内在能力建模

Ruifu Kang & Yirou Niu

2024-12-26

# 赋分规则

1. 运动（Short Physical Performance Battery Test）：2.5米步行速度≥ 1米/秒得1分；重复坐下5次≤12秒得1分；平衡：3个10秒完成2个及以上得1分。满分3分
2. 认知：情景记忆（基于延迟回忆得分）：0-10分；减7测试：5分；日期、月份、年份和季节：4分；绘画：1分。满分20分 进一步：范围18-20：得3分；范围14-17：得2分；范围7-13：得1分；范围0-6：得0分。满分3分
3. CES-D评分为0到9分：得1分；总睡眠时间在5到10.5小时之间，得1分；睡眠质量：一周内睡眠不安的频率0到2天之间得1分。满分3分
4. 感官：听力回答非常好、好、一般：得1分；视力回答非常好、好、一般：得1分（远和近两项）。满分3分
5. 活力：握力：男性≥ 35kg得1分，女性≥ 25kg得1分；FEV：男性≥ 400得1分，女性≥ 290得1分;血红蛋白先看是否可获取。 # 数据准备 ## 读取并合并数据 - 读取所有老年人的个人数据（暂不考虑家庭数据和整体数据权重） - 将所有数据以ID为键值进行匹配合并到一个数据框中

# 安装并加载所需要的包  
if (!requireNamespace("haven", quietly = TRUE)) {  
 install.packages("haven")  
}  
if (!requireNamespace("dplyr", quietly = TRUE)) {  
 install.packages("dplyr")  
}  
library(haven)  
library(dplyr)

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':  
  
 filter, lag

The following objects are masked from 'package:base':  
  
 intersect, setdiff, setequal, union

# 读取并合并数据  
files <- list.files("data\_raw/self",  
 pattern = "\\.dta$",  
 full.names = TRUE  
)  
data\_list <- lapply(files, read\_dta)  
my\_data\_raw <- Reduce(function(x, y) full\_join(x, y, by = "ID"), data\_list)  
print(my\_data\_raw)

# A tibble: 21,805 × 7,573  
 ID householdID.x communityID.x pa001 pa002 qa001s1 qa001s2 qa001s3  
 <chr> <chr> <chr> <dbl+l> <dbl+l> <dbl+l> <dbl+l> <dbl+l>  
 1 09400410… 0940041030 0940041 5 [5 N… 1 [1 Y… NA NA NA   
 2 09400411… 0940041100 0940041 5 [5 N… 1 [1 Y… NA NA NA   
 3 09400410… 0940041080 0940041 5 [5 N… 1 [1 Y… NA NA NA   
 4 09400411… 0940041120 0940041 5 [5 N… 1 [1 Y… NA NA NA   
 5 09400411… 0940041120 0940041 5 [5 N… 1 [1 Y… NA NA NA   
 6 09400411… 0940041140 0940041 5 [5 N… 1 [1 Y… NA NA NA   
 7 09400411… 0940041190 0940041 5 [5 N… 1 [1 Y… NA NA NA   
 8 09400411… 0940041170 0940041 5 [5 N… 1 [1 Y… NA NA NA   
 9 09400411… 0940041170 0940041 5 [5 N… 1 [1 Y… NA NA NA   
10 09400431… 0940043100 0940043 5 [5 N… 1 [1 Y… NA NA NA   
# ℹ 21,795 more rows  
# ℹ 7,565 more variables: qa001s4 <dbl+lbl>, qa001s5 <dbl+lbl>,  
# qa001s6 <dbl+lbl>, qa001s7 <dbl+lbl>, qa001s8 <dbl+lbl>,  
# qa001s97 <dbl+lbl>, qa002 <chr>, qa002\_1 <chr>, qa003 <dbl>, qa004 <dbl>,  
# qa005 <dbl>, qa006 <chr>, qa006\_1 <chr>, qa007 <dbl>, qa008 <dbl>,  
# qa009 <dbl>, qa010 <chr>, qa010\_1 <chr>, qa011 <dbl>, qa012 <dbl>,  
# qa013 <dbl>, qa014 <dbl+lbl>, qa015 <dbl+lbl>, qa016 <dbl+lbl>, …

## 变量重命名

* 将所有的变量重命名为可读性更好的名称
* 重命名后的变量名保存在一个新的数据框中

# 变量重命名  
my\_data\_rename <- my\_data\_raw %>%  
 rename(  
 # 基本信息重命名  
 id\_birth\_year = ba004\_w3\_1, # ID出生年份  
 actual\_birth\_year = ba002\_1, # 实际出生年份   
 death = died, # 是否死亡  
 # 内在能力运动维度重命名  
 walk\_time\_first = qg002, # 步行测试时间1  
 walk\_time\_second = qg003, # 步行测试时间2  
 stand\_test\_semi\_tandem = qd002, # 双脚半前后站立测试  
 stand\_test\_tandem = qe002, # 双脚一条线站立测试  
 stand\_test\_feet\_together = qf002, # 双脚并拢站立测试  
 sit\_stand\_test\_time = qh003, # 完成五次坐下起来测试的时间  
  
 # 内在能力认知维度重命名  
 subtraction\_test\_1 = dc019, # 减法测试1  
 subtraction\_test\_2 = dc020, # 减法测试2  
 subtraction\_test\_3 = dc021, # 减法测试3  
 subtraction\_test\_4 = dc022, # 减法测试4  
 subtraction\_test\_5 = dc023, # 减法测试5  
 drawing\_test = dc025, # 画图测试  
 recall\_year = dc001s1, # 回忆年份  
 recall\_month = dc001s2, # 回忆月份  
 recall\_day = dc001s3, # 回忆日期  
 recall\_weekday = dc002, # 回忆星期  
 recall\_season = dc003, # 回忆季节是否正确  
 recall\_word\_1 = dc006s1, # 回忆词汇1  
 recall\_word\_2 = dc006s2, # 回忆词汇2  
 recall\_word\_3 = dc006s3, # 回忆词汇3  
 recall\_word\_4 = dc006s4, # 回忆词汇4  
 recall\_word\_5 = dc006s5, # 回忆词汇5  
 recall\_word\_6 = dc006s6, # 回忆词汇6  
 recall\_word\_7 = dc006s7, # 回忆词汇7  
 recall\_word\_8 = dc006s8, # 回忆词汇8  
 recall\_word\_9 = dc006s9, # 回忆词汇9  
 recall\_word\_10 = dc006s10, # 回忆词汇10  
 recall\_none = dc006s11, # 是否一个都没回忆起来  
 recall\_refused = dc006s12, # 是否拒绝回忆  
 # 内在能力心理维度重命名  
 sleep\_time = da049, # 睡眠时间  
 depression\_scale\_1 = dc009, # 心理量表问题1  
 depression\_scale\_2 = dc010, # 心理量表问题2  
 depression\_scale\_3 = dc011, # 心理量表问题3  
 depression\_scale\_4 = dc012, # 心理量表问题4  
 depression\_scale\_5 = dc013, # 心理量表问题5  
 depression\_scale\_6 = dc014, # 心理量表问题6  
 poor\_sleep\_frequency = dc015, # 睡眠不佳频率  
 depression\_scale\_7 = dc016, # 心理量表问题7  
 depression\_scale\_8 = dc017, # 心理量表问题8  
 depression\_scale\_9 = dc018, # 心理量表问题9  
 # 内在能力感官维度重命名  
 far\_vision = da033, # 看远处视力情况  
 near\_vision = da034, # 看近处视力情况  
 hearing\_status = da039, # 听力情况  
 # 内在能力活力维度重命名  
 left\_hand\_grip\_1 = qc003, # 第一次左手握力测量  
 right\_hand\_grip\_1 = qc004, # 第一次右手握力测量  
 left\_hand\_grip\_2 = qc005, # 第二次左手握力测量  
 right\_hand\_grip\_2 = qc006, # 第二次右手握力测量  
 breath\_test\_1 = qb002, # 第一次呼吸功能测定  
 breath\_test\_2 = qb003, # 第二次呼吸功能测定  
 breath\_test\_3 = qb004 # 第三次呼吸功能测定  
 )

## 年龄数据清洗

* 计算年龄
* 按照年龄筛选出60岁及以上的老年人

# 去除实际出生年份和ID出生年份同时缺失的数据  
my\_data\_year\_na <- my\_data\_rename %>%  
 filter(!(is.na(actual\_birth\_year) & is.na(id\_birth\_year)))  
  
# 计算年龄值  
my\_data\_age <- my\_data\_year\_na %>%  
 mutate(age = 2015 - coalesce(actual\_birth\_year, id\_birth\_year))  
  
# 年龄清洗  
my\_data\_elder <- my\_data\_age %>%  
 filter(age >= 60)

## 内在能力分数初步计算

# 处理步行测试时间  
my\_data\_final <- my\_data\_elder %>%  
 mutate(  
 final\_walk\_time = case\_when(  
 is.na(walk\_time\_first) & is.na(walk\_time\_second) ~ NA\_real\_,  
 walk\_time\_first %in% c(NA, 993, 999) & walk\_time\_second %in% c(NA, 993, 999) ~ NA\_real\_,  
 walk\_time\_first %in% c(NA, 993, 999) ~ walk\_time\_second,  
 walk\_time\_second %in% c(NA, 993, 999) ~ walk\_time\_first,  
 TRUE ~ (walk\_time\_first + walk\_time\_second) / 2  
 )  
 )  
  
# 处理左手握力  
my\_data\_final <- my\_data\_final %>%  
 mutate(  
 final\_left\_hand\_grip = case\_when(  
 is.na(left\_hand\_grip\_1) & is.na(left\_hand\_grip\_2) ~ NA\_real\_,  
 left\_hand\_grip\_1 %in% c(NA, 993, 999) & left\_hand\_grip\_2 %in% c(NA, 993, 999) ~ NA\_real\_,  
 left\_hand\_grip\_1 %in% c(NA, 993, 999) ~ left\_hand\_grip\_2,  
 left\_hand\_grip\_2 %in% c(NA, 993, 999) ~ left\_hand\_grip\_1,  
 TRUE ~ (left\_hand\_grip\_1 + left\_hand\_grip\_2) / 2  
 )  
 )  
  
# 处理右手握力  
my\_data\_final <- my\_data\_final %>%  
 mutate(  
 final\_right\_hand\_grip = case\_when(  
 is.na(right\_hand\_grip\_1) & is.na(right\_hand\_grip\_2) ~ NA\_real\_,  
 right\_hand\_grip\_1 %in% c(NA, 993, 999) & right\_hand\_grip\_2 %in% c(NA, 993, 999) ~ NA\_real\_,  
 right\_hand\_grip\_1 %in% c(NA, 993, 999) ~ right\_hand\_grip\_2,  
 right\_hand\_grip\_2 %in% c(NA, 993, 999) ~ right\_hand\_grip\_1,  
 TRUE ~ (right\_hand\_grip\_1 + right\_hand\_grip\_2) / 2  
 )  
 )  
  
# 处理呼吸功能  
my\_data\_final <- my\_data\_final %>%  
 mutate(  
 final\_breath\_test = case\_when(  
 is.na(breath\_test\_1) & is.na(breath\_test\_2) & is.na(breath\_test\_3) ~ NA\_real\_,  
 breath\_test\_1 %in% c(NA, 993, 999) & breath\_test\_2 %in% c(NA, 993, 999) & breath\_test\_3 %in% c(NA, 993, 999) ~ NA\_real\_,  
 breath\_test\_1 %in% c(NA, 993, 999) & breath\_test\_2 %in% c(NA, 993, 999) ~ breath\_test\_3,  
 breath\_test\_1 %in% c(NA, 993, 999) & breath\_test\_3 %in% c(NA, 993, 999) ~ breath\_test\_2,  
 breath\_test\_2 %in% c(NA, 993, 999) & breath\_test\_3 %in% c(NA, 993, 999) ~ breath\_test\_1,  
 breath\_test\_1 %in% c(NA, 993, 999) ~ (breath\_test\_2 + breath\_test\_3) / 2,  
 breath\_test\_2 %in% c(NA, 993, 999) ~ (breath\_test\_1 + breath\_test\_3) / 2,  
 breath\_test\_3 %in% c(NA, 993, 999) ~ (breath\_test\_1 + breath\_test\_2) / 2,  
 TRUE ~ (breath\_test\_1 + breath\_test\_2 + breath\_test\_3) / 3  
 )  
 )  
   
# 计算最终回忆分数  
my\_data\_final <- my\_data\_final %>%  
 mutate(  
 final\_recall\_score = case\_when(  
 !is.na(recall\_refused) ~ 0,  
 !is.na(recall\_none) ~ NA\_real\_,  
 TRUE ~ rowSums(select(., starts\_with("recall\_word\_")), na.rm = TRUE)  
 )  
 )  
  
# 计算抑郁总分  
my\_data\_final <- my\_data\_final %>%  
 mutate(  
 total\_depression\_score = if\_else(  
 rowSums(select(., starts\_with("depression\_scale\_")), na.rm = TRUE) == 0 &   
 rowSums(is.na(select(., starts\_with("depression\_scale\_")))) == 9,  
 NA\_real\_,  
 rowSums(select(., starts\_with("depression\_scale\_")), na.rm = TRUE) +   
 rowSums(is.na(select(., starts\_with("depression\_scale\_"))))  
 )  
 )  
  
# 计算减法测试总分  
my\_data\_final <- my\_data\_final %>%  
 mutate(  
 total\_subtraction\_score = if\_else(  
 rowSums(is.na(select(., starts\_with("subtraction\_test\_")))) == 5,  
 NA\_real\_,  
 rowSums(!is.na(select(., starts\_with("subtraction\_test\_"))))  
 )  
 )

## 内在能力清洗

* 剔除内在能力指标缺失20%以上的个体

# 分出来前100个个体数据  
my\_data\_final\_top100 <- my\_data\_final %>%  
 slice\_head(n = 100)  
print(my\_data\_final\_top100)

# A tibble: 100 × 7,581  
 ID householdID.x communityID.x pa001 pa002 qa001s1 qa001s2 qa001s3  
 <chr> <chr> <chr> <dbl+l> <dbl+l> <dbl+l> <dbl+l> <dbl+l>  
 1 09400410… 0940041030 0940041 5 [5 N… 1 [1 Y… NA NA NA   
 2 09400411… 0940041100 0940041 5 [5 N… 1 [1 Y… NA NA NA   
 3 09400410… 0940041080 0940041 5 [5 N… 1 [1 Y… NA NA NA   
 4 09400411… 0940041120 0940041 5 [5 N… 1 [1 Y… NA NA NA   
 5 09400411… 0940041120 0940041 5 [5 N… 1 [1 Y… NA NA NA   
 6 09400411… 0940041190 0940041 5 [5 N… 1 [1 Y… NA NA NA   
 7 09400411… 0940041170 0940041 5 [5 N… 1 [1 Y… NA NA NA   
 8 09400411… 0940041170 0940041 5 [5 N… 1 [1 Y… NA NA NA   
 9 09400431… 0940043100 0940043 5 [5 N… 1 [1 Y… NA NA NA   
10 09400431… 0940043100 0940043 5 [5 N… 1 [1 Y… NA NA NA   
# ℹ 90 more rows  
# ℹ 7,573 more variables: qa001s4 <dbl+lbl>, qa001s5 <dbl+lbl>,  
# qa001s6 <dbl+lbl>, qa001s7 <dbl+lbl>, qa001s8 <dbl+lbl>,  
# qa001s97 <dbl+lbl>, qa002 <chr>, qa002\_1 <chr>, qa003 <dbl>, qa004 <dbl>,  
# qa005 <dbl>, qa006 <chr>, qa006\_1 <chr>, qa007 <dbl>, qa008 <dbl>,  
# qa009 <dbl>, qa010 <chr>, qa010\_1 <chr>, qa011 <dbl>, qa012 <dbl>,  
# qa013 <dbl>, qa014 <dbl+lbl>, qa015 <dbl+lbl>, qa016 <dbl+lbl>, …

# 导出前100个个体数据为CSV文件  
write.csv(my\_data\_final\_top100, "top100\_individuals.csv", row.names = FALSE)  
  
# 剔除除了ID和这些内在能力指标以外的变量保存到新数据里  
required\_columns <- c(  
 "ID", "final\_walk\_time", "stand\_test\_semi\_tandem", "stand\_test\_tandem", "stand\_test\_feet\_together", "sit\_stand\_test\_time",  
 "final\_recall\_score",   
 "sleep\_time", "poor\_sleep\_frequency", "total\_depression\_score",   
 "far\_vision", "near\_vision", "hearing\_status",  
 "final\_left\_hand\_grip", "final\_right\_hand\_grip", "final\_breath\_test"  
)  
  
my\_data\_final\_filtered <- my\_data\_final %>%  
 select(all\_of(required\_columns))  
  
print(my\_data\_final\_filtered)

# A tibble: 9,982 × 16  
 ID final\_walk\_time stand\_test\_semi\_tandem stand\_test\_tandem  
 <chr> <dbl> <dbl+lbl> <dbl+lbl>   
 1 094004103001 4.18 1 [1 Yes] 1 [1 Yes]   
 2 094004110001 3.22 1 [1 Yes] 1 [1 Yes]   
 3 094004108001 2.54 1 [1 Yes] 1 [1 Yes]   
 4 094004112001 3.39 1 [1 Yes] 1 [1 Yes]   
 5 094004112002 3.44 1 [1 Yes] 1 [1 Yes]   
 6 094004119002 4.38 1 [1 Yes] 1 [1 Yes]   
 7 094004117002 3.29 1 [1 Yes] 1 [1 Yes]   
 8 094004117001 3.20 1 [1 Yes] 5 [5 No]   
 9 094004310002 10.4 1 [1 Yes] 1 [1 Yes]   
10 094004310001 9.60 1 [1 Yes] 5 [5 No]   
# ℹ 9,972 more rows  
# ℹ 12 more variables: stand\_test\_feet\_together <dbl+lbl>,  
# sit\_stand\_test\_time <dbl>, final\_recall\_score <dbl>, sleep\_time <dbl>,  
# poor\_sleep\_frequency <dbl+lbl>, total\_depression\_score <dbl>,  
# far\_vision <dbl+lbl>, near\_vision <dbl+lbl>, hearing\_status <dbl+lbl>,  
# final\_left\_hand\_grip <dbl>, final\_right\_hand\_grip <dbl>,  
# final\_breath\_test <dbl>

# 筛选出所有内在能力指标非NA值大于80%的观察值数量  
threshold <- 0.8  
non\_na\_counts <- my\_data\_final\_filtered %>%  
 mutate(non\_na\_count = rowSums(!is.na(select(., -ID)))) %>%  
 filter(non\_na\_count / (ncol(my\_data\_final\_filtered) - 1) > threshold) %>%  
 nrow()  
# 输出非NA值大于80%的观察值数量（带文字）  
print(paste("覆盖内在能力80%以上的个体数量:", non\_na\_counts))

[1] "覆盖内在能力80%以上的个体数量: 7323"

# 将大于80%的数据放到一个新的数据框  
my\_data\_above\_80 <- my\_data\_final\_filtered %>%  
 mutate(non\_na\_count = rowSums(!is.na(select(., -ID)))) %>%  
 filter(non\_na\_count / (ncol(my\_data\_final\_filtered) - 1) > threshold) %>%  
 select(-non\_na\_count)  
  
# 打印新的数据框  
print(my\_data\_above\_80)

# A tibble: 7,323 × 16  
 ID final\_walk\_time stand\_test\_semi\_tandem stand\_test\_tandem  
 <chr> <dbl> <dbl+lbl> <dbl+lbl>   
 1 094004103001 4.18 1 [1 Yes] 1 [1 Yes]   
 2 094004110001 3.22 1 [1 Yes] 1 [1 Yes]   
 3 094004112001 3.39 1 [1 Yes] 1 [1 Yes]   
 4 094004119002 4.38 1 [1 Yes] 1 [1 Yes]   
 5 094004117002 3.29 1 [1 Yes] 1 [1 Yes]   
 6 094004117001 3.20 1 [1 Yes] 5 [5 No]   
 7 094004310002 10.4 1 [1 Yes] 1 [1 Yes]   
 8 094004310001 9.60 1 [1 Yes] 5 [5 No]   
 9 094004317002 3.20 1 [1 Yes] 5 [5 No]   
10 094004317001 3.86 1 [1 Yes] 5 [5 No]   
# ℹ 7,313 more rows  
# ℹ 12 more variables: stand\_test\_feet\_together <dbl+lbl>,  
# sit\_stand\_test\_time <dbl>, final\_recall\_score <dbl>, sleep\_time <dbl>,  
# poor\_sleep\_frequency <dbl+lbl>, total\_depression\_score <dbl>,  
# far\_vision <dbl+lbl>, near\_vision <dbl+lbl>, hearing\_status <dbl+lbl>,  
# final\_left\_hand\_grip <dbl>, final\_right\_hand\_grip <dbl>,  
# final\_breath\_test <dbl>

# 用筛选后的数据筛选my\_data\_final，用ID号匹配  
filtered\_ids <- my\_data\_above\_80 %>% select(ID)  
my\_data\_final\_matched <- my\_data\_final %>% semi\_join(filtered\_ids, by = "ID")  
  
# 打印匹配后的数据  
print(my\_data\_final\_matched)

# A tibble: 7,323 × 7,581  
 ID householdID.x communityID.x pa001 pa002 qa001s1 qa001s2 qa001s3  
 <chr> <chr> <chr> <dbl+l> <dbl+l> <dbl+l> <dbl+l> <dbl+l>  
 1 09400410… 0940041030 0940041 5 [5 N… 1 [1 Y… NA NA NA   
 2 09400411… 0940041100 0940041 5 [5 N… 1 [1 Y… NA NA NA   
 3 09400411… 0940041120 0940041 5 [5 N… 1 [1 Y… NA NA NA   
 4 09400411… 0940041190 0940041 5 [5 N… 1 [1 Y… NA NA NA   
 5 09400411… 0940041170 0940041 5 [5 N… 1 [1 Y… NA NA NA   
 6 09400411… 0940041170 0940041 5 [5 N… 1 [1 Y… NA NA NA   
 7 09400431… 0940043100 0940043 5 [5 N… 1 [1 Y… NA NA NA   
 8 09400431… 0940043100 0940043 5 [5 N… 1 [1 Y… NA NA NA   
 9 09400431… 0940043170 0940043 5 [5 N… 1 [1 Y… NA NA NA   
10 09400431… 0940043170 0940043 5 [5 N… 1 [1 Y… NA NA NA   
# ℹ 7,313 more rows  
# ℹ 7,573 more variables: qa001s4 <dbl+lbl>, qa001s5 <dbl+lbl>,  
# qa001s6 <dbl+lbl>, qa001s7 <dbl+lbl>, qa001s8 <dbl+lbl>,  
# qa001s97 <dbl+lbl>, qa002 <chr>, qa002\_1 <chr>, qa003 <dbl>, qa004 <dbl>,  
# qa005 <dbl>, qa006 <chr>, qa006\_1 <chr>, qa007 <dbl>, qa008 <dbl>,  
# qa009 <dbl>, qa010 <chr>, qa010\_1 <chr>, qa011 <dbl>, qa012 <dbl>,  
# qa013 <dbl>, qa014 <dbl+lbl>, qa015 <dbl+lbl>, qa016 <dbl+lbl>, …

# 导出匹配后的数据为CSV文件  
write.csv(my\_data\_final\_matched, "matched\_data.csv", row.names = FALSE)

# 内在能力维度分数计算

## 计算认知维度

## 计算活力维度

## 计算心理维度

## 计算感官维度

## 总分计算

# 模型建立

# 模型验证