标红部分为使用者需要调整参数的地方，未标红的部分使用者不要随意更改，以防模板无法正常运行。编号与原始R文件中编号一致。

1. *###清洗及描述部分包下载*
2. library(Hmisc)*#读取SPSS文件*
3. library(openxlsx)*#读取Excel文件*
4. library(pastecs)*#描述统计量*
5. library(psych)*#描述统计量*
6. library(VIM)*#总体评价*
7. library(mice)*#多重插补*
8. library(haven)*#写入spss(sav格式)*
9. *###数据读取及原始数据统计--------------------------------------------------------------------*
10. health\_data1 <- spss.get(file.choose())*#读取SPSS数据(若含有中文字符，需保存为本地编码格式)*
11. health\_data1 <- read.xlsx(file.choose())*#读取Excel数据*
12. View(head(health\_data1)) *#检查读取(Rstudio中使用)*
13. des1 <- summary(health\_data1)
14. des2 <- stat.desc(health\_data1,norm = T)
15. des3 <- Hmisc::describe(health\_data1)
16. des4 <- psych::describe(health\_data1)
17. describe\_list <- list(des1,des2,des3,des4)*#原始数据信息*
18. health\_data1 <- as.data.frame(apply(health\_data1, 2, as.character))
19. *#aggregate.data.frame(myda[,-c(1:4)],by=list(myda$性别,myda$吸烟情况),FUN = sd)#分组描述参考*
20. *#summary(health\_data1)#检查转化（去掉第一个“#”执行）*
21. *###-----------------------------------------------------------------------------------------*
22. *###完整性评价-------------------------------------------------------------------------------*
23. *##人员差缺率------------------------------------------------------------*
24. cqs <- length(health\_data1[is.na(health\_data1)])*#总指标差缺数*
25. zbr <- (length(health\_data1)-1)\*length(health\_data1[,1])*#指标\*人数（去掉不需要评价的列数）*
26. cqz <- cqs/zbr *#人员差缺率*
27. cqz
28. rm(cqs,zbr)
29. *##指标差缺率------------------------------------------------------------*
30. zcql <- c()*#指标差缺率*
31. zb <- colnames(health\_data1)[-1] #不需评价列
32. for (i in 2:length(health\_data1)) {#调整源数据列顺序，将不需要评价的列放到前面
33. zcql[i-1] <- length(health\_data1[i][is.na(health\_data1[i])])/length(health\_data1[,i])
34. }
35. zcqld <- data.frame(zb, zcql)
36. colnames(zcqld) <- c("指标","指标差缺率")
37. zcqld *#指标差缺率表格*
38. rm(zb,zcql,i)
39. *##缺失记录定位----------------------------------------------------------*
40. ra <- which(is.na(health\_data1))*#整体位置*
41. hang <- ifelse(ra %%length(health\_data1[,1])==0,
42. length(health\_data1[,1]),
43. ra %%length(health\_data1[,1]))*#行位置*
44. lie <- ifelse(ra %%length(health\_data1[,1])==0,
45. ra%/%length(health\_data1[,1]),
46. ra%/%length(health\_data1[,1])+1)*#列位置*
47. weizhi <- data.frame(hang,lie)*#准确位置*
48. colnames(weizhi) <- c("行","列")
49. weizhi
50. rm(ra,hang,lie)
51. *###----------------------------------------------------------------------------------------*
52. *###准确性评价-------------------------------------------------------------------------------*
53. *##---------------------------------------------------------------------*
54. *#health\_data1 <- spss.get(file.choose())#读取SPSS数据(若含有中文字符，需保存为本地编码格式)：测试用*
55. *#health\_data1 <- read.xlsx(file.choose())#读取Excel数据:测试用*
56. *#health\_data1 <- as.data.frame(apply(health\_data1, 2, as.character)) ：测试用*
57. *##分类变量是否混入无效值检查；比如1表示男，2表示女，数据中出现3----------*
58. View(head(health\_data1)) *#方便定位*
59. effectv <- list(as.character(c(1,0)),as.character(c(1,0)),as.character(1:5))*#有效值（按顺序输入，可扩展）*
60. dcl <- c(2,3,6)*#需检查列，要与有效值顺序对应*
61. for (i in 1:length(dcl)) {
62. autua <- unique(health\_data1[,dcl[i]])
63. uneffectv <- setdiff(autua,effectv[[i]])*#无效值*
64. health\_data1[,dcl[i]][health\_data1[,dcl[i]] %in% uneffectv] <- "10086" *#无效值赋值*
65. }
66. *##数字混进字符------------------------------------------------------------*
67. dclt <- c(2,3,6)*#需检查列*
68. for (i in dclt) {
69. zifu <- setdiff(health\_data1[,i],as.character(as.numeric(health\_data1[,i])))
70. health\_data1[,i][health\_data1[,i] %in% zifu] <- "10086"
71. } *#数字混进字符*
72. *##字符混进数字------------------------------------------------------------*
73. dclm <- c(7,8)*#需检查列*
74. for (i in dclm) {
75. shuzi <-intersect(health\_data1[,i],as.character(as.numeric(health\_data1[,i])))
76. shuzi <- shuzi[!is.na(shuzi)]
77. health\_data1[,i][health\_data1[,i] %in% shuzi] <- "10086"
78. }*#字符混进数字*
79. *##收缩压小于舒张压检查----------------------------------------------------*
80. higblood <- 5 *#收缩压数据所在列*
81. lowblood <- 4 *#舒张压数据所在列*
82. bloodd <- rownames(health\_data1)[as.numeric(health\_data1[,higblood])-as.numeric(health\_data1[,lowblood]) <= 0] *#血压无效值*
83. bloodd <- as.integer(bloodd)
84. health\_data1[,lowblood][bloodd] <- "10086"
85. health\_data1[,higblood][bloodd] <- "10086"
86. *##总指标无效值检查-------------------------------------------------------*
87. wxzs <- length(health\_data1[health\_data1=="10086"])*#总指标无效值数量*
88. zbr <- (length(health\_data1)-1)\*length(health\_data1[,1])*#指标\*人数*
89. xwsz <- wxzs/zbr *#总无效值比例*
90. xwsz
91. rm(wxzs,zbr)
92. *##单项指标无效值---------------------------------------------------------*
93. dzcql <- c()*#单项指标无效值比例*
94. zb <- colnames(health\_data1)[-1]
95. for (i in 2:length(health\_data1)) {
96. dzcql[i-1] <- length(health\_data1[i][health\_data1[i]=="10086"])/length(health\_data1[,i])
97. }
98. dzcqld <- data.frame(zb, dzcql)
99. colnames(dzcqld) <- c("指标","指标无效值比例")
100. dzcqld *#指标无效值比例表格*
101. rm(zb,dzcql,i)
102. *##无效值定位-------------------------------------------------------------*
103. wra <- which(health\_data1=="10086")*#整体位置*
104. whang <- wra%%length(health\_data1[,1])*#行位置*
105. wlie <- wra%/%length(health\_data1[,1])*#列位置*
106. wweizhi <- data.frame(whang,wlie)*#准确位置*
107. colnames(wweizhi) <- c("行","列")
108. wweizhi
109. rm(wra,whang,wlie)
110. *#无效值标记为NA---------------------------------------------------------*
111. health\_data1[health\_data1=="10086"] <- NA
112. *##唯一性检查--------------------------------------------------------------*
113. id\_u <- 1*#唯一身份识别信息所在列*
114. id\_NA <- which(is.na(health\_data1[,id\_u])==T)*#唯一身份识别信息缺失R\_dataframe索引*
115. id\_NA
116. for (i in 1:length(id\_NA)) {
117. health\_data1[id\_NA,id\_u][i] <- paste0("未知",i)
118. }*#标记未知列*
119. chongfux <- health\_data1[duplicated(health\_data1[,id\_u]),id\_u]*#所有重复*
120. health\_data1[which(health\_data1[,id\_u] %in% chongfux),]*#所有重复记录展示*
121. chongfuy <- health\_data1[duplicated(health\_data1),id\_u]*#完全重复*
122. health\_data1[which(health\_data1[,id\_u] %in% chongfuy),]*#完全重复记录展示*
123. bwqchongfu <- setdiff(chongfux,chongfuy)*#不完全重复*
124. health\_data1[which(health\_data1[,id\_u] %in% bwqchongfu),]*#不完全重复记录展示*
125. zcfl <- length(chongfux)/length(health\_data1[,id\_u])*#总重复率*
126. zcfl
127. wqcfl <- length(chongfuy)/length(health\_data1[,id\_u])*#完全重复率*
128. wqcfl
129. bwqcfl <- length(bwqchongfu)/length(health\_data1[,id\_u])*#不完全重复率*
130. bwqcfl
131. origin\_len <- length(health\_data1[,1])
132. health\_data1 <- unique(health\_data1[!(health\_data1[,id\_u] %in% bwqchongfu),])*#去除重复值（包括不完全）*
133. use\_record1 <- length(health\_data1[,1])/origin\_len*#可用值比例*
134. *##离群值检查--------------------------------------------------------------*
135. *#health\_data1 <- spss.get(file.choose())#读取SPSS数据(若含有中文字符，需保存为本地编码格式)：测试用*
136. *#health\_data1 <- read.xlsx(file.choose())#读取Excel数据:测试用*
137. *#health\_data1 <- as.data.frame(apply(health\_data1, 2, as.character)) ：测试用*
138. dcv <- c(5:12)*#待检测指标*
139. zoutlie <- list()
140. lqzbl <- c()*#离群值比例*
141. for (i in 1:length(dcv)) {
142. lie\_rank <- dcv[i]
143. jclz <- as.numeric(health\_data1[,lie\_rank])
144. qx <- quantile(jclz,na.rm = T)
145. IQR <- qx[4]-qx[2]
146. names(IQR) <- "IQR"
147. highlim <- qx[4]+1.5\*IQR
148. lowlim <- qx[2]-1.5\*IQR
149. outliem <- jclz[jclz>highlim | jclz<lowlim]*#四分位间距法离群值*
150. indm <- which(jclz>highlim | jclz<lowlim)*#离群值索引（四分位间距法）*
151. meand <- mean(jclz,na.rm = T)
152. sdd <- sd(jclz,na.rm = T)
153. minlim <- meand-1.96\*sdd
154. maxlim <- meand+1.96\*sdd
155. outlies <- jclz[jclz>maxlim | jclz<minlim]*#均数标准差法离群值*
156. inds <- which(jclz>maxlim | jclz<minlim)*#离群值索引（均数标准差法）*
157. zoutlie[[i]] <- jclz[intersect(indm,inds)]*#两种方法的交集（最终离群值）*
158. lqzbl[i] <- length(zoutlie[[i]])/length(jclz) *#离群值比例*
159. }
160. names(zoutlie) <- colnames(health\_data1)[dcv]
161. lqzbl\_df <- data.frame(colnames(health\_data1)[dcv],lqzbl)*#离群值比例列表*
162. zlqz <- mean(lqzbl)*#总离群值比例*
163. colnames(lqzbl\_df) <- c("指标","离群值比例")
164. lqzbl\_df*#离群值比例列表*
165. zlqz *#总离群值比例*
166. *#离群值标记--------------------------------------------------------------*
167. health\_data2 <- health\_data1
168. for (i in 1:length(dcv)) {
169. lie\_rank <- dcv[i]
170. health\_data2[,lie\_rank][health\_data2[,lie\_rank] %in% zoutlie[[i]]] <- "outlies"
171. print(length(which(health\_data2[,lie\_rank] %in% "outlies"))==length(zoutlie[[i]]))
172. }
173. hangx <- (which(health\_data2=="outlies")%%length(health\_data2[,1]))
174. hangx[hangx==0] <- length(health\_data2[,1])
175. health\_data2 <- health\_data2[-hangx,]*#删除离群值的数据集*
176. health\_data1[,dcv] <- apply(health\_data1[,dcv],2,as.numeric)
177. health\_data2[,dcv] <- apply(health\_data2[,dcv],2,as.numeric)
178. *#health\_data1为保留离群值数据；health\_data2为删除离群值数*
179. *###总体数据质量评价-----------------------------------------------------------------------------------------*
180. use\_record1  *#可用记录比例（保留离群值）*
181. use\_record2 <- length(health\_data2[,1])/origin\_len*##可用记录比例（删除离群值）*
182. *#保留离群值*
183. aggr\_plot1 <- aggr( health\_data1, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE, labels=names(health\_data1), cex.axis=.7, gap=3, ylab=c("Histogram of abnormal data","Pattern"))
184. parten1 <- md.pattern(health\_data1)
185. *#删除离群值*
186. aggr\_plot2 <- aggr( health\_data2, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE, labels=names(health\_data2), cex.axis=.7, gap=3, ylab=c("Histogram of abnormal data","Pattern"))
187. parten2 <- md.pattern(health\_data2)
188. *###---------------------------------------------------------------------------------------------------------*
189. *###数据处理-------------------------------------------------------------------------------------------------*
190. *#health\_data1 <- spss.get("D:/Desktop/jupfile/health\_test2.sav")#读取SPSS数据(若含有中文字符，需保存为本地编码格式)：测试用*
191. *#health\_data1 <- read.xlsx(file.choose())#读取Excel数据:测试用*
192. *#health\_data1 <- as.data.frame(apply(health\_data1, 2, as.character)) ：测试用*
193. *#health\_data1 <- unique(health\_data1[!(health\_data1[,id] %in% bwqchongfu),]) #重复值处理:数据检查时已做*
194. dec\_health <- health\_data1[,-c(1)] *#需填补数据集部分（去掉非研究变量）*
195. *#须填补列中有因子型变量需将其转换成数字,若后续需要再自行转化成因子*
196. iddz <- c(1:11)*#需转化成数字的列*
197. iddx <- c(1,4:11)*#均数填充列*
198. iddy <- c(2:3)*#众数填充列*
199. *#计数资料无空值不需要执行*
200. for(i in iddz){
201. dec\_health[,i] <- as.numeric(dec\_health[,i])
202. }
203. *#data：包含缺失值的数据框或矩阵。缺失值被编码为 NA。*
204. *#m：多重插补法的数量，默认为 5。*
205. *#method：指定数据中每一列的输入方法。1）默认 pmm；2）二分类数据适用 logreg；3）无序多类别数据适用 ployreg；4）有序多分类变量适用 polr。详细参考?mice*
206. *#maxit：迭代次数，默认为 5 。*
207. *#seed：一个整数，由set.seed()产生，作为偏移随机数生成器。默认情况下，不使用随机数生成器。*
208. mt <- 5 *#插补次数*
209. tempData <- mice(dec\_health,m=mt,maxit=50,seed=600,method = "pmm")#产生多种结果
210. summary(tempData)
211. *#寻找合适值*
212. densityplot(tempData)
213. densityplot(tempData,~`空腹血糖`+`HDL` | .imp)
214. *#自行选择合适值填充-------------------------------*
215. id\_dt <- 1*#原数据填充列*
216. tic <- 2*#打算填充次数*
217. dait <- dec\_health[,id\_dt]
218. dait <- complete(tempData,action = tic)[,id\_dt]
219. *#平均值填充(计量资料)-----------------------------*
220. ss <- 0
221. for(i in 1:mt){
222. ddaa <- complete(tempData,action = i)
223. ss <- ss + ddaa[,iddx]
224. }
225. dec\_health[,iddx] <- ss/mt
226. *#众数填补（计数资料）-----------------------------*
227. zhongshu <- iddy*#计数资料填补列*
228. health\_data1y <- complete(tempData,action = 1)
229. sdt <- complete(tempData,action = 1)
230. for (i in 2:mt) {
231. sdt <- rbind(sdt,complete(tempData,action = i))
232. }
233. sdt$group <- rep(c(1:length(complete(tempData,action = 1)[,1])),time=mt)
234. for (i in 1:length(complete(tempData,action = 1)[,1])) {
235. lgde <- sdt[sdt$group==i,]
236. for (m in zhongshu) {
237. ifelse(is.na(as.integer(names(which.max(table(lgde[,m])))))==T ,
238. health\_data1y[i,m] <- names(which.max(table(lgde[,m]))),
239. health\_data1y[i,m] <- as.integer(names(which.max(table(lgde[,m])))))
240. }
241. }
242. dec\_health[,iddy] <- health\_data1y[,iddy]
243. aggr\_plot3 <- aggr(dec\_health, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE, labels=names(dec\_health), cex.axis=.7, gap=3, ylab=c("Histogram of abnormal data","Pattern"))
244. save.image(file = "env\_data\_deal.RData") *#保存所有变量*
245. saveRDS(dec\_health,"处理完成数据.RDS")
246. *###结果描述-----------------------------------------------------------------------------*

特别注意：

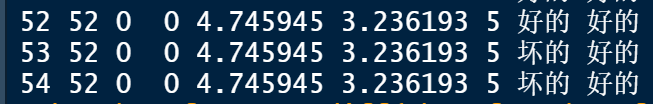
1.10.health\_data1 <- spss.get(file.choose())#读取SPSS数据(若含有中文字符，需保存为本地编码格式)如图：



保存好的文件用SPSS打开后，再次用导入R需要重新按上图保存一次。

分类变量用0表示时，R读取时可能无法识别0所对应的标签产生警告，可用其他数字来替代0表示对应标签。注意在原数据中将0替换成所选择的替代数字。例如用0表示不患病时报警告，就用2表示不患病，并将原数据中的0替换成2 。文件名不能包含中文

2.唯一性检查时，若同于身份识别ID下出现A1与A2完全相同，A2与A3不完全相同，所以记录会被全部归为完全相同。（基于此种情况出现概率不高，不对源码进行修改，使用者适当注意）



3.54-142行使用test.xlsx文件演示。其余所有部分使用health\_test2.sav演示。

4.描述+图表+图表参考数据.RData中包括图表演示数据及描述数据，对照注释选择需要变量即可。