Q1 :

a):

3 columns with all 0 data, which are mean less in datamining.

Drop it

b):

**Code :**

D1 <- read\_table("~/Desktop/Assigments/2023\_DM/file/Domotic1.txt")

summary(D1)

boxplot(D1[,3:24])

pca\_result = PCA(D1[,3:18,22,23,24], scale.unit=T, graph=T, ncp=18)

summary(pca\_result)

fviz\_screeplot(pca\_result, ncp=18)

communality(pca\_result)

A picture containing text

Description automatically generated

1. **Eigenvalue Criterion:** select dim 1~5, because dim 6 eigenvalue is below 1.
2. **Proportion of Variance Explained:** select dim 1~5, because dim 6 cumulative% is beyond 85%.
3. **Screeplot Criterion** : select dim 1~5, because after dim 5 it has significant leveling. Chart, histogram

   Description automatically generated
4. **Minimum Communality Criterion :** select dim 1~5, because before dim 6 every variable has a communality of at least 50%.

Table

Description automatically generated

c) Based on final PCA model, I will divide into 5 group :"Light&ME", "Temperature", "CO2", "Humedad", "Precipitacion ".

d)

The arrow direction represents the correlation between the original variable and the principal component. The arrow length represents the contribution rate of the original data to the principal component.

Diagram, engineering drawing

Description automatically generated

e)

**Dim1 (Light&ME)** = -0.072x1 + -0.031x2 + 0.325x3 +0.144x4 + 0.131x5 + 0.005 x6 + -0.065 x7 + 0.970x8 + 0.971x9 + -0.079x10 +0.555x11 + 0.450x12 + 0.307x13 + 0.459 x14 + 0.954x15 + 0.922x16 + 0.262\*x17 + -0.134 x18 + 0.040x19

**Dim2 (Temperature)** = -0.959x1 - 0.961x2 - 0.775x3 - 0.035x4 - 0.019x5 + 0.091x6 + 0.231x7 - 0.084x8 - 0.042x9 + 0.179x10 - 0.104x11 - 0.236x12 - 0.298x13 + 0.223x14 - 0.019x15 - 0.144x16 - 0.889x17 + 0.414x19

**Dim3 (Humedad)** = -0.202x1 - 0.203x2 - 0.113x3 - 0.002x4 - 0.006x5 + 0.976x6 + 0.958x7 - 0.039x8 - 0.017x9 - 0.007x10 - 0.025x11 - 0.106x12 - 0.054x13 - 0.055x14 - 0.061x15 - 0.030x16 - 0.090x17 + 0.729x19

**Dim4 (CO2)** = -0.019x1 - 0.033x2 - 0.020x3 - 0.967x4 - 0.971x5 + 0.006x6 + 0.008x7 - 0.114x8 - 0.076x9 + 0.043x10 - 0.029x11 + 0.049x12 - 0.061x13 + 0.034x14 - 0.121x15 - 0.082x16 - 0.015x17 - 0.014x18

**Dim5 (Precipitacion)** = -0.107x1 - 0.098x2 + 0.049x3 - 0.023x4 - 0.026x5 - 0.084x6 + 0.003x7 - 0.021x8 - 0.026x9 + 0.968x10 + 0.047x11 + 0.243x12 - 0.036x13 - 0.070x14 - 0.047x15 - 0.041x16 - 0.065x17 + 0.205x18

Code :

> round(pca\_result$var$coord[,1:5],digits = 3) # show 5 dimensions

Dim.1 Dim.2 Dim.3 Dim.4 Dim.5

Temperature\_Comedor\_Sensor -0.072 -0.959 -0.202 -0.019 -0.107

Temperature\_Habitacion\_Sensor -0.031 -0.961 -0.203 -0.033 -0.098

Weather\_Temperature 0.325 -0.775 -0.113 -0.020 0.049

CO2\_Comedor\_Sensor 0.144 -0.035 -0.002 -0.967 -0.023

CO2\_Habitacion\_Senso 0.131 -0.019 -0.006 -0.971 -0.026

Humedad\_Comedor\_Sensor 0.005 0.091 0.976 0.006 -0.084

Humedad\_Habitacion\_Sensor -0.065 0.231 0.958 0.008 0.003

Lighting\_Comedor\_Sensor 0.970 -0.084 -0.039 -0.114 -0.021

Lighting\_Habitacion\_Sensor 0.971 -0.042 -0.017 -0.076 -0.026

Precipitacion -0.079 0.179 -0.007 0.043 0.968

Meteo\_Exterior\_Crepusculo 0.555 -0.104 -0.025 -0.029 0.047

Meteo\_Exterior\_Viento 0.450 -0.236 -0.106 0.049 0.243

Meteo\_Exterior\_Sol\_Oest 0.307 -0.298 -0.054 -0.061 -0.036

Meteo\_Exterior\_Sol\_Est 0.459 0.223 -0.055 0.034 -0.070

Meteo\_Exterior\_Sol\_Sud 0.954 -0.019 -0.061 -0.121 -0.047

Meteo\_Exterior\_Piranometro 0.922 -0.144 -0.030 -0.082 -0.041

Temperature\_Exterior\_Sensor 0.262 -0.889 -0.090 -0.015 -0.065

Humedad\_Exterior\_Sensor -0.134 0.414 0.729 -0.014 0.205

Day\_Of\_Week 0.040 -0.004 -0.070 0.091 -0.084

Observation:

> D1\_PC = as.matrix(D1[,c(3:18,22,23,24)]) %\*% as.matrix(pca\_result$var$coord[,1:5]) #new dataset

> colnames(D1\_PC) = c("Light&ME", "Temperature","Humedad", "CO2", "Precipitacion ")

> D1\_PC

Light&ME Temperature Humedad CO2 Precipitacion

[1,] 101053.81390 -2.106864e+03 -6988.698812 -12269.2607 -5.686886e+03

[2,] 100634.22459 -2.916242e+03 -6962.696001 -12413.2929 -5.608920e+03

[3,] 100992.35550 -4.051132e+03 -7060.160916 -12622.9944 -5.636333e+03

[4,] 102340.44329 -5.415036e+03 -7303.983480 -12892.4016 -5.799352e+03

[5,] 104533.33044 -7.854630e+03 -7705.848464 -13379.6659 -6.054762e+03

[6,] 107403.77087 -1.100117e+04 -8228.292070 -14012.9364 -6.389966e+03

[7,] 111220.01997 -1.485654e+04 -8910.468172 -14792.7350 -6.842867e+03

[8,] 115086.54667 -1.867805e+04 -9611.801004 -15546.9673 -7.320289e+03

[9,] 114779.19349 -2.131196e+04 -9916.871960 -15690.5020 -7.513945e+03

[10,] 113057.82421 -2.375999e+04 -10109.817253 -15641.5846 -7.623682e+03

[11,] 109856.33795 -2.559144e+04 -10138.212228 -15363.7454 -7.617293e+03

[12,] 104425.29745 -2.601480e+04 -9863.474482 -14704.2500 -7.403608e+03

[13,] 98848.07865 -2.662187e+04 -9613.502890 -14031.0684 -7.209357e+03

[14,] 91083.13060 -2.613108e+04 -9091.621316 -12998.9696 -6.822775e+03

[15,] 82810.56555 -2.542404e+04 -8482.687727 -11928.1240 -6.360335e+03

[16,] 74848.12206 -2.570916e+04 -7972.130506 -11021.6222 -5.940814e+03

[17,] 66359.10544 -2.551018e+04 -7387.769054 -9995.2728 -5.477728e+03

[18,] 57310.79663 -2.451948e+04 -6690.426810 -8818.8477 -4.946677e+03

[19,] 50440.30354 -2.421898e+04 -6220.350432 -7953.2779 -4.583396e+03

[20,] 43479.01864 -2.240070e+04 -5560.144516 -6972.5948 -4.098378e+03

[21,] 38691.24025 -2.119432e+04 -5108.976179 -6311.7340 -3.763291e+03

[22,] 33127.40730 -1.867130e+04 -4438.222132 -5472.8217 -3.274891e+03

[23,] 21110.66829 -1.138067e+04 -2750.566193 -3548.7555 -2.062369e+03

[24,] 4365.28517 -5.712756e+02 -333.955902 -796.0140 -3.433323e+02

[25,] 3299.12160 -4.600063e+02 -219.622320 -703.4170 -2.428547e+02

[26,] 1131.22969 -3.803748e+02 -3.196103 -527.0618 -4.708639e+01

[27,] 405.08092 -1.145310e+02 86.770298 -433.5018 1.882644e+01

[28,] 161.92106 -6.751863e+01 100.579533 -420.8422 -1.299875e+00

[29,] 80.52585 -5.008988e+01 106.629713 -415.9856 -8.100632e+00

[30,] 77.64567 -4.858796e+01 108.280038 -415.5974 -8.034910e+00

[31,] 77.83385 -4.731317e+01 109.048038 -415.7113 -7.883313e+00

[32,] 76.28250 -4.579138e+01 110.223164 -416.3366 -7.899801e+00

[33,] 77.50759 -4.498586e+01 111.225755 -416.0749 -7.543427e+00

[34,] 76.57887 -4.376082e+01 111.923804 -415.6760 -7.424800e+00

[35,] 76.37711 -4.331151e+01 111.713200 -414.2707 -7.427105e+00

[36,] 76.63679 -4.295742e+01 111.751023 -414.5201 -7.580404e+00

[37,] 76.34734 -4.244058e+01 111.505297 -414.0641 -7.489441e+00

[38,] 75.82270 -4.194247e+01 111.292613 -414.0060 -7.430699e+00

[39,] 76.28540 -4.113161e+01 111.577811 -416.2302 -7.462205e+00

[40,] 76.28159 -4.028366e+01 111.678793 -418.2063 -7.457311e+00

[41,] 75.88680 -3.897194e+01 111.536283 -419.4367 -7.431314e+00

[42,] 75.90511 -3.778635e+01 111.972660 -421.4453 -7.457186e+00

[43,] 76.27865 -3.740358e+01 111.890703 -422.9740 -7.400390e+00

[44,] 76.33009 -3.665740e+01 112.011433 -423.3421 -7.358009e+00

[45,] 75.97915 -3.520181e+01 112.186143 -424.4895 -7.435019e+00

[46,] 75.50900 -3.490469e+01 111.907968 -424.1401 -7.303217e+00

[47,] 77.50569 -3.513949e+01 111.647919 -426.6490 -7.322467e+00

[48,] 76.97671 -3.483314e+01 111.389186 -428.2478 -7.315393e+00

[49,] 76.80727 -3.393208e+01 111.020984 -430.2949 -7.353732e+00

[50,] 76.67921 -3.339321e+01 111.019240 -430.9842 -7.383399e+00

[51,] 77.18185 -3.272422e+01 110.626444 -432.4008 -7.483817e+00

[52,] 76.57075 -3.171479e+01 110.918597 -433.8645 -7.421513e+00

[53,] 77.19511 -3.109715e+01 111.567903 -438.3969 -7.386219e+00

[54,] 76.49760 -3.063249e+01 111.839750 -437.7946 -7.288689e+00

[55,] 77.85325 -3.049547e+01 111.506908 -437.7072 -7.354836e+00

[56,] 77.31668 -3.037289e+01 110.991193 -437.4378 -7.424269e+00

[57,] 77.37553 -2.988974e+01 110.852654 -436.2147 -7.310812e+00

[58,] 77.33516 -2.936844e+01 110.754988 -438.2250 -7.338727e+00

[59,] 77.57476 -2.825581e+01 110.978432 -438.9676 -7.386794e+00

[60,] 76.90134 -2.780749e+01 111.003438 -437.0454 -7.281777e+00

[61,] 77.50091 -2.777133e+01 111.033788 -437.1713 -7.248773e+00

[62,] 76.70301 -2.704622e+01 110.814929 -433.3918 -7.243335e+00

[63,] 75.91667 -2.616947e+01 110.552172 -431.0767 -7.174964e+00

[64,] 75.66081 -2.600142e+01 109.979512 -425.3772 -6.975072e+00

[65,] 74.78408 -2.554058e+01 110.026794 -423.2976 -7.053137e+00

[66,] 75.15799 -2.526584e+01 110.054659 -421.8120 -7.020836e+00

[67,] 74.53262 -2.491538e+01 110.055166 -419.4088 -6.855508e+00

[68,] 74.20272 -2.463547e+01 109.948038 -416.6668 -6.632680e+00

[69,] 73.80451 -2.441712e+01 109.941978 -415.8552 -6.391114e+00

[70,] 73.61446 -2.367446e+01 110.092867 -413.7773 -6.499255e+00

[71,] 73.42612 -2.274615e+01 110.406079 -414.2701 -6.704745e+00

[72,] 73.71864 -2.239951e+01 110.792113 -415.0208 -6.606088e+00

[73,] 74.52200 -2.235029e+01 111.310752 -414.3435 -6.310661e+00

[74,] 110.05693 -2.887177e+01 109.720103 -416.6237 -3.339129e+00

[75,] 351.62173 -7.302882e+01 99.104653 -429.5192 1.697590e+01

[76,] 2114.76236 3.106744e+02 -69.323500 -464.1864 -1.604269e+02

[77,] 18555.40171 3.457951e+03 -1767.994477 -805.8042 -1.992197e+03

[78,] 39084.74306 6.957582e+03 -3764.319988 -1514.0892 -4.097257e+03

[79,] 53748.37161 8.754631e+03 -5057.603542 -2373.0007 -5.390076e+03

[80,] 66908.99727 9.867603e+03 -6155.419013 -3351.4428 -6.439485e+03

[81,] 78624.73216 1.072409e+04 -7109.937035 -4287.0030 -7.335824e+03

[82,] 87459.11777 1.094375e+04 -7783.427633 -5160.4592 -7.926145e+03

[83,] 95845.83660 1.099543e+04 -8402.610976 -6053.1365 -8.451850e+03

[84,] 104196.25880 1.079902e+04 -9005.309757 -7020.7273 -8.942164e+03

[85,] 110966.33996 1.024011e+04 -9505.876496 -7881.7302 -9.326724e+03

[86,] 117148.95857 9.802756e+03 -9966.192834 -8647.3679 -9.686855e+03

[87,] 123450.25812 1.045949e+04 -10392.409484 -9223.8506 -1.007714e+04

[88,] 125718.23182 1.027688e+04 -10396.894816 -9754.0587 -1.000707e+04

[89,] 128057.81723 9.844356e+03 -10407.518706 -10335.6098 -9.924998e+03

[90,] 127175.40052 9.054022e+03 -10175.373391 -10662.7879 -9.611528e+03

[91,] 122877.38952 7.462418e+03 -9640.310081 -10887.0667 -8.968271e+03

[92,] 118062.70561 5.446404e+03 -9051.808217 -11212.1614 -8.246459e+03

[93,] 112392.06960 2.923052e+03 -8363.171748 -11713.4243 -7.396734e+03

[94,] 107360.24774 7.142924e+02 -7750.480229 -12079.8239 -6.639641e+03

[95,] 103999.50287 -6.793174e+02 -7339.717345 -12294.0212 -6.137288e+03

[96,] 102385.75292 -1.378108e+03 -7140.257752 -12279.2370 -5.888052e+03

[97,] 101466.18206 -1.968079e+03 -7036.808737 -12349.9213 -5.748620e+03

[98,] 100895.43608 -2.955169e+03 -6994.769172 -12601.7652 -5.648541e+03

[99,] 101308.96067 -4.082358e+03 -7099.612362 -12917.7494 -5.687287e+03

[100,] 102741.48543 -5.470605e+03 -7348.987973 -13324.9757 -5.857248e+03

[101,] 104859.11516 -7.826482e+03 -7733.546990 -13848.2121 -6.103142e+03

[102,] 107753.22063 -1.113873e+04 -8268.774802 -14502.3171 -6.444703e+03

[103,] 111450.12541 -1.490692e+04 -8931.209285 -15217.8399 -6.880408e+03

[104,] 115411.08973 -1.880509e+04 -9644.811610 -16008.2100 -7.365300e+03

[105,] 115160.17160 -2.161470e+04 -9967.369344 -16125.2589 -7.564747e+03

[106,] 113101.30916 -2.403889e+04 -10134.277196 -15957.1943 -7.651071e+03

[107,] 110231.73525 -2.601977e+04 -10202.825894 -15659.8506 -7.671823e+03

[108,] 105380.92997 -2.671839e+04 -10001.492354 -15036.8063 -7.509508e+03

[109,] 99429.10863 -2.723424e+04 -9716.243620 -14267.9144 -7.287970e+03

[110,] 92211.49994 -2.691089e+04 -9247.080561 -13292.3779 -6.936734e+03

[111,] 83870.21373 -2.629956e+04 -8643.830513 -12204.6101 -6.473055e+03

[112,] 75824.91286 -2.661451e+04 -8139.492637 -11264.4376 -6.055989e+03

[113,] 67327.50830 -2.653153e+04 -7573.411849 -10227.4521 -5.604335e+03

[114,] 58155.98859 -2.554989e+04 -6870.539740 -9024.7188 -5.069453e+03

[115,] 51606.19678 -2.534395e+04 -6435.136898 -8194.5249 -4.732985e+03

[116,] 44826.46045 -2.354002e+04 -5792.597093 -7224.6087 -4.262625e+03

[117,] 40205.87541 -2.238194e+04 -5356.900364 -6578.0558 -3.939156e+03

[118,] 34849.85303 -1.997669e+04 -4715.190283 -5766.1333 -3.472524e+03

[119,] 24324.54649 -1.363834e+04 -3237.219572 -4089.0355 -2.407150e+03

[120,] 4302.58820 -4.630609e+02 -326.466207 -767.5555 -3.425138e+02

[121,] 3347.39264 -3.864120e+02 -220.696685 -688.8707 -2.493411e+02

[122,] 1421.93849 -3.493861e+02 -26.904891 -537.2609 -7.446450e+01

[123,] 404.11588 -1.106631e+02 91.626715 -430.6169 1.946291e+01

[124,] 138.69058 -5.914190e+01 107.209372 -415.0206 -2.258005e+00

[125,] 78.02941 -4.552716e+01 112.871330 -411.2688 -7.142955e+00

[126,] 75.88724 -4.403004e+01 115.122063 -411.3140 -7.126924e+00

[127,] 75.59637 -4.308490e+01 116.622300 -410.5890 -7.114603e+00

[128,] 75.89965 -4.216005e+01 118.286726 -411.9272 -6.963946e+00

[129,] 75.62103 -4.018115e+01 119.425167 -411.5139 -6.594676e+00

[130,] 74.20710 -3.884465e+01 121.136178 -410.7160 -6.683030e+00

[131,] 74.48926 -3.828653e+01 121.971259 -412.2744 -6.662963e+00

[132,] 74.63381 -3.787968e+01 122.156461 -411.5290 -6.630221e+00

[133,] 74.36187 -3.720227e+01 121.389445 -412.4400 -6.033902e+00

[134,] 74.40865 -3.563342e+01 121.325912 -413.3726 -5.872150e+00

[135,] 74.60713 -3.465936e+01 121.564387 -415.4779 -5.701615e+00

[136,] 73.48119 -3.416147e+01 121.712240 -415.2609 -5.760682e+00

[137,] 74.99133 -3.374355e+01 123.076972 -416.6732 -6.061095e+00

[138,] 74.27071 -3.315721e+01 123.904649 -416.8741 -6.082522e+00

[139,] 74.36879 -3.268532e+01 124.345429 -415.7727 -6.071252e+00

[140,] 74.21334 -3.243281e+01 123.629023 -416.9695 -6.027487e+00

[141,] 73.92934 -3.179731e+01 123.331153 -416.6239 -6.125457e+00

[142,] 73.88942 -3.145074e+01 121.788774 -416.9196 -6.251996e+00

[143,] 73.94131 -3.002416e+01 121.839860 -418.7680 -6.009192e+00

[144,] 74.12096 -2.953834e+01 120.687714 -418.9034 -5.860159e+00

[145,] 73.68494 -2.961634e+01 118.030443 -418.4943 -5.791552e+00

[146,] 73.64426 -2.976050e+01 117.004695 -419.1096 -5.862201e+00

[147,] 73.58988 -2.988724e+01 116.076247 -420.3078 -6.051059e+00

[148,] 74.46849 -2.999536e+01 115.313909 -421.3927 -6.189062e+00

[149,] 74.29414 -3.003735e+01 114.478151 -423.4094 -6.343985e+00

[150,] 75.67304 -2.985005e+01 113.865793 -426.3756 -6.423020e+00

[151,] 75.75148 -2.914729e+01 113.658915 -425.9052 -6.242592e+00

[152,] 74.60304 -2.716080e+01 116.242478 -423.8488 -5.547340e+00

[153,] 74.47681 -2.597738e+01 118.536295 -421.0702 -4.770640e+00

[154,] 73.70405 -2.556168e+01 119.325791 -418.7409 -4.555215e+00

[155,] 72.13931 -2.438394e+01 119.911187 -416.4801 -4.436818e+00

[156,] 73.41944 -2.418352e+01 119.985470 -414.4097 -4.412745e+00

[157,] 73.38360 -2.380930e+01 120.510612 -415.8801 -4.184233e+00

[158,] 72.72733 -2.315906e+01 120.662287 -415.5124 -4.300174e+00

[159,] 72.94199 -2.348670e+01 120.965902 -415.5788 -4.277456e+00

[160,] 72.71302 -2.291841e+01 121.351208 -414.6724 -4.149135e+00

[161,] 72.22683 -2.171465e+01 121.411971 -415.6621 -4.341707e+00

[162,] 71.42666 -2.149684e+01 121.846726 -415.9301 -4.288338e+00

[163,] 72.60775 -2.214802e+01 121.771190 -416.9607 -4.386404e+00

[164,] 71.78180 -2.166986e+01 122.315693 -417.0618 -4.324966e+00

[165,] 72.06885 -2.127143e+01 122.449709 -416.5821 -4.387404e+00

[166,] 72.16832 -2.071450e+01 123.047298 -415.9131 -4.294306e+00

[167,] 71.66300 -1.984589e+01 123.855487 -414.8495 -4.094686e+00

[168,] 71.48082 -1.988687e+01 124.084675 -413.7500 -4.057618e+00

[169,] 72.16227 -2.037123e+01 123.588403 -413.5301 -3.950104e+00

[170,] 114.64047 -2.811186e+01 121.717549 -417.9715 -3.527789e-01

[171,] 359.56825 -7.311745e+01 110.785998 -431.7033 2.016558e+01

[172,] 2279.10347 2.931688e+02 -73.212586 -485.4521 -1.714846e+02

[173,] 22361.47183 4.507552e+03 -2157.670111 -798.4782 -2.446070e+03

[174,] 40959.24294 7.361124e+03 -3953.222806 -1535.4270 -4.315118e+03

[175,] 55570.32856 9.146552e+03 -5234.253147 -2407.0333 -5.594077e+03

[176,] 70868.64257 1.056802e+04 -6513.384680 -3513.4139 -6.826401e+03

[177,] 79242.31434 1.077241e+04 -7160.226970 -4328.7760 -7.396551e+03

[178,] 89209.67512 1.113221e+04 -7929.959649 -5270.4932 -8.083008e+03

[179,] 98358.22004 1.125437e+04 -8614.093471 -6213.4692 -8.671492e+03

[180,] 105431.03292 1.093086e+04 -9109.159763 -7090.8329 -9.057776e+03

[181,] 111827.11195 1.038285e+04 -9583.351879 -7908.4432 -9.420676e+03

[182,] 117109.43500 9.921157e+03 -9966.474428 -8596.1297 -9.710071e+03

[183,] 123100.43045 1.051396e+04 -10367.019593 -9156.4362 -1.007407e+04

[184,] 122469.01830 9.953190e+03 -10180.041954 -9418.1824 -9.825703e+03

[185,] 118885.76413 8.895551e+03 -9785.734277 -9462.8132 -9.371594e+03

[186,] 99438.00335 5.966002e+03 -8185.375843 -8286.1904 -7.760245e+03

[187,] 118799.50233 6.857305e+03 -9494.540219 -10331.2451 -8.881834e+03

[188,] 119645.42627 5.146510e+03 -9283.049679 -11218.7663 -8.481424e+03

[189,] 114262.06842 2.806001e+03 -8627.609488 -11558.3862 -7.670624e+03

[190,] 109714.47693 7.378282e+02 -8070.756387 -11869.2785 -6.975766e+03

[191,] 106334.63704 -6.802013e+02 -7655.378037 -12075.3401 -6.465693e+03

[192,] 104577.13131 -1.464591e+03 -7442.034999 -12192.4219 -6.199610e+03

[193,] 103492.01551 -2.193679e+03 -7321.901460 -12307.8143 -6.031589e+03

[194,] 102980.97165 -3.314749e+03 -7290.994397 -12510.5089 -5.930740e+03

[195,] 103414.06233 -4.589652e+03 -7404.066854 -12748.8647 -5.967012e+03

[196,] 103086.19292 -5.753346e+03 -7455.225129 -12898.0822 -5.953380e+03

[197,] 105577.47907 -7.990808e+03 -7852.295075 -13402.6580 -6.211572e+03

[198,] 108832.97311 -1.139732e+04 -8435.737303 -14091.0306 -6.592910e+03

[199,] 112690.92076 -1.539238e+04 -9128.450441 -14897.7342 -7.047990e+03

[200,] 115656.83570 -1.905924e+04 -9744.143636 -15547.1376 -7.456302e+03

[ reached getOption("max.print") -- omitted 2564 rows ]

F)

First, this dataset have some missing value, which can not be analysis, so I removed it to make this model more accurated. And for the outliner, I think those outliners in each column still have some Continuity, those data don’t have huge gap between each others, therefore I didn’t removed it.

Second, during PCA, Dimension 6 ~12 are not necessary to keep it. Therefore I only keep dimension 1~5 for dimension reduction ,and rotated it to make variable be more related.

Finally, after PCA, I reduce 22 variables(not including column1 and 2) to 5 variables, those are "Light&ME", "Temperature","Humedad", "CO2", "Precipitacion ".

Q2:

A :

For factors analysis, this dataset is not appropriate. It has a lot of columns having high multicollinearity, and many variable’s uniqueness are too high. Therefore, during the factor analysis we have to removed those columns.

b :

factor 1 :Country&drug\_Le\_Ls, factor 2 : drug\_Am\_Co\_Ec\_Ke, factor 3 : drug\_Cr\_He\_Me

, factor 4 : drug\_Im\_SS, factor 5 : drug\_Ns\_Es\_Cs

Call:

factanal(x = DC\_rm\_miss[, c(-1, -2, -4, -5, -7, -15, -18, -19, -20, -21, -30, -32, -33)], factors = f\_num, rotation = "varimax")

Uniquenesses:

age country Nscore Escore Oscore Ascore Csore Impulsive SS Amphet Amyl Coke

0.711 0.282 0.600 0.399 0.661 0.834 0.583 0.369 0.314 0.524 0.646 0.398

Crack Ecstasy Heroin Ketamine Legalh LSD Meth Nicotine

0.555 0.283 0.409 0.589 0.489 0.481 0.582 0.740

Loadings:

Factor1 Factor2 Factor3 Factor4 Factor5

age

country -0.760

Nscore -0.619

Escore 0.699

Oscore

Ascore

Csore 0.570

Impulsive 0.737

SS 0.714

Amphet

Amyl 0.582

Coke 0.606

Crack 0.630

Ecstasy 0.681

Heroin 0.729

Ketamine 0.590

Legalh 0.535

LSD 0.607

Meth 0.537

Nicotine

Factor1 Factor2 Factor3 Factor4 Factor5

SS loadings 2.444 2.316 1.770 1.512 1.509

Proportion Var 0.122 0.116 0.089 0.076 0.075

Cumulative Var 0.122 0.238 0.327 0.402 0.478

Test of the hypothesis that 5 factors are sufficient.

The chi square statistic is 522.71 on 100 degrees of freedom.

The p-value is 1.76e-58

C :

unexplained variables :

age max : -0.468 in F1

Amphet max : 0.438. in F2

D :

In order to better comprehend the data, rotate the factors.

Orthogonal:

orthogonal rotation enables a clearer view of the factor loadings. But it will not keep correlation between factors and factors.

Oblique:

oblique rotation are rotations that retain the correlation between two variables, the axes are not kept perpendicular.

E :

None rotation:

Call:

factanal(x = DC\_rm\_miss[, c(-1, -2, -4, -5, -7, -15, -18, -19, -20, -21, -30, -32, -33)], factors = f\_num, rotation = "none")

Uniquenesses:

age country Nscore Escore Oscore Ascore Csore Impulsive SS Amphet Amyl Coke

0.711 0.282 0.600 0.399 0.661 0.834 0.583 0.369 0.314 0.524 0.646 0.398

Crack Ecstasy Heroin Ketamine Legalh LSD Meth Nicotine

0.555 0.283 0.409 0.589 0.489 0.481 0.582 0.740

Loadings:

Factor1 Factor2 Factor3 Factor4 Factor5

age

country -0.604

Nscore

Escore 0.714

Oscore

Ascore

Csore

Impulsive 0.515

SS 0.638

Amphet 0.669

Amyl

Coke 0.686

Crack

Ecstasy 0.745

Heroin 0.508

Ketamine 0.519

Legalh 0.675

LSD 0.656

Meth

Nicotine 0.502

Factor1 Factor2 Factor3 Factor4 Factor5

SS loadings 5.369 1.407 1.047 0.897 0.831

Proportion Var 0.268 0.070 0.052 0.045 0.042

Cumulative Var 0.268 0.339 0.391 0.436 0.478

Test of the hypothesis that 5 factors are sufficient.

The chi square statistic is 522.71 on 100 degrees of freedom.

The p-value is 1.76e-58

Orthogonal rotation (varimax):

Call:

factanal(x = DC\_rm\_miss[, c(-1, -2, -4, -5, -7, -15, -18, -19, -20, -21, -30, -32, -33)], factors = f\_num, rotation = "varimax")

Uniquenesses:

age country Nscore Escore Oscore Ascore Csore Impulsive SS Amphet Amyl Coke

0.711 0.282 0.600 0.399 0.661 0.834 0.583 0.369 0.314 0.524 0.646 0.398

Crack Ecstasy Heroin Ketamine Legalh LSD Meth Nicotine

0.555 0.283 0.409 0.589 0.489 0.481 0.582 0.740

Loadings:

Factor1 Factor2 Factor3 Factor4 Factor5

age

country -0.760

Nscore -0.619

Escore 0.699

Oscore

Ascore

Csore 0.570

Impulsive 0.737

SS 0.714

Amphet

Amyl 0.582

Coke 0.606

Crack 0.630

Ecstasy 0.681

Heroin 0.729

Ketamine 0.590

Legalh 0.535

LSD 0.607

Meth 0.537

Nicotine

Factor1 Factor2 Factor3 Factor4 Factor5

SS loadings 2.444 2.316 1.770 1.512 1.509

Proportion Var 0.122 0.116 0.089 0.076 0.075

Cumulative Var 0.122 0.238 0.327 0.402 0.478

Test of the hypothesis that 5 factors are sufficient.

The chi square statistic is 522.71 on 100 degrees of freedom.

The p-value is 1.76e-58

Oblique rotation (promax) :

Call:

factanal(x = DC\_rm\_miss[, c(-1, -2, -4, -5, -7, -15, -18, -19, -20, -21, -30, -32, -33)], factors = f\_num, rotation = "promax")

Uniquenesses:

age country Nscore Escore Oscore Ascore Csore Impulsive SS Amphet Amyl Coke

0.711 0.282 0.600 0.399 0.661 0.834 0.583 0.369 0.314 0.524 0.646 0.398

Crack Ecstasy Heroin Ketamine Legalh LSD Meth Nicotine

0.555 0.283 0.409 0.589 0.489 0.481 0.582 0.740

Loadings:

Factor1 Factor2 Factor3 Factor4 Factor5

age

country -0.901

Nscore -0.609

Escore 0.780

Oscore

Ascore

Csore 0.533

Impulsive 0.878

SS 0.783

Amphet

Amyl 0.696

Coke 0.632

Crack 0.656

Ecstasy 0.777

Heroin 0.747

Ketamine 0.686

Legalh

LSD 0.547

Meth

Nicotine

Factor1 Factor2 Factor3 Factor4 Factor5

SS loadings 2.639 1.973 1.754 1.466 1.438

Proportion Var 0.132 0.099 0.088 0.073 0.072

Cumulative Var 0.132 0.231 0.318 0.392 0.463

Factor Correlations:

Factor1 Factor2 Factor3 Factor4 Factor5

Factor1 1.000 -0.151 0.537 -0.530 0.373

Factor2 -0.151 1.000 -0.190 0.219 -0.328

Factor3 0.537 -0.190 1.000 -0.547 0.419

Factor4 -0.530 0.219 -0.547 1.000 -0.318

Factor5 0.373 -0.328 0.419 -0.318 1.000

Test of the hypothesis that 5 factors are sufficient.

The chi square statistic is 522.71 on 100 degrees of freedom.

The p-value is 1.76e-58

I choose Orthogonal rotation (varimax) for my final model. Because in this rotation, factors separate the variables better than None and Oblique rotation (promax) and no variables that are isolated in their own factor.

F :

Diagram:



Q3:

a)

Divide into 2 dataset by runif() Evenly Distributed Random Number.

mpg$rndnum = runif(194, 1,100)

> mpg\_train = mpg [ mpg[, "rndnum"] <= 80 , ]

> mpg\_test = mpg [ mpg[, "rndnum"] > 80 , ]

b)

after checking p-value, F-statistic, Adjusted R-squared and each variable’s p-value, mpg ~ weight+model\_year+amc +buick+chevrolet+chrysler+ford+mercury is my final equation.

And vif(mdl) shows that each variables have low multicollinearity.

hist(mdl$residuals) shows that the model is nearly like normal distribution.

f = "mpg ~ weight+model\_year+amc +buick+chevrolet+chrysler+ford+mercury"

> mdl = lm( f , data=mpg\_train) # use this one

> summary(mdl)

Call:

lm(formula = f, data = mpg\_train)

Residuals:

Min 1Q Median 3Q Max

-8.0093 -2.2371 -0.1319 1.9853 11.5732

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -1.628e+01 6.533e+00 -2.492 0.0138 \*

weight -6.355e-03 3.706e-04 -17.148 <2e-16 \*\*\*

model\_year 7.868e-01 8.005e-02 9.829 <2e-16 \*\*\*

amc -2.888e+00 1.391e+00 -2.075 0.0397 \*

buick -2.097e+00 1.323e+00 -1.585 0.1152

chevrolet -2.017e+00 1.011e+00 -1.995 0.0479 \*

chrysler -2.665e+00 1.661e+00 -1.604 0.1109

ford -2.117e+00 9.180e-01 -2.306 0.0225 \*

mercury -2.361e+00 1.307e+00 -1.807 0.0728 .

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.501 on 144 degrees of freedom

Multiple R-squared: 0.8355, Adjusted R-squared: 0.8264

F-statistic: 91.43 on 8 and 144 DF, p-value: < 2.2e-16

> # Regression Assumptions

> vif(mdl)

weight model\_year amc buick chevrolet chrysler ford mercury

1.337177 1.155601 1.055142 1.082978 1.127996 1.089142 1.144238 1.055971

> vif(mdl)

weight model\_year amc buick chevrolet chrysler ford mercury

1.337177 1.155601 1.055142 1.082978 1.127996 1.089142 1.144238 1.055971

> mean(mdl$residuals) # Approximately 0

[1] 2.554239e-16

>

> hist(mdl$residuals)

> shapiro.test(mdl$residuals) # W = 0.96747, p-value = 0.0008836 Ha

Shapiro-Wilk normality test

data: mdl$residuals

W = 0.96309, p-value = 0.0004098

> ad.test(mdl$residuals) # A = 1.1187, p-value = 0.006094 Ha

Anderson-Darling normality test

data: mdl$residuals

A = 1.1204, p-value = 0.006029

c)

ME = 0

MAE = 2.5

MPE = 0.01

MAPE = 0.12

> #ME

> #MAE

> #MPE

> #MAPE

> mean(mpg\_train$err) # == 0

[1] -8.485877e-14

> mean(mpg\_train$abs\_err)

[1] 2.570654

> mean(mpg\_train$pcterr)

[1] 0.01157451

> mean(mpg\_train$abs\_pcterr)

[1] 0.1192117

d)

ME = -0.34

MAE = 2.72

MPE = -0.03

MAPE = 0.14

> #ME

> #MAE

> #MPE

> #MAPE

>

> mean(mpg\_test$err, na.rm=T)

[1] -0.3498646

> mean(mpg\_test$abs\_err, na.rm=T)

[1] 2.720533

> mean(mpg\_test$pcterr, na.rm=T)

[1] -0.03058817

> mean(mpg\_test$abs\_pcterr, na.rm=T)

[1] 0.1410315

e)

average accuracy from train dataset:

Table

Description automatically generated

average accuracy from test dataset:

> paste("New dataset err average",mean(SA$err))

[1] "New dataset err average 0.16471130326216"

> paste("New dataset pcterr average",mean(SA$pcterr))

[1] "New dataset pcterr average 0.019225705870551"

> paste("New dataset abs\_err average",mean(SA$abs\_err))

[1] "New dataset abs\_err average 2.36116848628271"

> paste("New dataset abs\_pcter average",mean(SA$abs\_pcterr))

[1] "New dataset abs\_pcter average 0.113257097380802"

After built up a train regression model, and using stepwise regression row by row from the test set. The accuracy values have been increased. Which means, in this case, new data set is slightly different from the train set.

Chart, histogram

Description automatically generated

In other case, accuracy values have been decreased. Which means new dataset has improved the regression model.

Table

Description automatically generated

> paste("New dataset err average",mean(SA$err))

[1] "New dataset err average 0.913812842753099"

> paste("New dataset pcterr average",mean(SA$pcterr))

[1] "New dataset pcterr average 0.0524288063195208"

> paste("New dataset abs\_err average",mean(SA$abs\_err))

[1] "New dataset abs\_err average 2.18823358329297"

> paste("New dataset abs\_pcter average",mean(SA$abs\_pcterr))

[1] "New dataset abs\_pcter average 0.105772569612852"

Chart, histogram

Description automatically generated

Code :

library (readr)

library(car)

library(stringr)

library(nortest)

mpg <- read.csv("~/Desktop/Assigments/2023\_DM/file/mpg.csv", stringsAsFactors=FALSE)

mpg = as.data.frame(mpg)

# ---- separate car name ----

cars = unique(mpg[order(mpg$car.name),c(10)])

cars

cars\_string = ''

for(i in cars){

column\_name = str\_remove\_all(string = i,pattern = '"' )

mpg = cbind(mpg,i = as.numeric(mpg$car.name==i))

names(mpg)[ncol(mpg)] <- column\_name

cars\_string <- str\_c(cars\_string, column\_name , sep = "+")

}

# ---- clean column ----

mpg = mpg[,c(-1,-10,-11,-12)]

# ---- separate to two groups----

summary(mpg)

mpg$rndnum = runif(194, 1,100)

mpg\_Matrix = mpg [ mpg[, "rndnum"] <= 80 , ]

mpg\_New = mpg [ mpg[, "rndnum"] > 80 , ]

mpg\_Matrix = mpg\_Matrix[-36]

mpg\_New = mpg\_New[-36]

# ----stepwise regression function ----

stepwise\_regression <- function(model,df,row\_data){

# stepwise

model.final = step(model, na.action = na.exclude) # stepwise equation

# add new pre & error

df$pred = predict( model.final,newdata=df)

df$err = df$pred - df$mpg

df$pcterr = df$err / df$mpg

df$abs\_err = abs(df$err)

df$abs\_pcterr = abs(df$pcterr)

# check error

err = mean(df$err) # == 0

pctErr = mean(df$pcterr)

absErr = mean(df$abs\_err)

abspctErr = mean(df$abs\_pcterr)

records = c(

as.character(row\_data),

# vif(model.final),

# mean(model.final$residuals),

# shapiro.test(model.final$residuals),

# ad.test(model.final$residuals),

# bptest(model.final),

# durbinWatsonTest(model.final),

format(err, digits = 6, nsmall = 4, scientific = FALSE) ,

format(pctErr, digits = 6, nsmall = 4, scientific = FALSE) ,

format(absErr, digits = 6, nsmall = 4, scientific = FALSE),

format(abspctErr, digits = 6, nsmall = 4, scientific = FALSE) ,

paste(model.final$call,collapse = ",")

)

print(records)

# print(summary(model.final))

# print(paste(checks, collapse = ","))

new\_model = model.final

return(records)

}

# -------- start --------

# do 80% train set regression first

formula = as.formula(mpg ~. )

mdl = lm(formula , data=mpg\_Matrix) #fisrt full equation after train set

summary(mdl)

# store the first equation and accuracy data

pre\_val = list()

pred = predict( mdl,newdata=mpg\_Matrix)

err = pred - mpg\_Matrix$mpg

pcterr = err / mpg\_Matrix$mpg

abs\_err = abs(err)

abs\_pcterr = abs(pcterr)

pre\_matrix = mean(pred)

err = mean(err) # == 0

pctErr = mean(pcterr )

absErr = mean(abs\_err )

abspctErr = mean(abs\_pcterr )

# set up dataframe for accuracy values

SA = data.frame(

row\_data = c("80% train set"),

err = c(err),

pcterr = c(pctErr),

abs\_err = c(absErr),

abs\_pcterr = c(abspctErr),

regression\_mdl = c(paste(mdl$call,collapse = ","))

)

pre\_val = append(pre\_val,pre\_matrix)

# start to add rows

for (i in 1:nrow(mpg\_New)) {

add\_row = mpg\_New[i,] # pick out each rows

row\_str <- paste(mpg\_New[i, ], collapse = ",") # row's call

mpg\_Matrix <- rbind(mpg\_Matrix, add\_row) #add row to matrix set

formula = as.formula(mpg ~. ) #create formula

mdl = lm(formula , data=mpg\_Matrix) # set up formula

regression = stepwise\_regression(mdl,mpg\_Matrix,row\_str) #do train set regression

newrow\_record = NULL

# get formula and matrix accuracy values

for( i in (1:length(regression))){

# print(i)

# newrow\_record = c(newrow\_record, regression[i])

if (i == 6) {

formula\_str = regression[i]

formula = as.formula(strsplit(regression[i], ",")[[1]][[2]]) #get new formula

}

}

# set up test formula

mdl\_new = lm(formula , data=mpg\_Matrix)

newRow\_pred = predict(mdl\_new,newdata=add\_row) # predict new row & measure accuracy

pre\_val[[1]] = append(pre\_val[[1]],as.character(newRow\_pred))

newRow\_err = newRow\_pred - add\_row$mpg

newRow\_pcterr = newRow\_err / add\_row$mpg

newRow\_abs\_err = abs(newRow\_err)

newRow\_abs\_pcterr = abs(newRow\_pcterr)

newrow\_record = c(row\_str, newRow\_err, newRow\_pcterr, newRow\_abs\_err, newRow\_abs\_pcterr,formula\_str)

SA = rbind(SA,newrow\_record) #import accuracy values

}

# import predict values and index

SA<- cbind(SA, predict\_val= pre\_val)

SA<- cbind(SA, index = 1:nrow(SA))

# get average of new dataset

SA$err = as.numeric(SA$err)

SA$pcterr = as.numeric(SA$pcterr)

SA$abs\_err = as.numeric(SA$abs\_err)

SA$abs\_pcterr = as.numeric(SA$abs\_pcterr)

paste("New dataset err average",mean(SA$err))

paste("New dataset pcterr average",mean(SA$pcterr))

paste("New dataset abs\_err average",mean(SA$abs\_err))

paste("New dataset abs\_pcter average",mean(SA$abs\_pcterr))

# print chart

par(mfrow = c(2, 2))

# par(mar=c(3,3,2,1)) # set margin

plot(SA$index,format(SA$err,scientific=FALSE), type = "l", main = "err line chart", xlab ="index" , ylab = "values", col = "blue")

plot(SA$index,SA$abs\_err, type = "l", main = "abs\_err line chart", xlab ="index" , ylab = "values", col = "red")

plot(SA$index,SA$pcterr, type = "l", main = "abs\_err line chart", xlab = "index", ylab = "values", col = "orange")

plot(SA$index,SA$abs\_pcterr, type = "l", main = "abs\_err line chart", xlab ="index", ylab = "values", col = "green")