Supplement 4

Risk factors

Peter Kamerman

Last knitted: 27 June 2019

Contents

This script generates summaries of key demographic information for the full cohort, with and without conditioning on HIV status.

We present the data in tabular and graphical format, and calculate the precision of the estimates using bootstrap 95% confidence intervals.

To describe any differences between the HIV+ and HIV- groups, we have calculated 95% confidence intervals of the difference in mean/proportion.

Import data

```r
# Import data
pain <- read_rds('data-cleaned/wbpq.rds') %>%
    select(PID,
           pain_in_last_month,
           pain_worst) %>%
    mutate(pain = ifelse(pain_in_last_month == 'yes' & pain_worst > 0,
                         yes = 'yes',
                         no = 'no')) %>%
    select(PID, pain)

general <- read_rds('data-cleaned/general_info.rds') %>%
    select(PID, age, sex, educational_level, employment) %>%
    mutate(employment = fct_collapse(employment,
                                     employed = c('employed', 'employed (part time)'),
                                     unemployed = c('unemployed'),
                                     grant = c('disability grant', 'pension grant')))

mental_health <- read_rds('data-cleaned/hscl.rds') %>%
    select(PID, total_score)

# Join to core_info
data <- read_rds('data-cleaned/hiv_test.rds') %>%
    select(PID, test_result) %>%
    left_join(pain) %>%
    left_join(general) %>%
    left_join(mental_health)
```

Clean data

```r
# Remove participants without test results
data %<>%
    filter(!is.na(test_result))

# Remove participants with missing pain data
data %<>%
    filter(!is.na(pain))

# Convert character classes to factors
data %<>%
    mutate_if(is.character, factor)
```

Quick look

```
# Dataframe dimensions
dim(data)
```

```
## [1] 535   8
```

```
# Column names
names(data)
```

```
## [1] "PID"               "test_result"       "pain"
## [4] "age"               "sex"               "educational_level"
## [7] "employment"        "total_score"
```

```
# Glimpse data
glimpse(data)
```

```
## Observations: 535
## Variables: 8
## $ PID               <fct> 001, 003, 004, 005, 006, 007, 008, 009, 010,...
## $ test_result       <fct> HIV negative, HIV negative, HIV negative, HI...
## $ pain              <fct> no, yes, yes, yes, yes, no, yes, yes, yes, y...
## $ age               <dbl> 35, 50, 38, 37, 30, 25, 39, 27, 23, 32, 36, ...
## $ sex               <fct> male, female, male, male, male, male, male, ...
## $ educational_level <ord> secondary school, no/primary school, seconda...
## $ employment        <fct> unemployed, grant, employed, employed, emplo...
## $ total_score       <dbl> 3.40, 1.28, 1.92, 1.04, 2.72, 1.64, 1.76, 2....
```

Check missingness

Full cohort

```
data %>%
    profile_missing() %>%
    mutate(pct_missing = round(100 * pct_missing)) %>%
    arrange(pct_missing)
```

```
## # A tibble: 8 x 3
##   feature           num_missing pct_missing
##   <fct>                   <int>       <dbl>
## 1 PID                         0           0
## 2 test_result                 0           0
## 3 pain                        0           0
## 4 sex                         2           0
## 5 age                         3           1
## 6 employment                  3           1
## 7 total_score                 5           1
## 8 educational_level          14           3
```

HIV-

```
data %>%
    filter(test_result == 'HIV negative') %>%
    profile_missing() %>%
    mutate(pct_missing = round(100 * pct_missing)) %>%
    arrange(pct_missing)
```

```
## # A tibble: 8 x 3
##   feature           num_missing pct_missing
##   <fct>                   <int>       <dbl>
## 1 PID                         0           0
## 2 test_result                 0           0
## 3 pain                        0           0
## 4 age                         2           0
## 5 sex                         1           0
## 6 employment                  3           1
## 7 total_score                 5           1
## 8 educational_level          14           3
```

HIV+

```
data %>%
    filter(test_result == 'HIV positive') %>%
    profile_missing() %>%
    mutate(pct_missing = round(100 * pct_missing)) %>%
    arrange(pct_missing)
```

```
## # A tibble: 8 x 3
##   feature           num_missing pct_missing
##   <fct>                   <int>       <dbl>
## 1 PID                         0           0
## 2 test_result                 0           0
## 3 pain                        0           0
## 4 educational_level           0           0
## 5 employment                  0           0
## 6 total_score                 0           0
## 7 age                         1           1
## 8 sex                         1           1
```

HIV status

Build model

```
mod_hiv <- glm(pain ~ test_result,
               data = data,
               family = binomial(link = 'logit'))
```

Beta coefficients

```
# Coefficients
coef(mod_hiv)
```

```
##             (Intercept) test_resultHIV positive
##               0.4234189              -0.4234189
```

```
# 95% CI of the coefficients
confint(mod_hiv)
```

```
##                              2.5 %     97.5 %
## (Intercept)              0.2386153 0.61063071
## test_resultHIV positive -0.9293655 0.08220701
```

Odds ratio

```
# OR
exp(coef(mod_hiv))
```

```
##             (Intercept) test_resultHIV positive
##               1.5271739               0.6548043
```

```
# 95% CI of the OR
exp(confint(mod_hiv))
```

```
##                             2.5 %   97.5 %
## (Intercept)             1.2694900 1.841593
## test_resultHIV positive 0.3948041 1.085681
```

Overall model

```
# likelihood ratio test
Anova(mod_hiv,
      test = 'LR')
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: pain
##              LR Chisq Df Pr(>Chisq)
## test_result   2.6992  1     0.1004
```

Model terms

```
# Summary
summary(mod_hiv)
```

```
##
## Call:
## glm(formula = pain ~ test_result, family = binomial(link = "logit"),
##     data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.362  -1.362   1.004   1.004   1.177
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)              0.42342    0.09483   4.465 8.01e-06 ***
## test_resultHIV positive -0.42342    0.25717  -1.646   0.0997 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 723.98  on 534  degrees of freedom
## Residual deviance: 721.28  on 533  degrees of freedom
## AIC: 725.28
##
## Number of Fisher Scoring iterations: 4
```

```
# Wald test
Anova(mod_hiv,
      type = 'II',
      test = 'Wald')
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: pain
##             Df  Chisq Pr(>Chisq)
## test_result  1 2.7108    0.09967 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model fit

Pseudo-R^2

```
nagelkerke(mod_hiv)
```

```
## $Models
##
## Model: "glm, pain ~ test_result, binomial(link = \"logit\"), data"
## Null:  "glm, pain ~ 1, binomial(link = \"logit\"), data"
##
## $Pseudo.R.squared.for.model.vs.null
##                              Pseudo.R.squared
## McFadden                           0.00372828
## Cox and Snell (ML)                 0.00503255
## Nagelkerke (Cragg and Uhler)       0.00678609
##
## $Likelihood.ratio.test
##  Df.diff LogLik.diff  Chisq p.value
##       -1     -1.3496 2.6992  0.1004
##
## $Number.of.observations
##
## Model: 535
## Null:  535
##
## $Messages
## [1] "Note: For models fit with REML, these statistics are based on refitting with ML"
##
## $Warnings
## [1] "None"
```

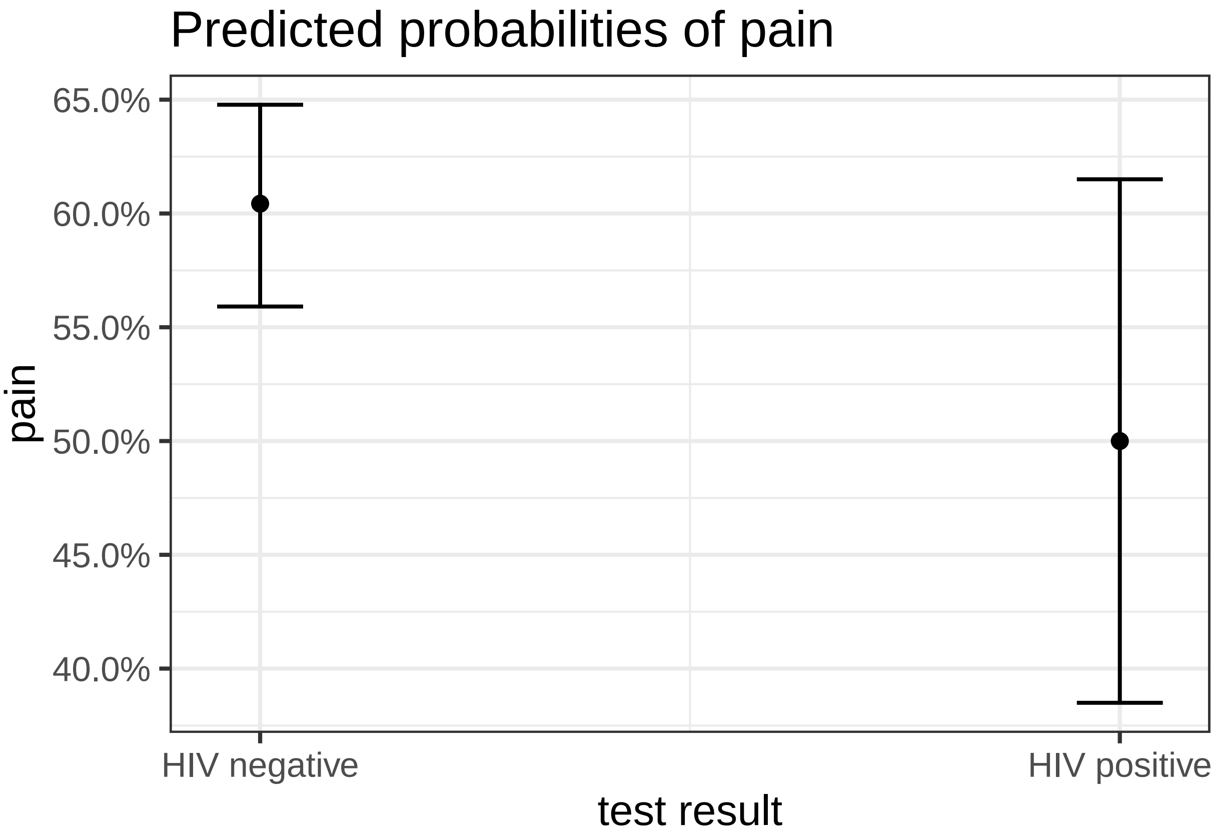
Hosmer-Lemeshow test

```
hoslem.test(x = mod_hiv$y,
            y = fitted(mod_hiv),
            g = 10)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  mod_hiv$y, fitted(mod_hiv)
## X-squared = 9.9918e-29, df = 8, p-value = 1
```

Plot predicted probabilities

```
plot_model(mod_hiv,
           type = 'pred')$test_result
```



```
# Publication plot
## Extract data
hiv <- plot_model(mod_hiv,
                  type = 'pred')$test_result

hiv_data <- tibble(x = factor(hiv$data$x),
                   pred = hiv$data$predicted,
                   low = hiv$data$conf.low,
                   high = hiv$data$conf.high)

## Plot
```

```
pp_hiv <- ggplot(data = hiv_data) +
    aes(x = x,
        y = pred,
        ymin = low,
        ymax = high) +
    geom_errorbar(width = 0.3,
                  size = 1) +
    geom_point(size = 3) +
    annotate(geom = 'text',
             label = 'HIV status*',
             size = 5,
             x = 0.5,
             y = 0.97,
             hjust = 0) +
    scale_y_continuous(limits = c(0, 1),
                       position = 'right') +
    scale_x_discrete(labels = c('Negative', 'Positive')) +
    labs(x = 'HIV test result') +
    theme(axis.title.y = element_blank(),
          axis.title.x = element_text(size = 17),
          panel.grid = element_blank(),
          axis.text = element_text(colour = '#000000'))
```

Age

Build model

```
mod_age <- glm(pain ~ age,
               data = data[!is.na(data$age), ],
               family = binomial(link = 'logit'))
```

Beta coefficients

```
# Coefficients
coef(mod_age)
```

```
## (Intercept)         age
## 0.194494016 0.004976257
```

```
# 95% CI of the coefficients
confint(mod_age)
```

```
##                   2.5 %     97.5 %
## (Intercept) -0.36882723 0.75671672
## age         -0.01059239 0.02075425
```

Odds ratios

```
# OR
exp(coef(mod_age))
```

```
## (Intercept)         age
##    1.214696    1.004989
```

```
# 95% CI of the OR
exp(confint(mod_age))
```

```
##                  2.5 %   97.5 %
## (Intercept) 0.6915449 2.131267
## age         0.9894635 1.020971
```

Overall model

```
# Likelihood ratio test
Anova(mod_age,
      test = 'LR')
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: pain
##     LR Chisq Df Pr(>Chisq)
## age  0.39027  1     0.5322
```

Model terms

```
# Summary
summary(mod_age)
```

```
##
## Call:
## glm(formula = pain ~ age, family = binomial(link = "logit"),
##     data = data[!is.na(data$age), ])
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.419  -1.324   1.002   1.034   1.065
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 0.194494   0.286725   0.678    0.498
## age         0.004976   0.007981   0.624    0.533
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 720.09  on 531  degrees of freedom
## Residual deviance: 719.70  on 530  degrees of freedom
## AIC: 723.7
##
## Number of Fisher Scoring iterations: 4
```

```
# Wald test
Anova(mod_age,
      type = 'II',
      test = 'Wald')
```

```
## Analysis of Deviance Table (Type II tests)
```

```
##
## Response: pain
##     Df  Chisq Pr(>Chisq)
## age  1 0.3888      0.533
```

Model fit

Pseudo-R^2

```
nagelkerke(mod_age)
```

```
## $Models
##
## Model: "glm, pain ~ age, binomial(link = \"logit\"), data[!is.na(data$age), ]"
## Null:  "glm, pain ~ 1, binomial(link = \"logit\"), data[!is.na(data$age), ]"
##
## $Pseudo.R.squared.for.model.vs.null
##                              Pseudo.R.squared
## McFadden                          0.000541979
## Cox and Snell (ML)                0.000733329
## Nagelkerke (Cragg and Uhler)      0.000988741
##
## $Likelihood.ratio.test
##  Df.diff LogLik.diff   Chisq p.value
##       -1    -0.19514 0.39027 0.53216
##
## $Number.of.observations
##
## Model: 532
## Null:  532
##
## $Messages
## [1] "Note: For models fit with REML, these statistics are based on refitting with ML"
##
## $Warnings
## [1] "None"
```

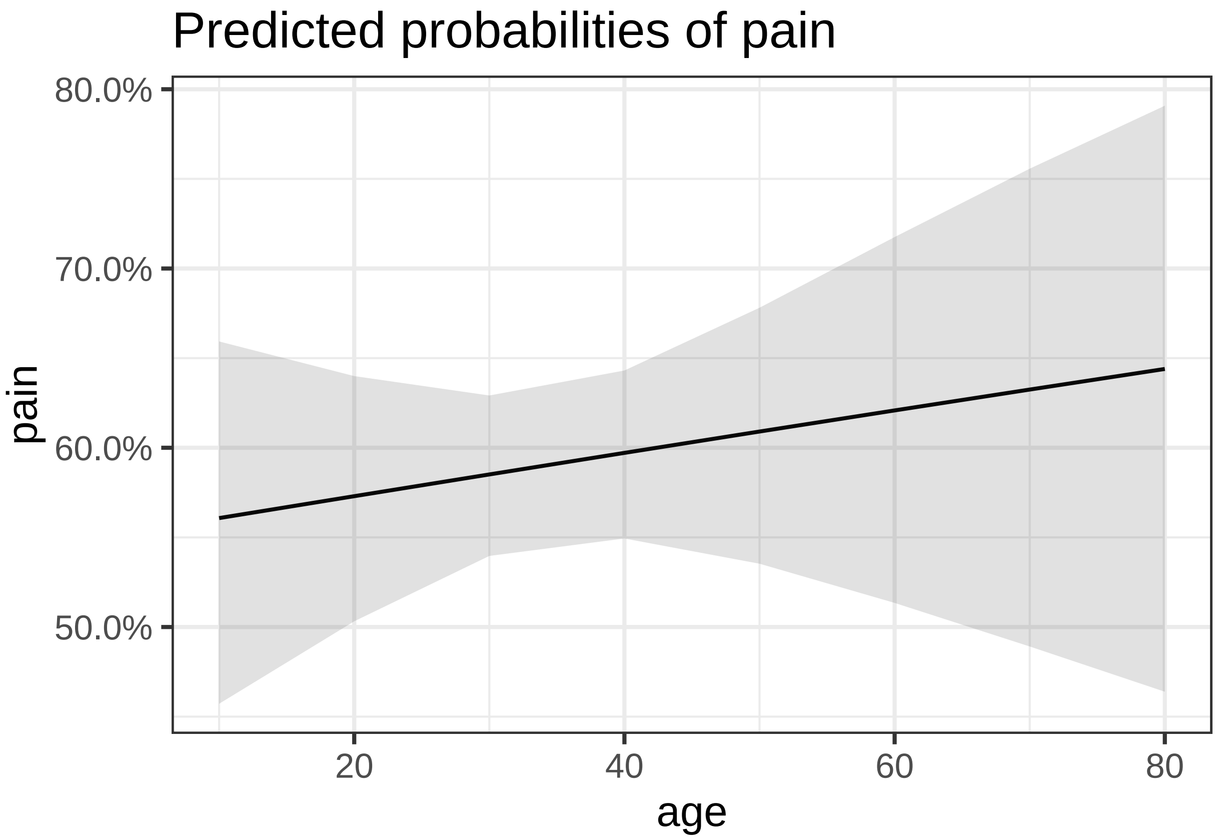
Hosmer-Lemeshow test

```
hoslem.test(x = mod_age$y,
            y = fitted(mod_age),
            g = 10)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  mod_age$y, fitted(mod_age)
## X-squared = 6.6654, df = 8, p-value = 0.5731
```

Plot predicted probabilities

```
plot_model(mod_age,
           type = 'pred')$age
```



```r
# Publication plot
## Extract data
age <- plot_model(mod_age,
                  type = 'pred')$age

age_data <- tibble(x = age$data$x,
                   pred = age$data$predicted,
                   low = age$data$conf.low,
                   high = age$data$conf.high)

## Plot
pp_age <- ggplot(data = age_data) +
    aes(x = x,
        y = pred,
        ymax = high,
        ymin = low) +
    geom_ribbon(fill = '#CCCCCC') +
    geom_line(size = 0.8) +
    annotate(geom = 'text',
             label = 'Age*',
             size = 5,
             x = 10,
             y = 0.97,
             hjust = 0) +
    scale_y_continuous(limits = c(0, 1),
                       position = 'left') +
    labs(x = 'Age (years)') +
    theme(axis.title.y = element_blank(),
          axis.title.x = element_text(size = 17),
          panel.grid = element_blank(),
          axis.text = element_text(colour = '#000000'))
```

Sex

Build model

```
mod_sex <- glm(pain ~ sex,
               data = data[!is.na(data$sex), ],
               family = binomial(link = 'logit'))
```

Beta coefficients

```
# Coefficients
coef(mod_sex)
```

```
## (Intercept)     sexmale
##   0.5920511  -0.5007013
```

```
# 95% CI of the coefficients
confint(mod_sex)
```

```
##                  2.5 %     97.5 %
## (Intercept)  0.3549933  0.8346266
## sexmale     -0.8502162 -0.1532804
```

Odds ratios

```
# OR
exp(coef(mod_sex))
```

```
## (Intercept)     sexmale
##   1.8076923   0.6061055
```

```
# 95% CI of the OR
exp(confint(mod_sex))
```

```
##                 2.5 %    97.5 %
## (Intercept) 1.4261710 2.3039535
## sexmale     0.4273226 0.8578891
```

Overall model

```
# Likelihood ratio test
Anova(mod_sex,
      test = 'LR')
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: pain
##     LR Chisq Df Pr(>Chisq)
## sex   7.9882  1   0.004708 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model terms

```
# Summary
summary(mod_sex)
```

```
##
## Call:
## glm(formula = pain ~ sex, family = binomial(link = "logit"),
##     data = data[!is.na(data$sex), ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4369  -1.2164   0.9384   0.9384   1.1389
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.5921     0.1222   4.845 1.27e-06 ***
## sexmale      -0.5007     0.1777  -2.818  0.00483 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 721.87  on 532  degrees of freedom
## Residual deviance: 713.88  on 531  degrees of freedom
## AIC: 717.88
##
## Number of Fisher Scoring iterations: 4
```

```
# Wald test
Anova(mod_sex,
      type = 'II',
      test = 'Wald')
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: pain
##     Df Chisq Pr(>Chisq)
## sex  1 7.942    0.00483 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model fit

Pseudo-R^2

```
nagelkerke(mod_sex)
```

```
## $Models
##
## Model: "glm, pain ~ sex, binomial(link = \"logit\"), data[!is.na(data$sex), ]"
## Null:  "glm, pain ~ 1, binomial(link = \"logit\"), data[!is.na(data$sex), ]"
##
## $Pseudo.R.squared.for.model.vs.null
##                              Pseudo.R.squared
## McFadden                            0.0110660
## Cox and Snell (ML)                  0.0148755
## Nagelkerke (Cragg and Uhler)        0.0200509
##
## $Likelihood.ratio.test
##  Df.diff LogLik.diff  Chisq   p.value
##       -1     -3.9941 7.9882 0.0047083
##
## $Number.of.observations
##
## Model: 533
## Null:  533
##
## $Messages
## [1] "Note: For models fit with REML, these statistics are based on refitting with ML"
##
## $Warnings
## [1] "None"
```

Hosmer-Lemeshow test

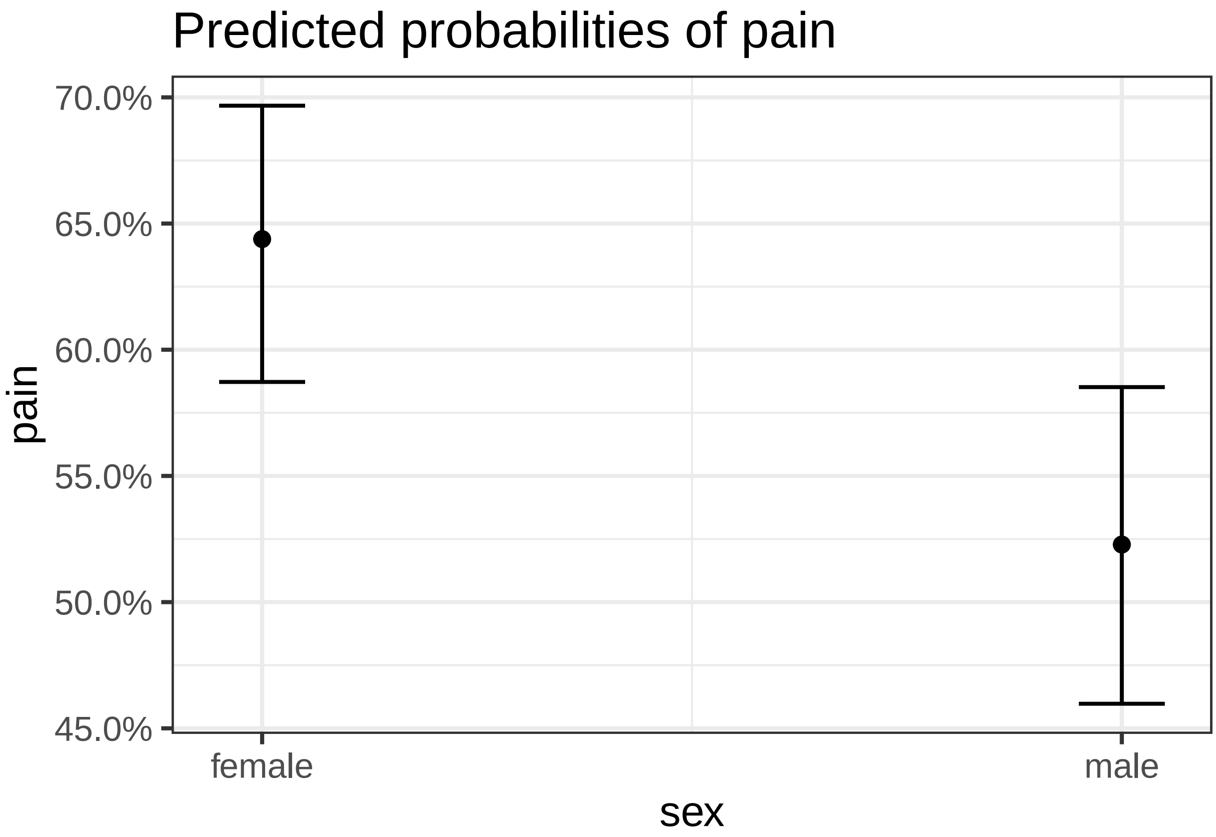
```
hoslem.test(x = mod_sex$y,
            y = fitted(mod_sex),
            g = 10)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  mod_sex$y, fitted(mod_sex)
## X-squared = 4.1499e-26, df = 8, p-value = 1
```

Plot predicted probabilities

```
plot_model(mod_sex,
           type = 'pred')
```

```
## $sex
```



```
# Publication plot
## Extract data
sex <- plot_model(mod_sex,
                  type = 'pred')$sex

sex_data <- tibble(x = factor(sex$data$x),
                   pred = sex$data$predicted,
                   low = sex$data$conf.low,
                   high = sex$data$conf.high)

## Plot
pp_sex <- ggplot(data = sex_data) +
    aes(x = x,
        y = pred,
        ymin = low,
        ymax = high) +
    geom_errorbar(width = 0.3,
                  size = 1) +
    geom_point(size = 3) +
    annotate(geom = 'text',
             label = 'Sex',
             size = 5,
             x = 0.5,
             y = 0.97,
             hjust = 0) +
    scale_y_continuous(limits = c(0, 1),
                       position = 'right') +
    scale_x_discrete(labels = c('Female', 'Male')) +
    labs(x = 'Sex') +
    theme(axis.title.y = element_blank(),
          axis.title.x = element_text(size = 17),
          panel.grid = element_blank(),
          axis.text = element_text(colour = '#000000'))
```

Educational level

Build model

```
mod_school <- glm(pain ~ educational_level,
                 data = data[!is.na(data$educational_level), ],
                 family = binomial(link = 'logit'))
```

Beta coefficients

```
# Coefficients
coef(mod_school)
```

```
##         (Intercept) educational_level.L educational_level.Q
##          0.31072492          0.21947500          0.02822161
```

```
# 95% CI of the coefficients
confint(mod_school)
```

```
##                           2.5 %    97.5 %
## (Intercept)          0.01507632 0.6117129
## educational_level.L -0.39956640 0.8277941
## educational_level.Q -0.36608050 0.4276322
```

Odds ratios

```
# OR
exp(coef(mod_school))
```

```
##         (Intercept) educational_level.L educational_level.Q
##            1.364414            1.245423            1.028624
```

```
# 95% CI of the OR
exp(confint(mod_school))
```

```
##                         2.5 %   97.5 %
## (Intercept)         1.0151905 1.843587
## educational_level.L 0.6706108 2.288265
## educational_level.Q 0.6934470 1.533622
```

Overall model

```
# Likelihood ratio test
Anova(mod_school,
      test = 'LR')
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: pain
##                   LR Chisq Df Pr(>Chisq)
## educational_level   1.1781  2     0.5548
```

Model terms

```
# Summary
summary(mod_school)
```

```
##
## Call:
## glm(formula = pain ~ educational_level, family = binomial(link = "logit"),
##     data = data[!is.na(data$educational_level), ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3857  -1.3018   0.9825   1.0579   1.1073
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          0.31072    0.15081   2.060   0.0394 *
## educational_level.L  0.21948    0.30985   0.708   0.4787
## educational_level.Q  0.02822    0.20114   0.140   0.8884
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 706.98  on 520  degrees of freedom
## Residual deviance: 705.80  on 518  degrees of freedom
## AIC: 711.8
##
## Number of Fisher Scoring iterations: 4
```

```
# Wald test
Anova(mod_school,
      type = 'II',
      test = 'Wald')
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: pain
##                   Df  Chisq Pr(>Chisq)
## educational_level  2 1.1729     0.5563
```

Model fit

Pseudo-R^2

```
nagelkerke(mod_school)
```

```
## $Models
```

```
##
## Model: "glm, pain ~ educational_level, binomial(link = \"logit\"), data[!is.na(data$educational_leve
## Null:  "glm, pain ~ 1, binomial(link = \"logit\"), data[!is.na(data$educational_level), ]"
##
## $Pseudo.R.squared.for.model.vs.null
##                              Pseudo.R.squared
## McFadden                           0.00166642
## Cox and Snell (ML)                 0.00225873
## Nagelkerke (Cragg and Uhler)       0.00304181
##
## $Likelihood.ratio.test
##  Df.diff LogLik.diff  Chisq p.value
##       -2    -0.58906 1.1781 0.55485
##
## $Number.of.observations
##
## Model: 521
## Null:  521
##
## $Messages
## [1] "Note: For models fit with REML, these statistics are based on refitting with ML"
##
## $Warnings
## [1] "None"
```

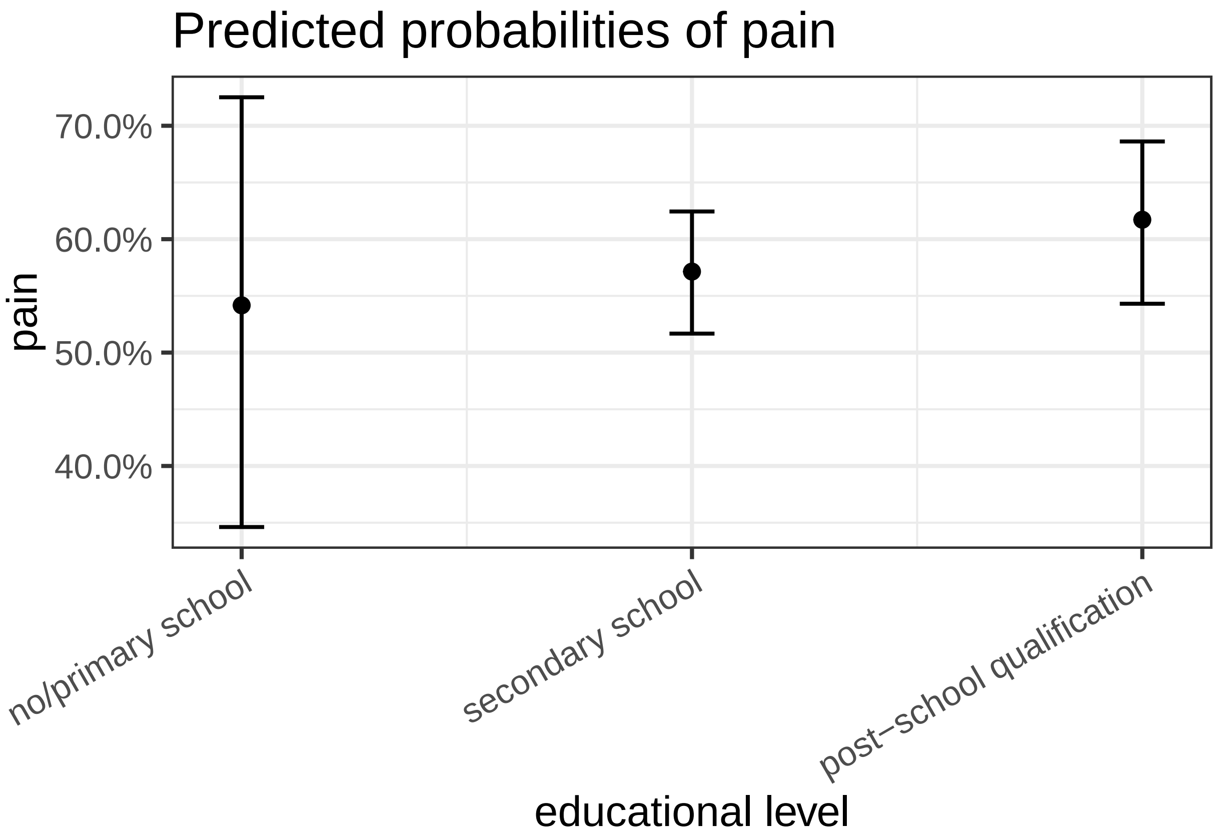
Hosmer-Lemeshow test

```
hoslem.test(x = mod_school$y,
            y = fitted(mod_school),
            g = 10)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  mod_school$y, fitted(mod_school)
## X-squared = 7.384e-28, df = 8, p-value = 1
```

Plot predicted probabilities

```
plot_model(mod_school,
           type = 'pred')$educational_level +
    theme(axis.text.x = element_text(angle = 30,
                                     hjust = 1))
```



```
# Publication plot
## Extract data
edu <- plot_model(mod_school,
                  type = 'pred')$educational_level

edu_data <- tibble(x = factor(edu$data$x),
                   pred = edu$data$predicted,
                   low = edu$data$conf.low,
                   high = edu$data$conf.high)

## Plot
pp_edu <- ggplot(data = edu_data) +
    aes(x = x,
        y = pred,
        ymin = low,
        ymax = high) +
    geom_errorbar(width = 0.3,
                  size = 1) +
    geom_point(size = 3) +
    annotate(geom = 'text',
             label = 'Education*',
             size = 5,
             x = 0.5,
             y = 0.97,
             hjust = 0) +
    scale_y_continuous(limits = c(0, 1),
                       position = 'left') +
    scale_x_discrete(labels = c('0-7', '8-12', '>12')) +
    labs(x = 'School grade') +
    theme(axis.title.y = element_blank(),
          axis.title.x = element_text(size = 17),
          panel.grid = element_blank(),
          axis.text = element_text(colour = '#000000'))
```

Employment

Build model

```
mod_employment <- glm(pain ~ employment,
                      data = data[!is.na(data$employment), ],
                      family = binomial(link = 'logit'))
```

Beta coefficients

```
# Coefficients
coef(mod_employment)
```

```
##          (Intercept)   employmentemployed employmentunemployed
##            1.0296194           -0.7467566           -0.6333710
```

```
# 95% CI of the coefficients
confint(mod_employment)
```

```
##                            2.5 %    97.5 %
## (Intercept)           0.06891139 2.1592545
## employmentemployed   -1.90174775 0.2485792
## employmentunemployed -1.78608566 0.3589340
```

Odds ratios

```
# OR
exp(coef(mod_employment))
```

```
##          (Intercept)   employmentemployed employmentunemployed
##            2.8000000            0.4739011            0.5307995
```

```
# 95% CI of the OR
exp(confint(mod_employment))
```

```
##                          2.5 %   97.5 %
## (Intercept)          1.0713413 8.664676
## employmentemployed   0.1493074 1.282202
## employmentunemployed 0.1676150 1.431802
```

Overall model

```
# Likelihood ratio test
Anova(mod_employment,
      test = 'LR')
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: pain
##            LR Chisq Df Pr(>Chisq)
## employment   2.2454  2     0.3254
```

Model terms

```
# Summary
summary(mod_employment)
```

```
##
## Call:
## glm(formula = pain ~ employment, family = binomial(link = "logit"),
##     data = data[!is.na(data$employment), ])
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.634  -1.300   1.014   1.060   1.060
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)            1.0296     0.5210   1.976   0.0481 *
## employmentemployed    -0.7468     0.5369  -1.391   0.1643
## employmentunemployed  -0.6334     0.5355  -1.183   0.2369
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 720.09  on 531  degrees of freedom
## Residual deviance: 717.84  on 529  degrees of freedom
## AIC: 723.84
##
## Number of Fisher Scoring iterations: 4
```

```
# Wald test
Anova(mod_employment,
      type = 'II',
      test = 'Wald')
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: pain
##            Df Chisq Pr(>Chisq)
## employment  2  2.09     0.3517
```

Model fit

Pseudo-R^2

```
nagelkerke(mod_employment)
```

```
## $Models
```

```
##
## Model: "glm, pain ~ employment, binomial(link = \"logit\"), data[!is.na(data$employment), ]"
## Null:  "glm, pain ~ 1, binomial(link = \"logit\"), data[!is.na(data$employment), ]"
##
## $Pseudo.R.squared.for.model.vs.null
##                              Pseudo.R.squared
## McFadden                           0.00311829
## Cox and Snell (ML)                 0.00421187
## Nagelkerke (Cragg and Uhler)       0.00567883
##
## $Likelihood.ratio.test
##  Df.diff LogLik.diff  Chisq p.value
##       -2     -1.1227 2.2454 0.32539
##
## $Number.of.observations
##
## Model: 532
## Null:  532
##
## $Messages
## [1] "Note: For models fit with REML, these statistics are based on refitting with ML"
##
## $Warnings
## [1] "None"
```

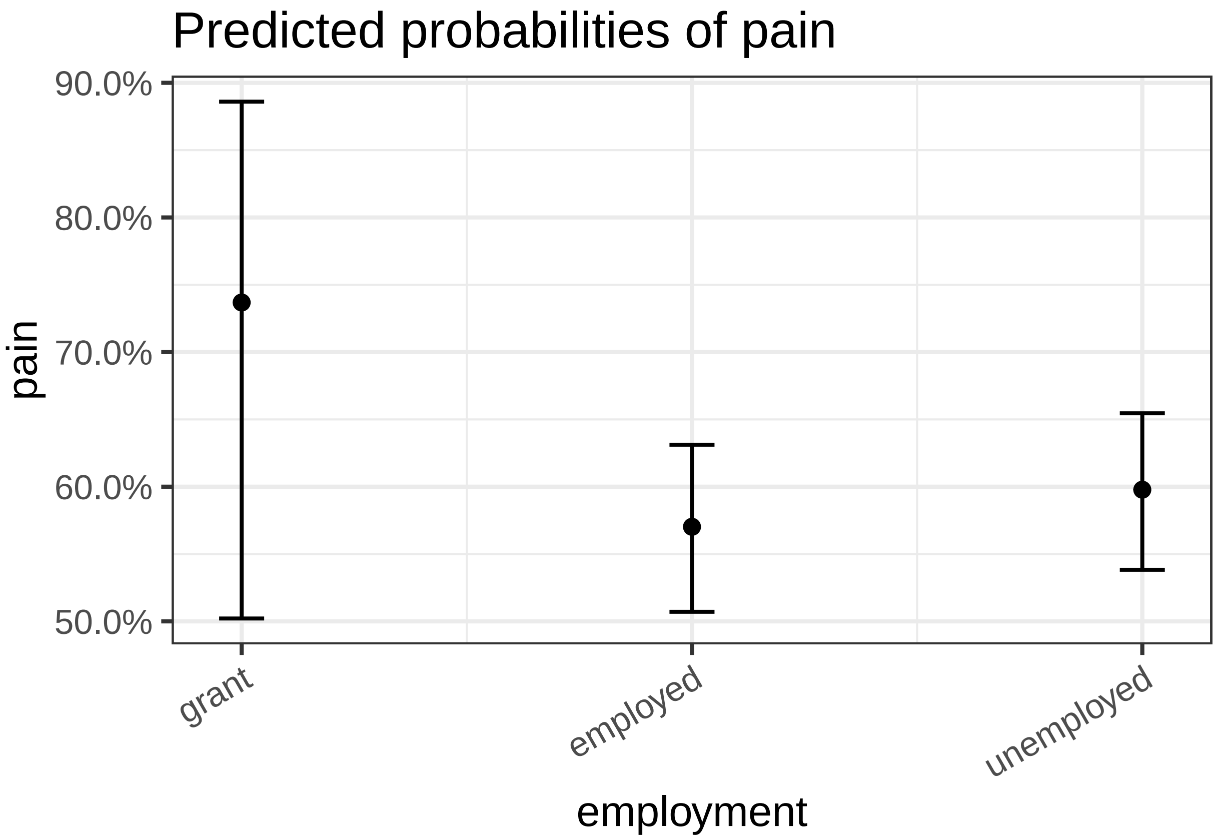
Hosmer-Lemeshow test

```
hoslem.test(x = mod_employment$y,
            y = fitted(mod_employment),
            g = 10)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  mod_employment$y, fitted(mod_employment)
## X-squared = 3.1778e-23, df = 8, p-value = 1
```

Plot predicted probabilities

```
plot_model(mod_employment,
           type = 'pred')$employment +
    theme(axis.text.x = element_text(angle = 30,
                                     hjust = 1))
```



```r
# Publication plot
## Extract data
emp <- plot_model(mod_employment,
                  type = 'pred')$employment

emp_data <- tibble(x = factor(emp$data$x),
                   pred = emp$data$predicted,
                   low = emp$data$conf.low,
                   high = emp$data$conf.high)

## Plot
pp_emp <- ggplot(data = emp_data) +
    aes(x = x,
        y = pred,
        ymin = low,
        ymax = high) +
    geom_errorbar(width = 0.3,
                  size = 1) +
    geom_point(size = 3) +
    annotate(geom = 'text',
             label = 'Employment',
             size = 5,
             x = 0.5,
             y = 0.97,
             hjust = 0) +
    scale_y_continuous(limits = c(0, 1),
                       position = 'right') +
    scale_x_discrete(labels = c('Grant', 'Employed', 'Unemployed')) +
    labs(x = 'Employment status') +
    theme(axis.title.y = element_blank(),
          axis.title.x = element_text(size = 17),
          panel.grid = element_blank(),
          axis.text = element_text(colour = '#000000'))
```

HSCL25 (total score)

Build model

```
mod_hscl <- glm(pain ~ total_score,
                data = data[!is.na(data$total_score), ],
                family = binomial(link = 'logit'))
```

Beta coefficients

```
# Coefficients
coef(mod_hscl)
```

```
## (Intercept) total_score
##   -1.856534    1.367989
```

```
# 95% CI of the coefficients
confint(mod_hscl)
```

```
##                 2.5 %    97.5 %
## (Intercept) -2.512401 -1.231110
## total_score  0.988773  1.775077
```

Odds ratios

```
# Odds ratio
exp(coef(mod_hscl))
```

```
## (Intercept) total_score
##   0.1562131   3.9274441
```

```
# 95% CI of the OR
exp(confint(mod_hscl))
```

```
##                  2.5 %    97.5 %
## (Intercept) 0.08107334 0.2919683
## total_score 2.68793445 5.9007362
```

Overall model

```
# Likelihood ratio test
Anova(mod_hscl,
      test = 'LR')
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: pain
##             LR Chisq Df Pr(>Chisq)
## total_score   59.271  1  1.374e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model terms

```
# Summary
summary(mod_hscl)
```

```
##
## Call:
## glm(formula = pain ~ total_score, family = binomial(link = "logit"),
##     data = data[!is.na(data$total_score), ])
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.389  -1.110   0.594   1.040   1.391
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.8565     0.3264  -5.687 1.29e-08 ***
## total_score   1.3680     0.2003   6.830 8.50e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 717.25  on 529  degrees of freedom
## Residual deviance: 657.98  on 528  degrees of freedom
## AIC: 661.98
##
## Number of Fisher Scoring iterations: 4
```

```
# Wald test
Anova(mod_hscl,
      type = 'II',
      test = 'Wald')
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: pain
##             Df  Chisq Pr(>Chisq)
## total_score  1 46.647  8.499e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model fit

Pseudo-R^2

```
nagelkerke(mod_hscl)
```

```
## $Models
##
## Model: "glm, pain ~ total_score, binomial(link = \"logit\"), data[!is.na(data$total_score), ]"
## Null:  "glm, pain ~ 1, binomial(link = \"logit\"), data[!is.na(data$total_score), ]"
##
## $Pseudo.R.squared.for.model.vs.null
##                              Pseudo.R.squared
## McFadden                             0.082637
## Cox and Snell (ML)                   0.105806
## Nagelkerke (Cragg and Uhler)         0.142670
##
## $Likelihood.ratio.test
##  Df.diff LogLik.diff  Chisq    p.value
##       -1     -29.636 59.271 1.3736e-14
##
## $Number.of.observations
##
## Model: 530
## Null:  530
##
## $Messages
## [1] "Note: For models fit with REML, these statistics are based on refitting with ML"
##
## $Warnings
## [1] "None"
```

Hosmer-Lemeshow test

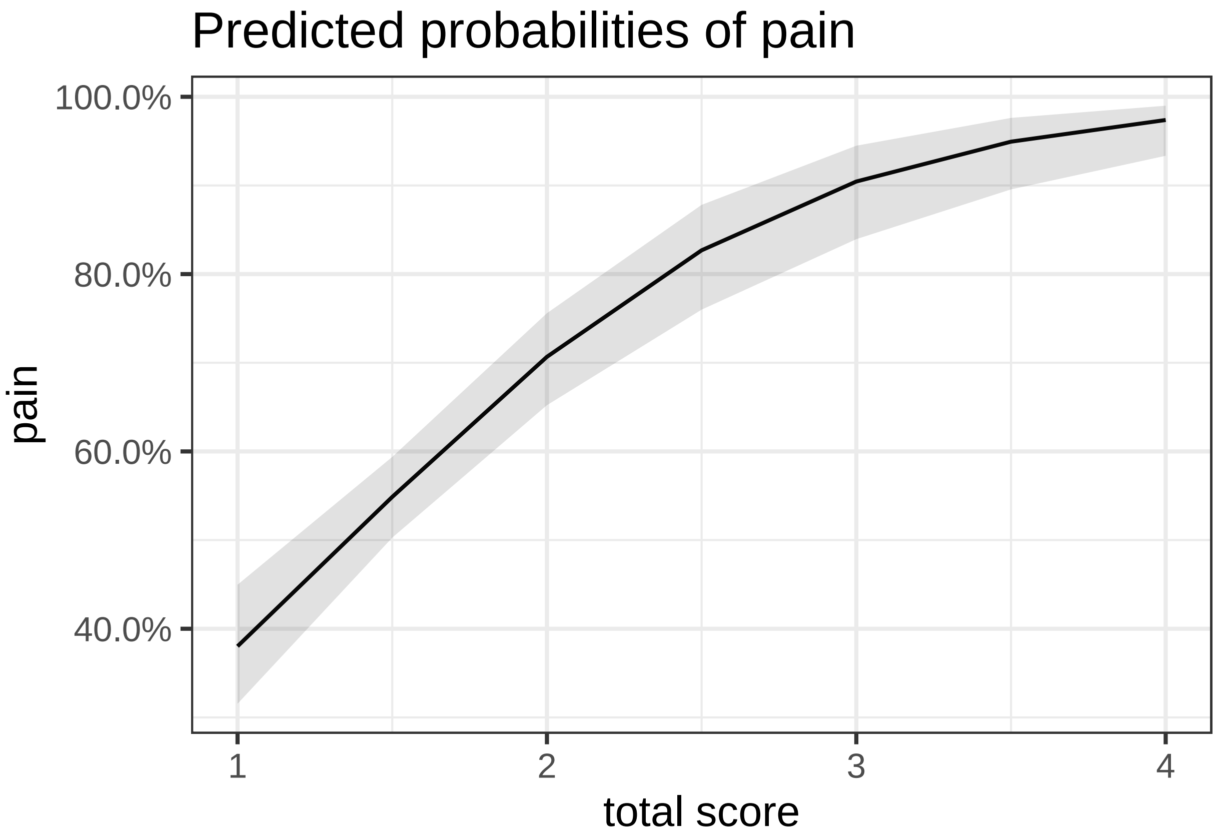
```
hoslem.test(x = mod_hscl$y,
            y = fitted(mod_hscl),
            g = 10)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  mod_hscl$y, fitted(mod_hscl)
## X-squared = 4.6332, df = 8, p-value = 0.796
```

Plot predicted probabilities

```
plot_model(mod_hscl,
           type = 'pred')
```

```
## $total_score
```



```
# Publication plot
## Extract data
hscl <- plot_model(mod_hscl,
                   type = 'pred')$total_score

hscl_data <- tibble(x = hscl$data$x,
                   pred = hscl$data$predicted,
                   low = hscl$data$conf.low,
                   high = hscl$data$conf.high)

## Plot
pp_hscl <- ggplot(data = hscl_data) +
    aes(x = x,
        y = pred,
        ymax = high,
        ymin = low) +
    geom_ribbon(fill = '#CCCCCC') +
    geom_line(size = 0.8) +
    annotate(geom = 'text',
             label = 'HSCL-25',
             size = 5,
             x = 1,
             y = 0.97,
             hjust = 0) +
    scale_y_continuous(limits = c(0, 1),
                       position = 'left') +
    labs(x = 'HSCL-25 total score') +
    theme(axis.title.y = element_blank(),
          axis.title.x = element_text(size = 17),
          panel.grid = element_blank(),
          axis.text = element_text(colour = '#000000'))
```

Variable selection

LASSO is a regression method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model it produces.

The process involves performing a 10-fold cross validation to find the optimal *lambda* (penalization parameter). And then running the analysis and extracting the model based on the best lambda.

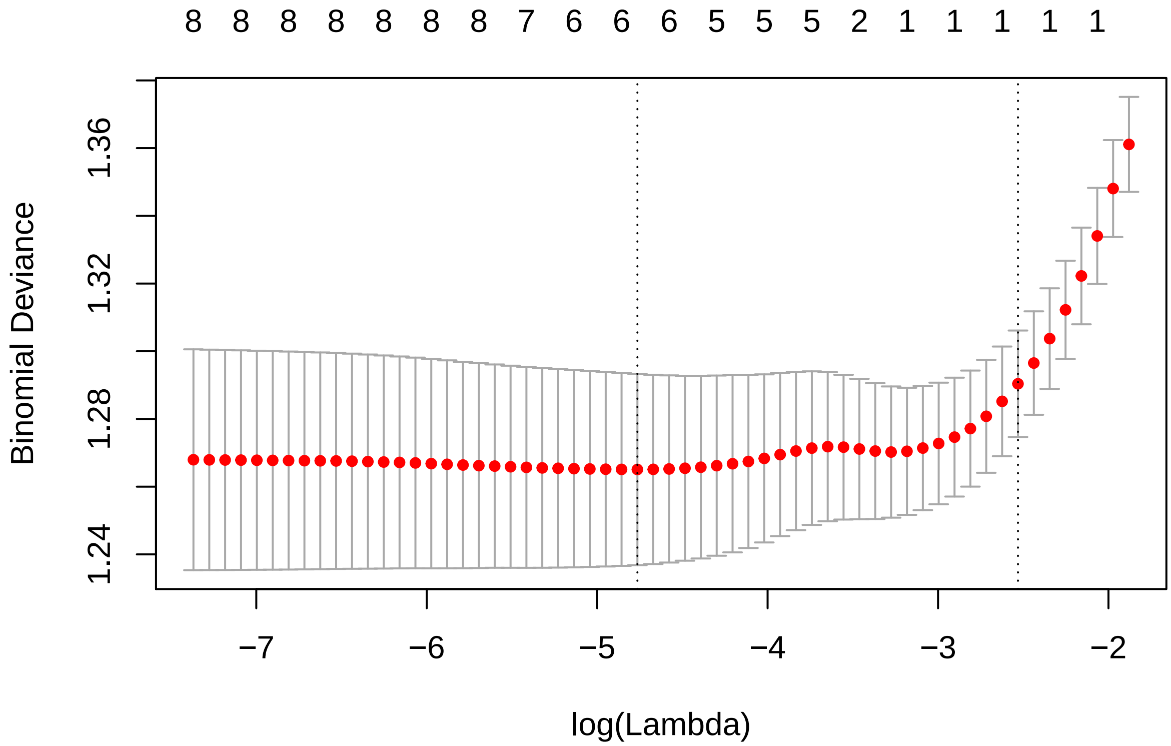
- *lambda.min* is the value of lambda that gives minimum mean cross-validated error.
- *lambda.1se*, is the value of lambda that gives the most regularized model such that error is within one standard error of the minimum

Generate a model matrix

```
# Extract complete cases
complete <- data %>%
    filter(complete.cases(.))

# Dependent variable
y <- complete$pain

# Predictor variables
## Factor variables
xfactor <- model.matrix(complete$pain ~ complete$test_result + complete$sex +
                            complete$educational_level + complete$employment)[, -1]

## Combine with continuous variables
x <- as.matrix(data.frame(complete$age, complete$total_score, xfactor))
```

Find the best minimum and 1SE lambda value using cross-validation

```
# Set seed
set.seed(2019)

# Calculate lambda (alpha = 1, lasso)
cv.lasso <- cv.glmnet(x = x, y = y,
                      nfolds = 10,
                      alpha = 1,
                      family = "binomial")

# Plot
plot(cv.lasso)
```



Lambda values

Lambda min

```
cv.lasso$lambda.min
```

```
## [1] 0.008532659
```

Lambda 1se

```
cv.lasso$lambda.1se
```

```
## [1] 0.07957586
```

Inspect the model coefficients

Lambda min

```
coef(cv.lasso, s = "lambda.min")
```

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                                          1
## (Intercept)                    -2.04768463
## complete.age                    0.01313481
## complete.total_score            1.24556316
## complete.test_resultHIV.positive -0.45009074
## complete.sexmale               -0.16187709
## complete.educational_level.L    0.23690209
## complete.educational_level.Q    0.02224888
## complete.employmentemployed       .
## complete.employmentunemployed     .
```

Lambda 1se

```
coef(cv.lasso, s = "lambda.1se")
```

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                                            1
## (Intercept)                       -0.5727692
## complete.age                       .
## complete.total_score               0.5481293
## complete.test_resultHIV.positive   .
## complete.sexmale                   .
## complete.educational_level.L       .
## complete.educational_level.Q       .
## complete.employmentemployed        .
## complete.employmentunemployed      .
```

Publication plot

```
composite_plot <- pp_hscl + pp_sex + pp_age + pp_hiv + pp_edu + pp_emp +
    plot_layout(ncol = 2, nrow = 3)

ggsave(filename = 'figures/figure2-original.png',
       plot = composite_plot,
       height = 10,
       width = 8)
```

Session information

```
sessionInfo()
```