

# multivariada

sergio

2022-03-26

```
#analisis de componentes principales
```

```
##(INTRODUCCION)
```

el analisis de componentes principales (*ACP*) es un metodo de reduccion de la dimensionalidad de las variables principales.

```
##Matriz de trabajo
```

##1.- se selecciona la matriz *mtautos*, checo que tipos de variables tengo y despues elimine variables que no necesito.

```
x<-datos::mtautos  
str(x)
```

```
## 'data.frame': 32 obs. of 11 variables:  
## $ millas : num 21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...  
## $ cilindros : num 6 6 4 6 8 6 8 4 4 6 ...  
## $ cilindrada : num 160 160 108 258 360 ...  
## $ caballos : num 110 110 93 110 175 105 245 62 95 123 ...  
## $ eje : num 3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...  
## $ peso : num 2.62 2.88 2.32 3.21 3.44 ...  
## $ velocidad : num 16.5 17 18.6 19.4 17 ...  
## $ forma : num 0 0 1 1 0 1 0 1 1 1 ...  
## $ transmision : num 1 1 1 0 0 0 0 0 0 0 ...  
## $ cambios : num 4 4 4 3 3 3 3 4 4 4 ...  
## $ carburadores: num 4 4 1 1 2 1 4 2 2 4 ...
```

```
x<-x[,1:7]  
x<-x[,-2]
```

```
##2.- Exploracion de la matriz
```

```
colnames(x)
```

```
## [1] "millas"      "cilindrada"   "caballos"     "eje"          "peso"  
## [6] "velocidad"
```

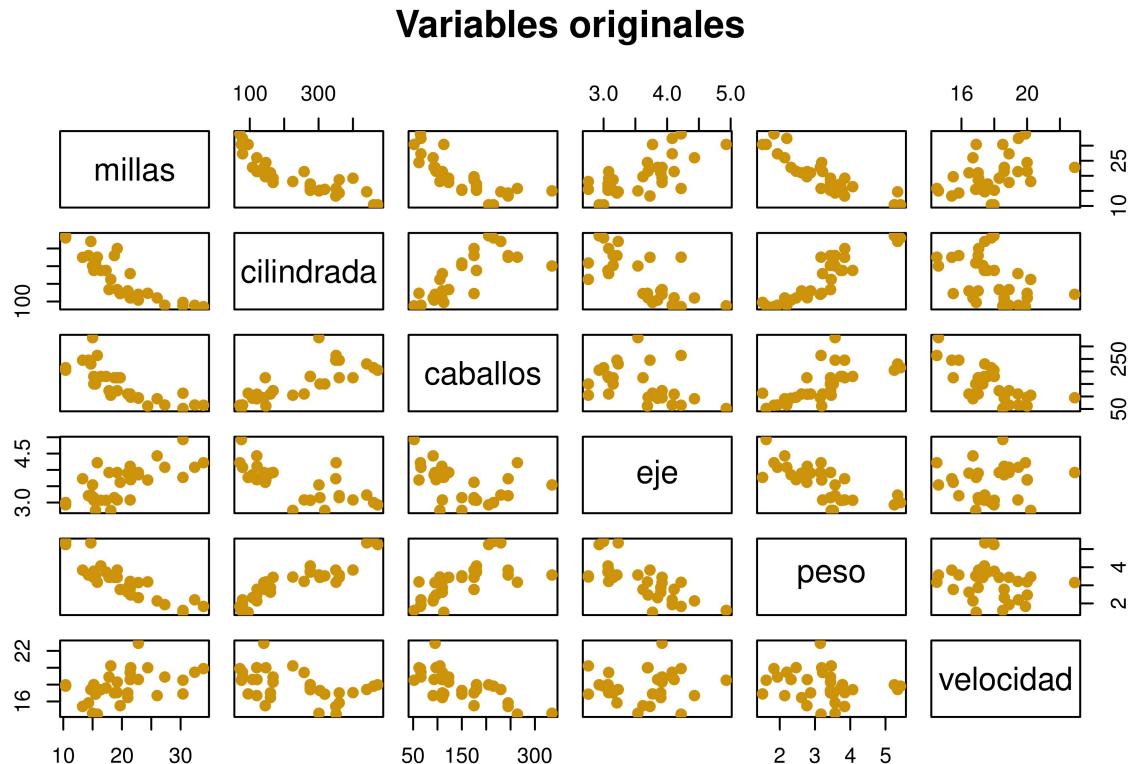
con esto podemos checar cuales son las variables que usaremos.

### 3.- Se definen n (numero de estados) y p (variables)

```
dim(x)  
  
## [1] 32 6  
  
n<-dim(x)[1]  
p<-dim(x)[2]
```

### 4.- Generación de un scatterplot de las variables originales

```
pairs(x,col="darkgoldenrod3", pch=19,  
      main="Variables originales")
```



#5.- Obtención de los componentes principales con base en la matriz de covarianza muestral

```
mu<-colMeans(x);mu
```

```
##      millas cilindrada    caballos        eje        peso  velocidad
##  20.090625 230.721875 146.687500  3.596563  3.217250 17.848750
```

```
s<-cov(x);s
```

```
##          millas cilindrada caballos      eje      peso
## millas      36.324103 -633.09721 -320.73206  2.19506351 -5.1166847
## cilindrada -633.097208 15360.79983 6721.15867 -47.06401915 107.6842040
## caballos    -320.732056 6721.15867 4700.86694 -16.45110887 44.1926613
## eje         2.195064   -47.06402 -16.45111  0.28588135 -0.3727207
## peso        -5.116685   107.68420  44.19266 -0.37272073  0.9573790
## velocidad  4.509149   -96.05168 -86.77008  0.08714073 -0.3054816
##          velocidad
## millas      4.50914919
## cilindrada -96.05168145
## caballos    -86.77008065
## eje         0.08714073
## peso        -0.30548161
## velocidad  3.19316613
```

#6.- Obtención de los componentes principales con base a la matriz de covarianza muestral

```
es<-eigen(s);es
```

```
## eigen() decomposition
## $values
## [1] 1.863762e+04 1.453896e+03 9.252216e+00 1.459916e+00 1.157971e-01
## [6] 8.679172e-02
##
## $vectors
##           [,1]          [,2]          [,3]          [,4]          [,5]
## [1,]  0.038121041  0.009182789  0.99197917  0.09118542 -0.0118855167
## [2,] -0.899662536  0.435416473  0.03128880 -0.00358661  0.0011277610
## [3,] -0.434814427 -0.899812970  0.02253072  0.02758290 -0.0020369196
## [4,]  0.002660333 -0.003899948  0.04006140 -0.05508924  0.9503409728
## [5,] -0.006240020  0.004865430 -0.08468926  0.17473521  0.3109736501
## [6,]  0.006671364  0.025018172 -0.07558782  0.97843974 -0.0008589936
##           [,6]
## [1,] -0.0773546682
## [2,]  0.0051664821
## [3,]  0.0006593723
## [4,]  0.3036280760
## [5,] -0.9303376342
## [6,]  0.1904494838
```

## 7.- Matriz de auto-valores(Lamdas)

```
eigen.val<-es$values;eigen.val
```

```
## [1] 1.863762e+04 1.453896e+03 9.252216e+00 1.459916e+00 1.157971e-01
## [6] 8.679172e-02
```

## 8.- Matriz de auto-vectores(a1)

```
eigen.vec<-es$vectors;eigen.vec

##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,]  0.038121041  0.009182789  0.99197917  0.09118542 -0.0118855167
## [2,] -0.899662536  0.435416473  0.03128880 -0.00358661  0.0011277610
## [3,] -0.434814427 -0.899812970  0.02253072  0.02758290 -0.0020369196
## [4,]  0.002660333 -0.003899948  0.04006140 -0.05508924  0.9503409728
## [5,] -0.006240020  0.004865430 -0.08468926  0.17473521  0.3109736501
## [6,]  0.006671364  0.025018172 -0.07558782  0.97843974 -0.0008589936
##           [,6]
## [1,] -0.0773546682
## [2,]  0.0051664821
## [3,]  0.0006593723
## [4,]  0.3036280760
## [5,] -0.9303376342
## [6,]  0.1904494838
```

Proporción de variabilidad para cada vector

```
pro.var<-eigen.val/sum(eigen.val);pro.var
```

```
## [1] 9.271326e-01 7.232441e-02 4.602537e-04 7.262388e-05 5.760355e-06
## [6] 4.317475e-06
```

Proporción de variabilidad acumulada

```
pro.var.acum<-cumsum(eigen.val)/sum(eigen.val);pro.var.acum
```

```
## [1] 0.9271326 0.9994570 0.9999173 0.9999899 0.9999957 1.0000000
```

## 9.- Obtencion de los componentes principales con base en la matriz de correlaciones muestrales

```
R<-cor(x);R

##           millas cilindrada    caballos       eje       peso     velocidad
## millas      1.0000000 -0.8475514 -0.7761684  0.68117191 -0.8676594  0.41868403
## cilindrada -0.8475514  1.0000000  0.7909486 -0.71021393  0.8879799 -0.43369788
## caballos    -0.7761684  0.7909486  1.0000000 -0.44875912  0.6587479 -0.70822339
## eje         0.6811719 -0.7102139 -0.4487591  1.00000000 -0.7124406  0.09120476
## peso        -0.8676594  0.8879799  0.6587479 -0.71244065  1.0000000 -0.17471588
## velocidad   0.4186840 -0.4336979 -0.7082234  0.09120476 -0.1747159  1.00000000
```

```

eR<-eigen(R);eR

## eigen() decomposition
## $values
## [1] 4.18739648 1.14811212 0.33335666 0.15436054 0.12479601 0.05197818
##
## $vectors
## [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] 0.4586835 -0.05867609 -0.19479235 0.78205878 0.1111533 -0.35249327
## [2,] -0.4660354  0.06065296  0.09688406 0.60001871 -0.2946297  0.56825752
## [3,] -0.4258534 -0.36147576  0.14613554 0.12301873 0.8057408 -0.04771555
## [4,]  0.3670963 -0.43652537  0.80049152 0.02259258 -0.1437714  0.11277675
## [5,] -0.4386179  0.29953457  0.41776208 0.10438337 -0.2301541 -0.69246040
## [6,]  0.2528320  0.76284877  0.34059066 0.04268124 0.4218755  0.24152663

```

##Obtención de auto-valores

```
eigen.val<-eR$values
```

##Obtención de auto-vectores

```
eigen.vec<-eR$vectors
```

##Proporcion de variancia

```
pro.var<-eigen.val/sum(eigen.val)
```

##Proporcion de variancia acumulada

```
pro.var.acum<-cumsum(eigen.val)/sum(eigen.val)
```

##Media de los auto-valores

```
mean(eigen.val)
```

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

## 9.1.- Obtencion de los coeficientes (nuevas variables)

### 9.1.1.- Centrar los datos con respecto a la media

```
ones<-matrix(rep(1,n), nrow=n, ncol=1)
```

### 9.1.2.- Construccion de la matriz centrada

```
X.cen<-as.matrix(x)-ones%*%mu
X.cen
```

	millas	cilindrada	caballos	eje	peso
##					
## Mazda RX4	0.909375	-70.721875	-36.6875	0.3034375	-0.59725
## Mazda RX4 Wag	0.909375	-70.721875	-36.6875	0.3034375	-0.34225
## Datsun 710	2.709375	-122.721875	-53.6875	0.2534375	-0.89725
## Hornet 4 Drive	1.309375	27.278125	-36.6875	-0.5165625	-0.00225
## Hornet Sportabout	-1.390625	129.278125	28.3125	-0.4465625	0.22275
## Valiant	-1.990625	-5.721875	-41.6875	-0.8365625	0.24275
## Duster 360	-5.790625	129.278125	98.3125	-0.3865625	0.35275
## Merc 240D	4.309375	-84.021875	-84.6875	0.0934375	-0.02725
## Merc 230	2.709375	-89.921875	-51.6875	0.3234375	-0.06725
## Merc 280	-0.890625	-63.121875	-23.6875	0.3234375	0.22275
## Merc 280C	-2.290625	-63.121875	-23.6875	0.3234375	0.22275
## Merc 450SE	-3.690625	45.078125	33.3125	-0.5265625	0.85275
## Merc 450SL	-2.790625	45.078125	33.3125	-0.5265625	0.51275
## Merc 450SLC	-4.890625	45.078125	33.3125	-0.5265625	0.56275
## Cadillac Fleetwood	-9.690625	241.278125	58.3125	-0.6665625	2.03275
## Lincoln Continental	-9.690625	229.278125	68.3125	-0.5965625	2.20675
## Chrysler Imperial	-5.390625	209.278125	83.3125	-0.3665625	2.12775
## Fiat 128	12.309375	-152.021875	-80.6875	0.4834375	-1.01725
## Honda Civic	10.309375	-155.021875	-94.6875	1.3334375	-1.60225
## Toyota Corolla	13.809375	-159.621875	-81.6875	0.6234375	-1.38225
## Toyota Corona	1.409375	-110.621875	-49.6875	0.1034375	-0.75225
## Dodge Challenger	-4.590625	87.278125	3.3125	-0.8365625	0.30275
## AMC Javelin	-4.890625	73.278125	3.3125	-0.4465625	0.21775
## Camaro Z28	-6.790625	119.278125	98.3125	0.1334375	0.62275
## Pontiac Firebird	-0.890625	169.278125	28.3125	-0.5165625	0.62775
## Fiat X1-9	7.209375	-151.721875	-80.6875	0.4834375	-1.28225
## Porsche 914-2	5.909375	-110.421875	-55.6875	0.8334375	-1.07725
## Lotus Europa	10.309375	-135.621875	-33.6875	0.1734375	-1.70425
## Ford Pantera L	-4.290625	120.278125	117.3125	0.6234375	-0.04725
## Ferrari Dino	-0.390625	-85.721875	28.3125	0.0234375	-0.44725
## Maserati Bora	-5.090625	70.278125	188.3125	-0.0565625	0.35275
## Volvo 142E	1.309375	-109.721875	-37.6875	0.5134375	-0.43725
##	velocidad				
## Mazda RX4		-1.38875			
## Mazda RX4 Wag		-0.82875			
## Datsun 710		0.76125			
## Hornet 4 Drive		1.59125			
## Hornet Sportabout		-0.82875			
## Valiant		2.37125			
## Duster 360		-2.00875			
## Merc 240D		2.15125			
## Merc 230		5.05125			
## Merc 280		0.45125			
## Merc 280C		1.05125			
## Merc 450SE		-0.44875			
## Merc 450SL		-0.24875			
## Merc 450SLC		0.15125			
## Cadillac Fleetwood		0.13125			
## Lincoln Continental		-0.02875			

```

## Chrysler Imperial      -0.42875
## Fiat 128              1.62125
## Honda Civic            0.67125
## Toyota Corolla         2.05125
## Toyota Corona          2.16125
## Dodge Challenger       -0.97875
## AMC Javelin            -0.54875
## Camaro Z28             -2.43875
## Pontiac Firebird       -0.79875
## Fiat X1-9              1.05125
## Porsche 914-2          -1.14875
## Lotus Europa            -0.94875
## Ford Pantera L         -3.34875
## Ferrari Dino           -2.34875
## Maserati Bora           -3.24875
## Volvo 142E              0.75125

```

### 9.1.3.- Construccion de la matriz diagonal de las varianzas

```

Dx<-diag(diag(s))
Dx

##          [,1]   [,2]   [,3]   [,4]   [,5]   [,6]
## [1,] 36.3241    0.0  0.000 0.000000 0.000000 0.000000
## [2,]  0.0000 15360.8    0.000 0.000000 0.000000 0.000000
## [3,]  0.0000    0.0 4700.867 0.000000 0.000000 0.000000
## [4,]  0.0000    0.0  0.000 0.2858814 0.000000 0.000000
## [5,]  0.0000    0.0  0.000 0.0000000 0.957379 0.000000
## [6,]  0.0000    0.0  0.000 0.0000000 0.000000 3.193166

```

### 9.1.4.- Construccion de la matriz centrada multiplicada

```

Dx^1/2

##          [,1]   [,2]   [,3]   [,4]   [,5]   [,6]
## [1,] 18.16205    0.0  0.000 0.000000 0.000000 0.000000
## [2,]  0.00000 7680.4    0.000 0.000000 0.000000 0.000000
## [3,]  0.00000    0.0 2350.433 0.000000 0.000000 0.000000
## [4,]  0.00000    0.0  0.000 0.1429407 0.000000 0.000000
## [5,]  0.00000    0.0  0.000 0.0000000 0.4786895 0.000000
## [6,]  0.00000    0.0  0.000 0.0000000 0.0000000 1.596583

```

```

Y<-X.cen%*%solve(Dx)^(1/2)
Y #datos normalizados

```

```

##          [,1]   [,2]   [,3]   [,4]
## Mazda RX4      0.15088482 -0.57061982 -0.53509284 0.56751369
## Mazda RX4 Wag  0.15088482 -0.57061982 -0.53509284 0.56751369
## Datsun 710     0.44954345 -0.99018209 -0.78304046 0.47399959

```

## Hornet 4 Drive	0.21725341	0.22009369	-0.53509284	-0.96611753
## Hornet Sportabout	-0.23073453	1.04308123	0.41294217	-0.83519779
## Valiant	-0.33028740	-0.04616698	-0.60801861	-1.56460776
## Duster 360	-0.96078893	1.04308123	1.43390296	-0.72298087
## Merc 240D	0.71501778	-0.67793094	-1.23518023	0.17475447
## Merc 230	0.44954345	-0.72553512	-0.75387015	0.60491932
## Merc 280	-0.14777380	-0.50929918	-0.34548584	0.60491932
## Merc 280C	-0.38006384	-0.50929918	-0.34548584	0.60491932
## Merc 450SE	-0.61235388	0.36371309	0.48586794	-0.98482035
## Merc 450SL	-0.46302456	0.36371309	0.48586794	-0.98482035
## Merc 450SLC	-0.81145962	0.36371309	0.48586794	-0.98482035
## Cadillac Fleetwood	-1.60788262	1.94675381	0.85049680	-1.24665983
## Lincoln Continental	-1.60788262	1.84993175	0.99634834	-1.11574009
## Chrysler Imperial	-0.89442035	1.68856165	1.21512565	-0.68557523
## Fiat 128	2.04238943	-1.22658929	-1.17683962	0.90416444
## Honda Civic	1.71054652	-1.25079481	-1.38103178	2.49390411
## Toyota Corolla	2.29127162	-1.28790993	-1.19142477	1.16600392
## Toyota Corona	0.23384555	-0.89255318	-0.72469984	0.19345729
## Dodge Challenger	-0.76168319	0.70420401	0.04831332	-1.56460776
## AMC Javelin	-0.81145962	0.59124494	0.04831332	-0.83519779
## Camaro Z28	-1.12671039	0.96239618	1.43390296	0.24956575
## Pontiac Firebird	-0.14777380	1.36582144	0.41294217	-0.96611753
## Fiat X1-9	1.19619000	-1.22416874	-1.17683962	0.90416444
## Porsche 914-2	0.98049211	-0.89093948	-0.81221077	1.55876313
## Lotus Europa	1.71054652	-1.09426581	-0.49133738	0.32437703
## Ford Pantera L	-0.71190675	0.97046468	1.71102089	1.16600392
## Ferrari Dino	-0.06481307	-0.69164740	0.41294217	0.04383473
## Maserati Bora	-0.84464392	0.56703942	2.74656682	-0.10578782
## Volvo 142E	0.21725341	-0.88529152	-0.54967799	0.96027290
##	[,5]	[,6]		
## Mazda RX4	-0.610399567	-0.77716515		
## Mazda RX4 Wag	-0.349785269	-0.46378082		
## Datsun 710	-0.917004624	0.42600682		
## Hornet 4 Drive	-0.002299538	0.89048716		
## Hornet Sportabout	0.227654255	-0.46378082		
## Valiant	0.248094592	1.32698675		
## Duster 360	0.360516446	-1.12412636		
## Merc 240D	-0.027849959	1.20387148		
## Merc 230	-0.068730634	2.82675459		
## Merc 280	0.227654255	0.25252621		
## Merc 280C	0.227654255	0.58829513		
## Merc 450SE	0.871524874	-0.25112717		
## Merc 450SL	0.524039143	-0.13920420		
## Merc 450SLC	0.575139986	0.08464175		
## Cadillac Fleetwood	2.077504765	0.07344945		
## Lincoln Continental	2.255335698	-0.01608893		
## Chrysler Imperial	2.174596366	-0.23993487		
## Fiat 128	-1.039646647	0.90727560		
## Honda Civic	-1.637526508	0.37564148		
## Toyota Corolla	-1.412682800	1.14790999		
## Toyota Corona	-0.768812180	1.20946763		
## Dodge Challenger	0.309415603	-0.54772305		
## AMC Javelin	0.222544170	-0.30708866		
## Camaro Z28	0.636460997	-1.36476075		

```

## Pontiac Firebird      0.641571082 -0.44699237
## Fiat X1-9            -1.310481114  0.58829513
## Porsche 914-2         -1.100967659 -0.64285758
## Lotus Europa          -1.741772228 -0.53093460
## Ford Pantera L        -0.048290296 -1.87401028
## Ferrari Dino          -0.457097039 -1.31439542
## Maserati Bora          0.360516446 -1.81804880
## Volvo 142E             -0.446876870  0.42041067

```

### 9.1.5.- Construccion de los coeficientes o scores eigen.vec matriz de autovectores

```
scores<-Y%*%eigen.vec
```

```
###Nombramos las columnas PC1...PC8
```

```
colnames(scores)<-c("PC1", "PC2", "PC3", "PC4", "PC5",
"PC6")
```

```
scores
```

	PC1	PC2	PC3	PC4
## Mazda RX4	0.842580639	-0.873469391	-0.228278346	-0.37427255
## Mazda RX4 Wag	0.807504121	-0.556341552	-0.012667800	-0.33369312
## Datsun 710	1.685044761	0.040006569	-0.156493687	-0.40571569
## Hornet 4 Drive	0.096444303	1.294377904	-0.570229748	0.25207881
## Hornet Sportabout	-1.291509555	0.006516693	-0.525074062	0.48131922
## Valiant	-0.218730892	2.005957905	-0.725839941	-0.31361702
## Duster 360	-2.245190880	-0.832631648	-0.313240442	0.02419056
## Merc 240D	1.550656769	1.197159848	0.152817779	0.05288819
## Merc 230	1.832265452	2.073858925	1.150258327	-0.04936365
## Merc 280	0.502753448	0.099432035	0.594300648	-0.41544992
## Merc 280C	0.481098961	0.369202815	0.753908731	-0.58278335
## Merc 450SE	-1.464571595	0.381740160	-0.284259438	-0.14288656
## Merc 450SL	-1.215365529	0.354274415	-0.420394087	-0.05759698
## Merc 450SLC	-1.341005273	0.560786296	-0.254933769	-0.31520557
## Cadillac Fleetwood	-3.357256928	1.127500626	0.521078742	0.20708364
## Lincoln Continental	-3.426823644	0.996718766	0.681607545	0.18462986
## Chrysler Imperial	-2.980802933	0.483262622	0.793341147	0.66442458
## Fiat 128	3.026916817	0.217176556	-0.090196261	0.66714798
## Honda Civic	3.684356739	-0.969613645	0.783991314	0.31880089
## Toyota Corolla	3.496444405	0.161656879	-0.011031261	0.80044813
## Toyota Corona	1.545862677	0.802010352	0.007683829	-0.46607822
## Dodge Challenger	-1.546688715	0.427782349	-1.086085997	-0.19363018
## AMC Javelin	-1.150170291	0.262992430	-0.457781132	-0.28265394
## Camaro Z28	-2.108550264	-1.353243809	0.523099358	-0.11347653
## Pontiac Firebird	-1.629232017	0.215163261	-0.436129194	0.78081432
## Fiat X1-9	2.675795416	-0.057483105	-0.146914711	-0.03506244
## Porsche 914-2	2.103412908	-1.318595692	0.172879784	0.02516077
## Lotus Europa	2.252616575	-1.057474693	-1.159836838	0.42357977
## Ford Pantera L	-1.532047829	-2.470900470	0.757669487	0.17734890
## Ferrari Dino	0.001011947	-1.346152342	-0.597578273	-0.51771241
## Maserati Bora	-2.477943625	-2.141594405	0.067555673	-0.02480245

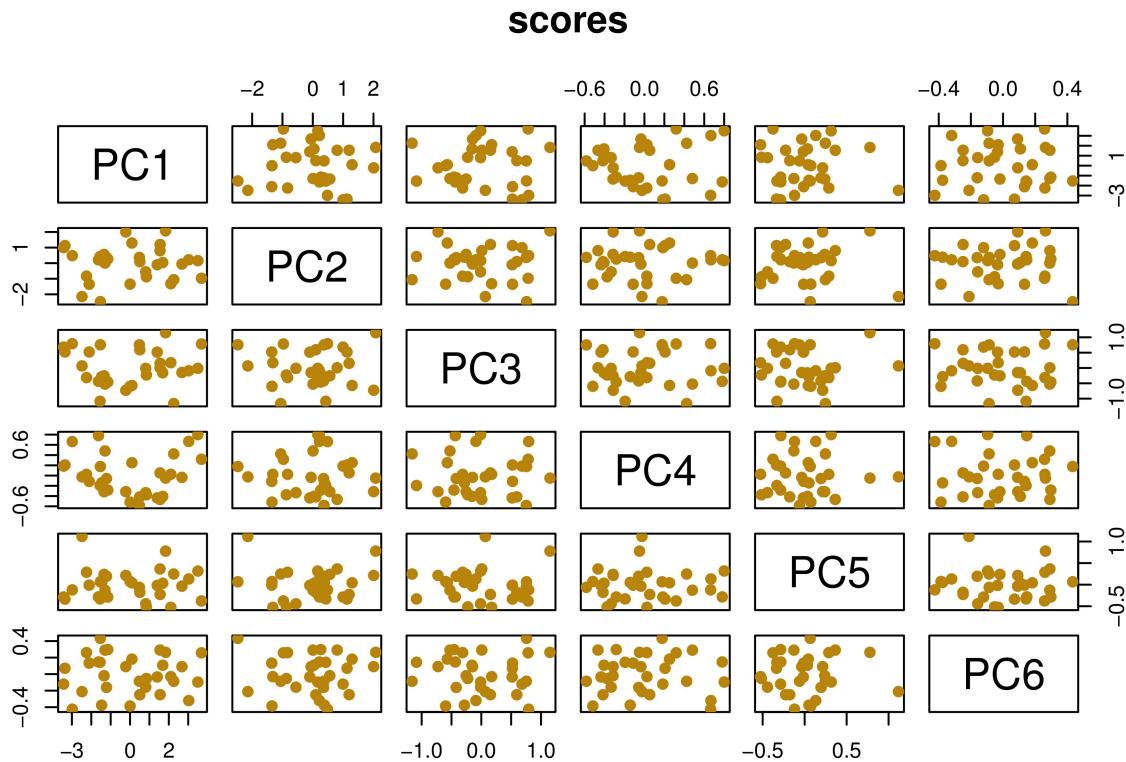
```

## Volvo 142E      1.401124031 -0.100076652  0.516772623 -0.43591506
##                   PC5          PC6
## Mazda RX4      -0.51522641 -0.05293884
## Mazda RX4 Wag   -0.44299870 -0.15771326
## Datsun 710      0.03340433  0.10756126
## Hornet 4 Drive  0.04326023  0.18173489
## Hornet Sportabout -0.12822104  0.29051949
## Valiant         0.21465335  0.09145688
## Duster 360      0.28796476  0.26030567
## Merc 240D       -0.22685092 -0.24858075
## Merc 230         0.77769555  0.26377009
## Merc 280         -0.17757395 -0.24926769
## Merc 280C        -0.06174108 -0.08628988
## Merc 450SE       0.05131736 -0.37586534
## Merc 450SL       0.19510864 -0.16085043
## Merc 450SLC      0.23905298 -0.01934996
## Cadillac Fleetwood -0.33493876  0.07099935
## Lincoln Continental -0.28641866 -0.12098198
## Chrysler Imperial -0.12099080 -0.42425486
## Fiat 128          0.13222362 -0.31977950
## Honda Civic        -0.37729414  0.25807107
## Toyota Corolla    0.31593125 -0.09569699
## Toyota Corona     0.36442162  0.29125853
## Dodge Challenger   -0.33055333  0.14335299
## AMC Javelin        -0.28616179  0.29724414
## Camaro Z28         -0.01155770  0.13342234
## Pontiac Firebird   -0.28344806  0.14734508
## Fiat X1-9          -0.03478377  0.09037547
## Porsche 914-2      -0.52486836 -0.03024333
## Lotus Europa        0.24689718 -0.08688085
## Ford Pantera L     0.06645790  0.43308775
## Ferrari Dino       0.07368942 -0.38588778
## Maserati Bora      1.11731392 -0.21177920
## Volvo 142E         -0.01576463 -0.03414438

```

###Generacion del grafico de los scores

```
pairs(scores, main="scores", col="darkgoldenrod", pch=19)
```



```
#10.- PCA sintetizado
```

```
head(x)
```

```
##          millas cilindrada caballos eje peso velocidad
## Mazda RX4      21.0       160     110 3.90 2.620      16.46
## Mazda RX4 Wag   21.0       160     110 3.90 2.875      17.02
## Datsun 710     22.8       108      93 3.85 2.320      18.61
## Hornet 4 Drive  21.4       258     110 3.08 3.215      19.44
## Hornet Sportabout 18.7       360     175 3.15 3.440      17.02
## Valiant        18.1       225     105 2.76 3.460      20.22
```

```
##Aplicar el cálculo de la varianza a las columnas 1=filas, 2=columnas
```

```
apply(x, 2, var)
```

```
##          millas    cilindrada      caballos         eje       peso velocidad
## 3.632410e+01 1.536080e+04 4.700867e+03 2.858814e-01 9.573790e-01 3.193166e+00
```

```
##centrado por la media y escalada por la desviacion standar (dividir entre sd).
```

```
acp<-prcomp(x, center=TRUE, scale=TRUE);acp
```

```
## Standard deviations (1, ..., p=6):
```

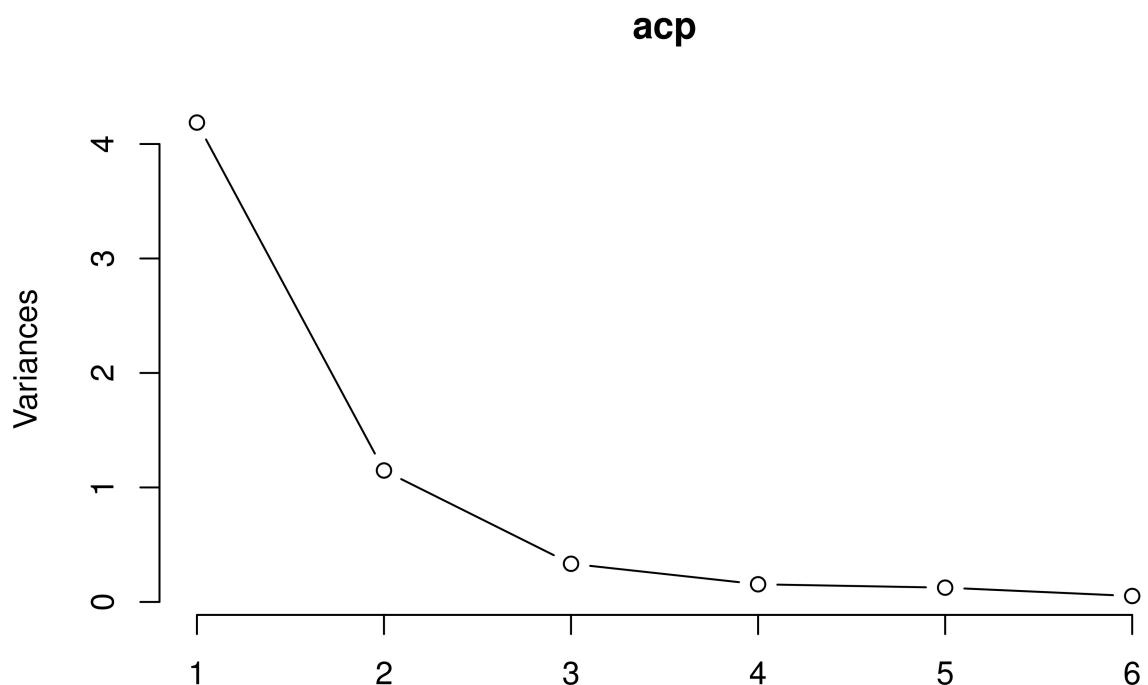
```

## [1] 2.0463129 1.0714999 0.5773705 0.3928874 0.3532648 0.2279872
##
## Rotation (n x k) = (6 x 6):
##          PC1        PC2        PC3        PC4        PC5        PC6
## millas    -0.4586835  0.05867609 -0.19479235  0.78205878 -0.1111533 -0.35249327
## cilindrada 0.4660354 -0.06065296  0.09688406  0.60001871  0.2946297  0.56825752
## caballos   0.4258534  0.36147576  0.14613554  0.12301873 -0.8057408 -0.04771555
## eje        -0.3670963  0.43652537  0.80049152  0.02259258  0.1437714  0.11277675
## peso       0.4386179 -0.29953457  0.41776208  0.10438337  0.2301541 -0.69246040
## velocidad -0.2528320 -0.76284877  0.34059066  0.04268124 -0.4218755  0.24152663

##Generación del gráfico screeplot

plot(acp, type="l")

```



```

##Visualizar el resumen

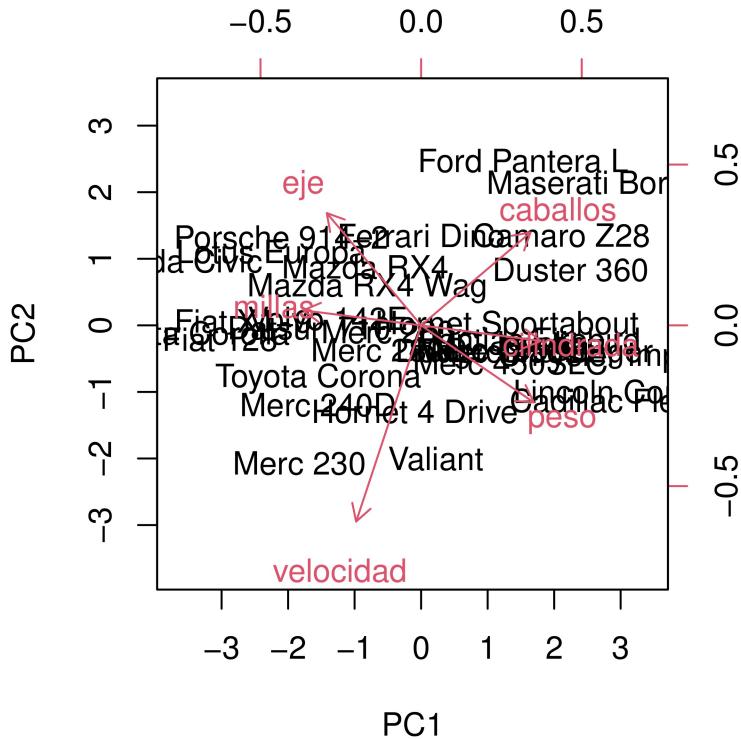
summary(acp)

## Importance of components:
##          PC1        PC2        PC3        PC4        PC5        PC6
## Standard deviation 2.0463 1.0715 0.57737 0.39289 0.3533 0.22799
## Proportion of Variance 0.6979 0.1913 0.05556 0.02573 0.0208 0.00866
## Cumulative Proportion 0.6979 0.8892 0.94481 0.97054 0.9913 1.00000

```

```
##Construcción del Biplot
```

```
biplot(acp, scale=0)
```



```
#Componente principal calculada Suma del producto de la matriz acp de cada uno de los componentes por el dato de la matriz original por filas filas = 1 columnas = 2
```

```
pc1<-apply(acp$rotation[,1]*x, 1, sum)
pc2<-apply(acp$rotation[,2]*x, 1, sum)
pc3<-apply(acp$rotation[,3]*x, 1, sum)
x$pc1<-pc1;x$pc1
```

```
## [1] 113.298680 -80.301247 8.543655 -14.706407 -75.347764 55.228566
## [7] 261.205065 -62.665353 20.439066 14.119147 -8.497850 53.041870
## [13] 196.366315 -144.179048 112.729777 -14.090812 -91.043340 -2.896770
## [19] 47.650902 -30.474336 10.820153 -10.226398 -61.721452 65.000679
## [25] 245.995864 -36.552184 17.262882 23.567456 -35.070637 -6.961840
## [31] 274.538997 -65.811978
```

```
x$pc2<-pc2;x$pc2
```

```
## [1] 22.365082 -27.308240 -16.862486 -196.432321 84.190984 24.891906
## [7] 54.318565 1.173447 -26.541458 -128.955451 54.823478 54.342498
## [13] 43.050522 -30.108424 -125.050805 -363.319910 110.196781 4.574276
## [19] 8.897347 -35.239974 -20.737576 -247.421344 73.017519 79.222813
```

```
## [25] 89.761869 -31.338970 -19.355890 -66.663856 115.455628 56.405762
## [31] 6.465243 -44.853650
```

```
x$pc3<-pc3;x$pc3
```

```
## [1] 71.744424 176.054295 28.241250 121.104095 -33.606285 129.476401
## [7] 158.688744 150.635581 41.075574 90.451708 -3.506859 191.763267
## [13] 119.431133 293.008441 157.890140 191.802201 -43.102472 88.680678
## [19] 33.877037 90.993216 32.092971 140.290728 -27.771578 248.537710
## [25] 134.907340 96.744150 34.618557 72.843520 -19.927264 174.752765
## [31] 186.946673 145.015721
```