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(prettyR)*#计算众数*
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*
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]
32. for (i in 2:length(health\_data1)) {
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 <- ra%%length(health\_data1[,1])*#行位置*
42. lie <- ra%/%length(health\_data1[,1])*#列位置*
43. weizhi <- data.frame(hang,lie)*#准确位置*
44. colnames(weizhi) <- c("行","列")
45. weizhi
46. rm(ra,hang,lie)
47. *###----------------------------------------------------------------------------------------*
48. *###准确性评价-------------------------------------------------------------------------------*
49. *##---------------------------------------------------------------------*
50. *#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*
51. *#health\_data1 <- read.xlsx(file.choose())#读取Excel数据:测试用*
52. *#health\_data1 <- as.data.frame(apply(health\_data1, 2, as.character)) ：测试用*
53. *##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----------*
54. View(head(health\_data1)) *#方便定位*
55. 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）*
56. dcl <- c(2,3,6)*#需检查列，要与有效值顺序对应*
57. for (i in 1:length(dcl)) {
58. autua <- unique(health\_data1[,dcl[i]])
59. uneffectv <- setdiff(autua,effectv[[i]])*#无效值*
60. health\_data1[,dcl[i]][health\_data1[,dcl[i]] %in% uneffectv] <- "10086" *#无效值赋值*
61. }
62. *##数字混进字符------------------------------------------------------------*
63. dclt <- c(2,3,6)*#需检查列*
64. for (i in dclt) {
65. zifu <- setdiff(health\_data1[,i],as.character(as.numeric(health\_data1[,i])))
66. health\_data1[,i][health\_data1[,i] %in% zifu] <- "10086"
67. } *#数字混进字符*
68. *##字符混进数字------------------------------------------------------------*
69. dclm <- c(7,8)*#需检查列*
70. for (i in dclm) {
71. shuzi <-intersect(health\_data1[,i],as.character(as.numeric(health\_data1[,i])))
72. shuzi <- shuzi[!is.na(shuzi)]
73. health\_data1[,i][health\_data1[,i] %in% shuzi] <- "10086"
74. }*#字符混进数字*
75. *##收缩压小于舒张压检查----------------------------------------------------*
76. higblood <- 5 *#收缩压数据所在列*
77. lowblood <- 4 *#舒张压数据所在列*
78. bloodd <- rownames(health\_data1)[as.numeric(health\_data1[,higblood])-as.numeric(health\_data1[,lowblood]) <= 0] *#血压无效值*
79. bloodd <- as.integer(bloodd)
80. health\_data1[,lowblood][bloodd] <- "10086"
81. health\_data1[,higblood][bloodd] <- "10086"
82. *##总指标无效值检查-------------------------------------------------------*
83. wxzs <- length(health\_data1[health\_data1=="10086"])*#总指标无效值数量*
84. zbr <- (length(health\_data1)-1)\*length(health\_data1[,1])*#指标\*人数*
85. xwsz <- wxzs/zbr *#总无效值比例*
86. xwsz
87. rm(wxzs,zbr)
88. *##单项指标无效值---------------------------------------------------------*
89. dzcql <- c()*#单项指标无效值比例*
90. zb <- colnames(health\_data1)[-1]
91. for (i in 2:length(health\_data1)) {
92. dzcql[i-1] <- length(health\_data1[i][health\_data1[i]=="10086"])/length(health\_data1[,i])
93. }
94. dzcqld <- data.frame(zb, dzcql)
95. colnames(dzcqld) <- c("指标","指标无效值比例")
96. dzcqld *#指标无效值比例表格*
97. rm(zb,dzcql,i)
98. *##无效值定位-------------------------------------------------------------*
99. wra <- which(health\_data1=="10086")*#整体位置*
100. whang <- wra%%length(health\_data1[,1])*#行位置*
101. wlie <- wra%/%length(health\_data1[,1])*#列位置*
102. wweizhi <- data.frame(whang,wlie)*#准确位置*
103. colnames(wweizhi) <- c("行","列")
104. wweizhi
105. rm(wra,whang,wlie)
106. *#无效值标记为NA---------------------------------------------------------*
107. health\_data1[health\_data1=="10086"] <- NA
108. *##唯一性检查--------------------------------------------------------------*
109. id\_unique <- health\_data1[,1]*#唯一身份识别信息所在列*
110. chongfux <- id\_unique[duplicated(id\_unique)]*#所有重复*
111. health\_data1[which(id\_unique %in% chongfux),]*#所有重复记录展示*
112. chongfuy <- id\_unique[duplicated(health\_data1)]*#完全重复*
113. health\_data1[which(id\_unique %in% chongfuy),]*#完全重复记录展示*
114. bwqchongfu <- setdiff(chongfux,chongfuy)*#不完全重复*
115. health\_data1[which(id\_unique %in% bwqchongfu),]*#不完全重复记录展示*
116. zcfl <- length(chongfux)/length(id\_unique)*#总重复率*
117. zcfl
118. wqcfl <- length(chongfuy)/length(id\_unique)*#完全重复率*
119. wqcfl
120. bwqcfl <- length(bwqchongfu)/length(id\_unique)*#不完全重复率*
121. bwqcfl
122. origin\_len <- length(health\_data1[,1])
123. health\_data1 <- unique(health\_data1[!(id\_unique %in% bwqchongfu),])*#去除重复值（包括不完全）*
124. use\_record1 <- length(health\_data1[,1])/origin\_len*#可用值比例*
125. *##离群值检查--------------------------------------------------------------*
126. *#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*
127. *#health\_data1 <- read.xlsx(file.choose())#读取Excel数据:测试用*
128. *#health\_data1 <- as.data.frame(apply(health\_data1, 2, as.character)) ：测试用*
129. dcv <- c(5:12)*#待检测指标*
130. zoutlie <- list()
131. lqzbl <- c()*#离群值比例*
132. for (i in 1:length(dcv)) {
133. lie\_rank <- dcv[i]
134. jclz <- as.numeric(health\_data1[,lie\_rank])
135. qx <- quantile(jclz,na.rm = T)
136. IQR <- qx[4]-qx[2]
137. names(IQR) <- "IQR"
138. highlim <- qx[4]+1.5\*IQR
139. lowlim <- qx[2]-1.5\*IQR
140. outliem <- jclz[jclz>highlim | jclz<lowlim]*#四分位间距法离群值*
141. indm <- which(jclz>highlim | jclz<lowlim)*#离群值索引（四分位间距法）*
142. meand <- mean(jclz,na.rm = T)
143. sdd <- sd(jclz,na.rm = T)
144. minlim <- meand-1.96\*sdd
145. maxlim <- meand+1.96\*sdd
146. outlies <- jclz[jclz>maxlim | jclz<minlim]*#均数标准差法离群值*
147. inds <- which(jclz>maxlim | jclz<minlim)*#离群值索引（均数标准差法）*
148. zoutlie[[i]] <- jclz[intersect(indm,inds)]*#两种方法的交集（最终离群值）*
149. lqzbl[i] <- length(zoutlie[[i]])/length(jclz) *#离群值比例*
150. }
151. names(zoutlie) <- colnames(health\_data1)[dcv]
152. lqzbl\_df <- data.frame(colnames(health\_data1)[dcv],lqzbl)*#离群值比例列表*
153. zlqz <- mean(lqzbl)*#总离群值比例*
154. colnames(lqzbl\_df) <- c("指标","离群值比例")
155. lqzbl\_df*#离群值比例列表*
156. zlqz *#总离群值比例*
157. *#离群值标记--------------------------------------------------------------*
158. health\_data2 <- health\_data1
159. for (i in 1:length(dcv)) {
160. lie\_rank <- dcv[i]
161. health\_data2[,lie\_rank][health\_data2[,lie\_rank] %in% zoutlie[[i]]] <- "outlies"
162. print(length(health\_data2[,lie\_rank][is.na(health\_data2[,lie\_rank])])==length(zoutlie[[i]]))
163. }
164. hangx <- (which(health\_data2=="outlies")%%length(health\_data2[,1]))
165. hangx[hangx==0] <- length(health\_data2[,1])
166. health\_data2 <- health\_data2[-hangx,]*#删除离群值的数据集*
167. health\_data1[,dcv] <- apply(health\_data1[,dcv],2,as.numeric)
168. health\_data2[,dcv] <- apply(health\_data2[,dcv],2,as.numeric)
169. *#health\_data1为保留离群值数据；health\_data2为删除离群值数*
170. *###总体数据质量评价-----------------------------------------------------------------------------------------*
171. use\_record1  *#可用记录比例（保留离群值）*
172. use\_record2 <- length(health\_data2[,1])/origin\_len*##可用记录比例（删除离群值）*
173. *#保留离群值*
174. 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"))
175. parten1 <- md.pattern(health\_data1)
176. *#删除离群值*
177. 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"))
178. parten2 <- md.pattern(health\_data2)
179. *###---------------------------------------------------------------------------------------------------------*
180. *###数据处理-------------------------------------------------------------------------------------------------*
181. *#health\_data1 <- unique(health\_data1[!(health\_data1[,id] %in% bwqchongfu),]) #重复值处理:数据检查时已做*
182. dec\_health <- health\_data1[,-c(1,2)] *#需填补数据列（也可整体输入）*
183. iddx <- c(5:12)*#需转化成数字的列*
184. iddy <- c()*#需转化成因子的列*
185. dec\_health[,iddx] <- as.numeric(dec\_health[,iddx])
186. dec\_health[,iddy] <- as.factor(dec\_health[,iddy])*#等级变量需要额外设置leves参数*
187. *#data：包含缺失值的数据框或矩阵。缺失值被编码为 NA。*
188. *#m：多重插补法的数量，默认为 5。*
189. *#method：指定数据中每一列的输入方法。1）数值型数据适用 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。默认方法为 pmm 。*
190. *#maxit：迭代次数，默认为 5 。*
191. *#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。*
192. mt <- 5 *#插补次数*
193. tempData <- tempData <- mice(dec\_health,m=mt,maxit=mt0,seed=mt00)
194. *#寻找合适值*
195. densityplot(tempData)
196. densityplot(tempData,~`空腹血糖`+`高密度脂蛋白胆固醇` | .imp)
197. *#自行选择合适值填充-------------------------------*
198. id\_dt <- 1
199. dait <- dec\_health[,id\_dt]
200. dait <- complete(tempData,action = mt)[,id\_dt]
201. *#平均值填充(计量资料)-----------------------------*
202. ss <- 0
203. dfc <- c(3:10)#计量资料填补列
204. for(i in 1:mt){
205. ddaa <- complete(tempData,action = 1)
206. ss <- ss + ddaa[,dfc]
207. }
208. health\_data1x <- ss/mt *#平均值填充完需自行与原数据核对后考虑合并*
209. *#众数填补（计数资料）-----------------------------*
210. zhongshu <- c(1,2)*#计数资料填补列*
211. health\_data1y <- complete(tempData,action = 1)
212. sdt <- complete(tempData,action = 1)
213. for (i in 2:mt) {
214. sdt <- rbind(sdt,complete(tempData,action = i))
215. }
216. sdt$group <- rep(c(1:length(complete(tempData,action = 1)[,1])),time=mt)
217. for (i in 1:length(complete(tempData,action = 1)[,1])) {
218. lgde <- sdt[sdt$group==i,]
219. for (m in zhongshu) {
220. ifelse(as.integer(names(which.max(table(lgde[,1])))) == NA,
221. health\_data1y[i,m] <- names(which.max(table(lgde[,m]))),
222. health\_data1y[i,m] <- as.integer(names(which.max(table(lgde[,1])))))
223. }
224. }
225. save.image(file = "env\_data\_deal.RData") *#保存结果*
226. *###结果描述-----------------------------------------------------------------------------*
227. library(reshape2)
228. library(ggplot2)
229. library(ggridges)
230. library(hrbrthemes)
231. myda <- health\_data1*#数据清洗完成的数据集*
232. aggregate.data.frame(myda[,-c(1:4)],by=list(myda$性别,myda$吸烟情况),FUN = sd)*#分组描述*
233. *#myda$性别 <- factor(myda$性别,levels = c(1,2),labels = c("男","女"))#读取Excel文件需要，SPSS文件不需要*
234. *#myda$吸烟情况 <- factor(myda$吸烟情况,levels = c(0,1),labels = c("不吸烟","吸烟"))#同上*
235. mydalo <- melt(myda,id=c("身份证号码","性别","吸烟情况","年龄"))*#根据自己数据决定，需要了解reshape2包详细使用方法*
236. *#疾病诊断-----------------------------------------------------------------------------------------------------------*
237. diagn <- data.frame(ID = myda[,1])*#患病情况*
238. zj <- length(myda[,1])
239. diagn$sex <- myda$性别
240. diagn$age <- myda$年龄
241. diagn$agepe <- cut(myda$年龄,2)
242. diagn$smoke <- myda$吸烟情况
243. diagn$hb <- rep(1,time=zj) *#高血压*
244. diagn$hb[myda$收缩压>=130 | myda$舒张压>=90] <- 2
245. diagn$hf <- rep(1,time=zj) *#高血脂*
246. diagn$hf[myda$总胆固醇>=6.2 | myda$低密度脂蛋白>=4.1 | myda$甘油三酯>=2.3 | myda$高密度脂蛋白<1] <- 2
247. diagn$hg <- rep(1,time=zj) *#高血糖*
248. diagn$hg[myda$空腹血糖>=7] <- 2
249. diagn$BMI <- rep(1,time=zj) *#肥胖*
250. diagn$BMI[myda$BMI>=28] <- 2
251. *#以上可根据诊断标准变化自行更改*
252. colnames(diagn) <- c("身份证","性别","年龄","年龄段","吸烟情况","高血压","高血脂","糖尿病","肥胖")
253. diagn\_lon <- melt(diagn,id=c("身份证","性别","吸烟情况","年龄","年龄段"))
254. colnames(diagn\_lon) <- c("身份证","性别","吸烟情况","年龄","年龄段","疾病类型","数值")
255. huanbin <- diagn\_lon[diagn\_lon$数值==2,]
256. *#jib <- aggregate.data.frame(huanbin$疾病类型,by=list(huanbin$年龄段,huanbin$吸烟情况,huanbin$疾病类型,huanbin$性别),FUN = length)#分组描述*
257. *#zjib <- aggregate.data.frame(diagn\_lon$疾病类型,by=list(diagn\_lon$年龄段,diagn\_lon$吸烟情况,diagn\_lon$疾病类型,diagn\_lon$性别),FUN = length)#分组描述*
258. *#山峦图绘制----------------------------------------------------------------------------------------------*
259. pla1 <- ggplot(huanbin, aes(x = `年龄`, y = `疾病类型`,fill=`性别`)) +
260. geom\_density\_ridges(alpha = .7,
261. rel\_min\_height = 0.01,
262. scale = 1.2,
263. color = "black",
264. from = 30,
265. to =90,
266. linetype = 1,
267. lwd = 0.5) +
268. theme\_ridges(grid = T) +
269. theme\_ipsum()+xlab("年龄")+ylab(" ")+
270. labs(title = "各疾病年龄分布")
271. *#图表美化，选择性执行*
272. pla1 + theme(plot.subtitle = element\_text(family = "serif"),
273. plot.caption = element\_text(family = "mono"),
274. axis.title = element\_text(family = "serif"),
275. axis.text = element\_text(family = "mono"),
276. plot.title = element\_text(family = "serif"))
277. *#---------------------------------------------------------------------------------------------------------*
278. *#保存结果*
279. *#saveRDS(huanbin,"huanbin\_x.RDS")*
280. *#saveRDS(diagn,"diagn\_x.RDS")*
281. *#saveRDS(diagn\_lon,"diag\_lon.RDS")*
282. *#saveRDS(jib,"jib\_x.RDS")*
283. *#saveRDS(zjib,"zjib\_x.RDS")*
284. *#saveRDS(pla,"pla.RDS")*
285. *#saveRDS(pla1,"pla1.RDS")*
286. *#setwd("D:/Desktop/描述结果")*
287. *#jib <- readRDS("jib\_x.RDS")*
288. *#zjib <- readRDS("zjib\_x.RDS")*
289. preda <- left\_join(zjib,jib,by=c("Group.1","Group.2", "Group.3" ,"Group.4")) *#总结表(每种疾病患病率情况)*
290. preda[is.na(preda)] <- 0
291. preda$hbl <- preda$x.y/preda$x.x
292. saveRDS(preda,"preda.RDS")
293. z\_age <- dcast(preda,Group.1~Group.3,value.var = "x.x",sum)
294. h\_age <- dcast(preda,Group.1~Group.3,value.var = "x.y",sum)
295. l\_age <- cbind(h\_age[,1],h\_age[,-1]/z\_age[,-1])
296. colnames(l\_age) <- c( "分组","高血压","高血脂","糖尿病","肥胖")
297. z\_smoke <- dcast(preda,Group.2~Group.3,value.var = "x.x",sum)
298. h\_smoke <- dcast(preda,Group.2~Group.3,value.var = "x.y",sum)
299. l\_smoke <- cbind(h\_smoke[,1],h\_smoke[,-1]/z\_smoke[,-1])
300. colnames(l\_smoke) <- c( "分组","高血压","高血脂","糖尿病","肥胖")
301. z\_sex <- dcast(preda,Group.4~Group.3,value.var = "x.x",sum)
302. h\_sex <- dcast(preda,Group.4~Group.3,value.var = "x.y",sum)
303. l\_sex <- cbind(h\_sex[,1],h\_sex[,-1]/z\_sex[,-1])
304. colnames(l\_sex) <- c( "分组","高血压","高血脂","糖尿病","肥胖")
305. saveRDS(z\_age,"z\_age.RDS")
306. saveRDS(h\_age,"h\_age.RDS")
307. saveRDS(l\_age,"l\_age.RDS")*#患病率（分组）可根据实际情况扩充*
308. saveRDS(z\_smoke,"z\_smoke.RDS")
309. saveRDS(h\_smoke,"h\_smoke.RDS")
310. saveRDS(l\_smoke,"l\_smoke.RDS")
311. saveRDS(z\_sex,"z\_sex.RDS")
312. saveRDS(h\_sex,"h\_sex.RDS")
313. saveRDS(l\_sex,"l\_sex.RDS")
314. *#雷达图和柱状图------------------------------------------------------------------------------*
315. devtools::install\_github("ricardo-bion/ggradar", dependencies=TRUE)
316. library(ggradar)
317. library(reshape2)
318. l\_age <- readRDS("D:/Desktop/描述结果/l\_age.RDS")
319. l\_age <- l\_age\*100
320. l\_age[,1] <- c("中年人","老年人")
321. lad <- ggradar(
322. l\_age,
323. values.radar = c("0", "50", "100"),
324. grid.min = 0, grid.mid = 50, grid.max = 100,
325. group.line.width = 1,
326. group.point.size = 3,
327. *#group.colours = c("#ffd200", "#304156"),*
328. background.circle.colour = "white",
329. gridline.mid.colour = "grey",
330. legend.position = "bottom"
331. )
332. l\_age\_long <- melt(l\_age)
333. bar <- ggplot(l\_age\_long, aes(fill=`分组`, x=variable, y=value)) +
334. geom\_bar(position="dodge", stat="identity")
335. library(ggpubr)
336. zhh <- ggarrange(lad,bar)
337. *#--------------------------------------------------------------------------------------------*
338. *#桑基图（不建议尝试！！！）------------------------------------------------------------------*
339. diaa <- readRDS("D:/Desktop/描述结果/diagn\_x.RDS")
340. View(diaa)
341. write.xlsx(diaa,"diaa1.xlsx")
342. nodess <- read.xlsx("diaa1.xlsx",sheet = 2)
343. head(raw\_link)
344. cnaa <- colnames(raw\_link)
345. link\_67 <- dcast(raw\_link,raw\_link[,6]~raw\_link[,7],value.var = cnaa[6])
346. link\_67 <- melt(link\_67,id="raw\_link[, 6]")
347. colnames(link\_67) <- c("source","target","value")
348. link\_67
349. link\_final <- rbind(link\_12,link\_23,link\_34,link\_45,link\_56,link\_67)
350. write.xlsx(link\_final,"link\_final.xlsx")
351. link\_final <- read.xlsx("link\_final.xlsx")
352. link\_final$target <- as.integer(link\_final$target)
353. link\_final$source <- link\_final$source-1
354. link\_final$target <- link\_final$target-1
355. link\_final$type <- c(1:24)
356. *#length(raw\_link$性别[raw\_link$性别==1])*
357. *#sta <- data.frame(source = 0,target = 1,value = 1852)*
358. *#link\_final <- rbind(sta,link\_final)*
359. *# 节点分组的情况下：*
360. sankeyNetwork(Links = link\_final, Nodes =nodess, Source = 'source',
361. Target = 'target', Value = 'value', NodeID = 'node',
362. units = 'Twh', fontSize = 28, nodeWidth = 45,nodePadding = 30)
363. *# 节点分组的情况下：*
364. sankeyNetwork(Links = energy$links, Nodes =energy$nodes, Source = 'source',
365. Target = 'target', Value = 'value', NodeID = 'name',
366. units = 'TWh', fontSize = 12, nodeWidth = 30)
367. link\_final$type <- sub(' .\*','',
368. nodess[link\_final$source + 1, 'node'])
369. save.image(file = "env\_data\_describe.RData") #保存结果

特别注意：

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 。