

```

1  ###dummy変数による重回帰分析###
2  ###makedummies関数を使ってダミー変数化###
3  library(makedummies)
4  dat <- read.csv("data/mtcars.csv",
5                header = T ,
6                row.names = 1 )
7  names(dat)
8  fc <- lapply(dat[,c(2,8,9)],          #lapply()関数で一括してfactor型に変換
9             as.factor)
10 dat.f <- data.frame(dat , fc)        #元data最終列にマージ
11 names(dat.f)                        #factor変換後変数は変数名.1になっている
12 dum <- makedummies(dat.f[,c(12:14)] , #makedummies()関数でdummy変数化
13                basal_level = T)     #defaultはF:基準カテゴリー除外
14 head(dum , 2)                       #dummy変数化されているか確認
15 df <- data.frame(dum)
16 names(df)
17 names(df) [1:7]<-c("cyl.4","cyl.6","cyl.8",  #項目名をリネーム:必須ではない
18                "v.shaped","straight",
19                "automatic","manual")
20 head(df)
21 dat.d <- data.frame(dat , df)        #dummy変数をマージし保存
22 names(dat.d)
23 write.csv(dat.d,"data/mtcars.dum.csv")
24 dat.dum <- read.csv("data/mtcars.dum.csv",
25                  row.names=1 , header = T)
26 out<-lm( mpg ~ cyl.6+cyl.8+          #重回帰分析
27          v.shaped+
28          automatic,                 #vs:straight,am:manualを基準カテゴリー
29          data = dat.dum)
30 summary(out)
31
32 ###正規化scoreの計算(基準カテゴリーの値は0)になっている###
33 #基準カテゴリーをどのカテゴリーにしたか要注意
34 names(df)                            #項目名確認
35 out$coefficients                      #係数確認
36 #cylの正規化score
37 coef <- c(0,                          #1番目は0(基準カテゴリー)
38          out$coefficients[2:3])       #2,3列の係数
39 coef                                  #偏回帰係数を返す
40 freq <- xtabs(~cyl ,                  #変数:cylの度数分布確認
41             data = dat)               #元data
42 freq                                  #度数分布を返す
43 c.coef <- coef-                        ##偏回帰係数-平均値
44   (c(coef%*%freq/sum(freq)))
45 c.coef                                #正規化scoreを返す
46 #vsの正規化score
47 coef <- c(out$coefficients[4],0)     #基準カテゴリー2番目
48 coef                                  #4列の係数,2番目は0を返す
49 freq <- xtabs(~vs ,                  #変数:cylの度数分布確認
50             data = dat)               #元data
51 freq                                  #度数分布を返す
52 v.coef <- coef-                       ##偏回帰係数-平均値
53   (c(coef%*%freq/sum(freq)))         #正規化scoreを返す
54 v.coef
55 #amの正規化score
56 coef <- c(out$coefficients[5],0)     #基準カテゴリー2番目
57 coef                                  #5列の係数,2番目は0を返す
58 freq <- xtabs(~am ,                  #変数:cylの度数分布確認
59             data = dat)               #元data
60 freq                                  #度数分布を返す

```

```

61 a.coef <- coef- ##偏回帰係数-平均値
62 (c(coef**%freq/sum(freq)))
63 a.coef ##正規化scoreを返す
64
65 ###偏相関係数の計算###
66 ##偏相関係数計算用dataはExcelで作成(作成したdataは103行目から)
67 #data作成のために正規化scoreをvector化
68 vc <- c(c.coef,v.coef,a.coef) #まとめてvector化
69 vc
70 t <- t(c(c.coef,v.coef,a.coef)) #転置
71 t
72 d <- data.frame(vc) #data.frame化
73 dimnames(d) <- list(c("cyl.4","cyl.6","cyl.8", #名前を付ける
74 "v.shaped","straight",
75 "automatic","manual"),
76 c("cate.score"))
77 d #正規化scoreをvector化
78
79 pcor <- read.csv("data/mtcars.pcor.csv",
80 row.names = 1 , header = T)
81 head(pcor,3)
82 cor <- cor(pcor[,c(1:4)])
83 round(cor,3)
84 ###偏相関係数行列の計算
85 library(corpcor)
86 p.cor <- cor2pcor(cor) #cor2pcor()関数
87 p.cor #2行1列:cyl,3行1列:vs,4行1列:am
88 ###参考:逆行列を求め偏相関係数を計算_1
89 #相関行列の逆行列の対角成分の外戚の平方根で相関係数行列の逆行列を除算しマイナス
90 s.cor <- solve(cor)
91 d <- diag(s.cor)
92 -(s.cor/sqrt(outer(d , d)))
93 ###参考:逆行列を求め偏相関係数を計算_2
94 det(cor) #行列式
95 tcor <- solve(cor) #逆行列
96 tcor
97 round(cor**%solve(cor),1) #単位行列子イック
98
99 -(tcor[1,2])/sqrt(tcor[1,1]*tcor[2,2]) #偏相関係数:cyl
100 -(tcor[1,3])/sqrt(tcor[1,1]*tcor[3,3]) #偏相関係数:vs
101 -(tcor[1,4])/sqrt(tcor[1,1]*tcor[4,4]) #偏相関係数:am
102
103 #<引用/参考文献>
104 #青木繁伸(2005):「数量化I類はダミー変数を用いた重回帰分析である」
105
106 #偏相関係数計算用data(mtcars.pcor.csv)
107 # mpg cyl vs am
108 #Mazda RX4 21 -0.6471711 -0.7472771 1.8791548
109 #Mazda RX4 Wag 21 -0.6471711 -0.7472771 1.8791548
110 #Datsun 710 22.8 4.7515025 0.9607849 1.8791548
111 #Hornet 4 Drive 21.4 -0.6471711 0.9607849 -1.2857375
112 #Hornet Sportabout 18.7 -3.4097378 -0.7472771 -1.2857375
113 #Valiant 18.1 -0.6471711 0.9607849 -1.2857375
114 #Duster 360 14.3 -3.4097378 -0.7472771 -1.2857375
115 #Merc 240D 24.4 4.7515025 0.9607849 -1.2857375
116 #Merc 230 22.8 4.7515025 0.9607849 -1.2857375
117 #Merc 280 19.2 -0.6471711 0.9607849 -1.2857375
118 #Merc 280C 17.8 -0.6471711 0.9607849 -1.2857375
119 #Merc 450SE 16.4 -3.4097378 -0.7472771 -1.2857375
120 #Merc 450SL 17.3 -3.4097378 -0.7472771 -1.2857375

```

```
121 #Merc 450SLC      15.2 -3.4097378 -0.7472771 -1.2857375
122 #Cadillac Fleetwo 10.4 -3.4097378 -0.7472771 -1.2857375
123 #Lincoln Continent 10.4 -3.4097378 -0.7472771 -1.2857375
124 #Chrysler Imperial 14.7 -3.4097378 -0.7472771 -1.2857375
125 #Fiat 128         32.4  4.7515025  0.9607849  1.8791548
126 #Honda Civic     30.4  4.7515025  0.9607849  1.8791548
127 #Toyota Corolla  33.9  4.7515025  0.9607849  1.8791548
128 #Toyota Corona   21.5  4.7515025  0.9607849 -1.2857375
129 #Dodge Challenger 15.5 -3.4097378 -0.7472771 -1.2857375
130 #AMC Javelin     15.2 -3.4097378 -0.7472771 -1.2857375
131 #Camaro Z28      13.3 -3.4097378 -0.7472771 -1.2857375
132 #Pontiac Firebird 19.2 -3.4097378 -0.7472771 -1.2857375
133 #Fiat X1-9       27.3  4.7515025  0.9607849  1.8791548
134 #Porsche 914-2   26    4.7515025 -0.7472771  1.8791548
135 #Lotus Europa    30.4  4.7515025  0.9607849  1.8791548
136 #Ford Pantera L  15.8 -3.4097378 -0.7472771  1.8791548
137 #Ferrari Dino    19.7 -0.6471711 -0.7472771  1.8791548
138 #Maserati Bora   15    -3.4097378 -0.7472771  1.8791548
139 #Volvo 142E     21.4  4.7515025  0.9607849  1.8791548
140
141
142
```