main

2023-12-17

library(magrittr)  
library(dplyr)

## Warning: package 'dplyr' was built under R version 4.3.2

##   
## 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

library(broom)

## Warning: package 'broom' was built under R version 4.3.2

library(effectsize)

## Warning: package 'effectsize' was built under R version 4.3.2

df = readxl::read\_xlsx("./poverty\_brazil.xlsx")

## Warning: Expecting numeric in F6988 / R6988C6: got 'NA'

# Discard singular entries: NA ethnicity and ill-defined work.  
  
df %<>% na.omit() %>% filter(work != 12)  
  
df1 = df %>% mutate(across(-c(2), .fns = factor),  
 gender = factor(ifelse(woman == "0", "F","M")),  
 education = factor(case\_when(  
 education == 1 ~ "1No education",  
 education == 2 ~ "2Incomplete primary",  
 education == 3 ~ "3Complete primary",  
 education == 4 ~ "4Incomplete secondary",  
 education == 5 ~ "5Complete secondary",  
 education == 6 ~ "6Incomplete higher",  
 education == 7 ~ "7Complete higher")  
 ),  
 work = factor(case\_when(  
 work == 1 ~ "Agriculture/aquaculture",  
 work == 2 ~ "Industry",  
 work == 3 ~ "Construction",  
 work == 4 ~ "Auto repair",  
 work == 5 ~ "Logistics",   
 work == 6 ~ "Hospitality",   
 work == 7 ~ "White collar business",  
 work == 8 ~ "Public administration",  
 work == 9 ~ "Education, health/social service",  
 work == 10 ~ "Other",  
 work == 11 ~ "Home services",  
 work == 12 ~ "Ill-defined")  
 ),  
 metropolitan\_area = factor(  
 ifelse(metropolitan\_area == 0, "No", "Yes")  
 ),  
 ethnicity = factor(  
 ifelse(non\_white == 0, "white", "non-white")  
 ),  
 urban = factor(  
 ifelse(urban == 0, "No", "Yes")  
 ),  
 work\_permit = factor(case\_when(  
 work\_permit == 0 ~ "No permit",  
 work\_permit == 1 ~ "Permit",  
 work\_permit == 2 ~ "Other situation")  
 ),  
 poverty = factor(  
 ifelse(poverty == 0, 1, 0))) %>%  
 select(-c(woman, non\_white))

df1$work %<>% relevel(ref = "Hospitality")  
df1$metropolitan\_area %<>% relevel(ref = "Yes")  
df1$gender %<>% relevel(ref = "M")  
  
# Create model function.  
model = function(link\_item){  
 glm(poverty ~ age + education + work + metropolitan\_area + urban + work\_permit + gender + ethnicity,data = df1, family=binomial(link=link\_item))  
}

logit = model("logit")   
logit\_output = logit %>% tidy %>% select(-c(std.error, statistic))  
  
logit\_output

## # A tibble: 24 × 3  
## term estimate p.value  
## <chr> <dbl> <dbl>  
## 1 (Intercept) -2.73 4.18e-57  
## 2 age 0.0206 2.91e-34  
## 3 education2Incomplete primary 0.736 1.08e-10  
## 4 education3Complete primary 1.13 2.93e-19  
## 5 education4Incomplete secondary 1.15 5.62e-18  
## 6 education5Complete secondary 1.47 2.46e-35  
## 7 education6Incomplete higher 2.02 5.68e-41  
## 8 education7Complete higher 2.78 7.02e-90  
## 9 workAgriculture/aquaculture 0.0329 7.37e- 1  
## 10 workAuto repair 0.233 1.02e- 2  
## # ℹ 14 more rows

# Respectively: degrees of freedom is number of coefficients - 1, parameters is number of coefficients - 1, and n is number of rows in the original data - 1.  
# Since the models are identical except for their predictors' values, this need only run once.  
deg\_f = as.integer(length(logit$coefficients)) - 1   
p = as.integer(length(logit$coefficients))  
n = nrow(df) - 1  
  
# The null model has an identical logl-likelihood value regardless of the link function.  
null\_loglike = logLik(glm(poverty ~ 1, data=df1, family=binomial(link=logit)))  
  
# Create empty data frame and set column and row names.  
comparison = data.frame(matrix(ncol = 3, nrow = 3))  
models = c("Logit", "Probit", "Cloglog")  
stats = c("AIC","AICC","BIC")  
names(comparison) = models  
row.names(comparison) = stats  
# Discard.  
rm(stats, models)

# Calculate AICC.  
AICC = -2\*logLik(logit)+2\*p\*n/(n-p-1)  
  
comparison['Logit'] = as.numeric(c(logit$aic,  
 AICC,  
 BIC(logit)))  
  
# Deviance test and print.  
deviance = -2\*(null\_loglike-logLik(logit))  
deviance\_pval = pchisq(deviance,deg\_f,lower.tail = FALSE)  
  
cat(sprintf("The deviance test value is %s. \n", deviance))

## The deviance test value is 4212.46682784777.

cat(sprintf("The deviance test p-value is %s.", deviance\_pval))

## The deviance test p-value is 0.

probit = model("probit")   
probit\_output = probit %>% tidy %>% select(-c(std.error))  
  
probit\_output

## # A tibble: 24 × 4  
## term estimate statistic p.value  
## <chr> <dbl> <dbl> <dbl>  
## 1 (Intercept) -1.60 -15.8 1.60e-56  
## 2 age 0.0118 12.3 1.29e-34  
## 3 education2Incomplete primary 0.450 6.54 6.18e-11  
## 4 education3Complete primary 0.689 9.08 1.12e-19  
## 5 education4Incomplete secondary 0.694 8.70 3.24e-18  
## 6 education5Complete secondary 0.884 12.4 2.86e-35  
## 7 education6Incomplete higher 1.19 13.7 1.51e-42  
## 8 education7Complete higher 1.55 19.7 1.50e-86  
## 9 workAgriculture/aquaculture 0.00566 0.0978 9.22e- 1  
## 10 workAuto repair 0.132 2.49 1.26e- 2  
## # ℹ 14 more rows

# Calculate AICC.  
AICC = -2\*logLik(probit)+2\*p\*n/(n-p-1)  
  
comparison['Probit'] = as.numeric(c(probit$aic,  
 AICC,  
 BIC(probit)))  
  
# Deviance test and print.  
deviance = -2\*(null\_loglike-logLik(probit))  
deviance\_pval = pchisq(deviance,deg\_f,lower.tail = FALSE)  
  
cat(sprintf("The deviance test value is %s. \n", deviance))

## The deviance test value is 4190.42044779456.

cat(sprintf("The deviance test p-value is %s.", deviance\_pval))

## The deviance test p-value is 0.

df1$work %<>% relevel(ref = "Agriculture/aquaculture")  
  
cloglog = model("cloglog")   
cloglog\_output = cloglog %>% tidy %>% select(-c(std.error))  
  
cloglog\_output

## # A tibble: 24 × 4  
## term estimate statistic p.value  
## <chr> <dbl> <dbl> <dbl>  
## 1 (Intercept) -1.89 -19.0 1.62e-80  
## 2 age 0.0103 11.9 1.73e-32  
## 3 education2Incomplete primary 0.509 6.20 5.54e-10  
## 4 education3Complete primary 0.736 8.46 2.79e-17  
## 5 education4Incomplete secondary 0.736 8.14 3.98e-16  
## 6 education5Complete secondary 0.908 10.9 9.53e-28  
## 7 education6Incomplete higher 1.16 12.6 1.74e-36  
## 8 education7Complete higher 1.41 16.4 2.61e-60  
## 9 workHospitality 0.0185 0.329 7.42e- 1  
## 10 workAuto repair 0.138 3.38 7.23e- 4  
## # ℹ 14 more rows

# Calculate AICC.  
AICC = -2\*logLik(cloglog)+2\*p\*n/(n-p-1)  
  
comparison['Cloglog'] = as.numeric(c(cloglog$aic,  
 AICC,  
 BIC(cloglog)))  
  
# Deviance test and print.  
deviance = -2\*(null\_loglike-logLik(cloglog))  
deviance\_pval = pchisq(deviance,deg\_f,lower.tail = FALSE)  
  
cat(sprintf("The deviance test value is %s. \n", deviance))

## The deviance test value is 4095.17681937631.

cat(sprintf("The deviance test p-value is %s.", deviance\_pval))

## The deviance test p-value is 0.

# Summon comparison dataframe.  
comparison

## Logit Probit Cloglog  
## AIC 17959.98 17982.02 18077.27  
## AICC 17960.04 17982.08 18077.33  
## BIC 18150.54 18172.59 18267.83

# Call the one that has the lowest AIC, AICc, and BIC.  
names(comparison)[apply(comparison, 1, which.min)]

## [1] "Logit" "Logit" "Logit"

logit\_output %>% arrange(desc(estimate))

## # A tibble: 24 × 3  
## term estimate p.value  
## <chr> <dbl> <dbl>  
## 1 education7Complete higher 2.78 7.02e- 90  
## 2 education6Incomplete higher 2.02 5.68e- 41  
## 3 education5Complete secondary 1.47 2.46e- 35  
## 4 workPublic administration 1.29 5.14e- 23  
## 5 work\_permitPermit 1.23 2.44e-113  
## 6 education4Incomplete secondary 1.15 5.62e- 18  
## 7 education3Complete primary 1.13 2.93e- 19  
## 8 ethnicitywhite 0.757 3.58e- 80  
## 9 education2Incomplete primary 0.736 1.08e- 10  
## 10 workEducation, health/social service 0.660 1.39e- 9  
## # ℹ 14 more rows

temp = glm(poverty ~ age + education + work + metropolitan\_area + urban + work\_permit + gender + ethnicity,data = df1, family=binomial(link=logit))  
  
effects = effectsize::standardize\_parameters(temp, exp = T)

## Profiled confidence intervals may take longer time to compute.  
## Use `ci\_method="wald"` for faster computation of CIs.

effects %>% arrange(desc(Std\_Odds\_Ratio))

## # Standardization method: refit  
##   
## Parameter | Std\_Odds\_Ratio | 95% CI  
## ----------------------------------------------------------------------  
## education7Complete higher | 16.18 | [12.35, 21.25]  
## education6Incomplete higher | 7.53 | [ 5.61, 10.13]  
## education5Complete secondary | 4.37 | [ 3.46, 5.52]  
## workPublic administration | 3.52 | [ 2.80, 4.44]  
## work\_permitPermit | 3.42 | [ 3.07, 3.80]  
## education4Incomplete secondary | 3.16 | [ 2.43, 4.10]  
## education3Complete primary | 3.11 | [ 2.43, 3.99]  
## ethnicitywhite | 2.13 | [ 1.97, 2.31]  
## education2Incomplete primary | 2.09 | [ 1.67, 2.61]  
## urbanYes | 1.87 | [ 1.70, 2.07]  
## workEducation, health/social service | 1.87 | [ 1.55, 2.26]  
## workWhite collar business | 1.77 | [ 1.45, 2.17]  
## work\_permitOther situation | 1.62 | [ 1.47, 1.80]  
## workLogistics | 1.37 | [ 1.11, 1.68]  
## workHome services | 1.35 | [ 1.12, 1.63]  
## age | 1.30 | [ 1.24, 1.35]  
## workIndustry | 1.28 | [ 1.10, 1.49]  
## workOther | 1.24 | [ 1.02, 1.51]  
## workAuto repair | 1.22 | [ 1.06, 1.40]  
## genderF | 1.14 | [ 1.04, 1.25]  
## workConstruction | 1.05 | [ 0.89, 1.23]  
## metropolitan\_areaNo | 1.01 | [ 0.92, 1.10]  
## workHospitality | 0.97 | [ 0.80, 1.17]  
## (Intercept) | 0.16 | [ 0.12, 0.20]  
##   
## - Response is unstandardized.