teniendo en cuenta el número de empleados de la empresa, el tipo de páginas web que visitan, cuántos de ellos chatean, etc… Otra forma de abordar el problema es recoger el número de mensajes “spam” que pasan por el servidor durante un conjunto de días. El primer método es muy complejo debido a la cantidad de variables que hay que tener en cuenta; en cambio, el segundo método es muy simple pero tenemos que aprender técnicas para poder alcanzar nuestro objetivo. La estadística es la disciplina que se dedica a resolver problemas como el anterior usando métodos como el que hemos mencionado de recogimiento de datos.
En todo estudio estadístico, existen dos fases bien diferenciadas:

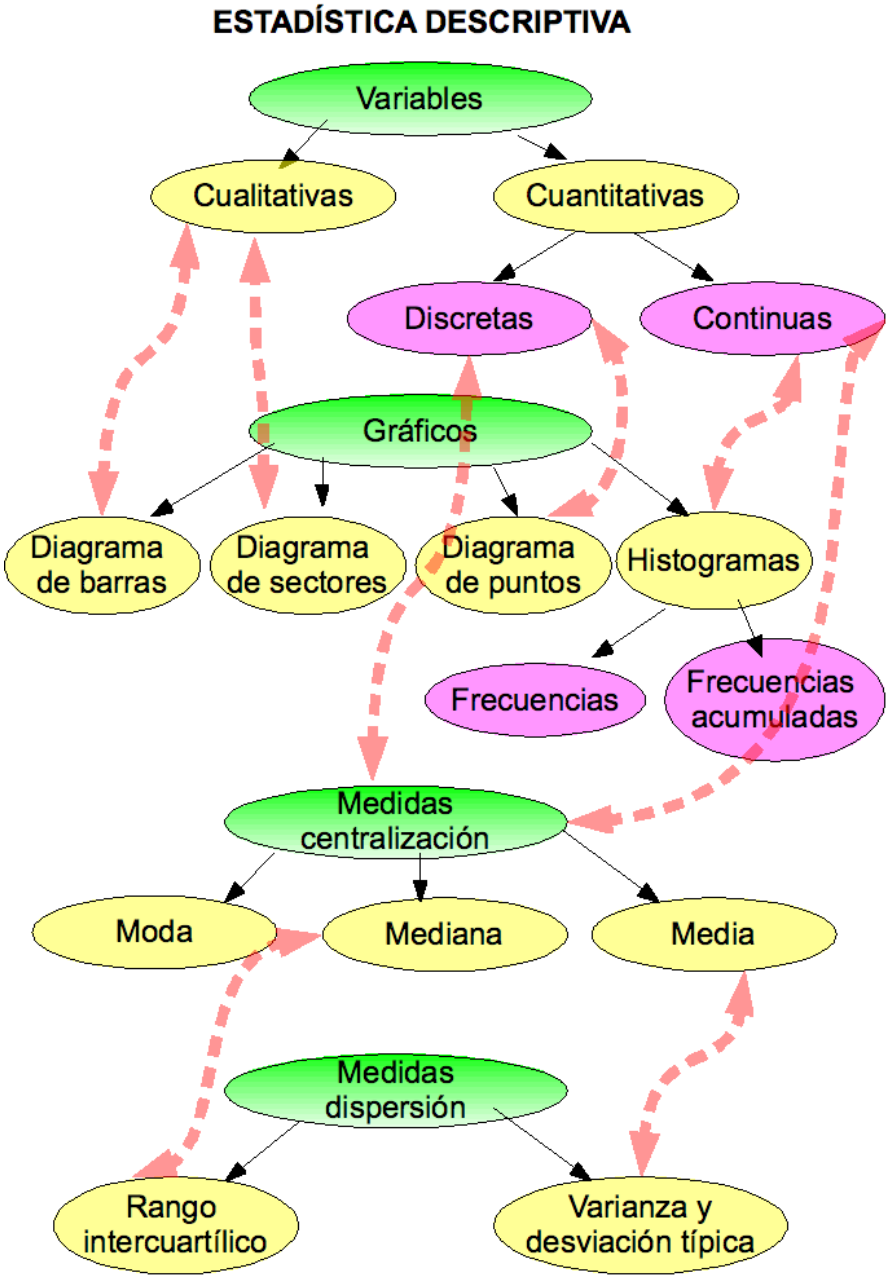
- Fase 1: recogida de datos y,
- Fase 2: análisis de dichos datos.

En la fase 1, los datos se recogen, agrupan y se caracterizan. La parte de la estadística encargada de llevar a cabo la fase 1 se denomina Estadística Descriptiva. En la fase 2, se realiza el análisis de dichos datos con el fin de sacar conclusiones a partir de dicho análisis. La parte de la estadística encargada de llevar a cabo la fase 2 se denomina Estadística Inferencial. ¿Qué tipo de conclusiones esperamos obtener? Básicamente conocer información de toda la población a partir del estudio realizado de una muestra de datos. En el ejemplo anterior, una vez tengamos caracterizada la muestra correspondiente al número de mensajes “spam” que recibe el servidor durante 30 días, ¿qué podemos decir sobre el número de mensajes “spam” que recibe diariamente dicho servidor? Concretando un poco más, ¿podemos afirmar que la media de mensajes “spam” que recibe el servidor de correo de la empresa es representativa de la media de todos los mensajes “spam” que recibe diariamente dicho servidor?

La estadística intenta resolver problemas más complejos. Por ejemplo, siguiendo con el ejemplo anterior, imaginemos que la empresa anterior no sólo está interesada en conocer el número de mensajes “spam” que recibe cada día sino el tipo de mensajes “spam”. O sea, queremos estudiar el conjunto de mensajes “spam” que recibe el servidor de correo de dicha empresa. ¿Qué significa exactamente conocer dicho conjunto? La respuesta a dicha pregunta puede ser muy amplia. Por ejemplo, clasificar los mensajes “spam” usando una serie de características comunes; o intentar estudiar a qué horas se reciben más “spam”, etc. Todo ello para intentar hacer una predicción del comportamiento de los mensajes “spam” de dicha empresa. Las técnicas anteriores son ejemplos de herramientas estadísticas denominadas “data mining”, herramientas muy importantes en ciencias de la computación.

En este módulo vamos a presentar un conjunto de ejemplos dedicados a la recolección de datos e intentar caracterizar dichos datos. Los datos se pueden caracterizar hallando valores que los representen y hallando valores que nos indiquen lo dispersos que están. Los valores que representan a los datos reciben el nombre de medidas de centralización y los valores que indican la dispersión de los mismos reciben el nombre de medidas de dispersión.
Las medidas de centralización más importantes son la media aritmética, la mediana y los percentiles y las medidas de dispersión más importantes son la varianza, la desviación típica y la desviación estándar.

Mapa conceptual



Actividades

Estadística Descriptiva

Actividad 1: Cómputo del tiempo de CPU.

Medidas de tendencia central. Medidas de dispersión. Histogramas. Diagramas de caja.

En la tabla siguiente se muestran los resultados de un test que consiste en ejecutar aleatoriamente diferentes programas en un ordenador y medir el tiempo de CPU consumido (en milisegundos) para cada programa (variable TEMP_CPU). También conocemos la longitud del código de cada uno de los programas ejecutados (Variable LONG_CODI). En este problema estudiaremos la variable TEMP_CPU.

TEMP_CPU	LONG_CODI
127	146
83	80
85	60
93	90
103	58
80	88
71	106
112	150
116	38
62	195
123	121
90	109
103	148
63	96
116	92
103	50
98	19
71	84
103	90
101	72
91	97
125	147
117	108
126	135
112	111
91	79
89	121
105	194
110	41
149	110
98	120
112	45
131	109
147	155
92	169
85	268
55	97
89	81
52	78
91	47
66	108

76	180
102	40
97	184
87	29
111	192
70	63
143	117
108	73
81	53
107	103
103	44
99	91
135	131
123	107
103	36
129	56
115	85
80	23
93	71
117	133
90	48
94	48
70	74
83	82
109	80
65	38
115	107
100	51
86	78
89	200
134	96
96	155
67	61
138	78
117	31
75	54
111	87
111	152
104	140
66	56
126	112
121	136
101	41
118	148
67	171
117	114
92	73
107	71
122	196
48	39
97	67
94	40
125	81
120	169

112	39
97	85
89	58
112	37
87	48

a) Indicad el tipo de variable considerada.
b) Calculad la media, mediana, desviación típica y los cuartiles, el máximo y el mínimo.
c) Dibujad un histograma de la variable y comentad su forma.
d) Construid un diagrama de caja de la variable y comentad su forma. Indicad si hay datos anómalos o atípicos.
e) Comentad el estudio realizado.

Solución

a) La variable TEMP_CPU es una variable cuantitativa continua.
b) En primer lugar metemos los datos en R en dos variables, una para la variable TEMP_CPU y la otra, para la variable LONG_CODI:

```
TEMP_CPU =
c(127,83,85,93,103,80,71,112,116,62,123,90,103,63,116,103,98,71,103,101,91,125,117,126,1
12,91,89,105,110,149,98,112,131,147,92,85,55,89,52,91,66,76,102,97,87,111,70,143,108,81,
107,103,99,135,123,103,129,115,80,93,117,90,94,70,83,109,65,115,100,86,89,134,96,67,138,
117,75,111,111,104,66,126,121,101,118,67,117,92,107,122,48,97,94,125,120,112,97,89,112,8
7)
LONG_CODI =
c(146,80,60,90,58,88,106,150,38,195,121,109,148,96,92,50,19,84,90,72,97,147,108,135,111,
79,121,194,41,110,120,45,109,155,169,268,97,81,78,47,108,180,40,184,29,192,63,117,73,53,
103,44,91,131,107,36,56,85,23,71,133,48,48,74,82,80,38,107,51,78,200,96,155,61,78,31,54,
87,152,140,56,112,136,41,148,171,114,73,71,196,39,67,40,81,169,39,85,58,37,48)
```

Hallemos la media, la mediana y la desviación típica de la variable TEMP_CPU:

```
mean(TEMP_CPU)

[1] 99.87

median(TEMP_CPU)

[1] 101

sd(TEMP_CPU)

[1] 21.55831
```

Por tanto, la media de la variable vale 99,87, la mediana, 101, la desviación típica, 21,56.

Los cuartiles primero y tercero y el mínimo y el máximo se hallan de la forma siguiente:

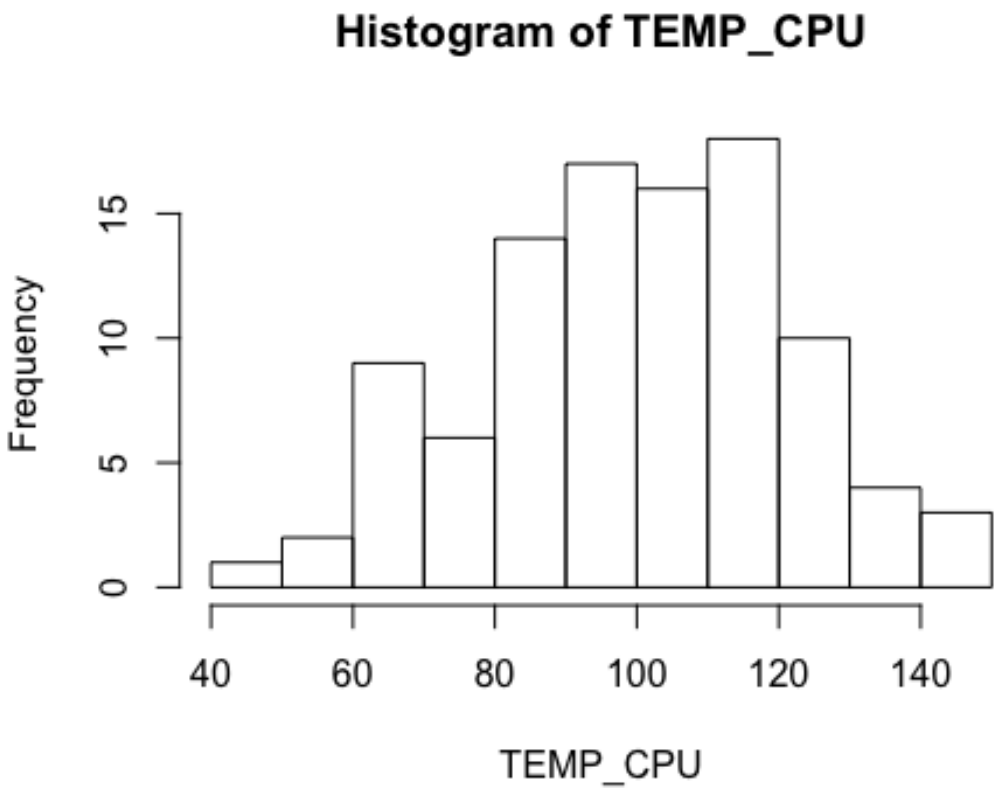
```
quantile(TEMP_CPU,0.25)

25%
 87

quantile(TEMP_CPU,0.75)

   75%
115.25
```

```
min(TEMP_CPU)

[1] 48

max(TEMP_CPU)

[1] 149
```

Los cuantiles primero y tercero valen 87 y 115,75, respectivamente y el máximo y el mínimo, 149 y 48, respectivamente.

c) Para hacer el histograma de la variable, usamos la función hist de R:

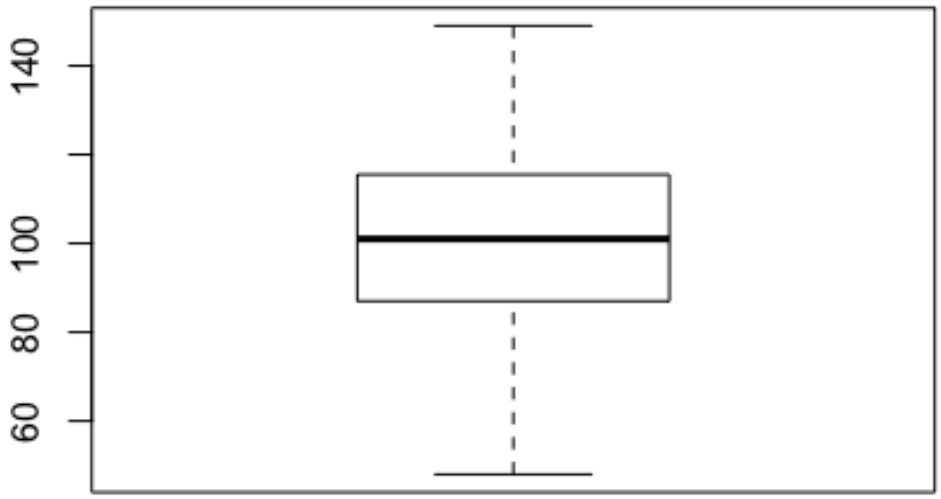
```
hist(TEMP_CPU)
```



Vemos que tiene una forma bastante simétrica, con ningún valor atípico.

d) Para realizar un boxplot, usamos la función boxplot de R.

```
boxplot(TEMP_CPU)
```



Comprobamos una vez más la simetría de la variable viendo cómo la caja del gráfico anterior es simétrica y la inexistencia de datos atípicos.

e) Como conclusión, podemos afirmar que la variable TEMP_CPU es una variable continua con una distribución bastante simétrica, y pocos datos atípicos.

Actividad 2: Cómputo del tiempo de CPU agrupado.

Agrupamiento de datos estadísticos.

Con los datos de la actividad anterior, queremos tabular los datos para estudiar mejor la variable. Para hacerlo distribuiremos los tiempos de ejecución en 3 categorías: “T_corto” (tiempo en el intervalo [48,81]) , “T_medio” (tiempo en el intervalo (81,114]), “T_largo” (tiempo en el intervalo (114,149]) creando la variable CLAS_TEMP. Para estudiar la variable CLAS_TEMP, se pide los resúmenes numéricos que ayuden a entender la distribución de la variable y un gráfico explicativo de la variable.

Solución

Como indica el enunciado de la actividad, agrupamos la variable TEMP_CPU usando la función cut de R indicando los intervalos de agrupamiento de la forma siguiente:

```
CLAS_TEMP =
cut(TEMP_CPU,breaks=c(48,81,114,149),labels=c('T_corto','T_medio','T_largo'),include.lowest = TRUE)
```

Observemos que hemos creador una variable nueva CLAS_TEMP que representa la variable agrupada del tiempo de CPU:

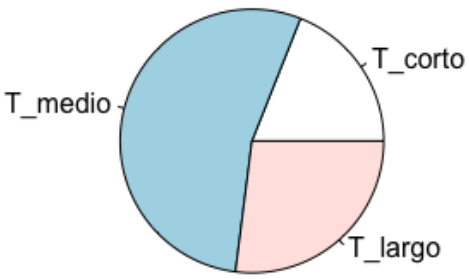
```
CLAS_TEMP

  [1] T_largo T_medio T_medio T_medio T_medio T_corto T_corto T_medio
  [9] T_largo T_corto T_largo T_medio T_medio T_corto T_largo T_medio
 [17] T_medio T_corto T_medio T_medio T_medio T_largo T_largo T_largo
 [25] T_medio T_medio T_medio T_medio T_medio T_largo T_medio T_medio
 [33] T_largo T_largo T_medio T_medio T_corto T_medio T_corto T_medio
 [41] T_corto T_corto T_medio T_medio T_medio T_medio T_corto T_largo
 [49] T_medio T_corto T_medio T_medio T_medio T_largo T_largo T_medio
 [57] T_largo T_largo T_corto T_medio T_largo T_medio T_medio T_corto
 [65] T_medio T_medio T_corto T_largo T_medio T_medio T_medio T_largo
 [73] T_medio T_corto T_largo T_largo T_corto T_medio T_medio T_medio
 [81] T_corto T_largo T_largo T_medio T_largo T_corto T_largo T_medio
 [89] T_medio T_largo T_corto T_medio T_medio T_largo T_largo T_medio
 [97] T_medio T_medio T_medio T_medio
```

Los resúmenes numéricos para la variable CLAS_TEMP será una tabla de frecuencias. Para poder realizarla, usamos la función table de R:

```
 table(CLAS_TEMP)

CLAS_TEMP
T_corto T_medio T_largo
     19      54      27
```

Vemos que los programas de media duración son los más abundantes y los de duración corta, los menos abundantes.
Un posible gráfico explicativo de la variable anterior podría ser un gráfico de sectores. Para ello, usamos la función pie de R:

```
pie(table(CLAS_TEMP))
```



Como podemos observar, las conclusiones son las mismas que hemos comentado antes.

Actividad 3: Inmersión de las tecnologías de la información y comunicación en los municipios.

Medidas de tendencia central. Medidas de dispersión. Histogramas. Diagramas de caja.

En la taba siguiente se recogen los resultados de unas encuestas a diferentes municipios sobre el uso de las TICs el año 2007. De cada municipio tenemos 4 valores: LLAR_ORD (proporción de hogares que tienen ordenador), LLAR_BA (proporción de hogares que tienen banda ancha), USU_ORD (proporción de habitantes que han utilizado el último mes el ordenador) y USU_COR (proporción de habitantes que han utilizado el último mes el correo electrónico). En este problema estudiaremos y compararemos las variables LLAR_ORD y USU_ORD.

LLAR_ORD	LLAR_BA	USU_ORD	USU_COR
64,6188	39,6013	58,9384	45,4938
70,6249	45,6342	64,395	47,4064
64,2484	43,5412	60,4109	48,6097
71,2663	33,7878	52,4718	35,7512
57,6435	32,1987	50,8143	36,3483
64,7489	44,0721	61,5918	43,5046
66,7538	40,5427	56,4004	40,8341
60,4349	41,1113	61,7787	45,5469
60,0854	24,4543	51,2101	33,9002
58,7004	42,2022	57,772	42,7404
63,9532	51,1428	65,7906	50,6901
69,3859	41,2327	61,8521	44,8027
65,0687	49,9265	58,7896	49,5121
60,9454	40,1733	55,2561	40,1754
57,5605	32,8864	62,7563	39,3537
77,3762	41,3325	60,6667	45,946
59,2103	45,6849	63,6107	50,0219
64,2766	32,5023	53,0574	37,4951
67,4305	45,7729	60,4322	42,6967
72,1898	41,6095	62,6842	47,4671
75,7838	47,3866	65,0358	50,2751
63,6186	26,5791	54,3564	36,6763
61,9027	30,9928	59,2544	43,6985
65,3377	43,654	61,1756	53,0282
74,99	34,9094	51,7311	36,2705
65,1809	33,6252	56,6723	38,1024
65,3551	36,09	60,0942	41,2713
68,4741	47,8383	62,4837	49,5373
59,2229	35,9583	56,1627	42,2148
61,2693	30,5968	50,1595	35,9874
62,2441	39,0514	56,7887	41,2898
64,6091	42,3303	60,1923	45,2172
67,4023	38,2239	62,53	46,0961
65,2089	38,4087	62,6382	45,2434
62,7509	37,3973	54,9593	37,0856
71,8195	41,1555	65,9579	47,5231
56,7604	23,373	39,5549	25,8601
63,3384	34,6439	67,8957	45,6673
58,2328	34,372	56,5405	41,8765
64,2704	48,2378	63,2496	50,1087
62,503	46,8367	62,343	48,9784

1) Calculad la media, la mediana, la desviación típica y los cuartiles, el máximo y el mínimo de estas dos variables.

2) Dibujad un histograma de cada una de las dos variables.
3) Construid diagramas de caja de las dos variables. Indicad si hay datos anómalos o atípicos.
4) Comentad los resultados comentando las diferencies-semejanzas entre las dos variables. Indicad que gráfico o que resumen numérico es mas útil en este caso.

Solución

1) En primer lugar, definimos las 4 variables en cuestión en R:

```
LLAR_ORD =
c(64.6188,70.6249,64.2484,71.2663,57.6435,64.7489,66.7538,60.4349,60.0854,58.7004,63.953
2,69.3859,65.0687,60.9454,57.5605,77.3762,59.2103,64.2766,67.4305,72.1898,75.7838,63.618
6,61.9027,65.3377,74.99,65.1809,65.3551,68.4741,59.2229,61.2693,62.2441,64.6091,67.4023,
65.2089,62.7509,71.8195,56.7604,63.3384,58.2328,64.2704,62.503)
LLAR_BA=
c(39.6013,45.6342,43.5412,33.7878,32.1987,44.0721,40.5427,41.1113,24.4543,42.2022,51.142
8,41.2327,49.9265,40.1733,32.8864,41.3325,45.6849,32.5023,45.7729,41.6095,47.3866,26.579
1,30.9928,43.654,34.9094,33.6252,36.09,47.8383,35.9583,30.5968,39.0514,42.3303,38.2239,3
8.4087,37.3973,41.1555,23.373,34.6439,34.372,48.2378,46.8367)
USU_ORD =
c(58.9384,64.395,60.4109,52.4718,50.8143,61.5918,56.4004,61.7787,51.2101,57.772,65.7906,
61.8521,58.7896,55.2561,62.7563,60.6667,63.6107,53.0574,60.4322,62.6842,65.0358,54.3564,
59.2544,61.1756,51.7311,56.6723,60.0942,62.4837,56.1627,50.1595,56.7887,60.1923,62.53,62
.6382,54.9593,65.9579,39.5549,67.8957,56.5405,63.2496,62.343)
USU_COR =
c(45.4938,47.4064,48.6097,35.7512,36.3483,43.5046,40.8341,45.5469,33.9002,42.7404,50.690
1,44.8027,49.5121,40.1754,39.3537,45.946,50.0219,37.4951,42.6967,47.4671,50.2751,36.6763
,43.6985,53.0282,36.2705,38.1024,41.2713,49.5373,42.2148,35.9874,41.2898,45.2172,46.0961
,45.2434,37.0856,47.5231,25.8601,45.6673,41.8765,50.1087,48.9784)
```

La media, la mediana, la desviación típica y los cuartiles, el máximo y el mínimo de la variable LLAR_ORD valen:

```
mean(LLAR_ORD)

[1] 64.79993

median(LLAR_ORD)

[1] 64.2766

sd(LLAR_ORD)

[1] 5.060672

quantile(LLAR_ORD,0.25)

    25%
61.2693

quantile(LLAR_ORD,0.75)

    75%
67.4023

max(LLAR_ORD)

[1] 77.3762

min(LLAR_ORD)

[1] 56.7604
```

Y para la variable USU_ORD:

```
mean(USU_ORD)

[1] 58.79159
```

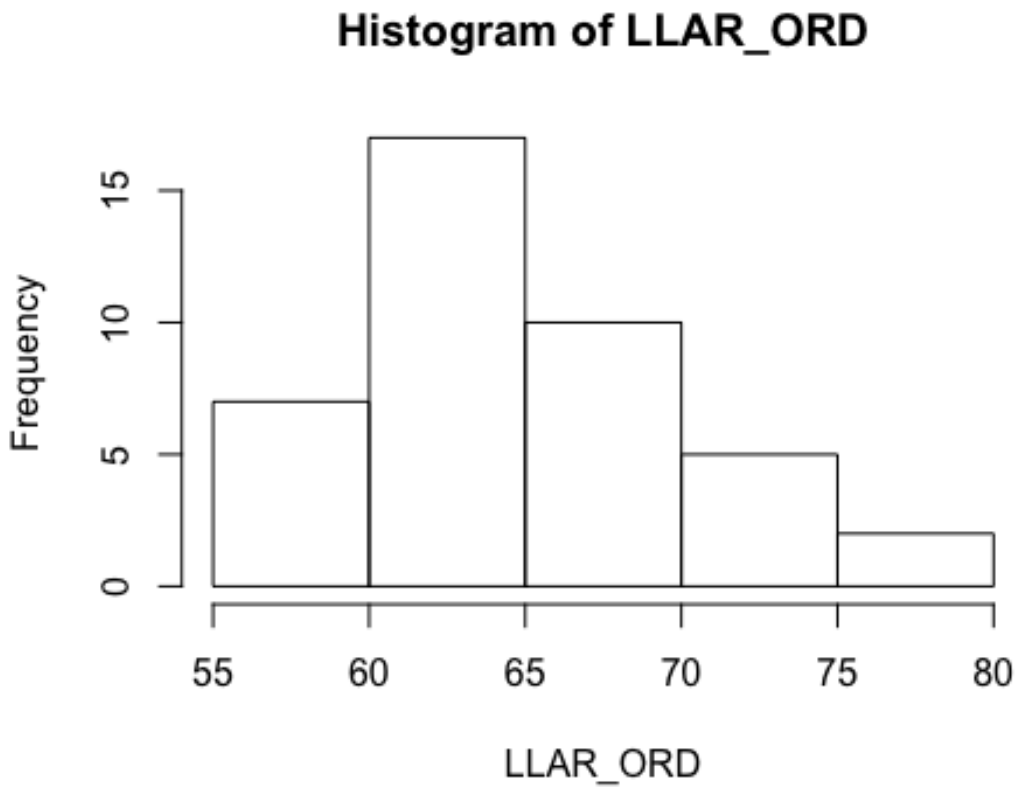
```
median(USU_ORD)

[1] 60.1923

sd(USU_ORD)

[1] 5.453406

quantile(USU_ORD,0.25)

    25%
56.1627

quantile(USU_ORD,0.75)

  75%
62.53

max(USU_ORD)

[1] 67.8957

min(USU_ORD)

[1] 39.5549
```

Podemos escribir los resultados anteriores en forma de tabla usando la función data.frame de R de la forma siguiente:

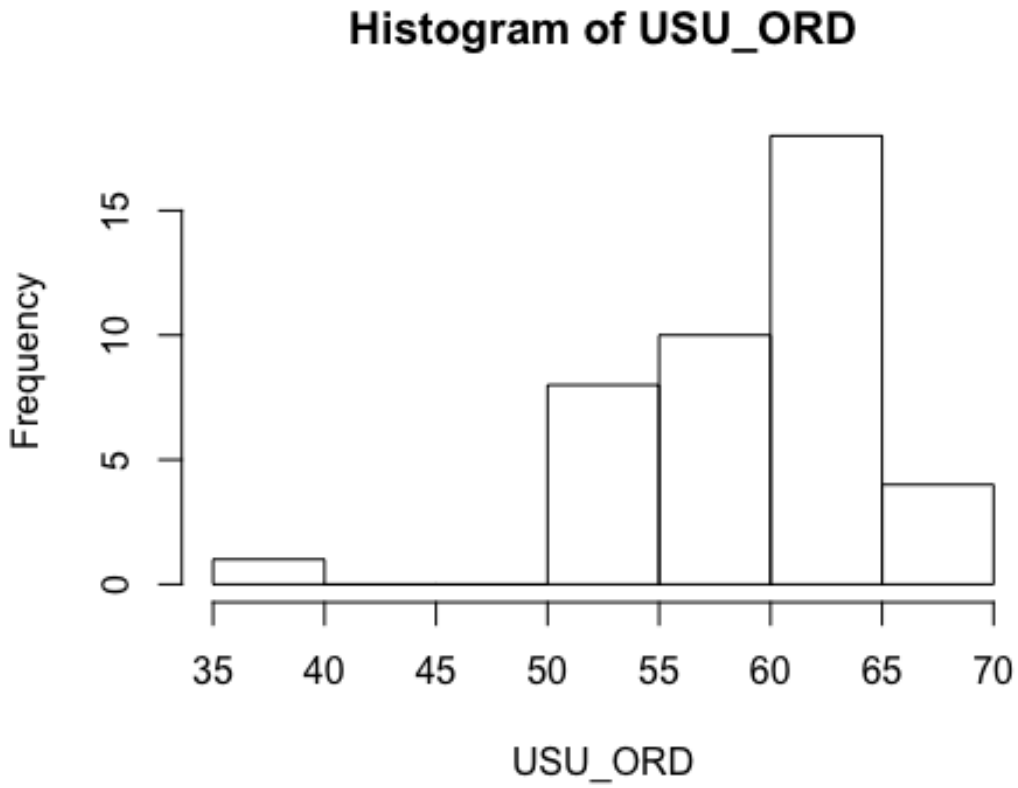
```
resultados=data.frame(c("media","mediana","desviación típica","cuantil 25", "cuantil
75","máximo", "mínimo"),
          c(mean(LLAR_ORD),median(LLAR_ORD),sd(LLAR_ORD),
            quantile(LLAR_ORD,0.25),quantile(LLAR_ORD,0.75),
max(LLAR_ORD), min(LLAR_ORD)),
c(mean(USU_ORD),median(USU_ORD),sd(USU_ORD),
            quantile(USU_ORD,0.25),quantile(USU_ORD,0.75),
max(USU_ORD),min(USU_ORD)))


names(resultados)=c("Estadísticos","LLAR_ORD","USU_ORD")
resultados

       Estadísticos  LLAR_ORD   USU_ORD
1             media 64.799934 58.791588
2           mediana 64.276600 60.192300
3 desviación típica  5.060672  5.453406
4        cuantil 25 61.269300 56.162700
5        cuantil 75 67.402300 62.530000
6            máximo 77.376200 67.895700
7            mínimo 56.760400 39.554900
```

2) Los histogramas se realizan usando la función hist de R:

```
hist(LLAR_ORD)
```


Histogram of LLAR_ORD
Frequency
0
5
10
15
55
60
65
70
75
80
LLAR_ORD

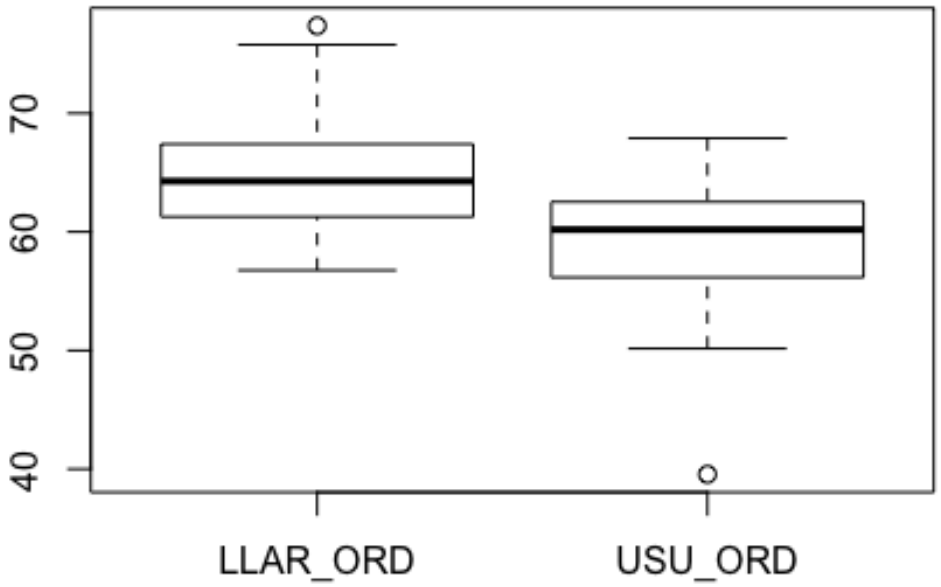
```
hist(USU_ORD)
```


Histogram of USU_ORD
Frequency
0
5
10
15
35
40
45
50
55
60
65
70
USU_ORD

Vemos que el histograma de la variable LLAR_ORD tiene una cierta simetría mientras que el histograma de la variable USU_ORD es asimétrico con cola hacia la izquierda.

3) Las diagramas de caja se construyen usando la función boxplot de R:

```
boxplot(LLAR_ORD,USU_ORD,names=c("LLAR_ORD","USU_ORD"))
```



En dicho gráfico, volvemos a comprobar cierta simetría en la variable LLAR_ORD y la asimetría de la variable USU_ORD con unos pocos datos atípicos en las dos variables.

4) En los gráficos anteriores vemos que la proporción de hogares con ordenador (LLA_ORD) es mayor que la proporción de usuarios que han usado el ordenador en el último mes (USU_ORD) con una cierta simetría en el primer caso y una asimetría en el segundo. El gráfico más conveniente para realizar dicha comparación es el diagrama de caja.

Actividad 4: Tiempo de computación de programas informáticos.

Agrupamiento de datos estadísticos. Medidas de tendencia central. Medidas de dispersión. Regla de Txebyschev.

El tiempo de computación (en segundos) de un determinado programa informático ejecutado de forma independiente 100 veces en una misma máquina vale: 4.67, 4.94, 5.09, 4.74, 4.63, 4.62, 4.53, 4.89, 5.12, 4.78, 4.51, 5.17, 4.53, 4.64, 4.57, 4.92, 5.15, 4.51, 4.57, 4.86, 4.64, 4.66, 4.98, 4.71, 5.07, 5.14, 4.54, 4.90, 4.88, 4.91, 5.16, 4.99, 5.19, 4.62, 4.56, 4.81, 5.10, 5.12, 4.69, 4.77, 5.04, 4.61, 4.72, 4.85, 5.20, 4.55, 4.52, 4.83, 5.09, 4.76, 4.64, 4.86, 4.68, 5.03, 4.57, 5.17, 4.56, 4.99, 4.95, 4.92, 4.70, 4.89, 5.01, 4.60, 4.65, 4.95, 4.79, 4.55, 5.01, 4.92, 4.60, 4.63, 4.77, 4.93, 4.85, 4.70, 4.78, 4.68, 5.02, 4.87, 4.72, 4.66, 4.66, 4.83, 4.87, 4.66, 5.08, 4.83, 4.75, 5.11, 4.81, 4.66, 4.68, 5.03, 5.02, 5.04, 4.82, 4.62, 4.92, 4.90.

Se pide:

a) Agrupad la variable “tiempo de computación” en intervalos de amplitud 0.138 empezando con el valor 4.51. Calculad una tabla de frecuencias donde se indique frecuencias absolutas, relativas, absolutas acumuladas y relativas acumuladas.
b) Calculad la media y la mediana de la variable agrupada “tiempo de computación agrupado”.
c) Calculad la desviación típica también de la variable agrupada.
d) Suponemos ahora que ejecutamos de forma independiente 1000 veces el programa obteniendo la misma media y la misma desviación típica que en los apartados anteriores. ¿Entre qué valores se encuentran como mínimo el 88.8% de los tiempos de computación?

Solución

a) Primero definimos la variable tiempo de computación en R:

```
TEMP_COMP= c(4.67, 4.94, 5.09, 4.74, 4.63, 4.62, 4.53, 4.89, 5.12, 4.78, 4.51, 5.17,
4.53, 4.64, 4.57, 4.92, 5.15, 4.51, 4.57, 4.86, 4.64, 4.66, 4.98, 4.71, 5.07, 5.14,
4.54, 4.90, 4.88, 4.91, 5.16, 4.99, 5.19, 4.62, 4.56, 4.81, 5.10, 5.12, 4.69, 4.77,
5.04, 4.61, 4.72, 4.85, 5.20, 4.55, 4.52, 4.83, 5.09, 4.76, 4.64, 4.86, 4.68, 5.03,
4.57, 5.17, 4.56, 4.99, 4.95, 4.92, 4.70, 4.89, 5.01, 4.60, 4.65, 4.95, 4.79, 4.55,
5.01, 4.92, 4.60, 4.63, 4.77, 4.93, 4.85, 4.70, 4.78, 4.68, 5.02, 4.87, 4.72, 4.66,
4.66, 4.83, 4.87, 4.66, 5.08, 4.83, 4.75, 5.11, 4.81, 4.66, 4.68, 5.03, 5.02, 5.04,
4.82, 4.62, 4.92, 4.90)
```

Seguidamente vamos a hallar el máximo de la columna anterior. Para hacerlo, usamos la función max de R:

```
max(TEMP_COMP)

[1] 5.2
```

Por tanto, el máximo de la variable “tiempo de computación” es 5,2. Los intervalos serán los siguientes:

[4,51, 4,648], (4.648, 4,786], (4,786, 4,924], (4,924, 5,062], (5,062, 5,2].

Para agrupar la variable usamos la función cut de R de la forma siguiente:

```
TEMP_COMP_AGRUP= cut(TEMP_COMP,breaks=seq(from=4.51,to=5.2,by=0.138),include.lowest=
TRUE)

TEMP_COMP_AGRUP

  [1] (4.65,4.79] (4.92,5.06] (5.06,5.2]  (4.65,4.79] [4.51,4.65]
  [6] [4.51,4.65] [4.51,4.65] (4.79,4.92] (5.06,5.2]  (4.65,4.79]
 [11] [4.51,4.65] (5.06,5.2]  [4.51,4.65] [4.51,4.65] [4.51,4.65]
 [16] (4.79,4.92] (5.06,5.2]  [4.51,4.65] [4.51,4.65] (4.79,4.92]
 [21] [4.51,4.65] (4.65,4.79] (4.92,5.06] (4.65,4.79] (5.06,5.2]
 [26] (5.06,5.2]  [4.51,4.65] (4.79,4.92] (4.79,4.92] (4.79,4.92]
 [31] (5.06,5.2]  (4.92,5.06] (5.06,5.2]  [4.51,4.65] [4.51,4.65]
 [36] (4.79,4.92] (5.06,5.2]  (5.06,5.2]  (4.65,4.79] (4.65,4.79]
 [41] (4.92,5.06] [4.51,4.65] (4.65,4.79] (4.79,4.92] (5.06,5.2]
 [46] [4.51,4.65] [4.51,4.65] (4.79,4.92] (5.06,5.2]  (4.65,4.79]
 [51] [4.51,4.65] (4.79,4.92] (4.65,4.79] (4.92,5.06] [4.51,4.65]
 [56] (5.06,5.2]  [4.51,4.65] (4.92,5.06] (4.92,5.06] (4.79,4.92]
 [61] (4.65,4.79] (4.79,4.92] (4.92,5.06] [4.51,4.65] (4.65,4.79]
 [66] (4.92,5.06] (4.79,4.92] [4.51,4.65] (4.92,5.06] (4.79,4.92]
 [71] [4.51,4.65] [4.51,4.65] (4.65,4.79] (4.92,5.06] (4.79,4.92]
 [76] (4.65,4.79] (4.65,4.79] (4.65,4.79] (4.92,5.06] (4.79,4.92]
 [81] (4.65,4.79] (4.65,4.79] (4.65,4.79] (4.79,4.92] (4.79,4.92]
 [86] (4.65,4.79] (5.06,5.2]  (4.79,4.92] (4.65,4.79] (5.06,5.2]
 [91] (4.79,4.92] (4.65,4.79] (4.65,4.79] (4.92,5.06] (4.92,5.06]
 [96] (4.92,5.06] (4.79,4.92] [4.51,4.65] (4.79,4.92] (4.79,4.92]
Levels: [4.51,4.65] (4.65,4.79] (4.79,4.92] (4.92,5.06] (5.06,5.2]
```

Para hallar la tabla de frecuencias pedida, hallamos las frecuencias absolutas usando la función table:

```
table(TEMP_COMP_AGRUP)

TEMP_COMP_AGRUP
[4.51,4.65] (4.65,4.79] (4.79,4.92] (4.92,5.06]  (5.06,5.2]
         24          23          23          15          15
```

las frecuencias relativas usando la función prop.table:

```
prop.table(table(TEMP_COMP_AGRUP))

TEMP_COMP_AGRUP
[4.51,4.65] (4.65,4.79] (4.79,4.92] (4.92,5.06]  (5.06,5.2]
       0.24        0.23        0.23        0.15        0.15
```

las frecuencias absolutas acumulados con la función cumsum:

```
cumsum(table(TEMP_COMP_AGRUP))

[4.51,4.65] (4.65,4.79] (4.79,4.92] (4.92,5.06]  (5.06,5.2]
         24          47          70          85         100
```

y las frecuencias relativas acumuladas con la función cumsum aplicada al resultados de la función prop.table:

```
cumsum(prop.table(table(TEMP_COMP_AGRUP)))

[4.51,4.65] (4.65,4.79] (4.79,4.92] (4.92,5.06]  (5.06,5.2]
       0.24        0.47        0.70        0.85        1.00
```

Para escribir todos los resultados anteriores en forma de tabla resumen, podemos usar la función cbind de R de la forma siguiente:

```
resultados2=cbind(table(TEMP_COMP_AGRUP),prop.table(table(TEMP_COMP_AGRUP)),cumsum(table
(TEMP_COMP_AGRUP)),cumsum(prop.table(table(TEMP_COMP_AGRUP))))
colnames(resultados2) = c("Frec. abs.","Frec. rel.","Frec. abs. acum.","Frec. rel.
acum.")
resultados2

            Frec. abs. Frec. rel. Frec. abs. acum. Frec. rel. acum.
[4.51,4.65]         24       0.24               24             0.24
(4.65,4.79]         23       0.23               47             0.47
(4.79,4.92]         23       0.23               70             0.70
(4.92,5.06]         15       0.15               85             0.85
(5.06,5.2]          15       0.15              100             1.00
```

b) y c) Para poder hallar la media, mediana y desviación típica de la variable agrupada anterior, necesitamos hallar la marca de clase para todos los intervalos anteriores. Primero hallamos los extremos de la izquierda y derecha de los intervalos considerados, a continuación hallamos las marcas de clase y por último, mostramos la tabla de frecuencias anterior incluyendo las marcas de clase halladas:

```
extremos_izquierda = seq(from=4.51,to=5.2-0.138,by=0.138)
extremos_derecha = seq(from=4.51+0.138,to=5.2,by=0.138)
marcas_clase = (extremos_izquierda+extremos_derecha)/2
marcas_clase

[1] 4.579 4.717 4.855 4.993 5.131

resultados3=cbind(marcas_clase,table(TEMP_COMP_AGRUP),prop.table(table(TEMP_COMP_AGRUP))
,cumsum(table(TEMP_COMP_AGRUP)),cumsum(prop.table(table(TEMP_COMP_AGRUP))))
colnames(resultados3) = c("Marcas clase","F. abs.","F. rel.","F. abs. acum.","F. rel.
acum.")
resultados3

            Marcas clase F. abs. F. rel. F. abs. acum. F. rel. acum.
[4.51,4.65]        4.579      24    0.24            24          0.24
(4.65,4.79]        4.717      23    0.23            47          0.47
(4.79,4.92]        4.855      23    0.23            70          0.70
(4.92,5.06]        4.993      15    0.15            85          0.85
(5.06,5.2]         5.131      15    0.15           100          1.00
```

Para hallar la media, media, la mediana y la desviación típica, redefinimos la variable TEMP_COMP_AGRUP usando como identificador de cada intervalo la marca de clase:

```
TEMP_COMP_AGRUP2=cut(TEMP_COMP,breaks=seq(from=4.51,to=5.2,by=0.138),include.lowest =
TRUE,labels=marcas_clase)
TEMP_COMP_AGRUP2=as.numeric(as.character(TEMP_COMP_AGRUP2))
```

El resultado de la función cut es una variable tipo factor. Por tanto, antes de hallar la media, mediana y varianza, hemos tenido que transformar la variable tipo factor TEMP_COMP_AGRUP2 en una variable tipo numérico usando las funciones as.character y as.numeric de R. Por último, hallemos la media, mediana y desviación típica pedidas:

```
mean(TEMP_COMP_AGRUP2)

[1] 4.81912

median(TEMP_COMP_AGRUP2)

[1] 4.855

sd(TEMP_COMP_AGRUP2)

[1] 0.1897845
```

Por tanto, la media vale 4,8191, la mediana, 4,8550 y la desviación típica 0,1898.

d) Para realizar dicho apartado, hemos de usar la desigualdad de Txebyxev. El intervalo pedido es de la forma $\left(\bar{x}-ms_x,\bar{x}+ms_x\right)$, donde m cumple: $1-\frac{1}{m^2}=0,889$. Hallando m de la igualdad anterior, tenemos que m=3. Seguidamente hemos de calcular la media y la desviación típica de la variable no agrupada:

```
mean(TEMP_COMP)

[1] 4.8199

sd(TEMP_COMP)

[1] 0.1960442
```

La media será 4,8199 y la desviación típica, 0,1960.
El intervalo será:

$$(4.8199-3\cdot 0.1960, 4.8199+3\cdot 0.1960)=(4.2318, 5.4080)$$

Por tanto, podemos afirmar que 889 de los 1000 tiempos de ejecución estarán entre 4,2318 y 5,4080.

Actividad 5: Número de mensajes no deseados que recibe una empresa.

Datos estadísticos discretos. Tabla de frecuencias. Diagrama de puntos. Medidas de tendencia central. Medidas de dispersión. Regla de Txebyschev.

La siguiente tabla nos indica el número de mensajes "SPAM" que reciben en un día cualquiera los empleados de una determinada empresa:

Número "SPAM"	0	1	2	3	4	5	6	7
Número de empleados	7	11	10	7	1	2	1	1

La tabla anterior se ha de interpretar así: 7 empleados no reciben ningún "SPAM" en el día considerado, 11 empleados reciben 1 "SPAM" en el día considerado, etc. Consideramos la variable X= "número de mensajes "SPAM" que recibe un empleado cualquiera de esta empresa por día". Se pide:

a) Calculad una tabla de frecuencias donde se indique las frecuencias absolutas, relativas, absolutas acumuladas y relativas acumuladas.
b) Haced un diagrama de puntos de la variable X.
c) En base a los apartados anteriores comentad cómo es la variable X (forma, distribución...)
d) ¿Qué porcentaje de empleados reciben entre 2 y 6 mensajes SPAM cada día?
e) Calcular la media, moda y la mediana de X.
f) La desviación típica de X.
g) Si se duplica el número de mensajes "SPAM" que recibe cada empleado, ¿cuáles serán las nuevas media y variancia del número de mensajes "SPAM" que recibe un empleado cualquiera por día? Este apartado se ha de hacer aplicando las propiedades de la media y la desviación típica sin volver a calcular la media y la desviación típica de la nueva variable.
h) En otra empresa, en la que la media y la desviación típica del nombre de mensajes "SPAM" que recibe cada empleado son las mismas que en los apartados a) y b), ¿entre qué valores se encuentran como mínimo el 75% de las observaciones?

Solución

a) En primer lugar, introducimos las variables que nos dan el número de mensajes SPAM y el número de empleados respectivamente:

```
SPAM=0:7
EMPLEADOS=c(7,11,10,7,1,2,1,1)
```

A continuación, creamos la variable de estudio, "número de SPAMS que recibe un empleado" llamándola SPAM_EMPLEADOS, definiéndola de la forma siguiente: repetimos los valores de la columna "SPAM" tantas veces como indica la variable "EMPLEADOS":

```
SPAM_EMPLEADOS=rep(SPAM,EMPLEADOS)
SPAM_EMPLEADOS
```

```
[1] 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3  4 5 5 6 7
```

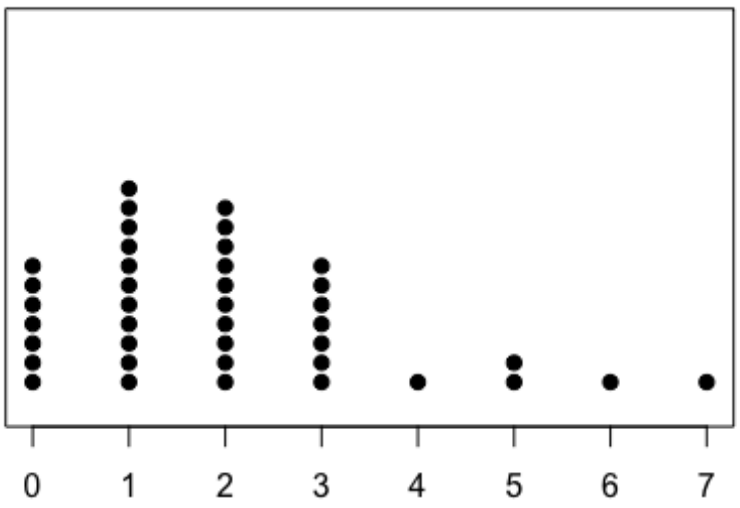
O sea, repetimos el valor 0, 7 veces, el valor 1, 11 veces y así sucesivamente.
Para hallar la tabla de frecuencias, usamos el mismo procedimiento que hemos usado en el apartado a) de la actividad 4:

```
resultados=cbind(table(SPAM_EMPLEADOS),prop.table(table(SPAM_EMPLEADOS)),cumsum(table(SP
AM_EMPLEADOS)),cumsum(prop.table(table(SPAM_EMPLEADOS))))


colnames(resultados) = c("Frec. abs.","Frec. rel.","Frec. abs. acum.","Frec. rel.
acum.")


resultados

  Frec. abs. Frec. rel. Frec. abs. acum. Frec. rel. acum.
0          7      0.175                 7            0.175
1         11      0.275                18            0.450
2         10      0.250                28            0.700
3          7      0.175                35            0.875
4          1      0.025                36            0.900
5          2      0.050                38            0.950
6          1      0.025                39            0.975
7          1      0.025                40            1.000
```

Las dos primeras columnas coinciden con las variables SPAM y EMPLEADOS que indican los valores de la variable y las frecuencias absolutas, respectivamente.

b) Para poder realizar un diagrama de puntos, podemos usar la función stripchart de R:

```
stripchart(SPAM_EMPLEADOS, method = "stack", offset = .5, at = .15, pch = 19)
```



c) Vemos que la variable de estudio X tiene una distribución asimétrica con cola a la derecha donde los valores más repetidos son 0, 1, 2 y 3. Por tanto, la mayoría de los empleados reciben o ningún, o uno, o dos o tres SPAMS diarios.

d) El número de empleados que reciben entre 2 y 6 SPAMS son 21=39-18. O sea, frecuencia absoluta acumulada de 6 menos frecuencia absoluta acumulada de 1. El porcentaje de empleados será: $\frac{21}{40}\cdot 100=52,5\%$.

e) y f) La media, la mediana y la desviación típica de X serán:

```
mean(SPAM_EMPLEADOS)

[1] 1.975

median(SPAM_EMPLEADOS)

[1] 2

sd(SPAM_EMPLEADOS)

[1] 1.671595
```

Por tanto, la media de mensajes SPAMS recibidos por día es de 1,975, la mediana es de 2 y la desviación típica, de 1,672. La moda, como puede observarse en el diagrama de puntos anterior, vale 1.

g) Si se duplican el número de mensajes recibidos por día, también se duplicaran la media y la desviación típica. Por tanto, éstas serán: $\bar{x}=1,975\cdot 2=3,95, s_x=1,672\cdot 2=3,344.$ Comprobemos dichos resultados en R definiendo una nueva variable que sea el doble de la variable SPAM_EMPLEADOS:

```
SPAM_EMPLEADOS2 = 2*SPAM_EMPLEADOS
mean(SPAM_EMPLEADOS2)

[1] 3.95

sd(SPAM_EMPLEADOS2)
```

```
[1] 3.343191
```

Podemos comprobar que los valores de la media y la desviación típica coinciden con los valores calculados anteriormente.

h) Para realizar dicho apartado, hemos de usar la desigualdad de Txebyxev. El intervalo pedido es de la forma $\left(\bar{x}-ms_x,\bar{x}+ms_x\right)$, donde m cumple: $1-\frac{1}{m^2}=0,75$ Hallando m de la igualdad anterior, tenemos que m=2. El intervalo será: $\left(1,975-2\cdot 1,672,1,975+2\cdot 1,672\right)=\left(-1,369;5,319\right).$

Actividad 6: Conocimiento de los lenguajes de programación actuales por parte de los estudiantes de ciencias de la computación.

Agrupamiento de datos estadísticos. Medidas de tendencia central. Medidas de dispersión. Histogramas. Regla de Txebyschev.

La siguiente tabla nos indica el conocimiento de los lenguajes de programación Java, Perl y Python por parte de 25 estudiantes de ciencias de la computación de 0 (ningún conocimiento) a 100 (máximo dominio del lenguaje):

Java	Perl	Python
50.27	75.85	20.30
50.51	74.04	19.68
49.58	74.32	20.41
50.09	75.04	20.00
50.27	74.68	20.43
50.72	74.93	19.78
49.22	75.19	20.13
50.09	74.34	20.21
51.37	75.79	20.29
49.93	75.53	19.71
50.58	75.64	20.38
48.92	75.19	19.71
50.21	75.20	19.62
49.81	75.29	19.31
50.69	74.40	20.25
48.50	75.59	19.63
48.77	73.36	19.88
48.56	73.22	20.42
49.39	74.44	19.58
50.79	75.42	20.19
48.67	74.29	20.04
51.03	75.38	20.57
49.73	74.91	20.22
52.20	75.38	20.28
50.32	73.83	19.86

Se pide:

a) Agrupad las variables “conocimiento del lenguaje de programación Java, Perl y Python” en 5 intervalos de igual amplitud. Calculad una tabla de frecuencias donde se indique frecuencias absolutas, relativas, absolutas acumuladas y relativas acumuladas.
b) Calculad la media y la mediana de las variables agrupadas.
c) Calculad la desviación típica también de las variable agrupadas.
d) Haced un histograma de cada una de las variables.
e) Comentad los resultados comentando las diferencias-semejanzas entre las tres variables.

Solución

a) En primer lugar metemos los datos en tres variables cuyos nombres serán java, perl y python:

```
> java <-
c(50.27,50.51,49.58,50.09,50.27,50.72,49.22,50.09,51.37,49.93,50.58,48.92,50.21,49.81,50.69,48.50,48.77,48.56,49.39,50.79
,48.67,51.03,49.73,52.20,50.32)
> perl <-
c(75.85,74.04,74.32,75.04,74.68,74.93,75.19,74.34,75.79,75.53,75.64,75.19,75.20,75.29,74.40,75.59,73.36,73.22,74.44,75.42
,74.29,75.38,74.91,75.38,73.83)
> python <-
c(20.30,19.68,20.41,20.00,20.43,19.78,20.13,20.21,20.29,19.71,20.38,19.71,19.62,19.31,20.25,19.63,19.88,20.42,19.58,20.19
,20.04,20.57,20.22,20.28,19.86)
```

A continuación agrupamos las variables. En primer lugar, calculamos los intervalos de agrupamiento de cada variable:

```
> (intervalos_java <- seq(from=min(java),to=max(java),by=(max(java)-min(java))/5))
[1] 48.50 49.24 49.98 50.72 51.46 52.20
> (intervalos_perl <- seq(from=min(perl),to=max(perl),by=(max(perl)-min(perl))/5))
[1] 73.220 73.746 74.272 74.798 75.324 75.850
> (intervalos_python <- seq(from=min(python),to=max(python),by=(max(python)-min(python))/5))
[1] 19.310 19.562 19.814 20.066 20.318 20.570
>
```

Por ejemplo, para la variable java, los intervalos serán: [48.5,49.24), [49.24,49.98), [49.98,50.72), [50.72,51.46) y [51.46,52.20). Los demás se calculan de forma similar.

A continuación calculamos las marcas de clase de los intervalos:

```
>
> marcas_clase_java <- c()
> for (i in 1:5){marcas_clase_java <- c(marcas_clase_java,(intervalos_java[i+1]+intervalos_java[i])/2)}
> marcas_clase_java
[1] 48.87 49.61 50.35 51.09 51.83
> marcas_clase_perl <- c()
> for (i in 1:5){marcas_clase_perl <- c(marcas_clase_perl,(intervalos_perl[i+1]+intervalos_perl[i])/2)}
> marcas_clase_perl
[1] 73.483 74.009 74.535 75.061 75.587
> marcas_clase_python <- c()
> for (i in 1:5){marcas_clase_python <- c(marcas_clase_python,(intervalos_python[i+1]+intervalos_python[i])/2)}
> marcas_clase_python
[1] 19.436 19.688 19.940 20.192 20.444
>
```

A continuación, calculamos las variables agrupadas:

```
> (java_agrup <- cut(java,breaks=intervalos_java,right=F,include.lowest=T))
 [1] [50,50.7)   [50,50.7)   [49.2,50)   [50,50.7)   [50,50.7)   [50.7,51.5) [48.5,49.2)
 [8] [50,50.7)   [50.7,51.5) [49.2,50)   [50,50.7)   [48.5,49.2) [50,50.7)   [49.2,50)
[15] [50,50.7)   [48.5,49.2) [48.5,49.2) [48.5,49.2) [49.2,50)   [50.7,51.5) [48.5,49.2)
[22] [50.7,51.5) [49.2,50)   [51.5,52.2] [50,50.7)
Levels: [48.5,49.2) [49.2,50) [50,50.7) [50.7,51.5) [51.5,52.2]
> (perl_agrup <- cut(perl,breaks=intervalos_perl,right=F,include.lowest=T))
 [1] [75.3,75.8] [73.7,74.3) [74.3,74.8) [74.8,75.3) [74.3,74.8) [74.8,75.3) [74.8,75.3)
 [8] [74.3,74.8) [75.3,75.8] [75.3,75.8] [75.3,75.8] [74.8,75.3) [74.8,75.3) [74.8,75.3)
[15] [74.3,74.8) [75.3,75.8] [73.2,73.7) [73.2,73.7) [74.3,74.8) [75.3,75.8] [74.3,74.8)
[22] [75.3,75.8] [74.8,75.3) [75.3,75.8] [73.7,74.3)
Levels: [73.2,73.7) [73.7,74.3) [74.3,74.8) [74.8,75.3) [75.3,75.8]
> (python_agrup <- cut(python,breaks=intervalos_python,right=F,include.lowest=T))
 [1] [20.1,20.3) [19.6,19.8) [20.3,20.6] [19.8,20.1) [20.3,20.6] [19.6,19.8) [20.1,20.3)
 [8] [20.1,20.3) [20.1,20.3) [19.6,19.8) [20.3,20.6] [19.6,19.8) [19.6,19.8) [19.3,19.6)
[15] [20.1,20.3) [19.6,19.8) [19.8,20.1) [20.3,20.6] [19.6,19.8) [20.1,20.3) [19.8,20.1)
[22] [20.3,20.6] [20.1,20.3) [20.1,20.3) [19.8,20.1)
Levels: [19.3,19.6) [19.6,19.8) [19.8,20.1) [20.1,20.3) [20.3,20.6]
> |
```

Las frecuencias absolutas se calculan con la instrucción table:

```
> (frec_abs_java <- table(java_agrup))
java_agrup
[48.5,49.2)   [49.2,50)   [50,50.7) [50.7,51.5) [51.5,52.2]
          6           5           9           4           1
> (frec_abs_perl <- table(perl_agrup))
perl_agrup
[73.2,73.7) [73.7,74.3) [74.3,74.8) [74.8,75.3) [75.3,75.8]
          2           2           6           7           8
> (frec_abs_python <- table(python_agrup))
python_agrup
[19.3,19.6) [19.6,19.8) [19.8,20.1) [20.1,20.3) [20.3,20.6]
          1           7           4           8           5
> |
```

Las frecuencias relativas serán:

```
> (frec_rel_java <- table(java_agrup)/sum(table(java_agrup)))
java_agrup
[48.5,49.2)   [49.2,50)   [50,50.7) [50.7,51.5) [51.5,52.2]
       0.24        0.20        0.36        0.16        0.04
> (frec_rel_perl <- table(perl_agrup)/sum(table(perl_agrup)))
perl_agrup
[73.2,73.7) [73.7,74.3) [74.3,74.8) [74.8,75.3) [75.3,75.8]
       0.08        0.08        0.24        0.28        0.32
> (frec_rel_python <- table(python_agrup)/sum(table(python_agrup)))
python_agrup
[19.3,19.6) [19.6,19.8) [19.8,20.1) [20.1,20.3) [20.3,20.6]
       0.04        0.28        0.16        0.32        0.20
> |
```

Las frecuencias absolutas acumuladas serán:

```
>
> (frec_abs_acum_java <- cumsum(frec_abs_java))
[48.5,49.2)   [49.2,50)   [50,50.7) [50.7,51.5) [51.5,52.2]
          6          11          20          24          25
> (frec_abs_acum_perl <- cumsum(frec_abs_perl))
[73.2,73.7) [73.7,74.3) [74.3,74.8) [74.8,75.3) [75.3,75.8]
          2           4          10          17          25
> (frec_abs_acum_python <- cumsum(frec_abs_python))
[19.3,19.6) [19.6,19.8) [19.8,20.1) [20.1,20.3) [20.3,20.6]
          1           8          12          20          25
> |
```

Las frecuencias relativas acumuladas serán:

```
>
> (frec_rel_acum_java <- cumsum(frec_rel_java))
[48.5,49.2)   [49.2,50)   [50,50.7) [50.7,51.5) [51.5,52.2]
       0.24        0.44        0.80        0.96        1.00
> (frec_rel_acum_perl <- cumsum(frec_rel_perl))
[73.2,73.7) [73.7,74.3) [74.3,74.8) [74.8,75.3) [75.3,75.8]
       0.08        0.16        0.40        0.68        1.00
> (frec_rel_acum_python <- cumsum(frec_rel_python))
[19.3,19.6) [19.6,19.8) [19.8,20.1) [20.1,20.3) [20.3,20.6]
       0.04        0.32        0.48        0.80        1.00
> |
```

Por último, calculamos la tabla de frecuencias para cada variable:

```
>
> (tabla_frec_java <- cbind(marcas_clase_java,frec_abs_java,frec_rel_java,frec_abs_acum_java,frec_rel_acum_java))
          marcas_clase_java frec_abs_java frec_rel_java frec_abs_acum_java frec_rel_acum_java
[48.5,49.2)             48.87            6         0.24                 6               0.24
[49.2,50)               49.61            5         0.20                11               0.44
[50,50.7)               50.35            9         0.36                20               0.80
[50.7,51.5)             51.09            4         0.16                24               0.96
[51.5,52.2]             51.83            1         0.04                25               1.00
> (tabla_frec_perl <- cbind(marcas_clase_perl,frec_abs_perl,frec_rel_perl,frec_abs_acum_perl,frec_rel_acum_perl))
          marcas_clase_perl frec_abs_perl frec_rel_perl frec_abs_acum_perl frec_rel_acum_perl
[73.2,73.7)            73.483            2         0.08                 2               0.08
[73.7,74.3)            74.009            2         0.08                 4               0.16
[74.3,74.8)            74.535            6         0.24                10               0.40
[74.8,75.3)            75.061            7         0.28                17               0.68
[75.3,75.8]            75.587            8         0.32                25               1.00
> (tabla_frec_python <-
cbind(marcas_clase_python,frec_abs_python,frec_rel_python,frec_abs_acum_python,frec_rel_acum_python))
          marcas_clase_python frec_abs_python frec_rel_python frec_abs_acum_python
[19.3,19.6)              19.436              1            0.04                    1
[19.6,19.8)              19.688              7            0.28                    8
[19.8,20.1)              19.940              4            0.16                   12
[20.1,20.3)              20.192              8            0.32                   20
[20.3,20.6]              20.444              5            0.20                   25
          frec_rel_acum_python
[19.3,19.6)                 0.04
[19.6,19.8)                 0.32
[19.8,20.1)                 0.48
[20.1,20.3)                 0.80
[20.3,20.6]                 1.00
>
```

b) y c) Para calcula la media, mediana y la desviación típica de la variable agrupada, volvemos a calcular las variables agrupadas repitiendo las marcas de clase tantas veces como indican las correspondientes frecuencias absolutas:

```
> (java_agrup2 <- rep(marcas_clase_java,frec_abs_java))
 [1] 48.87 48.87 48.87 48.87 48.87 48.87 49.61 49.61 49.61 49.61 49.61 50.35 50.35 50.35 50.35
[16] 50.35 50.35 50.35 50.35 50.35 51.09 51.09 51.09 51.09 51.83
> (perl_agrup2 <- rep(marcas_clase_perl,frec_abs_perl))
 [1] 73.483 73.483 74.009 74.009 74.535 74.535 74.535 74.535 74.535 74.535 75.061 75.061 75.061
[14] 75.061 75.061 75.061 75.061 75.587 75.587 75.587 75.587 75.587 75.587 75.587 75.587
> (python_agrup2 <- rep(marcas_clase_python,frec_abs_python))
 [1] 19.436 19.688 19.688 19.688 19.688 19.688 19.688 19.688 19.940 19.940 19.940 19.940 20.192
[14] 20.192 20.192 20.192 20.192 20.192 20.192 20.444 20.444 20.444 20.444 20.444
> |
```

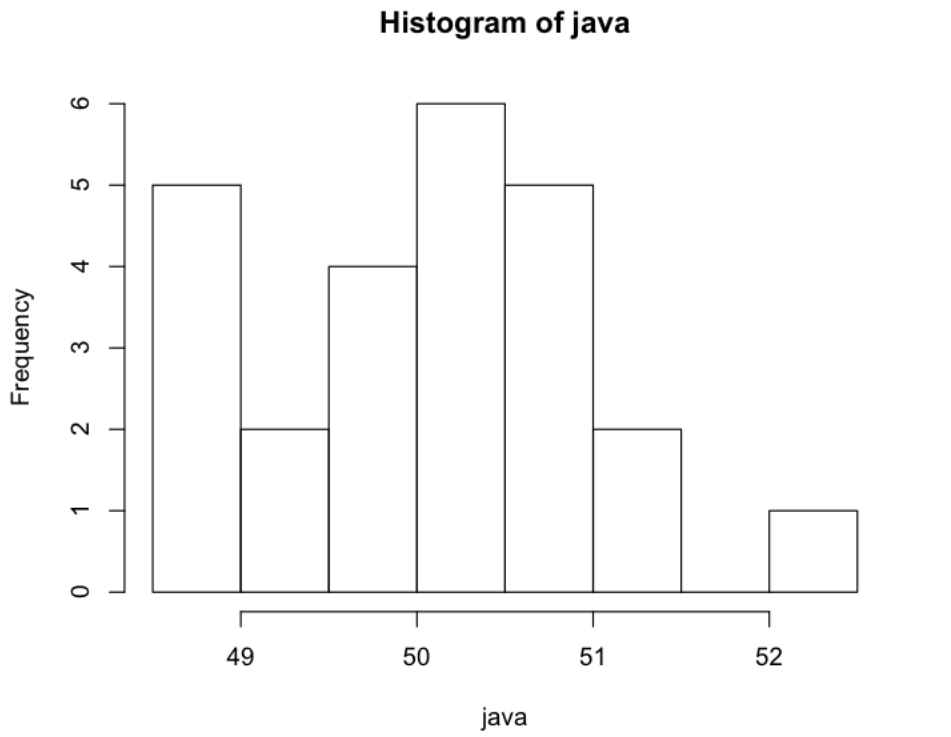
La instrucción summary nos da la media y la mediana de las variables agrupadas:

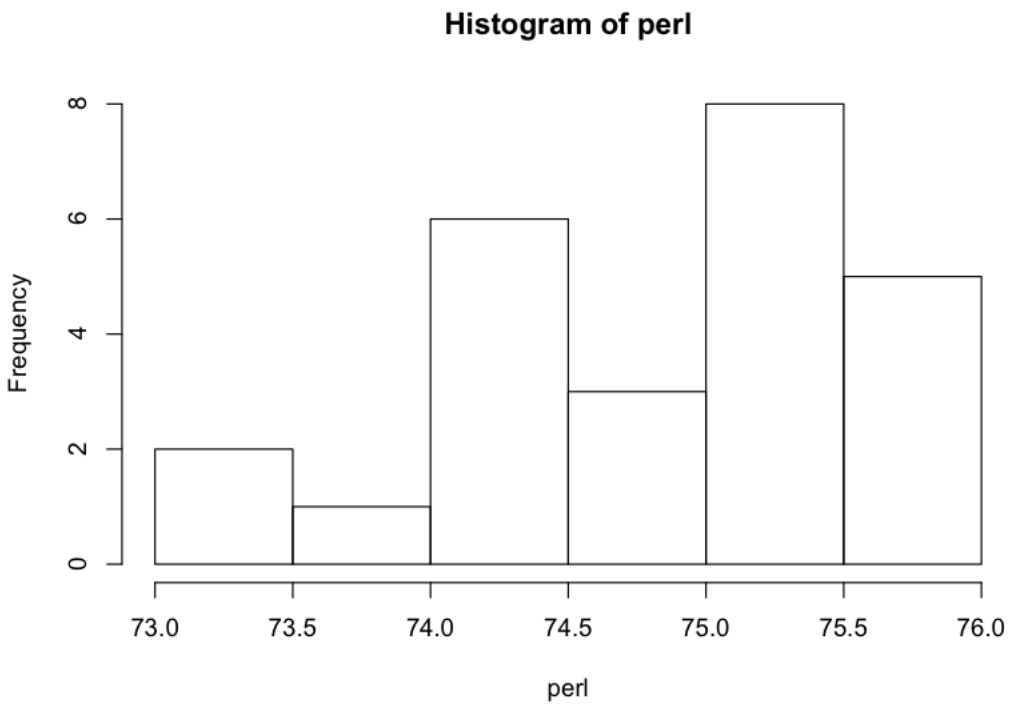
```
> summary(java_agrup2)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  48.87   49.61   50.35   50.02   50.35   51.83
> summary(perl_agrup2)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  73.48   74.54   75.06   74.89   75.59   75.59
> summary(python_agrup2)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  19.44   19.69   20.19   20.03   20.19   20.44
> |
```

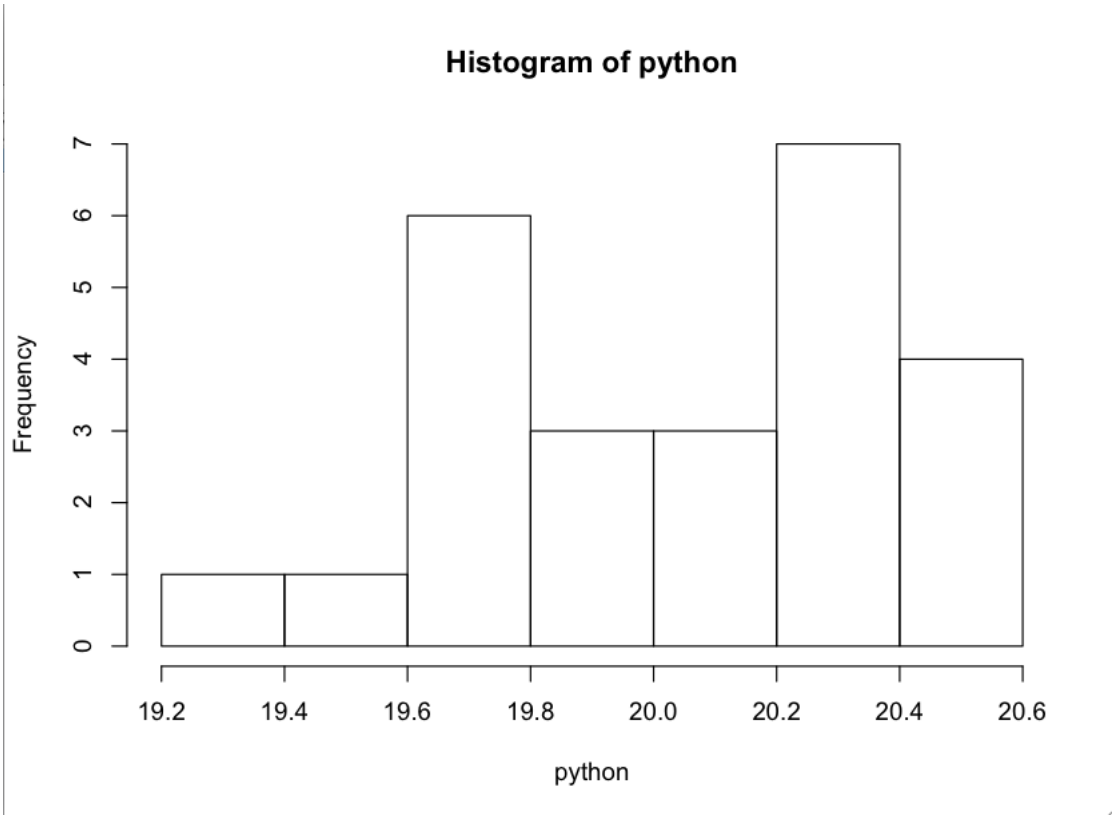
La desviación estándar se calcula con la instrucción sd:

```
> sd(java_agrup2)
[1] 0.856612
> sd(perl_agrup2)
[1] 0.6569738
> sd(python_agrup2)
[1] 0.3076052
> |
```

d) Los histogramas se muestran usando la instrucción hist:







Actividad 7: Número de cortes en la red de una empresa de servicios de Internet.

Datos estadísticos discretos. Tabla de frecuencias. Diagrama de puntos. Medidas de tendencia central. Medidas de dispersión.

Una pequeña empresa que se dedica a dar servicio de Internet tiene durante 50 días el número de cortes siguientes en la red: 2, 1, 0, 0, 1, 1, 1, 2, 0, 1, 0, 0, 0, 0, 0, 1, 2, 0, 1, 2, 0, 0, 0, 2, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 2, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0.
Se pide:
a) Calculad una tabla de frecuencias del número de cortes en la red por día donde se indique las frecuencias absolutas, relativas, absolutas acumuladas y relativas acumuladas.
b) Haced un diagrama de puntos de la variable anterior.
c) En base a los apartados anteriores comentad cómo es la variable que estamos estudiando (forma, distribución...)
d) Calcular la media, moda y la mediana del número de cortes diarios de Internet.
e) Calcular también la desviación típica.
f) Comentad el estudio realizado.

Solución

a) En primer lugar, introducimos la variable CORTES que nos da el número de cortes durante los 50 días:

```
CORTES=c(2,1,0,0,1,1,1,2,0,1,0,0,0,0,0,1,2,0,1,2,0,0,0,2,0,1,0,1,0,1,0,0,0,0,0,2,0,0,2,0
,0,0,0,0,0,0,0,0,1,0)
```

Para hallar las tablas de frecuencias, usamos el mismo procedimiento utilizado en las actividades 4 y 5:

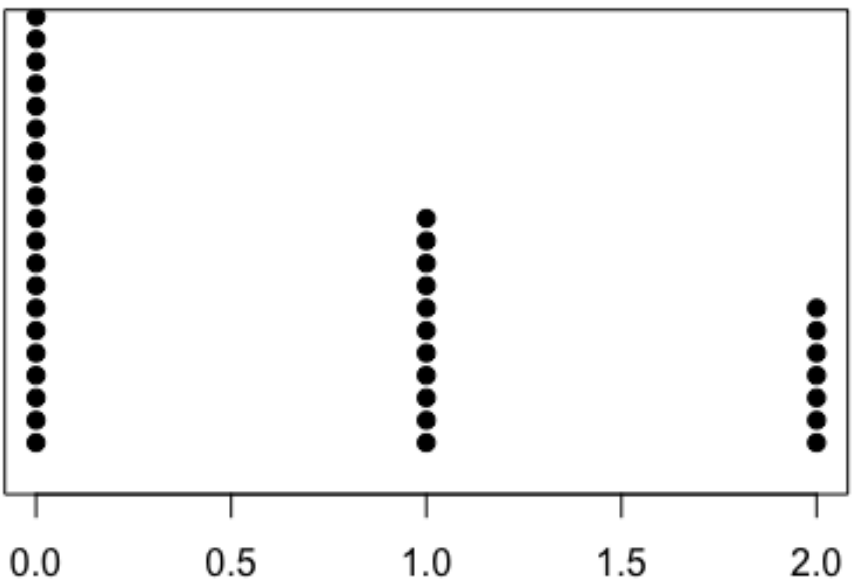
```
resultados=cbind(table(CORTES),prop.table(table(CORTES)),cumsum(table(CORTES)),cumsum(pr
op.table(table(CORTES))))
```

```
colnames(resultados) = c("Frec. abs.","Frec. rel.","Frec. abs. acum.","Frec. rel.
acum.")
resultados

  Frec. abs. Frec. rel. Frec. abs. acum. Frec. rel. acum.
0         32       0.64               32             0.64
1         11       0.22               43             0.86
2          7       0.14               50             1.00
```

Vemos en la tabla anterior que en 32 días no ha habido cortes, en 11 días ha habido un corte y en 7 días ha habido 2 cortes.

b) Para poder realizar un diagrama de puntos, podemos usar la función stripchart de R:

```
stripchart(CORTES, method = "stack", offset = .5, at = .15, pch = 19)
```



c) La variable anterior tiene una forma asimétrica con una asimetría con cola hacia la derecha.

d) y e) La media, la mediana y la desviación típica de la variable CORTES valen:

```
mean(CORTES)

[1] 0.5

median(CORTES)

[1] 0
```

```
sd(CORTES)

[1] 0.7354022
```

Vemos que la media vale 0,5, la mediana 0 y la desviación típica vale 0,735. La moda vale claramente 0 como puede observarse en el diagrama de puntos.

f) Concluimos que la mayoría de los días no hay cortes en Internet, lo que se manifiesta en los valores de la mediana y la moda aunque hay que comentar que la distribución del número de cortes de Internet tiene una desviación típica bastante elevada.

Actividad 8: Tiempo de infección de un ordenador por parte de un virus.

Agrupamiento de datos estadísticos. Medidas de tendencia central. Medidas de dispersión. Histogramas.

El tiempo de infección de una muestra de 25 ordenadores con el sistema operativo VENT por parte del virus MALASOMBRA es el siguiente (en segundos): 33.19, 31.59, 30.48, 30.35, 29.44, 29.73, 28.94, 28.57, 32.90, 30.64, 29.32, 30.43, 28.66, 29.02, 29.56, 27.70, 30.93, 30.26, 32.03, 29.28, 31.36, 29.58, 29.74, 28.92, 28.97. Se pide:

a) Agrupad la variable T: “tiempo de infección del ordenador” en 5 intervalos de igual amplitud. Calculad una tabla de frecuencias donde se indique frecuencias absolutas, relativas, absolutas acumuladas y relativas acumuladas.
b) Calculad la media y la mediana de la variable agrupada.
c) Calculad la desviación típica también de las variable agrupada.
d) Haced un histograma de la variable.
e) Comentad los resultados obtenidos.

Solución

a) En primer lugar, metemos los datos en una variable a la que llamamos tiempo:

```
> tiempo <- c(33.19, 31.59, 30.48, 30.35, 29.44, 29.73, 28.94, 28.57, 32.90, 30.64, 29.32,
30.43, 28.66, 29.02, 29.56, 27.70, 30.93, 30.26, 32.03, 29.28, 31.36, 29.58, 29.74, 28.92,
28.97)
```

A continuación, vamos a agrupar la variable anterior en 5 intervalos. Primeramente hallamos los intervalos a agrupar:

```
>
>
> intervalos <- seq(from=min(tiempo),to=max(tiempo),by=(max(tiempo)-min(tiempo))/5)
>
>
```

Los intervalos serán:

```
> intervalos
[1] 27.700 28.798 29.896 30.994 32.092 33.190
>
```

O sea [27.7,28.798), [28.798,29.896), [29.896,30.994), [30.994,32.092) y [32.092,33.190).

Para agrupar la variable, usamos la instrucción cut:

```
> (tiempo_agrup <- cut(tiempo,breaks <- intervalos,right=F,include.lowest=T))
 [1] [32.1,33.2] [31,32.1)   [29.9,31)   [29.9,31)   [28.8,29.9) [28.8,29.9) [28.8,29.9)
 [8] [27.7,28.8) [32.1,33.2] [29.9,31)   [28.8,29.9) [29.9,31)   [27.7,28.8) [28.8,29.9)
[15] [28.8,29.9) [27.7,28.8) [29.9,31)   [29.9,31)   [31,32.1)   [28.8,29.9) [31,32.1)
[22] [28.8,29.9) [28.8,29.9) [28.8,29.9) [28.8,29.9)
Levels: [27.7,28.8) [28.8,29.9) [29.9,31) [31,32.1) [32.1,33.2]
>
```

Las frecuencias absolutas se hallan usando la instrucción table:

```
>
> freq_abs <- table(tiempo_agrup)
> freq_abs
tiempo_agrup
[27.7,28.8) [28.8,29.9)   [29.9,31)   [31,32.1) [32.1,33.2]
          3          11           6           3           2
> |
```

Aquí aparecen las frecuencias relativas:

```
> freq_rel <- table(tiempo_agrup)/sum(table(tiempo_agrup))
> freq_rel
tiempo_agrup
[27.7,28.8) [28.8,29.9)   [29.9,31)   [31,32.1) [32.1,33.2]
       0.12        0.44        0.24        0.12        0.08
```

A continuación aparecen las frecuencias absolutas acumuladas:

```
> freq_abs_acum <- cumsum(table(tiempo_agrup))
> freq_abs_acum
[27.7,28.8) [28.8,29.9)   [29.9,31)   [31,32.1) [32.1,33.2]
          3          14          20          23          25
```

Y por último, las frecuencias relativas acumuladas:

```
> freq_rel_acum <- cumsum(table(tiempo_agrup))/sum(table(tiempo_agrup))
> freq_rel_acum
[27.7,28.8) [28.8,29.9)   [29.9,31)   [31,32.1) [32.1,33.2]
       0.12        0.56        0.80        0.92        1.00
```

A continuación, se enseña la tabla de frecuencias:

```
>
> cbind(freq_abs,freq_rel,freq_abs_acum,freq_rel_acum)
           freq_abs freq_rel freq_abs_acum freq_rel_acum
[27.7,28.8)        3     0.12            3          0.12
[28.8,29.9)       11     0.44           14          0.56
[29.9,31)          6     0.24           20          0.80
[31,32.1)          3     0.12           23          0.92
[32.1,33.2]        2     0.08           25          1.00
> |
```

b) Para hallar los estadísticos pedidos de la variable agrupada, necesitamos hallar las marcas de clase de los intervalos:

```
> marcas_clase <- c()
> for (i in 1:length(intervalos)-1){marcas_clase <- c(marcas_clase,(intervalos[i
+1]+intervalos[i])/2)}
> marcas_clase
[1] 28.249 29.347 30.445 31.543 32.641
```

A continuación, creamos la variable tiempo_agrupado repitiendo cada marca de clase según su frecuencia absoluta:

```
> tiempo_agrupado <- rep(marcas_clase,freq_abs)
> tiempo_agrupado
 [1] 28.249 28.249 28.249 29.347 29.347 29.347 29.347 29.347 29.347 29.347 29.347 29.347
[13] 29.347 29.347 30.445 30.445 30.445 30.445 30.445 30.445 31.543 31.543 31.543 32.641
[25] 32.641
>
```

La media de la variable agrupada valdrá:

```
> mean(tiempo_agrupado)
[1] 30.0058
> |
```

La mediana valdrá:

```
> median(tiempo_agrupado)
[1] 29.347
> |
```

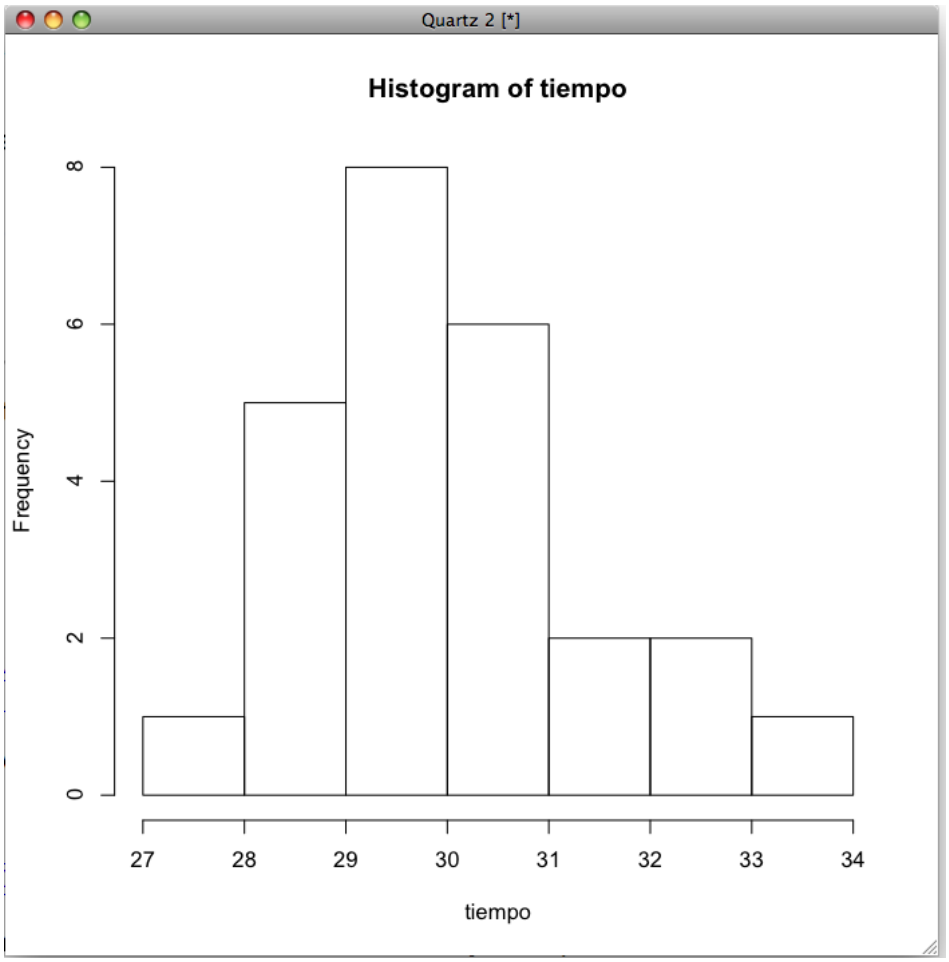
c) La varianza valdrá:

```
> varianza <- var(tiempo_agrupado)*(length(tiempo_agrupado)-1)/length(tiempo_agrupado)
> varianza
[1] 1.446725
> 
```

Hemos multiplicado por (n-1)/n, donde n es el número de datos porque R calcula la casi-varianza en lugar de la varianza donde la casi-varianza se define como: $\tilde{s}^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1}$, donde n es el número de datos.
La desviación típica será la raíz cuadrada de la varianza:

```
>
> desv_tipica <- sqrt(varianza)
> desv_tipica
[1] 1.202799
> 
```

d) El histograma se realiza con la instrucción hist(tiempo):



e) Concluimos que la mayoría de las veces, el tiempo de infección del ordenador está sobre 29-30 segundos, la distribución del tiempo de infección tiene un distribución un poco asimétrica a la derecha y no hay ningún dato atípico.

Direcciones de interés

http://wainu.ii.uned.es:8081/WAINU/ingenierias-tecnicas/segundo/estadistica-i/apuntes/desc_spanish.pdf/view

Apuntes en pdf de Estadística Descriptiva de los alumnos de la Facultad de Informática de la UNED.

http://home.ubalt.edu/ntsbarsh/Business-stat/otherapplets/Descriptive.htm

Web donde hay un applet de java que calcula las medidas de tendencia central y de dispersión.

http://en.wikipedia.org/wiki/Descriptive_statistics

Web donde se introduce la definición y los conceptos más importantes relacionados con la estadística descriptiva.

http://en.wikipedia.org/wiki/Portal:Statistics

Portal de estadística de la Wikipedia.

http://www.pitt.edu/~super1/lecture/lec0421/index.htm

Web que ofrece un pequeño curso de estadística descriptiva aplicada a la epidemiología.