

Exploratory Data Analysis in R

joseangeldiazg

November 25 2017

Ejemplo 1, hip dataset

Descargate el dataset hip con el siguiente comando

```
hip <-read.table("http://astrostatistics.psu.edu/datasets/HIP_star.dat", header=T, fill=T)
```

- Una vez descargado comprueba la dimensión y los nombres de las columnas del dataset. ¿Qué dimensión tiene? ¿qué datos alberga?

```
str(hip)
```

```
## 'data.frame': 2719 obs. of 9 variables:
## $ HIP : int 2 38 47 54 74 81 110 135 143 149 ...
## $ Vmag : num 9.27 8.65 10.78 10.57 9.93 ...
## $ RA : num 0.0038 0.111 0.1352 0.1517 0.2219 ...
## $ DE : num -19.5 -79.1 -56.8 18 35.8 ...
## $ Plx : num 21.9 23.8 24.4 21 24.2 ...
## $ pmRA : num 181.2 162.3 -44.2 367.1 157.7 ...
## $ pmDE : num -0.93 -62.4 -145.9 -19.49 -40.31 ...
## $ e_Plx: num 3.1 0.78 1.97 1.71 1.36 1.28 1.91 1.22 1.64 2.17 ...
## $ B.V : num 0.999 0.778 1.15 1.03 1.068 ...
```

Con el comando **str** vemos la información solicitada concretamente 2719 observaciones con 9 características, además vemos el tipo de dato de cada característica.

- Muestra por pantalla la columna de la variable RA

```
hip[,3]
```

```
## [1] 0.003797 0.111047 0.135192 0.151656 0.221873 0.243864
## [7] 0.348708 0.426746 0.455182 0.478685 0.612287 0.696411
## [13] 0.972063 1.099309 1.102623 1.244275 1.281668 1.369764
## [19] 1.423333 1.468617 1.843365 1.966150 2.261459 2.315143
## [25] 2.352249 2.431558 2.768701 2.878592 2.898287 2.906145
## [31] 3.125492 3.136756 3.287636 3.470117 3.499989 3.517613
## [37] 3.542473 3.561294 3.584257 3.588865 3.768990 4.047722
## [43] 4.101227 4.179702 4.342874 4.375348 4.376308 4.386210
## [49] 4.387165 4.401851 4.457699 4.582106 4.608357 4.901928
## [55] 5.280108 5.398303 5.604551 5.644953 5.735995 5.750338
## [61] 5.770342 5.907064 5.913140 5.947756 6.036948 6.091006
## [67] 6.256391 6.266819 6.285408 6.667964 6.930394 6.957396
## [73] 7.292119 7.368555 7.427643 7.465869 7.575263 7.608228
## [79] 7.689831 7.801700 7.885670 8.033768 8.182476 8.613389
## [85] 8.622360 8.712882 8.739422 8.807178 8.908298 8.967503
## [91] 8.970857 9.007174 9.016931 9.028761 9.156804 9.342176
## [97] 9.487066 9.561303 9.638220 9.803365 9.919210 9.934979
## [103] 9.989965 10.182398 10.214448 10.281968 10.313306 10.393825
## [109] 10.505973 10.617905 10.638702 10.711497 10.802331 10.922355
## [115] 11.199058 11.248836 11.295022 11.340305 11.448401 11.461630
```

##	[121]	11.641352	11.659293	11.951857	12.613796	12.861334	12.956579
##	[127]	13.195597	13.215928	13.395191	13.418159	13.747963	13.799435
##	[133]	13.800465	14.071556	14.080133	14.086786	14.187907	14.230759
##	[139]	14.334854	14.359748	14.810534	14.902862	15.094161	15.172868
##	[145]	15.257317	15.328935	15.648212	15.665018	15.683459	15.718225
##	[151]	15.914661	16.114874	16.167828	16.384181	16.455093	16.464240
##	[157]	16.508003	16.552217	16.608722	16.795620	16.817872	16.851366
##	[163]	16.942168	16.987958	16.998252	17.005018	17.240862	17.485876
##	[169]	17.685560	17.774711	17.883744	17.914945	18.124238	18.274262
##	[175]	18.290284	18.328914	18.432510	18.452612	18.543673	18.704919
##	[181]	18.937290	19.177982	19.185476	19.344598	19.496615	20.012387
##	[187]	20.115638	20.137260	20.265717	20.651504	20.736256	20.827316
##	[193]	20.838142	20.856737	20.961761	21.050790	21.085354	21.204147
##	[199]	21.301454	21.368489	21.557218	21.577496	21.705117	21.775895
##	[205]	21.863460	22.079648	22.184722	22.255183	22.359756	22.452853
##	[211]	22.463379	22.685087	22.795906	22.812423	22.830857	23.067525
##	[217]	23.136649	23.472117	23.480113	23.495490	23.603017	23.742165
##	[223]	23.811290	23.906500	23.917583	23.919763	23.924260	23.970427
##	[229]	24.042201	24.355127	24.428132	24.558431	24.694243	24.781368
##	[235]	24.923974	24.926077	24.976913	25.093014	25.142788	25.183597
##	[241]	25.265384	25.278081	25.363685	25.487784	25.525434	25.560728
##	[247]	25.654124	25.796793	25.808926	25.912488	25.937314	26.058869
##	[253]	26.120282	26.460976	26.504143	27.084421	27.361362	27.398006
##	[259]	27.563815	27.688127	27.716725	27.737869	27.833990	27.879742
##	[265]	27.941418	28.150791	28.213507	28.365926	28.380999	28.490802
##	[271]	28.807768	28.841750	28.943608	28.959378	29.160189	29.182608
##	[277]	29.416312	29.453802	29.480758	29.482398	30.022118	30.197706
##	[283]	30.202588	30.236395	30.449293	30.468324	30.511669	30.611289
##	[289]	30.756650	30.789233	30.859126	30.867524	30.886683	30.940466
##	[295]	30.979076	31.069481	31.146008	31.222648	31.329711	31.381960
##	[301]	31.405060	31.421853	31.858408	31.925780	32.165945	32.532800
##	[307]	32.716716	32.838177	32.842678	32.991939	33.086270	33.197180
##	[313]	33.232392	33.281209	33.439303	33.510536	33.817351	33.923787
##	[319]	33.928477	33.941399	33.980981	34.006963	34.065453	34.127085
##	[325]	34.197081	34.357188	34.558477	34.678044	34.727048	35.407083
##	[331]	35.437596	35.496517	35.629458	35.709842	35.810793	36.224977
##	[337]	36.300264	36.539614	36.561617	36.564190	36.567476	36.868438
##	[343]	36.871668	37.007042	37.007352	37.041626	37.095529	37.266617
##	[349]	37.850319	38.096250	38.272437	38.472245	38.514713	38.548152
##	[355]	38.644692	38.653526	38.683132	38.897663	38.948203	39.128737
##	[361]	39.160204	39.251816	39.351104	39.896765	39.948195	40.013098
##	[367]	40.164916	40.166382	40.244132	40.260256	40.276125	40.357674
##	[373]	40.392219	40.535147	40.587033	40.744524	40.856286	40.897930
##	[379]	41.003599	41.017642	41.060453	41.120743	41.687736	41.765011
##	[385]	42.181572	42.356519	42.495789	42.767495	42.965454	42.969977
##	[391]	43.166246	43.319831	43.497058	43.763774	44.050151	44.057202
##	[397]	44.058148	44.106682	44.157446	44.312089	44.348755	44.367928
##	[403]	44.565482	44.884889	45.070690	45.082105	45.635924	45.717560
##	[409]	45.719254	45.807011	45.875471	45.883382	45.898022	45.903766
##	[415]	46.068655	46.907415	46.913451	47.184434	47.425976	47.498874
##	[421]	47.517497	47.580105	47.807529	47.968815	47.984327	48.039766
##	[427]	48.106996	48.126158	48.162929	48.261105	48.402455	48.519994
##	[433]	48.556492	48.838748	48.972341	49.023753	49.169269	49.234452
##	[439]	49.359535	49.386169	49.514128	49.562612	49.574012	49.646313

##	[445]	49.680357	49.914871	50.177034	50.188236	50.232840	50.348931
##	[451]	50.360863	50.405685	50.468216	50.475217	50.635627	50.902536
##	[457]	51.119287	51.379709	51.640888	52.008780	52.086975	52.150018
##	[463]	52.266093	52.607873	52.626386	52.658680	52.666896	52.718640
##	[469]	52.814556	53.047775	53.165926	53.208232	53.248789	53.438571
##	[475]	53.486722	53.534668	53.551447	53.688409	53.747931	53.997149
##	[481]	54.001367	54.013961	54.222178	54.235377	54.242298	54.476360
##	[487]	54.726279	54.738916	54.744758	54.954120	54.994352	55.021698
##	[493]	55.030169	55.196725	55.201126	55.306241	55.316535	55.496224
##	[499]	55.501753	55.890994	55.913099	55.947529	56.164410	56.189388
##	[505]	56.316288	56.625853	56.976306	57.002769	57.049001	57.124673
##	[511]	57.149262	57.162728	57.432673	57.567101	57.595410	57.603705
##	[517]	57.762573	57.934751	57.988076	58.291494	58.363311	58.566090
##	[523]	58.594254	58.776715	58.817391	58.955626	59.119238	59.151816
##	[529]	59.216803	59.718422	60.112986	60.172306	60.202858	60.525629
##	[535]	60.747723	60.825362	60.912353	61.084441	61.128308	61.405433
##	[541]	61.566899	61.752643	61.754794	61.836520	61.924609	62.064424
##	[547]	62.110817	62.342339	62.360045	62.500933	62.530789	62.590083
##	[553]	62.676201	63.539787	63.606610	63.613275	63.634326	63.642792
##	[559]	63.716101	63.740554	63.937619	63.942560	64.126154	64.333916
##	[565]	64.411969	64.507366	64.579998	64.623547	64.741269	64.783078
##	[571]	64.903221	64.939589	64.948058	64.978269	64.990145	65.053732
##	[577]	65.104316	65.151020	65.195160	65.219406	65.514374	65.683765
##	[583]	65.694862	65.733447	65.833914	65.842782	65.854160	65.855020
##	[589]	65.876420	65.884447	65.972485	65.976429	66.023724	66.051658
##	[595]	66.060417	66.117739	66.136625	66.155934	66.199969	66.237749
##	[601]	66.251245	66.288472	66.342078	66.348869	66.353936	66.372156
##	[607]	66.378603	66.405223	66.424206	66.440008	66.447877	66.465247
##	[613]	66.478943	66.488676	66.508653	66.576645	66.576805	66.586132
##	[619]	66.602270	66.608825	66.666891	66.700800	66.726165	66.882743
##	[625]	66.899281	66.941704	66.969553	67.003007	67.018217	67.143468
##	[631]	67.153889	67.154802	67.165312	67.165753	67.200966	67.208754
##	[637]	67.219338	67.248822	67.335371	67.376191	67.381418	67.490262
##	[643]	67.535554	67.639871	67.655411	67.661778	67.694736	67.737960
##	[649]	67.815117	67.904325	67.965388	68.019759	68.064228	68.071794
##	[655]	68.126726	68.392082	68.404636	68.424440	68.444085	68.461897
##	[661]	68.493652	68.625531	68.646870	68.821827	68.913473	69.107364
##	[667]	69.121115	69.169437	69.383039	69.431418	69.539068	69.539174
##	[673]	69.713438	69.738551	69.788322	69.818549	69.881624	69.962130
##	[679]	70.050642	70.105891	70.176845	70.715178	70.755700	70.807281
##	[685]	70.882015	70.901358	70.956585	71.046134	71.107373	71.140897
##	[691]	71.200997	71.452649	71.474723	71.578080	71.705813	71.825743
##	[697]	71.923861	72.000903	72.115216	72.135536	72.209430	72.241699
##	[703]	72.264460	72.305278	72.599450	72.637067	72.640679	72.802006
##	[709]	72.819906	72.958017	72.987353	73.232419	73.481566	73.986055
##	[715]	74.064966	74.276840	74.553955	74.608976	74.649107	74.755991
##	[721]	74.814064	74.902875	74.934460	75.141364	75.165899	75.175267
##	[727]	75.216582	75.225224	75.315279	75.434431	75.773767	76.025879
##	[733]	76.198001	76.222877	76.366019	76.378138	76.479926	76.826306
##	[739]	76.947985	77.441187	77.508801	77.515585	77.678629	77.741562
##	[745]	77.746629	77.818783	77.903417	78.199742	78.354422	78.357215
##	[751]	78.743675	78.827178	79.037753	79.146074	79.250993	79.356732
##	[757]	79.364876	79.417693	79.420753	79.525740	80.158411	80.212086
##	[763]	80.677967	80.762950	80.915213	81.185074	81.478781	81.499263

```

## [769] 81.530864 81.572908 81.619289 81.874966 81.941051 82.061360
## [775] 82.246582 82.263636 82.539424 82.626258 82.749355 82.754931
## [781] 82.871202 83.140333 83.279070 83.400674 83.861587 83.914743
## [787] 84.279966 84.318722 84.404291 84.436013 84.721216 84.859388
## [793] 84.879805 84.934280 85.007073 85.072965 85.631773 85.744584
## [799] 85.885949 85.969469 86.193413 86.251512 86.305940 86.374124
## [805] 86.645661 86.956396 87.083659 87.153295 87.442438 87.458438
## [811] 88.248618 88.313517 88.758413 89.154664 89.369979 89.881460
## [817] 89.957026 90.089080 90.113105 90.255586 90.595793 90.729655
## [823] 91.030069 91.242432 91.425335 91.480137 91.624376 91.749374
## [829] 91.996030 92.026830 92.485177 92.503926 92.709214 92.763089
## [835] 93.058664 93.502694 93.701386 93.912731 94.178786 94.210573
## [841] 94.390382 94.656328 94.783547 94.803741 94.905795 95.365396
## [847] 95.405740 95.431407 95.483128 96.128320 96.209174 96.318090
## [853] 96.781608 96.836142 96.850440 97.190596 97.234706 97.265719
## [859] 97.416519 97.790108 98.096374 98.147895 98.154700 98.549053
## [865] 98.839081 98.890227 98.975302 99.127635 99.139219 99.584875
## [871] 99.798563 99.832964 100.095186 100.204336 100.293110 100.601344
## [877] 100.691993 101.005987 101.058605 101.228600 101.370258 101.503433
## [883] 102.050968 102.079533 102.149098 102.224706 102.339020 102.408803
## [889] 102.877039 102.918208 102.940005 103.008637 103.737618 104.143434
## [895] 104.155936 104.323388 104.443450 104.697982 104.872581 104.930818
## [901] 105.268821 105.619024 105.620362 105.891863 106.043628 106.557741
## [907] 106.570251 107.001137 107.050065 107.176423 107.186769 107.271051
## [913] 107.342304 107.528284 107.538987 107.806306 107.811656 107.831504
## [919] 108.017163 108.072192 108.204492 108.786094 108.976772 109.151639
## [925] 109.410489 109.551236 109.735395 109.867006 110.508258 110.541299
## [931] 110.610856 110.769287 110.868976 110.948562 111.010713 111.039515
## [937] 111.238172 111.248005 111.249978 111.325386 111.515211 111.630569
## [943] 112.018199 112.025655 112.050096 112.066974 112.107630 112.139573
## [949] 112.335264 112.597320 112.609382 112.620829 112.633443 112.665028
## [955] 112.966807 113.298647 113.318200 113.620490 113.685181 113.780877
## [961] 114.144875 114.553595 114.559845 114.746220 114.791485 114.793545
## [967] 114.840337 115.213835 115.311986 115.454983 115.537950 115.602343
## [973] 116.040597 116.082348 116.086031 116.111952 116.196451 116.885726
## [979] 117.194844 117.205802 117.222763 117.328882 117.475280 117.498819
## [985] 117.534148 117.925179 118.199463 118.700299 118.992561 119.303134
## [991] 120.336448 120.522664 120.608474 120.795588 120.824626 120.870037
## [997] 120.920310 121.007818 121.178227 121.786595
## [ reached getOption("max.print") -- omitted 1719 entries ]

```

- Calcula las tendencias centrales de todos los datos del dataset (mean, media) utilizando la function apply

```
apply(hip, 2, mean, na.rm=T)
```

```

##          HIP      Vmag        RA        DE       Plx
## 56549.4828981    8.2593858  173.4529975 -0.1397663 22.1980213
##          pmRA      pmDE      e_Plx        B.V
## 5.3761346   -63.9419934   1.6267929   0.7615299

```

```
apply(hip, 2, median, na.rm=T)
```

```

##          HIP      Vmag        RA        DE       Plx
## 56413.000000    8.280000  173.369788  3.254234 22.100000
##          pmRA      pmDE      e_Plx        B.V

```

```
##      10.550000 -49.480000    1.140000    0.710500
```

- Haz lo mismo para las medidas de dispersión mínimo y máximo.

```
apply(hip,2,min, na.rm=T)
```

```
##          HIP      Vmag        RA        DE        Plx
##      2.000000  0.450000  0.003797 -87.202730 20.000000
##      pmRA      pmDE      e_Plx      B.V
## -868.010000 -1392.300000  0.450000 -0.158000
```

```
apply(hip,2,max, na.rm=T)
```

```
##          HIP      Vmag        RA        DE        Plx
## 120003.00000 12.74000  359.95468 88.30268 25.000000
##      pmRA      pmDE      e_Plx      B.V
## 781.34000  481.19000  46.91000  2.80000
```

- ¿Sería posible hacerlo con un único comando? ¿Qué hace la función range()

Si que sería posible, para ello usaremos la función range que nos da el rango de valores comprendido para una determinada característica del dataset.

```
apply(hip,2,range, na.rm=T)
```

```
##      HIP  Vmag        RA        DE  Plx      pmRA      pmDE e_Plx      B.V
## [1,]    2  0.45  0.003797 -87.20273 20 -868.01 -1392.30  0.45 -0.158
## [2,] 120003 12.74 359.954685 88.30268 25  781.34  481.19 46.91  2.800
```

- Sin embargo las medidas más populares de dispersión son la varianza (var()), su desviación standard (sd()) y la desviación absoluta de la mediana o MAD. Calcula estas medidas para los valores de RA.

```
apply(hip,2,var, na.rm=T)
```

```
##          HIP      Vmag        RA        DE        Plx
## 1.266456e+09 3.552207e+00 1.156632e+04 1.515575e+03 2.008437e+00
##      pmRA      pmDE      e_Plx      B.V
## 2.591451e+04 1.985011e+04 4.896779e+00 1.012434e-01
```

```
apply(hip,2,sd, na.rm=T)
```

```
##          HIP      Vmag        RA        DE        Plx
## 3.558731e+04 1.884730e+00 1.075468e+02 3.893039e+01 1.417193e+00
##      pmRA      pmDE      e_Plx      B.V
## 1.609799e+02 1.408904e+02 2.212867e+00 3.181876e-01
```

```
apply(hip,2,mad, na.rm=T)
```

```
##          HIP      Vmag        RA        DE        Plx
## 4.909037e+04 1.882902e+00 1.469334e+02 4.398403e+01 1.764294e+00
##      pmRA      pmDE      e_Plx      B.V
## 1.416476e+02 9.949729e+01 4.892580e-01 2.809527e-01
```

- Imagina que quieres calcular dos de estos valores de una sola vez. ¿Te serviría este código?

```
f = function(x) c(median(x), mad(x))
f(hip[,1])
```

```
## [1] 56413.00 49090.37
```

Si que valdría y calcularía para la columna 1 la mediana y la desviación absoluta de la mediana.

- ¿Cuál sería el resultado de aplicar apply(hip,2,f)?

El resultado de esta función sería aplicar para cada columna la mediana y la desviación absoluta de la mediana.

```
apply(hip,2,f)
```

```
##          HIP      Vmag       RA        DE       Plx      pmRA      pmDE
## [1,] 56413.00 8.280000 173.3698 3.254234 22.100000 10.5500 -49.48000
## [2,] 49090.37 1.882902 146.9334 43.984032 1.764294 141.6476  99.49729
##          e_Plx B.V
## [1,] 1.140000 NA
## [2,] 0.489258 NA
```

Vamos a medir la dispersión de la muestra utilizando el concepto de cuartiles. El percentil 90 es aquel dato que excede en un 10% a todos los demás datos. El cuartil (quantile) es el mismo concepto, solo que habla de proporciones en vez de porcentajes. De forma que el percentil 90 es lo mismo que el cuartil 0.90. La mediana “median” de un dataset es el valor más central, en otras palabras exactamente la mitad del dataset excede la mediana.

- Calcula el cuartil .10 y .50 para la columna RA del dataset hip. Sugerencia: quantile()

```
quantile(hip$RA, probs = c(0.1, 0.5))
```

```
##      10%      50%
## 28.92324 173.36979
```

- Los cuantiles 0.25 y 0.75 se conocen como el first quartile y el third quartile, respectivamente. Calcula los cuatro cuartiles para RA con un único comando.

```
quantile(hip$RA, probs=c(0.25,0.5,0.75))
```

```
##      25%      50%      75%
## 70.14137 173.36979 266.92332
```

- Otra medida de dispersión es la diferencia entre el primer y el tercer cuartil conocida como rango intercuartil (IQR) Inter Quantile Range. ¿Obtienes ese valor con la función summary()?

El valor como tal no es ofrecido por la función summary, pero si que obtenemos los valores necesarios para calcularlo. Si queremos tenerlo directamente debemos usar la función **IQR**.

```
summary(hip$RA, na.rm=T)
```

```
##   Min. 1st Qu. Median Mean 3rd Qu. Max.
## 0.0038 70.1414 173.3698 173.4530 266.9233 359.9547
```

```
IQR(hip$RA)
```

```
## [1] 196.782
```

Hasta ahora has ignorado la presencia de valores perdidos NA. La función any() devuelve TRUE si se encuentra al menos un TRUE en el vector que damos como argumento. Su combinación con is.na es muy útil.

- ¿Qué obtienes cuando ejecutas el siguiente comando? ¿Cómo lo interpretas?

El siguiente comando, lo que hace es crear una función hasNA que nos dice si algún elemento de los datos que se le pasan como argumento es un *missing value*. Con la función apply, lo que hacemos es usarlo para cada columna del dataset hip y este nos devuelve que en la variable B.V, tenemos valores perdidos.

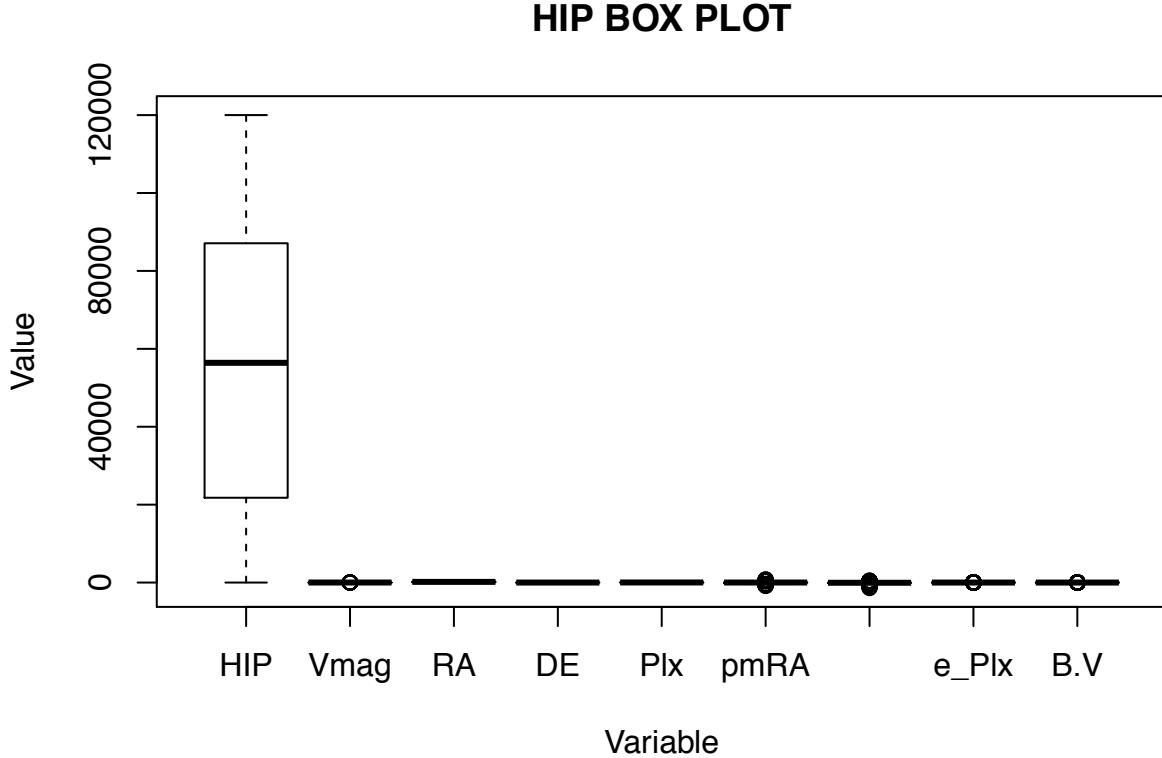
```
hasNA = function(x) any(is.na(x))
apply(hip,2,hasNA)
```

```
##    HIP Vmag     RA     DE     Plx      pmRA      pmDE e_Plx B.V
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE
```

- Normalmente querríamos poder usar las funciones sobre el resto de datos que no son NA: Para ello podemos utilizar la función na.omit. ¿Qué ocurre cuando lo hacemos?. Usando apply calcula la media para hip y hip1. Intenta calcular la media de forma que solo cambie la de B.V cuando ignores los valores NA.

- Obten una idea aproximada de tus datos mediante la creación de un boxplot del hip dataset

```
library(ggplot2)
boxplot(hip, main="HIP BOX PLOT", ylab="Value", xlab="Variable")
```



Vemos que el gráfico no es muy revelador esto es porque tenemos variables de rangos muy distintos, podemos normalizarlas para ver algo mejor la dispersión de los datos. De todas formas este gráfico ya nos aportaría información como que la variable HIP está muy por encima de los valores de las demás que se encuentran en rangos similares en función a esta y que tenemos presencia de bastantes outliers en pmRA y pmDE. Vamos a normalizar para ver si el resultado mejora:

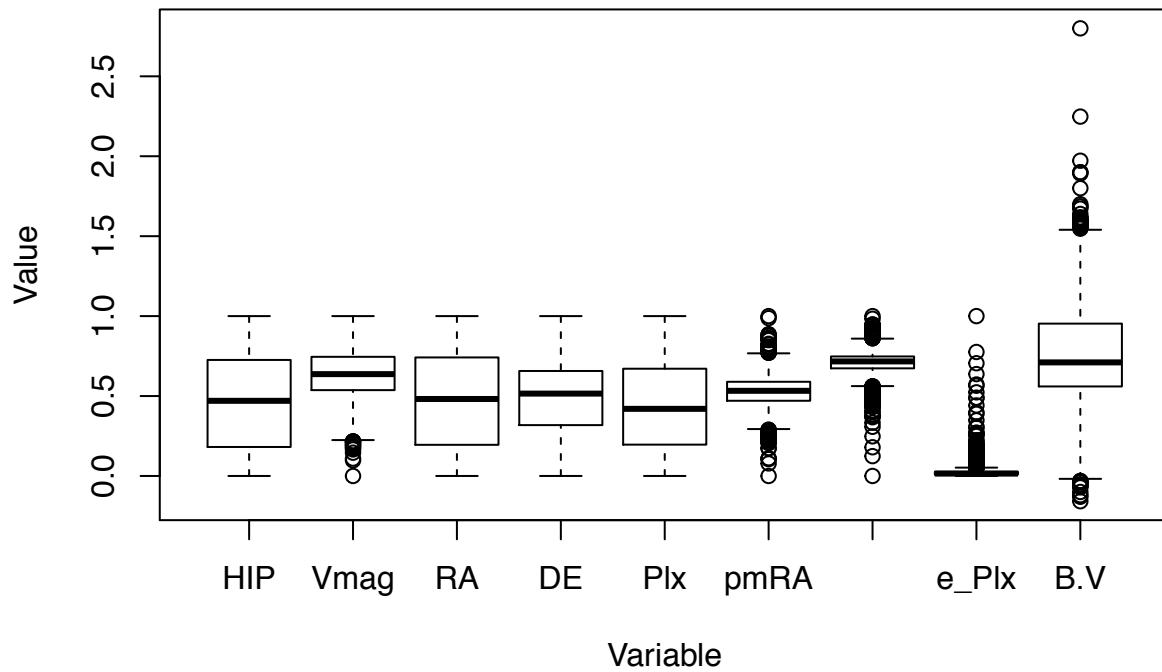
```
library(scales)

hip_norm<-hip

hip_norm$HIP <- rescale(hip$HIP)
hip_norm$Vmag <- rescale(hip$Vmag)
hip_norm$RA <- rescale(hip$RA)
hip_norm$DE <-rescale(hip$DE)
hip_norm$Plx<-rescale(hip$Plx)
hip_norm$pmRA<-rescale(hip$pmRA)
hip_norm$pmDE<-rescale(hip$pmDE)
hip_norm$e_Plx<-rescale(hip$e_Plx)

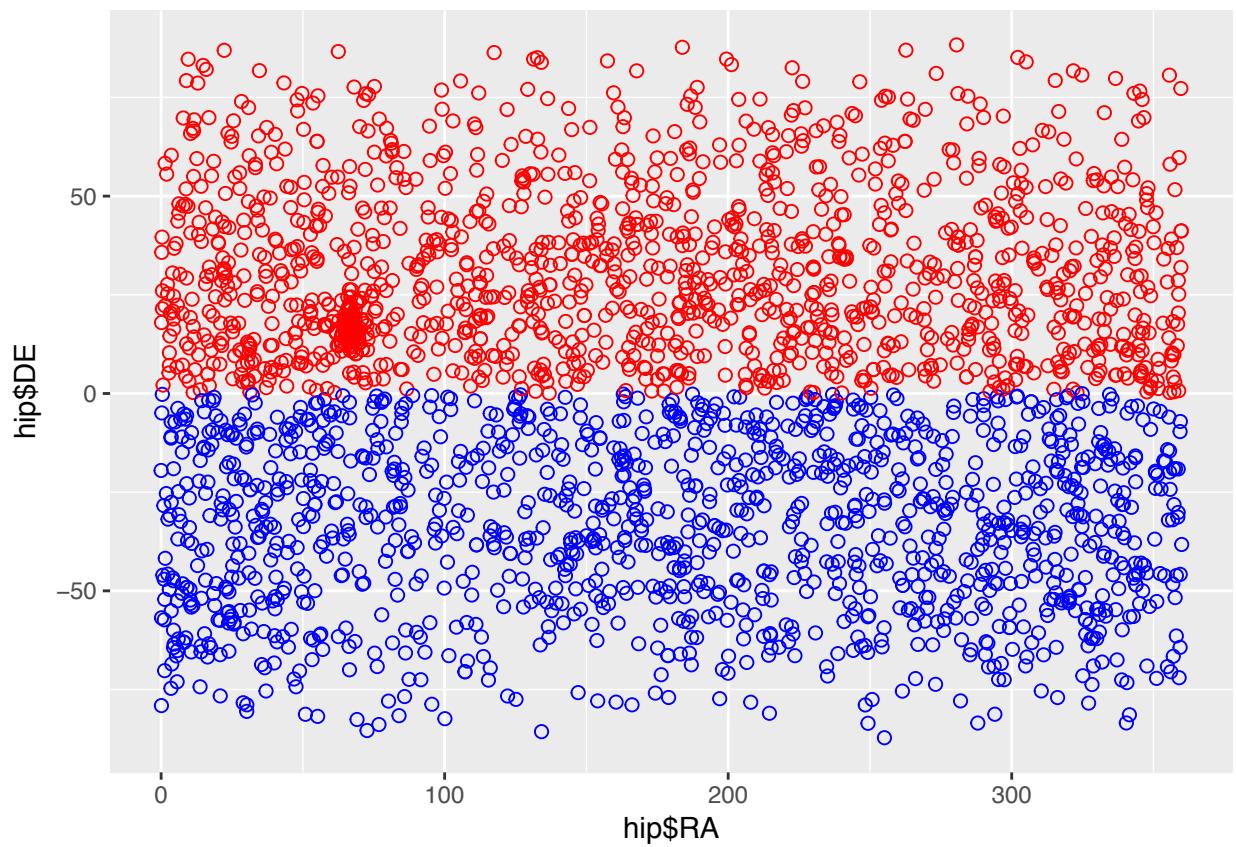
boxplot(hip_norm, main="HIP BOX PLOT NORMALIZADO", ylab="Value", xlab="Variable")
```

HIP BOX PLOT NORMALIZADO



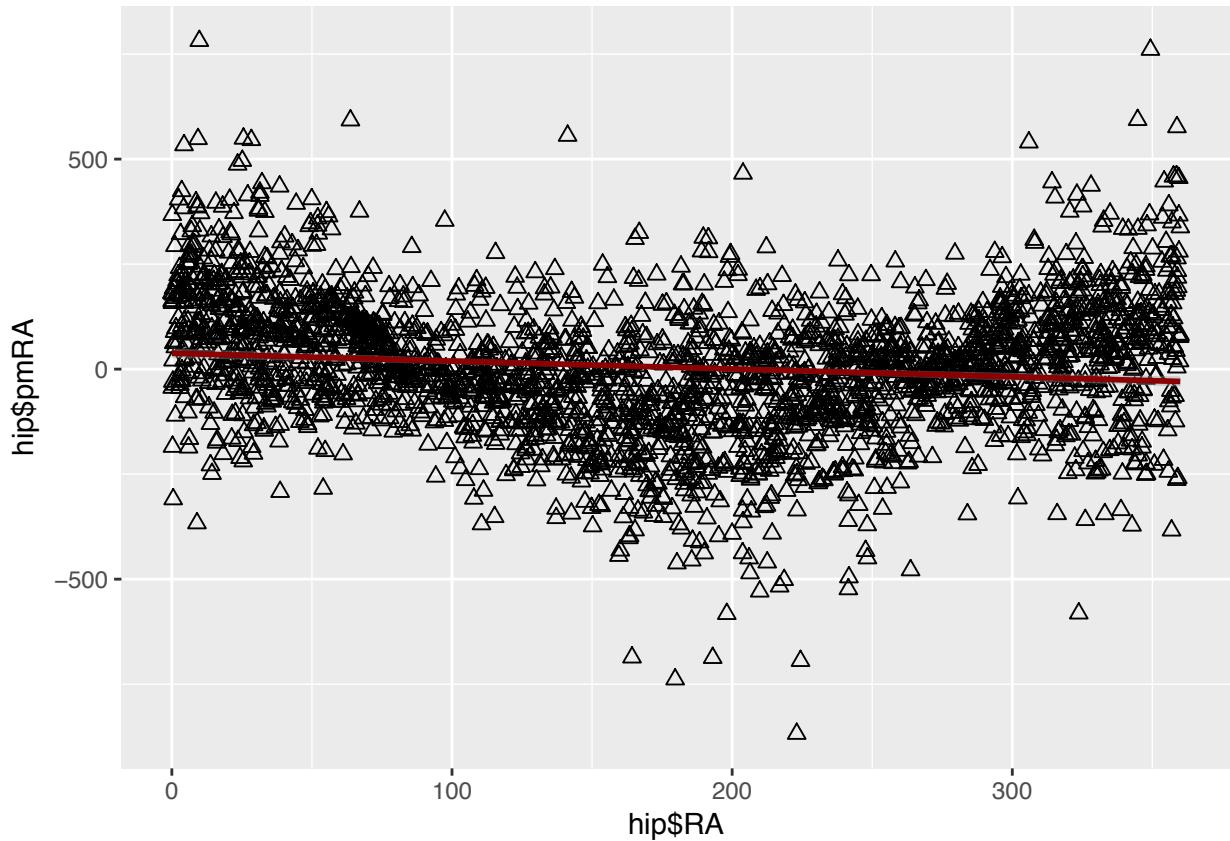
- Crea un scatterplot que te compare los valores de RA y DE. Representa los puntos con el símbolo ‘?’ Y que estos puntos sean de color rojo si DE excede de 0. Sugerencia ifelse()

```
ggplot(hip, aes(x=hip$RA, y=hip$DE)) + geom_point(size=2,color=ifelse(hip$DE>0,"red","blue"),shape=1)
```



- Haz un scatterplot de RA y pmRA. ¿Ves algún patrón?

```
ggplot(hip, aes(x=hip$RA, y=hip$pmRA)) + geom_point(size=2, shape=2) + geom_smooth(method=lm, color="darkred")
```



Aunque parece que a valores entre 100 y 150 de RA obtenemos valores más pequeños de pmRA la distribución de estos datos es bastante compleja y esto que apreciamos no es suficientemente concluyente para poder afirmar que existen relaciones entre ambas variables.

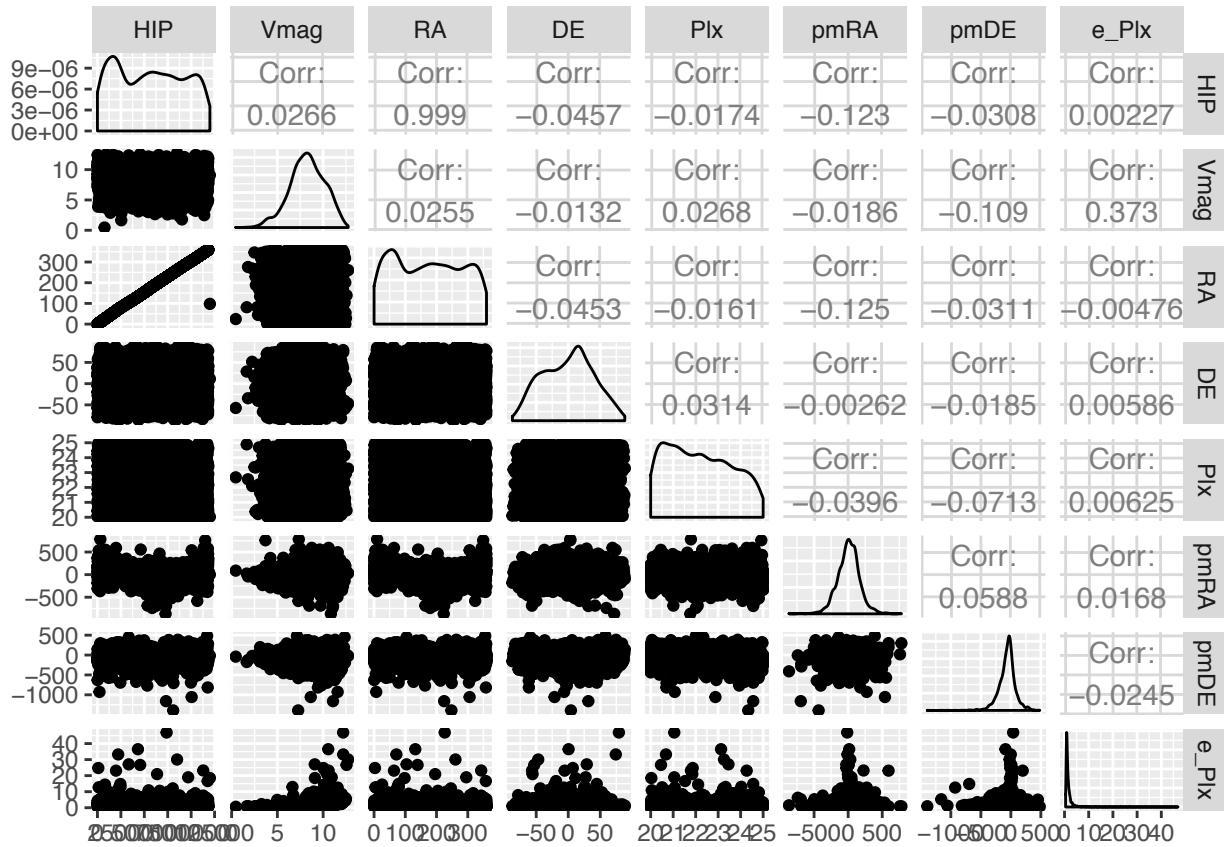
- En vez de crear los plots por separado para cada par de columnas, hazlos con un solo comando con el scatterplot matrix.

Para esto podemos usar pairs, pero si usamos el comando ggpairs nos ofrecerá mas informacion como correlaciones entre variables e incluso distribuciones de las mismas.

```
install.packages("GGally")

## Warning in install.packages :
##   package 'GGally' is not available (for R version 3.4.2)
## Warning in install.packages :
##   Perhaps you meant 'GGally' ?

library(GGally)
ggpairs(hip[1:8])
```



- Para poder acceder a las variables por su nombre usa attach(hip). Vamos a seleccionar las estrellas Hyadas del dataset aplicando los siguientes filtros:
 1. RA in the range (50,100)
 2. DE in the range (0,25)
 3. pmRA in the range (90,130)
 4. pmDE in the range (-60,-10)
 5. e_Plx <5
 6. Vmag >4 OR B.V <0.2

Crea un nuevo dataset con la aplicación de estos filtro. El Nuevo dataset se llama hyades.

```
hip <- hip[complete.cases(hip), ]
hyades<-hip[hip$RA > 50 & hip$RA < 100 &
  hip$DE>0 & hip$DE<25 &
  hip$pmRA>90 & hip$pmRA<130 &
  hip$pmDE> -60 & hip$pmDE< -10 &
  hip$e_Plx < 5 &
  (hip$Vmag>4|hip$B.V<0.2),]
```

hyades

```
##      HIP   Vmag      RA      DE     Plx    pmRA    pmDE  e_Plx    B.V
##  533 18735  5.89 60.20286 18.19407 21.99 129.49 -28.27  0.81 0.319
##  537 18946 10.12 60.91235 19.45509 23.07 119.02 -34.19  2.12 1.095
##  541 19148  7.85 61.56690 15.69817 21.41 118.53 -19.59  1.47 0.593
##  543 19207 10.49 61.75479 15.33508 23.57 122.63 -18.96  2.26 1.180
##  545 19261  6.02 61.92461 15.16284 21.27 127.06 -22.75  1.03 0.397
##  547 19316 11.28 62.11082 12.19187 24.90 115.35 -13.05  2.59 1.327
```

```

## 553 19504 6.61 62.67620 18.42333 23.22 123.92 -31.41 0.92 0.427
## 555 19781 8.45 63.60661 14.62508 21.91 105.61 -19.86 1.27 0.693
## 557 19793 8.05 63.63433 23.57506 21.69 121.09 -48.34 1.14 0.657
## 559 19808 10.69 63.71610 13.05500 22.67 114.46 -19.28 2.30 1.204
## 562 19877 6.31 63.94256 15.40075 22.51 114.38 -22.07 0.82 0.400
## 565 20019 8.32 64.41197 16.94791 21.40 113.07 -21.39 1.24 0.756
## 566 20056 7.53 64.50737 18.25688 21.84 115.65 -31.23 1.14 0.681
## 567 20082 9.57 64.58000 16.08839 20.01 121.88 -19.72 1.91 0.980
## 569 20130 8.62 64.74127 19.90679 23.53 113.95 -36.39 1.25 0.745
## 570 20146 8.47 64.78308 17.52482 21.24 112.80 -29.89 1.32 0.721
## 574 20215 6.85 64.97827 16.52268 23.27 121.27 -36.65 1.14 0.509
## 575 20219 5.58 64.99014 14.03525 22.31 115.42 -19.91 0.92 0.283
## 576 20237 7.46 65.05373 19.23356 22.27 115.67 -34.02 0.93 0.560
## 577 20255 6.11 65.10432 18.74273 21.12 119.59 -48.17 0.77 0.404
## 578 20261 5.26 65.15102 15.09550 21.20 108.79 -20.67 0.99 0.225
## 580 20284 6.15 65.21941 13.86447 21.80 105.29 -17.97 0.85 0.456
## 581 20400 5.72 65.51437 14.07725 21.87 114.04 -21.40 0.96 0.315
## 582 20440 6.97 65.68376 15.05614 21.45 111.98 -19.88 2.76 0.518
## 586 20480 8.84 65.84278 21.37919 20.63 99.77 -39.07 1.34 0.758
## 587 20484 5.64 65.85416 16.77733 21.17 105.09 -27.62 0.80 0.310
## 588 20485 10.47 65.85502 15.76319 21.08 126.22 -30.50 2.69 1.231
## 589 20491 7.18 65.87642 24.40551 20.04 93.11 -45.08 0.89 0.462
## 590 20492 9.11 65.88445 14.67052 21.23 107.57 -18.24 1.80 0.855
## 592 20527 10.89 65.97643 14.05215 22.57 115.90 -15.21 2.78 1.288
## 593 20542 4.80 66.02372 17.44421 22.36 109.99 -33.47 0.88 0.154
## 594 20553 7.58 66.05166 14.75829 22.25 97.38 -33.51 1.52 0.604
## 595 20557 7.13 66.06042 21.73636 24.47 119.10 -46.63 1.06 0.518
## 596 20577 7.79 66.11774 16.88623 20.73 110.89 -24.72 1.29 0.599
## 600 20614 5.97 66.23775 19.04209 20.40 110.73 -32.50 0.74 0.378
## 603 20635 4.21 66.34208 22.29398 21.27 105.49 -44.14 0.80 0.136
## 605 20641 5.27 66.35394 22.20011 22.65 112.45 -47.06 0.84 0.250
## 606 20648 4.30 66.37216 17.92799 22.05 108.26 -32.47 0.77 0.049
## 608 20661 6.44 66.40522 15.94108 21.47 104.62 -28.83 0.97 0.509
## 611 20679 8.99 66.44788 18.01736 20.79 112.62 -35.79 1.83 0.935
## 612 20686 8.07 66.46525 18.86415 23.08 110.87 -33.75 1.22 0.680
## 616 20711 4.28 66.57664 22.81369 21.07 108.66 -45.83 0.80 0.263
## 617 20712 7.36 66.57680 21.47052 21.54 105.82 -36.48 0.97 0.557
## 618 20713 4.48 66.58613 15.61835 20.86 114.66 -33.30 0.84 0.262
## 619 20719 8.04 66.60227 16.85337 21.76 103.64 -17.58 1.46 0.651
## 621 20741 8.10 66.66689 16.74697 21.42 110.29 -27.82 1.54 0.664
## 622 20751 9.45 66.70080 10.87111 23.03 111.07 -16.26 1.66 1.033
## 623 20762 10.48 66.72616 13.13822 21.83 104.54 -18.10 2.29 1.146
## 625 20815 7.41 66.89928 15.58925 21.83 103.54 -25.16 1.01 0.537
## 626 20826 7.49 66.94170 11.73645 21.18 110.28 -12.11 1.04 0.560
## 628 20842 5.72 67.00301 21.62001 20.85 98.82 -40.59 0.86 0.270
## 629 20850 9.02 67.01822 13.86798 21.29 106.16 -17.59 1.91 0.839
## 632 20890 8.62 67.15480 19.74078 20.09 99.83 -39.82 1.11 0.741
## 633 20894 3.40 67.16531 15.87095 21.89 108.66 -26.39 0.83 0.179
## 635 20899 7.83 67.20097 17.28554 21.09 105.58 -30.05 1.08 0.609
## 636 20901 5.02 67.20875 13.04764 20.33 105.17 -15.08 0.84 0.215
## 638 20916 6.59 67.24882 16.15915 20.58 90.28 -25.47 1.74 0.536
## 639 20935 7.02 67.33537 17.54501 23.25 104.88 -31.65 1.04 0.526
## 640 20948 6.90 67.37619 17.86324 21.59 105.72 -32.56 1.09 0.451
## 641 20951 8.95 67.38142 17.89326 24.19 107.00 -33.31 1.76 0.831

```

```

## 642 20978 9.08 67.49026 16.67291 24.71 105.04 -28.33 1.27 0.865
## 643 20995 5.58 67.53555 15.63790 22.93 107.59 -23.92 1.25 0.324
## 644 21029 4.78 67.63987 16.19408 22.54 104.98 -25.14 0.77 0.170
## 645 21036 5.40 67.65541 13.72445 21.84 108.06 -19.71 0.89 0.263
## 646 21039 5.47 67.66178 15.69194 22.55 104.17 -24.29 1.09 0.258
## 647 21053 6.50 67.69474 16.14875 24.28 98.20 -22.75 0.79 0.428
## 648 21066 7.03 67.73796 10.75179 22.96 104.19 -10.52 0.99 0.472
## 649 21099 8.59 67.81512 20.13326 21.81 102.78 -41.08 1.25 0.734
## 650 21123 9.53 67.90433 17.70985 23.41 105.81 -30.97 1.65 0.987
## 651 21137 6.01 67.96539 15.85164 22.25 107.59 -32.38 1.14 0.338
## 657 21256 10.69 68.40464 21.15096 24.98 109.30 -45.31 1.95 1.237
## 658 21261 10.74 68.42444 19.01412 21.06 102.30 -34.96 2.21 1.197
## 659 21267 6.62 68.44409 13.25193 22.80 101.77 -17.93 0.98 0.429
## 660 21273 4.65 68.46190 14.84449 21.39 103.69 -25.94 1.24 0.255
## 661 21280 8.48 68.49365 15.16370 24.02 101.93 -33.75 1.68 0.847
## 663 21317 7.90 68.64687 15.50469 23.19 100.66 -28.04 1.30 0.631
## 667 21459 6.01 69.12112 23.34099 22.60 109.97 -53.86 0.76 0.380
## 668 21474 6.64 69.16944 15.86938 22.99 93.78 -23.02 0.95 0.442
## 671 21588 5.78 69.53907 16.03339 21.96 113.05 -40.40 1.04 0.312
## 672 21589 4.27 69.53917 12.51087 21.79 101.73 -14.90 0.79 0.122
## 673 21637 7.51 69.71344 23.15010 22.60 104.30 -55.25 0.91 0.576
## 674 21654 7.96 69.73855 14.10563 20.81 103.31 -21.62 1.30 0.655
## 678 21723 10.04 69.96213 12.72852 23.95 102.00 -17.70 1.63 1.073
## 680 21762 9.47 70.10589 16.51373 23.65 91.94 -30.69 2.53 1.096
## 689 22044 5.39 71.10737 11.14617 20.73 98.87 -13.47 0.88 0.251
## 695 22224 9.60 71.70581 17.74841 24.11 96.93 -33.93 1.72 0.967
## 705 22496 7.10 72.59945 17.20274 22.96 102.78 -29.70 1.17 0.563
## 710 22607 6.30 72.95802 13.65519 23.91 106.84 -16.00 1.04 0.502

```

- ¿Que dimensiones tiene el dataset creado? Grafica un scatterplot de Vmag vs B.V

Tiene 88 observaciones con 9 variables cada una.

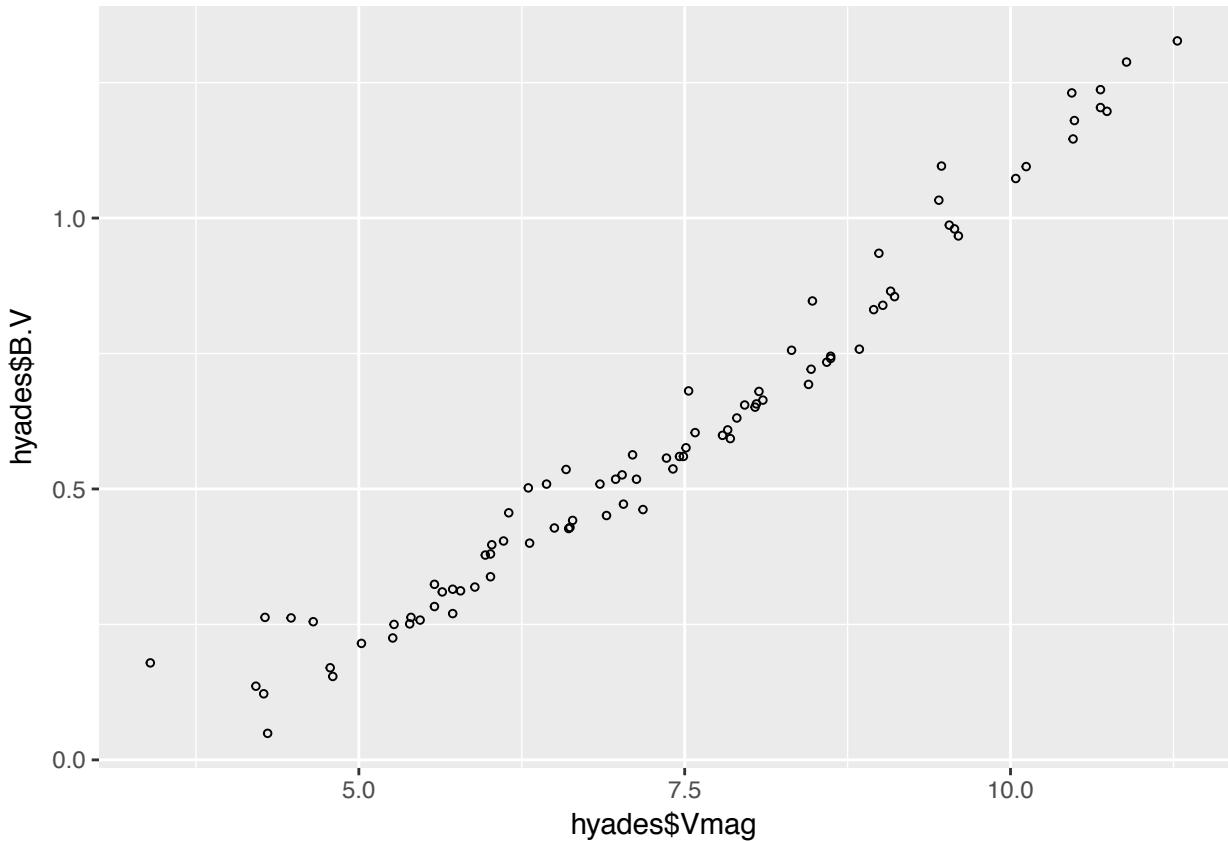
```
str(hyades)
```

```

## 'data.frame':   88 obs. of  9 variables:
## $ HIP  : int  18735 18946 19148 19207 19261 19316 19504 19781 19793 19808 ...
## $ Vmag : num  5.89 10.12 7.85 10.49 6.02 ...
## $ RA   : num  60.2 60.9 61.6 61.8 61.9 ...
## $ DE   : num  18.2 19.5 15.7 15.3 15.2 ...
## $ Plx  : num  22 23.1 21.4 23.6 21.3 ...
## $ pmRA : num  129 119 119 123 127 ...
## $ pmDE : num  -28.3 -34.2 -19.6 -19 -22.8 ...
## $ e_Plx: num  0.81 2.12 1.47 2.26 1.03 2.59 0.92 1.27 1.14 2.3 ...
## $ B.V  : num  0.319 1.095 0.593 1.18 0.397 ...

```

```
ggplot(hyades, aes(x=hyades$Vmag, y=hyades$B.V)) + geom_point(size=1, shape=1)
```



Analizando el gráfico podemos ver que la relación de las variables es totalmente lineal, lo que podría ayudarnos por ejemplo a poder predecir mediante regresión posibles valores perdidos en una de ambas. También podremos descartar una de ellas en función de otra en nuestros modelos ya que ambas pueden aportar información muy similar al estar tan correlacionadas.

Ejemplo 2, iris dataset

Vamos a utilizar el ejemplo del dataset iris que está incluido en la distribución de R. Este dataset fue creado por **Douglas Fisher**. Consta de tres clases y tipos de 3 clases de tipos de flores:

1. Setosa
2. Virginica
3. Versicolor

Cada una de ellas con cuatro atributos:

1. sepal width
 2. sepal length
 3. petal width
 4. petal length
- Inspecciona las primeras filas del dataset y calcula el `summary()` del mismo con cada atributo del dataset.

Esto nos ayuda a tener una idea inicial de como son los datos, en este caso están perfectamente equilibradas las clases, los rangos de valores son mas o menos similares y el IRQ es similar.

```

iris.data<-iris
head(iris.data)

##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa

summary(iris.data)

##   Sepal.Length     Sepal.Width    Petal.Length    Petal.Width
## Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
## 1st Qu.:5.100  1st Qu.:2.800  1st Qu.:1.600  1st Qu.:0.300
## Median  :5.800  Median  :3.000  Median  :4.350  Median  :1.300
## Mean    :5.843  Mean    :3.057  Mean    :1.758  Mean    :1.199
## 3rd Qu.:6.400  3rd Qu.:3.300  3rd Qu.:5.100  3rd Qu.:1.800
## Max.    :7.900  Max.    :4.400  Max.    :6.900  Max.    :2.500

##           Species
## setosa      :50
## versicolor:50
## virginica :50
##
## 
## 

str(iris.data)

## 'data.frame': 150 obs. of  5 variables:
## $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
## $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
## $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
## $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
## $ Species     : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...

```

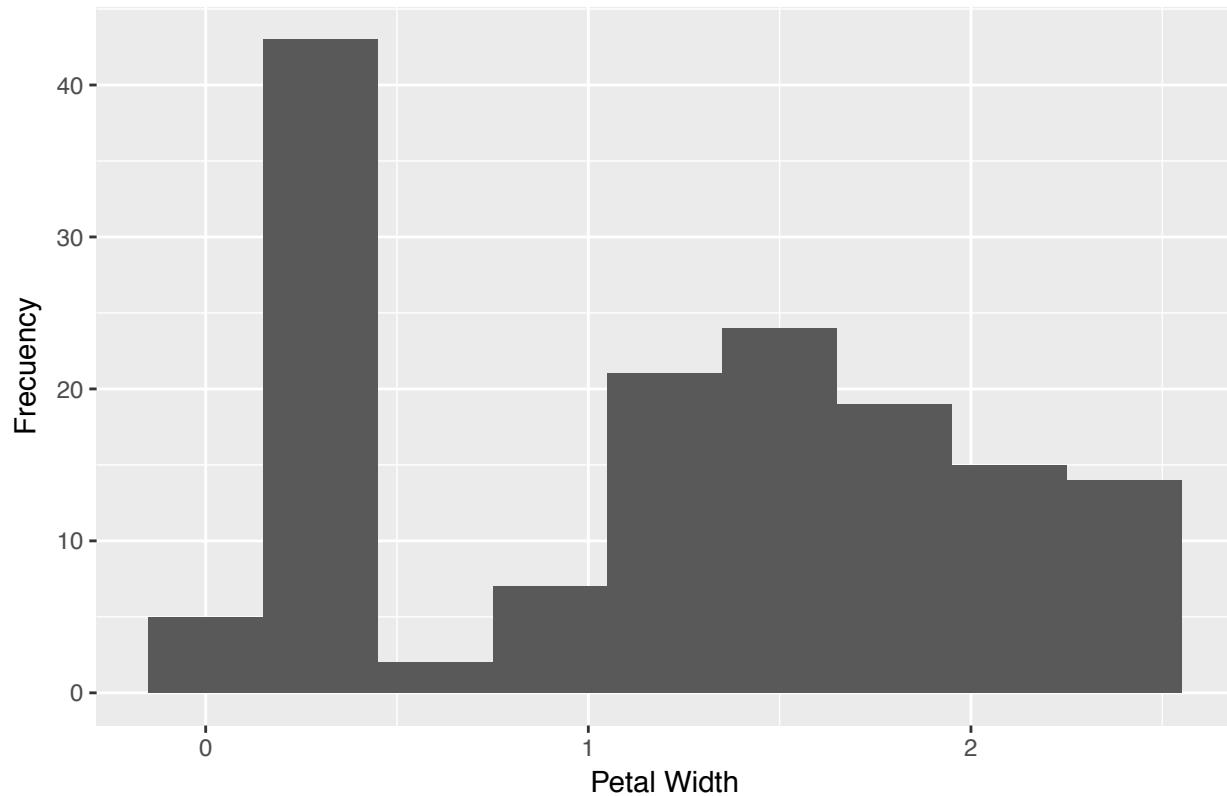
- Crea un histograma de petal.width , teniendo en cuenta que el número de bins es variable fija este a 9. Añádele color y nombres al eje x “Petal Width”y al gráfico dale el nombre de “Histogram of Petal Width”.

```

library(ggplot2)
ggplot(data=iris.data, aes(iris.data$Petal.Width)) + geom_histogram(binwidth = 0.3) + labs(title="Histogram of Petal Width")

```

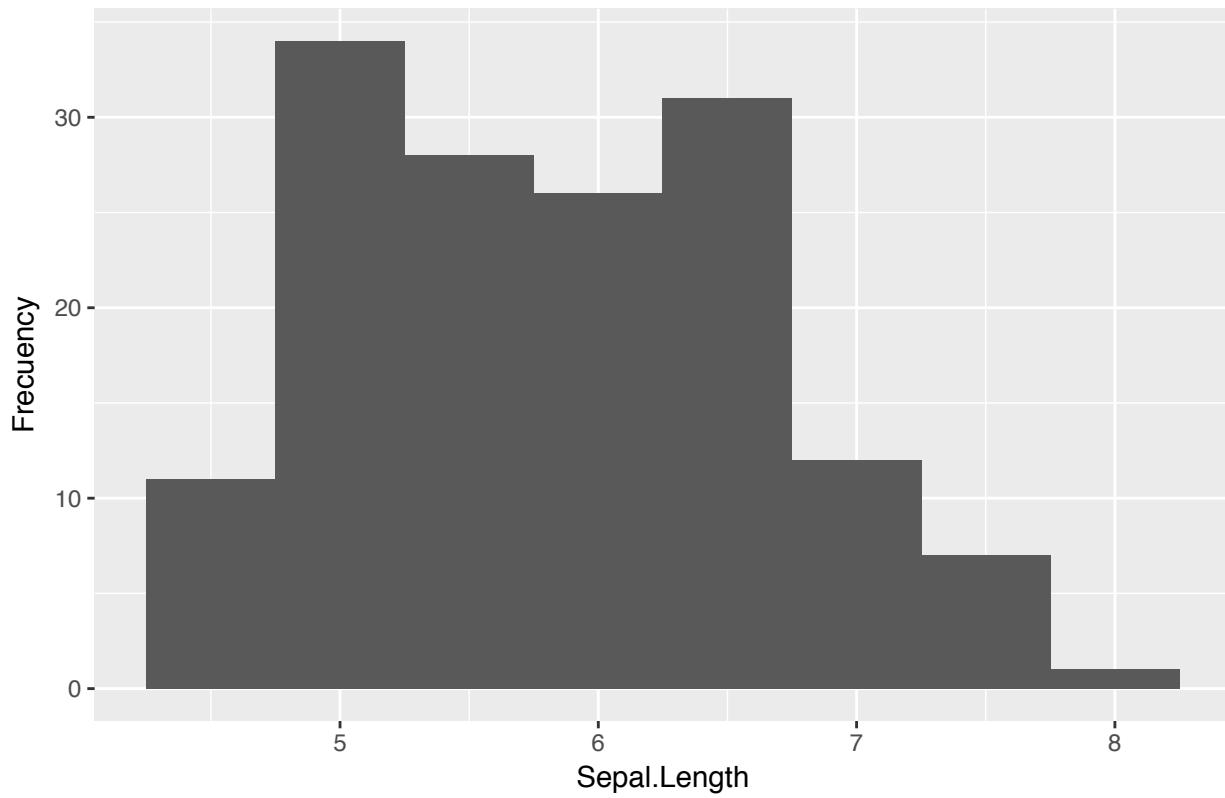
Histogram of Petal Width



- Crea un histograma para cada variable.

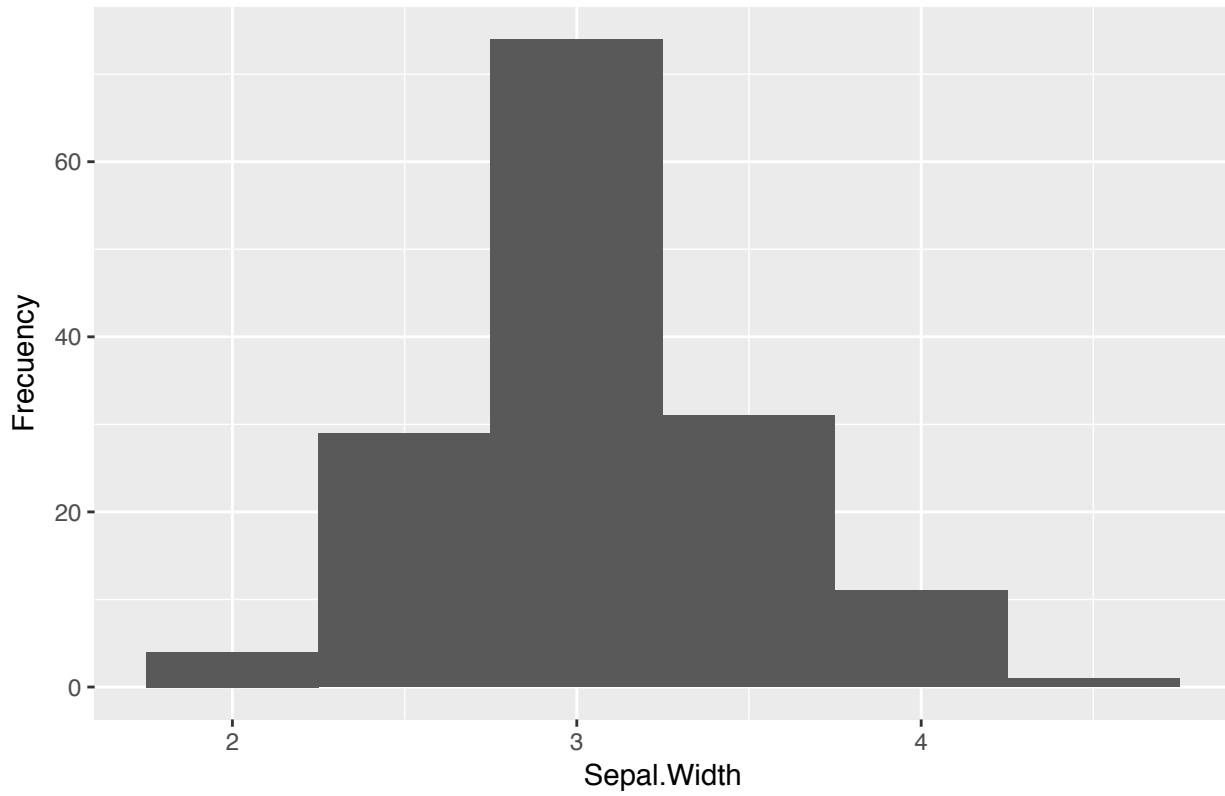
```
ggplot(data=iris.data, aes(iris.data$Sepal.Length)) + geom_histogram(binwidth = 0.5) + labs(title="Histograma de Sepal Length")
```

Histogram of Sepal.Length



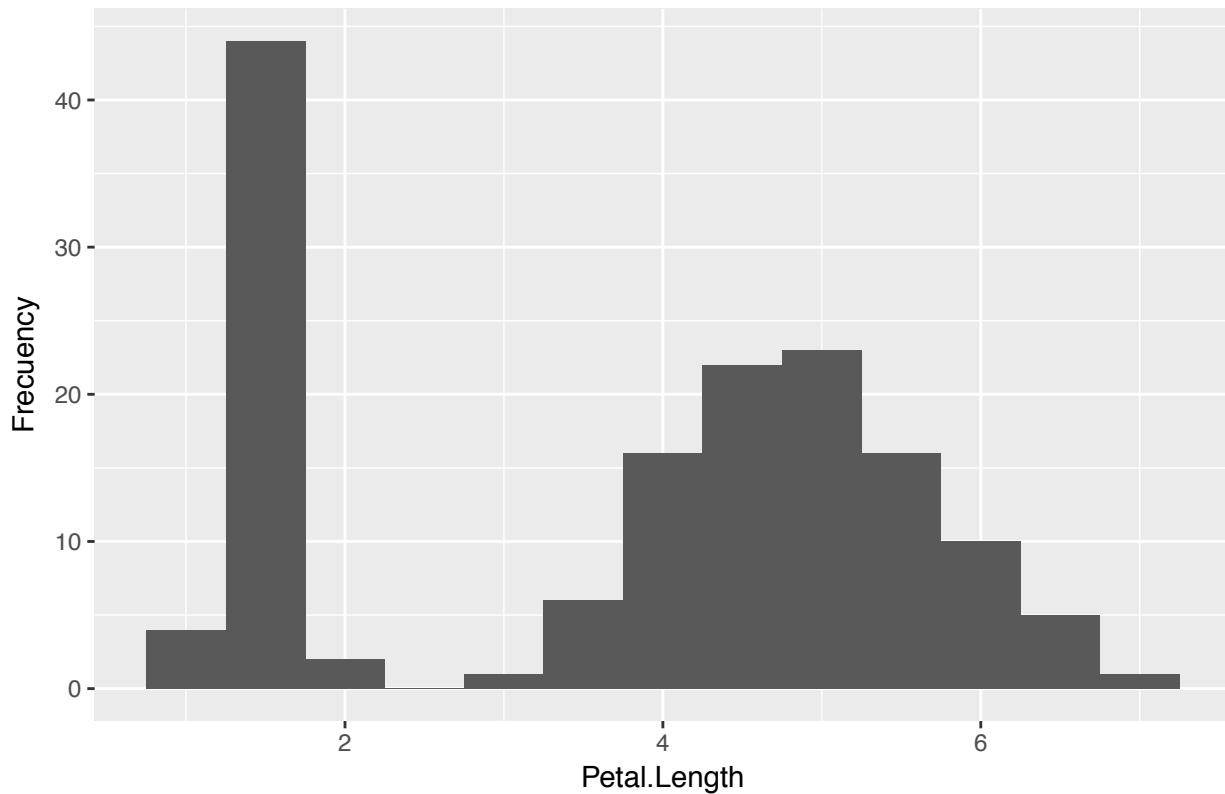
```
ggplot(data=iris.data, aes(iris.data$Sepal.Width)) + geom_histogram(binwidth = 0.5) + labs(title="Histogram of Sepal.Length")
```

Histogram of Sepal.Width



```
ggplot(data=iris.data, aes(iris.data$Petal.Length)) + geom_histogram(binwidth = 0.5) + labs(title="Histogram of Sepal.Width")
```

Histogram of Petal.Length



- Crea los cuartiles del dataset

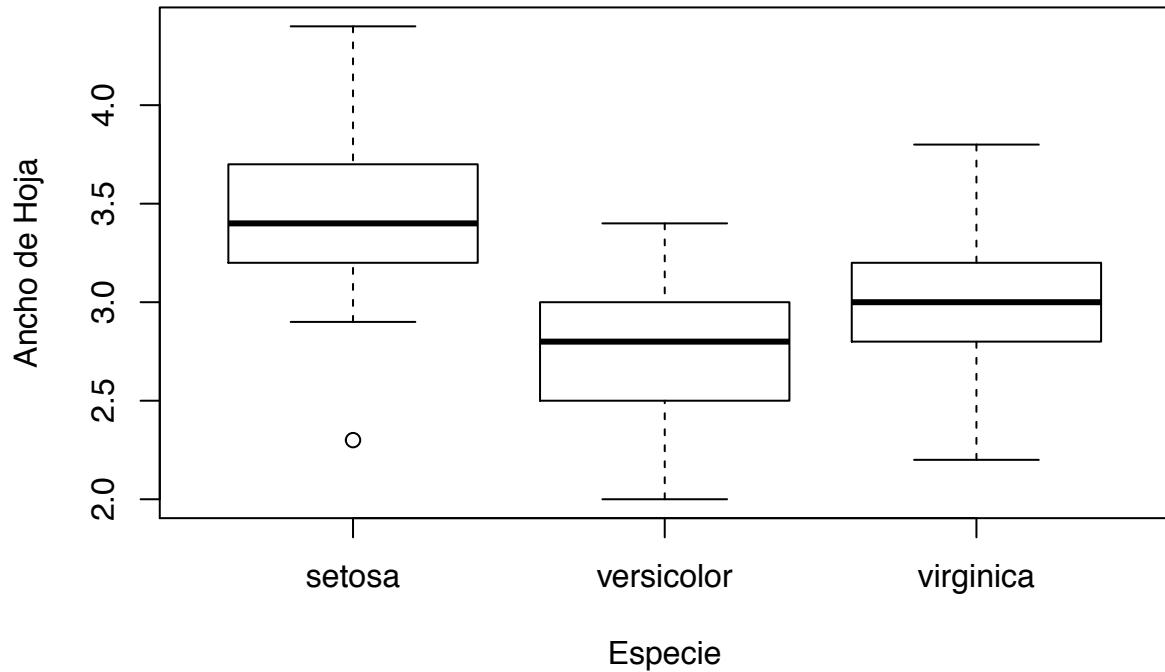
```
summary(iris.data, na.rm=T)
```

```
##   Sepal.Length   Sepal.Width   Petal.Length   Petal.Width
## Min.    :4.300   Min.    :2.000   Min.    :1.000   Min.    :0.100
## 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
## Median  :5.800   Median  :3.000   Median  :4.350   Median  :1.300
## Mean    :5.843   Mean    :3.057   Mean    :3.758   Mean    :1.199
## 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
## Max.    :7.900   Max.    :4.400   Max.    :6.900   Max.    :2.500
## 
##   Species
##   setosa    :50
##   versicolor:50
##   virginica :50
## 
```

- Representa en un boxplot la variable de ancho de hoja dependiendo del tipo de hoja que tengan.

```
boxplot(iris.data$Sepal.Width~iris.data$Species, main="Ancho de Hoja por Especie", ylab="Ancho de Hoja")
```

Ancho de Hoja por Especie

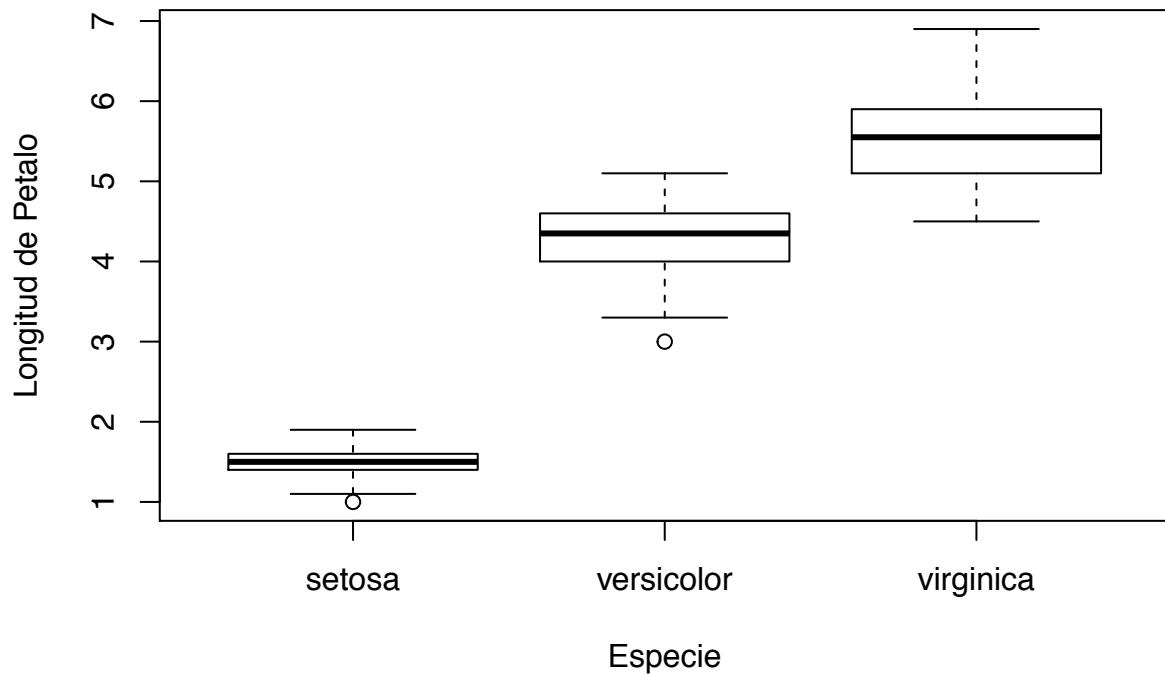


Este gráfico no podemos ver una tendencia clara, más haya de que encontramos los cuartiles bien distribuidos y salvo un caso la existencia de outliers es nula.

- Crea los boxplot de la longitud del pétalo en función de la especie de Iris.

```
boxplot(iris.data$Petal.Length~iris.data$Species, main="Longitud Petalo de Hoja por Especie", ylab="Longitud de Petalo")
```

Longitud Petalo de Hoja por Especie



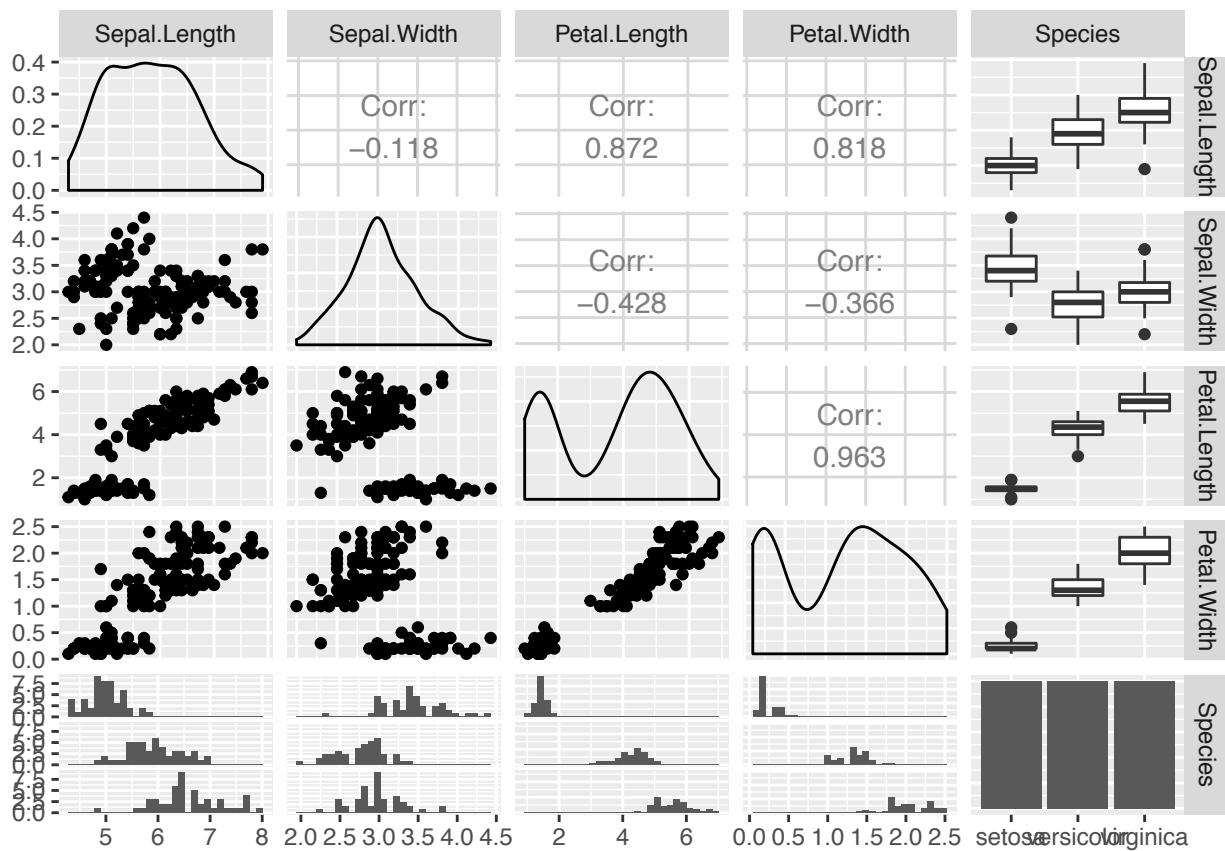
Este gráfico ya nos ofrece mucha información, vemos una tendencia entre los datos de setosa a virginica, en función del Sepal Length muy clara, ya que cuanto mayor es cambia la especie. También vemos un outlier en la especie versicolor y setosa.

- Compara con scatter plots las variables entre sí.

Vamos a volver a usar ggpairs, que nos ofrece mucha más información.

```
ggpairs(iris.data)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



SWISS dataset

El conjunto de datos “swiss” contiene una medida estandarizada de fecundidad y varios indicadores socioeconómicos para cada una de las 47 provincias francófonas de Suiza.

```
head(swiss)
```

```
##          Fertility Agriculture Examination Education Catholic
## Courtelary      80.2       17.0         15      12    9.96
## Delemont       83.1       45.1          6      9    84.84
## Franches-Mnt   92.5       39.7          5      5    93.40
## Moutier        85.8       36.5         12      7    33.77
## Neuveville     76.9       43.5         17      15    5.16
## Porrentruy     76.1       35.3          9      7    90.57
```

```

##           Infant.Mortality
## Courtelary                  22.2
## Delemont                   22.2
## Franches-Mnt                20.2
## Moutier                     20.3
## Neuveville                  20.6
## Porrentruy                  26.6

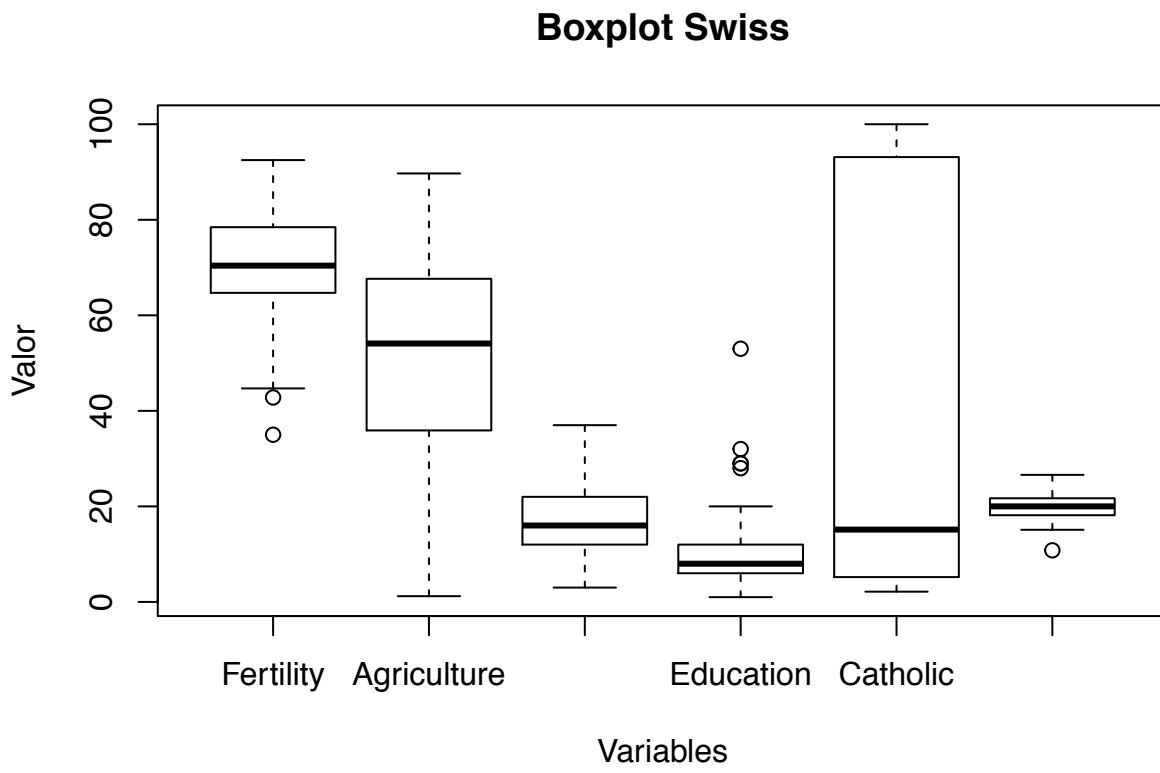
```

- ¿Qué diagrama dibujaría para mostrar la distribución de todos los valores? ¿Qué conclusiones sacarías?

Para obtener la distribución de todos los valores realizaría un Boxplot. En base al diagrama, podemos concluir que hay un valor que claramente es un %, **Catholic**, este ademas ofrece una dispersión entre los que están por encima de la mediana muy alta, frente a los que están por debajo de la misma.

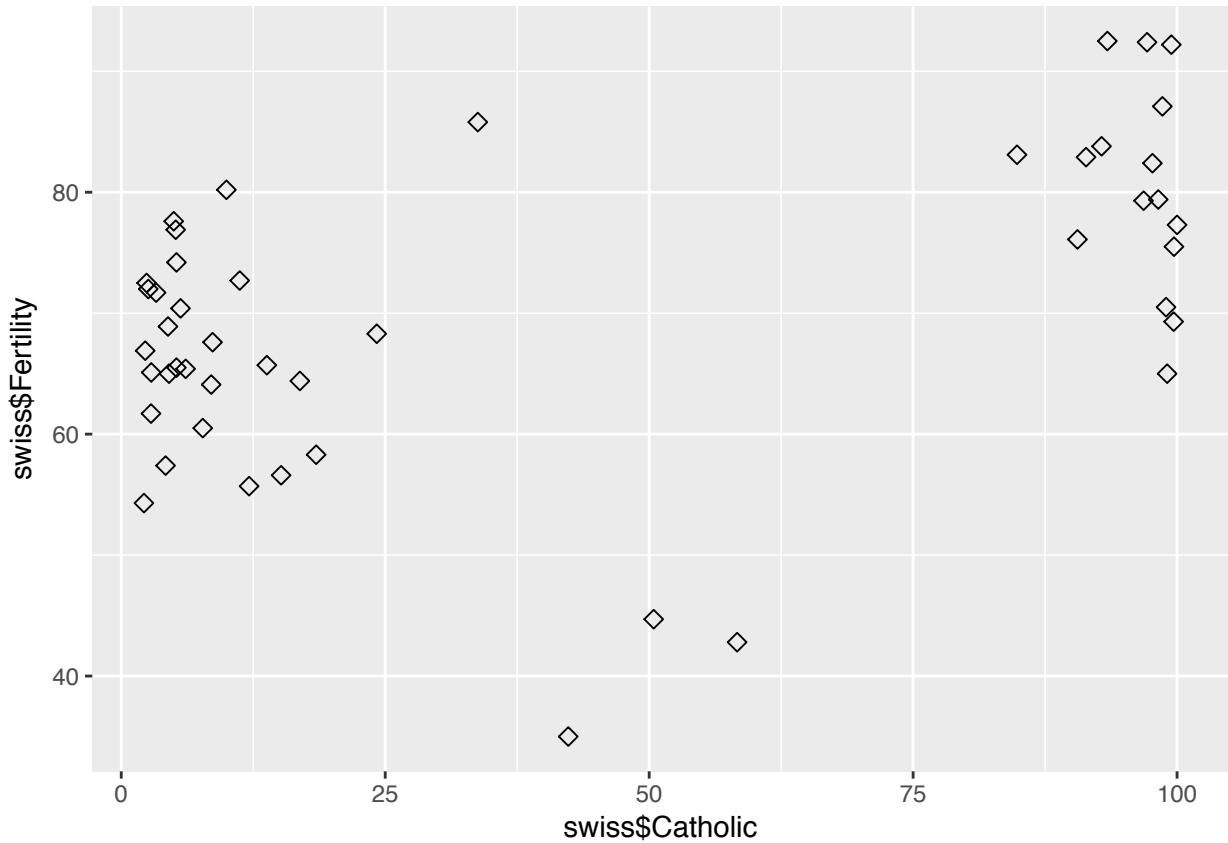
Por otro lado, tenemos bastantes outliers en **Education**.

```
boxplot(swiss, main="Boxplot Swiss", xlab="Variables", ylab="Valor")
```



- Dibuje gráficos para cada variable. ¿Qué puede concluir de las distribuciones con respecto a su forma y posibles valores atípicos?
- Dibuja un diagrama de dispersión de Fertilidad frente a % Catholic. ¿Qué tipo de áreas tienen las tasas de fertilidad más bajas?

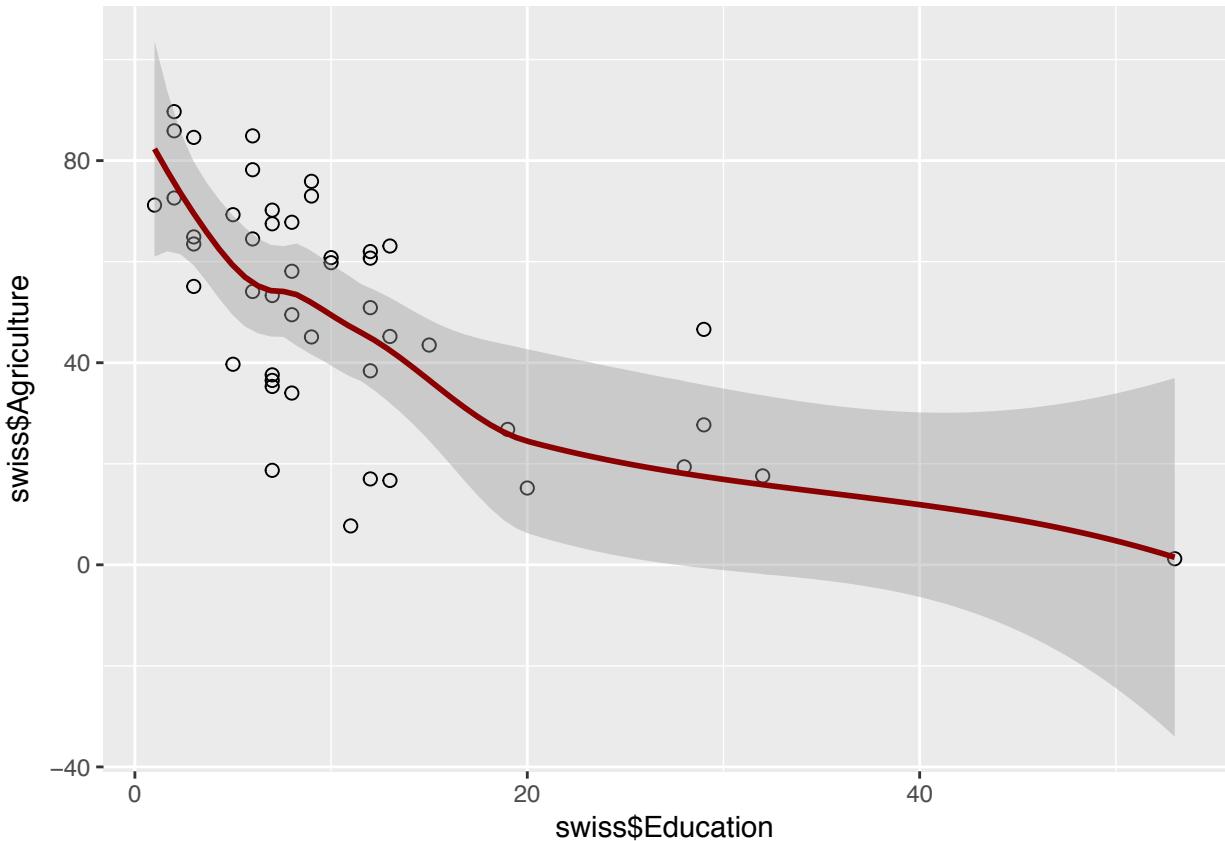
```
ggplot(swiss, aes(x=swiss$Catholic, y=swiss$Fertility))+
  geom_point(size=2, shape=5)
```



Las áreas con menor fertilidad son aquellas que tienen un % de católicos entre el 50% y 60%.

- ¿Qué tipo de relación existe entre las variables Educación y Agricultura?

```
ggplot(swiss, aes(x=swiss$Education, y=swiss$Agriculture))+
  geom_point(size=2, shape=1)+
  geom_smooth(method=loess, color="darkred")
```



Aunque la relación no es muy clara, una vez visto el gráfico podemos ver que a menores valores de educación, mayores de educación por lo que el gráfico se asemeja a una función exponencial con una correlación negativa bastante fuerte.

Aceites de Oliva

El conjunto de datos de aceites de oliva es bien conocido y se puede encontrar en varios paquetes, por ejemplo, como **olives** en **extracat**. La fuente original de los datos es el artículo [Forina et al., 1983].

Vamos a obtener el dataset:

```
install.packages("extracat")

## Error in install.packages : Updating loaded packages
library(extracat)
head(olives)

##           Area Region palmitic palmitoleic stearic oleic linoleic
## 1 North-Apulia   South     1088        73    224  7709    781
## 2 North-Apulia   South      911        54    246  8113    549
## 3 North-Apulia   South     966        57    240  7952    619
## 4 North-Apulia   South    1051        67    259  7771    672
## 5 North-Apulia   South      911        49    268  7924    678
## 6 North-Apulia   South    1100        61    235  7728    734
##   linolenic arachidic eicosenoic Test.Training
## 1       31       61       29     Training
## 2       31       63       29     Training
```

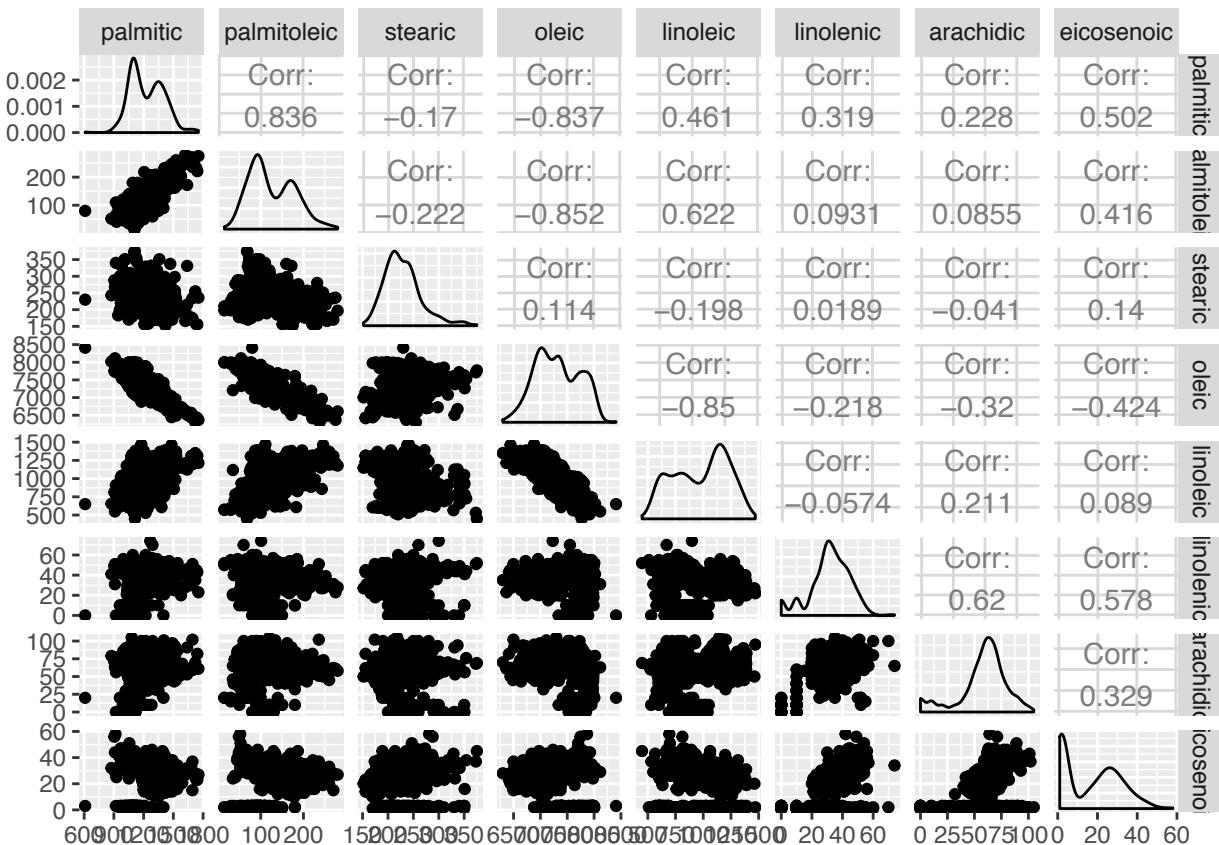
```

## 3      50      78      35 Training
## 4      50      80      46 Training
## 5      51      70      44 Training
## 6      39      64      35 Training

```

- Dibuje un scatterplot de las ocho variables continuas. ¿Cuáles de los ácidos grasos están fuertemente asociados positivamente y cuáles fuertemente asociados negativamente?

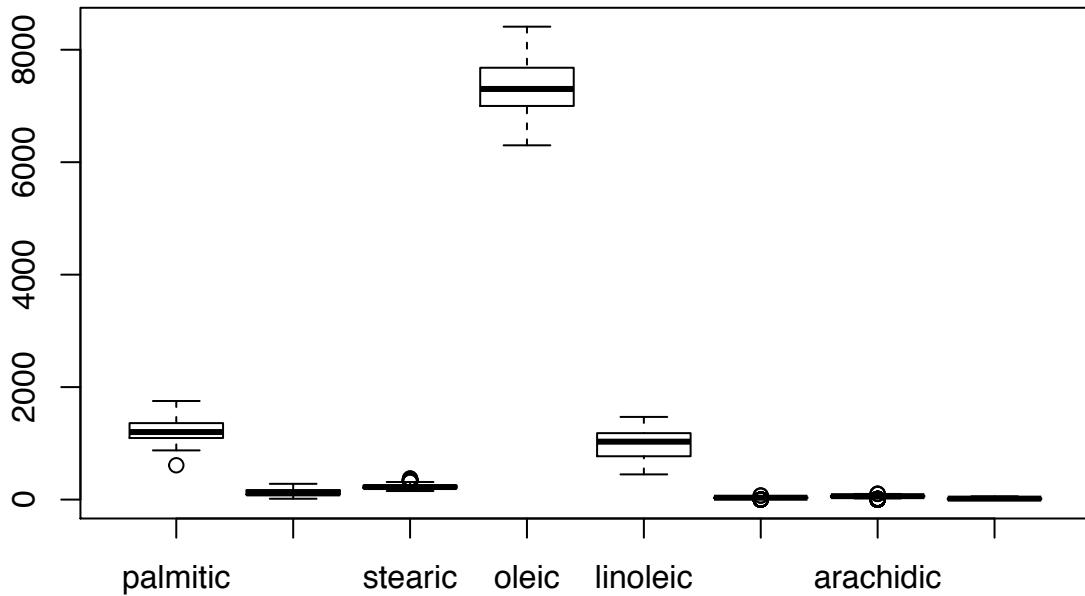
```
ggpairs(olives[3:10])
```



Nuevamente con ggpairs obtenemos la información que deseamos, ya que nos ofrece las correlaciones negativas y positicas entre cada uno de los gráficos.

- ¿Hay valores atípicos u otras características que valga la pena mencionar?

```
boxplot(olives[3:10])
```



Aunque deberíamos usar normalización para ver los datos representados en una misma escala podemos concluir que hay ciertos valores atípicos o outliers, en **stearic**, **linolenic** sobre todo.

HSAUR2.

El conjunto de datos se llama Lanza del paquete HSAUR2, por ello, lo primero será obtener los datos.

```
install.packages("HSAUR2")
```

```
## Error in install.packages : Updating loaded packages
```

```
library("HSAUR2")
```

```
Lanza
```

```
##   study treatment classification
## 1      I Misoprostol      1
## 2      I Misoprostol      1
## 3      I Misoprostol      1
## 4      I Misoprostol      1
## 5      I Misoprostol      1
## 6      I Misoprostol      1
## 7      I Misoprostol      1
## 8      I Misoprostol      1
## 9      I Misoprostol      1
## 10     I Misoprostol      1
## 11     I Misoprostol      1
## 12     I Misoprostol      1
## 13     I Misoprostol      1
## 14     I Misoprostol      1
## 15     I Misoprostol      1
## 16     I Misoprostol      1
## 17     I Misoprostol      1
## 18     I Misoprostol      1
## 19     I Misoprostol      1
## 20     I Misoprostol      1
## 21     I Misoprostol      1
```

## 22	I Misoprostol	2
## 23	I Misoprostol	2
## 24	I Misoprostol	3
## 25	I Misoprostol	3
## 26	I Misoprostol	3
## 27	I Misoprostol	3
## 28	I Misoprostol	4
## 29	I Misoprostol	4
## 30	I Placebo	1
## 31	I Placebo	1
## 32	I Placebo	2
## 33	I Placebo	2
## 34	I Placebo	3
## 35	I Placebo	3
## 36	I Placebo	3
## 37	I Placebo	3
## 38	I Placebo	4
## 39	I Placebo	4
## 40	I Placebo	4
## 41	I Placebo	4
## 42	I Placebo	4
## 43	I Placebo	4
## 44	I Placebo	4
## 45	I Placebo	4
## 46	I Placebo	4
## 47	I Placebo	5
## 48	I Placebo	5
## 49	I Placebo	5
## 50	I Placebo	5
## 51	I Placebo	5
## 52	I Placebo	5
## 53	I Placebo	5
## 54	I Placebo	5
## 55	I Placebo	5
## 56	I Placebo	5
## 57	I Placebo	5
## 58	I Placebo	5
## 59	I Placebo	5
## 60	II Misoprostol	1
## 61	II Misoprostol	1
## 62	II Misoprostol	1
## 63	II Misoprostol	1
## 64	II Misoprostol	1
## 65	II Misoprostol	1
## 66	II Misoprostol	1
## 67	II Misoprostol	1
## 68	II Misoprostol	1
## 69	II Misoprostol	1
## 70	II Misoprostol	1
## 71	II Misoprostol	1
## 72	II Misoprostol	1
## 73	II Misoprostol	1
## 74	II Misoprostol	1
## 75	II Misoprostol	1

## 76	II Misoprostol	1
## 77	II Misoprostol	1
## 78	II Misoprostol	1
## 79	II Misoprostol	1
## 80	II Misoprostol	2
## 81	II Misoprostol	2
## 82	II Misoprostol	2
## 83	II Misoprostol	2
## 84	II Misoprostol	3
## 85	II Misoprostol	3
## 86	II Misoprostol	3
## 87	II Misoprostol	3
## 88	II Misoprostol	3
## 89	II Misoprostol	3
## 90	II Placebo	1
## 91	II Placebo	1
## 92	II Placebo	1
## 93	II Placebo	1
## 94	II Placebo	1
## 95	II Placebo	1
## 96	II Placebo	1
## 97	II Placebo	1
## 98	II Placebo	2
## 99	II Placebo	2
## 100	II Placebo	2
## 101	II Placebo	2
## 102	II Placebo	3
## 103	II Placebo	3
## 104	II Placebo	3
## 105	II Placebo	3
## 106	II Placebo	3
## 107	II Placebo	3
## 108	II Placebo	3
## 109	II Placebo	3
## 110	II Placebo	3
## 111	II Placebo	4
## 112	II Placebo	4
## 113	II Placebo	4
## 114	II Placebo	4
## 115	II Placebo	5
## 116	II Placebo	5
## 117	II Placebo	5
## 118	II Placebo	5
## 119	II Placebo	5
## 120	III Misoprostol	1
## 121	III Misoprostol	1
## 122	III Misoprostol	1
## 123	III Misoprostol	1
## 124	III Misoprostol	1
## 125	III Misoprostol	1
## 126	III Misoprostol	1
## 127	III Misoprostol	1
## 128	III Misoprostol	1
## 129	III Misoprostol	1

## 130	III Misoprostol	1
## 131	III Misoprostol	1
## 132	III Misoprostol	1
## 133	III Misoprostol	1
## 134	III Misoprostol	1
## 135	III Misoprostol	1
## 136	III Misoprostol	1
## 137	III Misoprostol	1
## 138	III Misoprostol	1
## 139	III Misoprostol	1
## 140	III Misoprostol	2
## 141	III Misoprostol	2
## 142	III Misoprostol	2
## 143	III Misoprostol	2
## 144	III Misoprostol	3
## 145	III Misoprostol	3
## 146	III Misoprostol	3
## 147	III Misoprostol	4
## 148	III Misoprostol	5
## 149	III Misoprostol	5
## 150	III Placebo	2
## 151	III Placebo	2
## 152	III Placebo	3
## 153	III Placebo	3
## 154	III Placebo	3
## 155	III Placebo	3
## 156	III Placebo	3
## 157	III Placebo	4
## 158	III Placebo	4
## 159	III Placebo	4
## 160	III Placebo	4
## 161	III Placebo	4
## 162	III Placebo	5
## 163	III Placebo	5
## 164	III Placebo	5
## 165	III Placebo	5
## 166	III Placebo	5
## 167	III Placebo	5
## 168	III Placebo	5
## 169	III Placebo	5
## 170	III Placebo	5
## 171	III Placebo	5
## 172	III Placebo	5
## 173	III Placebo	5
## 174	III Placebo	5
## 175	III Placebo	5
## 176	III Placebo	5
## 177	III Placebo	5
## 178	III Placebo	5
## 179	IV Misoprostol	1
## 180	IV Misoprostol	2
## 181	IV Misoprostol	2
## 182	IV Misoprostol	2
## 183	IV Misoprostol	2

```

## 184    IV Misoprostol      3
## 185    IV Misoprostol      3
## 186    IV Misoprostol      3
## 187    IV Misoprostol      3
## 188    IV Misoprostol      3
## 189    IV     Placebo      4
## 190    IV     Placebo      4
## 191    IV     Placebo      4
## 192    IV     Placebo      4
## 193    IV     Placebo      5
## 194    IV     Placebo      5
## 195    IV     Placebo      5
## 196    IV     Placebo      5
## 197    IV     Placebo      5
## 198    IV     Placebo      5

```

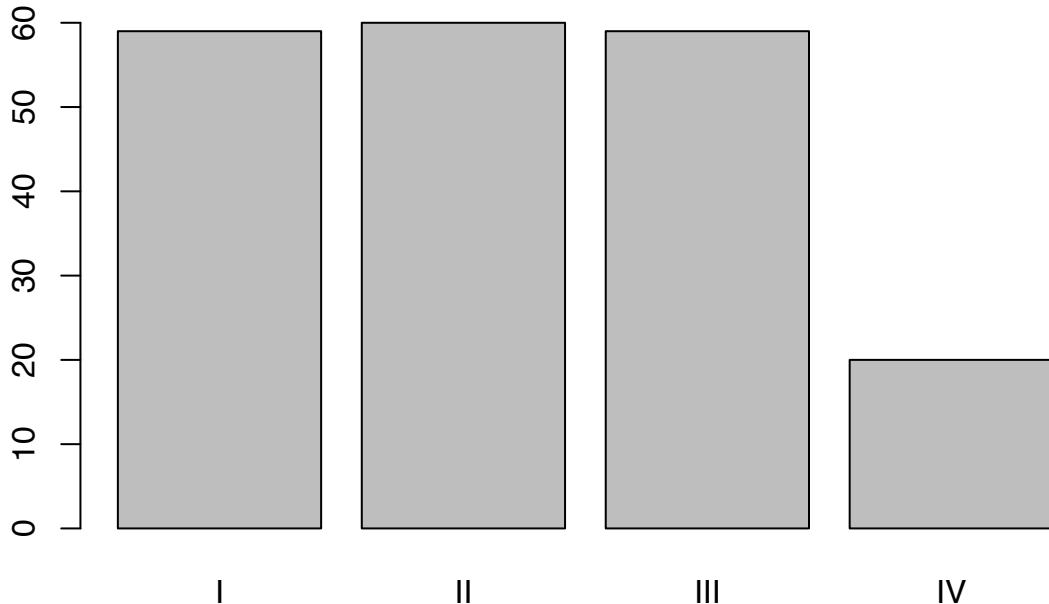
*Se informan los datos de cuatro estudios. Dibuje un diagrama para mostrar si los cuatro estudios son igualmente grandes.

Para obtener la información sobre el número de elementos de una determinada variable el mejor gráfico es un gráfico de barras.

```

estudy.freq <- table(Lanza$study)
barplot(estudy.freq)

```



Vemos que los estudios de tipo IV, están en clara minoría, por lo que podremos tener sesgos en estas clases a la hora de clasificar a los que tendremos que prestar mucha atención.

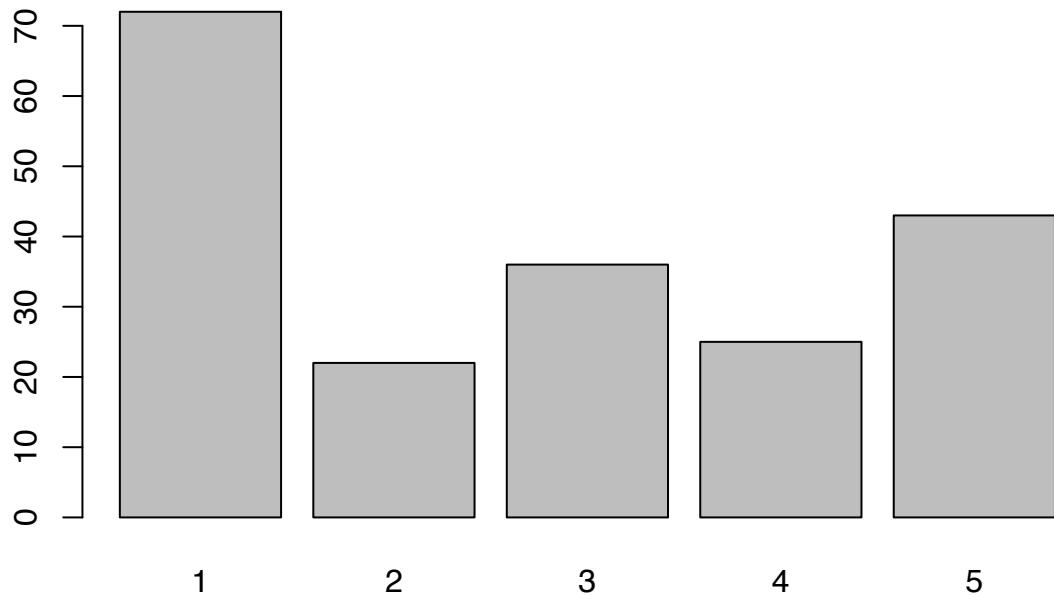
- El resultado se mide por la clasificación de la variable con puntuaciones de 1 (mejor) a 5 (peor). ¿Cómo describirías la distribución?

Pues hay una clara diferencia entre las clasificadas como 1 y las demás, que con ciertos matices, están bastante bien distribuidas.

```

class.freq <- table(Lanza$classification)
barplot(class.freq)

```



Cáncer de mama

El paquete vcdExtra incluye datos de un viejo estudio de cáncer de mama sobre la supervivencia o muerte de 474 pacientes.

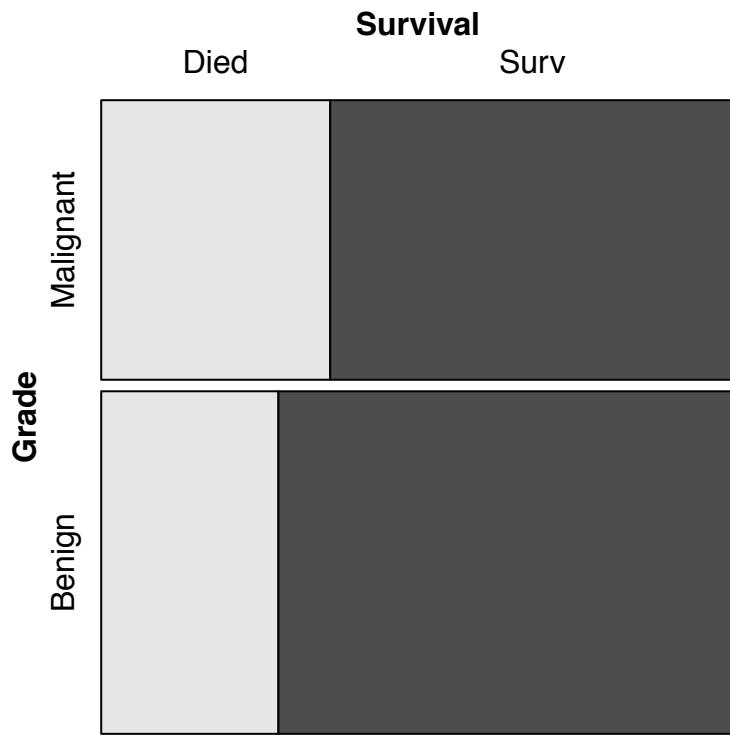
```
install.packages("vcdExtra")
```

```
## Error in install.packages : Updating loaded packages
library("vcdExtra")
cancer.dataframe<-as.data.frame(Cancer)
```

- Convierta los datos en un data frame y dibuje gráficos para comparar las tasas de supervivencia, primero, por grado de malignidad y, en segundo lugar, por centro de diagnóstico.

Para este caso lo mejor es un gráfico de mosaico.

```
library(vcd)
mosaic(Survival ~ Grade, data = cancer.dataframe, shade=TRUE)
```



```
mosaic(Survival ~ Center, data = cancer.dataframe, shade=TRUE)
```

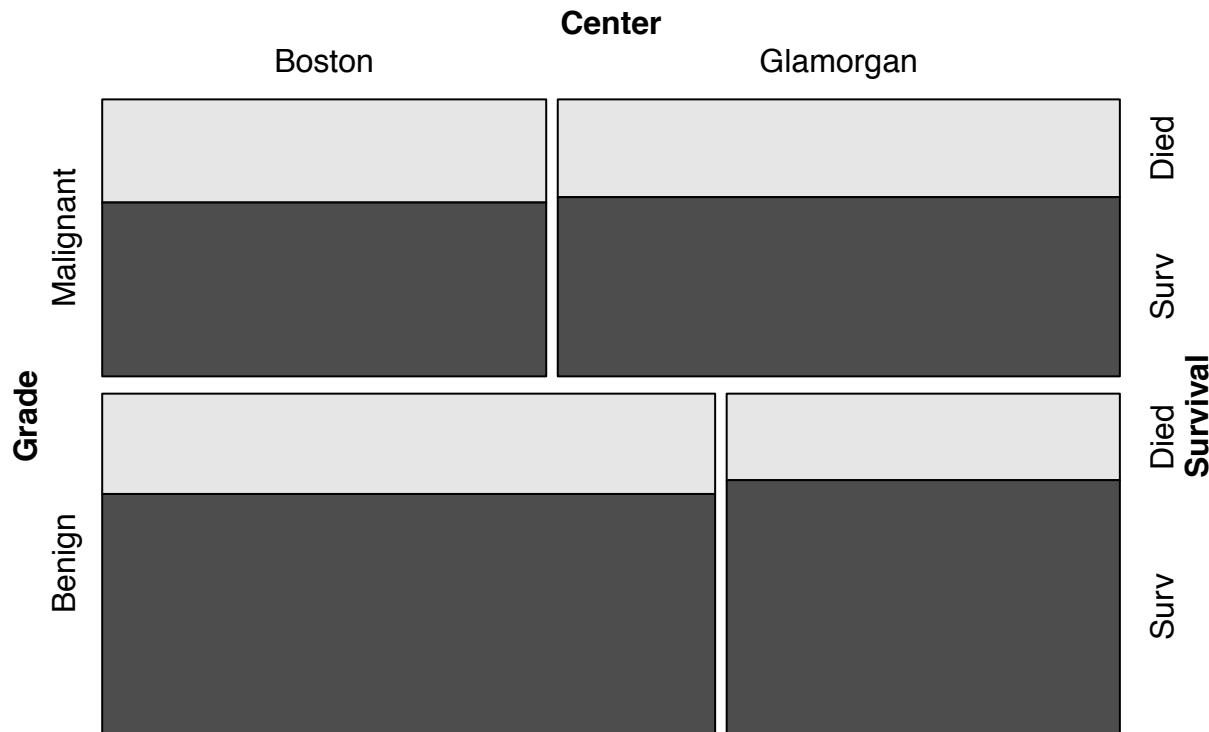


En cuanto a centros las tasas son muy parecidas en cambio, cuando vemos si es benigno o maligno si que, como la propia logica nos diría, al ser maligno los supervivientes son menos.

- ¿Qué diagrama dibujaría para comparar las tasas de supervivencia tanto por grado de malignidad como por centro de diagnóstico? ¿Importa el orden de las variables explicativas?

Para meter todo dentro de un gráfico volvería a usar el gráfico de mosaico.

```
mosaic(Survival~Grade+Center, data = cancer.dataframe, shade=TRUE)
```



Crabs

- Dataset Crabs (del paquete MASS) [Venables y Ripley, 2002]. Los autores inicialmente se transforman a una escala logarítmica y luego escriben que:

“The data are very highly correlated and scatterplot matrices and brush plots [i.e. interactive graphics] are none too revealing.”.

Utilizando gráficos generales, comente si la transformación logarítmica fue una buena idea y si está de acuerdo con su afirmación sobre las correlaciones.

```
install.packages("MASS")
```

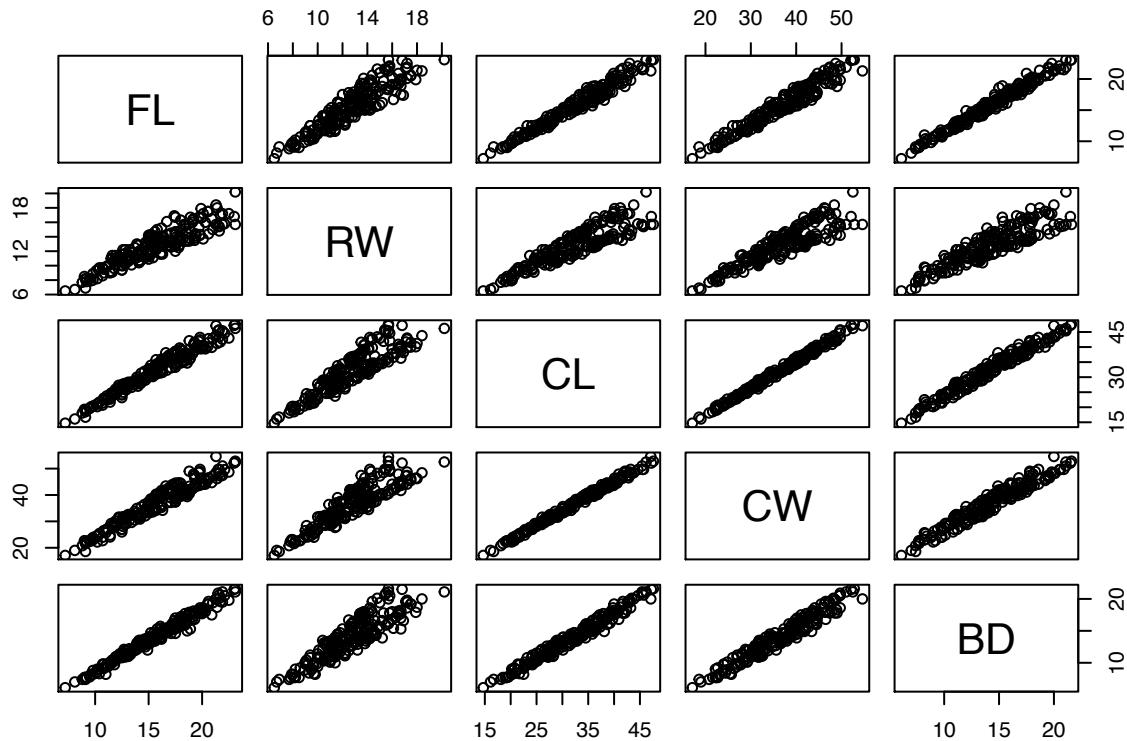
```
## Error in install.packages : Updating loaded packages
```

```
library(MASS)
head(crabs)
```

```
##   sp sex index   FL   RW   CL   CW   BD
## 1  B   M      1  8.1  6.7 16.1 19.0  7.0
## 2  B   M      2  8.8  7.7 18.1 20.8  7.4
## 3  B   M      3  9.2  7.8 19.0 22.4  7.7
## 4  B   M      4  9.6  7.9 20.1 23.1  8.2
## 5  B   M      5  9.8  8.0 20.3 23.0  8.2
## 6  B   M      6 10.8  9.0 23.0 26.5  9.8
```

En base a los datos, lo primero que tendremos que hacer es pintar los pairs de las variables continuas, ya que las demás no aportan información para lo que queremos comprobar.

```
pairs(crabs[4:8])
```



Definitivamente si que fue una buena idea la transformación logarítmica con las variables, ya que todas presenta una correlación lineal total.

Como crear subgrupos de datos en R

Busca información sobre la función `cut()`. Para ilustrar su uso vamos a utilizar el dataset `state.x77`. Si no lo tienes instalado instala el paquete R-Datasets. Usa la función `head()` para ver como son tus datos.

```
head(state.x77)
```

```
##          Population Income Illiteracy Life_Exp Murder HS_Grad Frost
## Alabama      3615    3624      2.1   69.05   15.1    41.3    20
## Alaska       365     6315      1.5   69.31   11.3    66.7   152
## Arizona     2212    4530      1.8   70.55    7.8    58.1    15
## Arkansas    2110    3378      1.9   70.66   10.1    39.9    65
## California   21198   5114      1.1   71.71   10.3    62.6    20
## Colorado     2541    4884      0.7   72.06    6.8    63.9   166
##          Area
## Alabama     50708
## Alaska      566432
## Arizona    113417
## Arkansas    51945
## California  156361
## Colorado    103766
```

```
state<-as.data.frame(state.x77)
```

- Extrae la columna Frost y asigna el resultado a la variable frost

```
frost<-state$Frost
```

- Tu nuevo objeto es un vector numérico. Ahora intenta agrupar los datos en frost en tres niveles. Para crear bins en tus datos puedes utilizar la función cut().

```
levels.frost<-cut(frost, breaks=3)
levels.frost
```

```
## [1] (-0.188,62.7] (125,188]      (-0.188,62.7] (62.7,125]      (-0.188,62.7]
## [6] (125,188]      (125,188]      (62.7,125]      (-0.188,62.7] (-0.188,62.7]
## [11] (-0.188,62.7] (125,188]      (125,188]      (62.7,125]      (125,188]
## [16] (62.7,125]      (62.7,125]      (-0.188,62.7] (125,188]      (62.7,125]
## [21] (62.7,125]      (62.7,125]      (125,188]      (-0.188,62.7] (62.7,125]
## [26] (125,188]      (125,188]      (125,188]      (125,188]      (62.7,125]
## [31] (62.7,125]      (62.7,125]      (62.7,125]      (125,188]      (62.7,125]
## [36] (62.7,125]      (-0.188,62.7] (125,188]      (125,188]      (62.7,125]
## [41] (125,188]      (62.7,125]      (-0.188,62.7] (125,188]      (125,188]
## [46] (62.7,125]      (-0.188,62.7] (62.7,125]      (125,188]      (125,188]
## Levels: (-0.188,62.7] (62.7,125] (125,188]
```

- ¿Qué obtienes como nombres de los niveles?

Los nombres de los niveles son los rangos de valores. Es decir, el mínimo y máximo valor entre el que se encontraba un valor anterior del vector.

- En la realidad no existen estados que tengan frost en días negativos. Esto es porque R añade un poco de padding. Prueba a solucionar el problema utilizando el parámetro include.lowest=TRUE en cut().

```
levels.frost<-cut(frost, breaks=3, labels = NULL, include.lowest=T)
```

- Los nombres de los niveles no son demasiado informativos, especifica nuevos nombres para los niveles.

```
levels(levels.frost)<-list("Bajo"="[-0.188,62.7]", "Medio"="(62.7,125]", "Alto"="(125,188]")
levels.frost
```

```
## [1] Bajo Alto Bajo Medio Bajo Alto Alto Medio Bajo Bajo Bajo
## [12] Alto Alto Medio Alto Medio Medio Bajo Alto Medio Medio Medio
## [23] Alto Bajo Medio Alto Alto Alto Medio Medio Medio Medio Medio
## [34] Alto Medio Medio Bajo Alto Alto Medio Alto Medio Bajo Alto
## [45] Alto Medio Bajo Medio Alto Alto
## Levels: Bajo Medio Alto
```

- Después de este paso has creado un factor que clasifica los estados en bajo, medio y alto según el número de heladas. Ahora cuenta el número de estados que hay en cada uno de los niveles. PISTA: utiliza la función table()

```
table(levels.frost)
```

```
## levels.frost
## Bajo Medio Alto
##    11    19    20
```