Replication scripts - Models fitted in main paper

Method

All analyses were performed in **R** (version 4.1.0, R Core Team, 2021) We fitted (penalized) logistic regression models using **R** package **glmnet** (version 4.1.2, Friedman et al., 2010); generalized additive models (GAMs) with smoothing splines suing pacakge **mgcv** (version 1.8.35, Wood, 2017); conditional inference trees using package **partykit** (Hothorn et al., 2006; version 1.2.13, Hothorn & Zeileis, 2015); gradient boosted tree ensembles using package **gbm** (version 2.1.8, Greenwell et al., 2020); random forests using package **ranger** (version 0.13.1, Wright & Ziegler, 2017); prediction rule ensembles using package **pre** (version 1.0.1, Fokkema, 2020); *k* nearest neighbours using package **class** (version 7.3.19, Venables & Ripley, 2002).

Dataset

```r
data <- read.delim("data.csv", header = TRUE)

## Items should be scored 1-5, 0 may be missings
data[ , 1:48][sapply(data[ , 1:48], function(x) x == 0)] <- NA
data <- data[complete.cases(data[ , 1:48]), ]

## Select only university students
data <- data[data$education >= 3, ]
# table(data$major)
psych_ids <- rowSums(sapply(c("psych", "psyhcology", "psycotherapy", "couns",
                              "behavior", "behaviour", "neuro"),
                            function(x) grepl(x, data$major, ignore.case = TRUE)))
anim_ids <- grepl("anim", data$major, ignore.case = TRUE) ## exclude animal psych
data$major <- factor(ifelse(psych_ids > 0, "psychology", "other"))
data$major[anim_ids > 0 & psych_ids > 0] <- "other"

set.seed(42)
test_ids <- sample(1:nrow(data), ceiling(nrow(data)/4))
train_ids <- which(!1:nrow(data) %in% test_ids)
train_y <- as.numeric(data$major)[train_ids] - 1
test_y <- as.numeric(data$major)[test_ids] - 1

data$Real <- rowSums(data[ , paste0("R", 1:8)])
data$Inve <- rowSums(data[ , paste0("I", 1:8)])
data$Arti <- rowSums(data[ , paste0("A", 1:8)])
data$Soci <- rowSums(data[ , paste0("S", 1:8)])
```

```
data$Ente <- rowSums(data[ , paste0("E", 1:8)])
data$Conv <- rowSums(data[ , paste0("C", 1:8)])
```

```
format(nrow(data), nsmall = 0, big.mark = ",")
```

```
## [1] "55,593"
```

```
format(unname(round(100*table(data$major)[2] / nrow(data), digits = 2)),
       nsmall = 0, big.mark = ",")
```

```
## [1] "19.42"
```

```
format(unname(round(100*table(data$major)[1] / nrow(data), digits = 2)),
       nsmall = 0, big.mark = ",")
```

```
## [1] "80.58"
```

Model fitting and evaluation

```
format(length(train_ids), nsmall = 0, big.mark = ",")
```

```
## [1] "41,694"
```

```
format(unname(round(100*table(data$major[train_ids])[2] / length(train_ids),
                    digits = 2)), nsmall = 0, big.mark = ",")
```

```
## [1] "19.46"
```

```
format(length(test_ids), nsmall = 0, big.mark = ",")
```

```
## [1] "13,899"
```

```
format(unname(round(100*table(data$major[test_ids])[2] / length(test_ids),
                    digits = 2)), nsmall = 0, big.mark = ",")
```

```
## [1] "19.3"
```

```
paste0(R.Version()$major, ".", R.Version()$minor)
```

```
## [1] "4.1.0"
```

Results

```
varnames_i <- paste0(rep(c("R", "I", "A", "S", "E", "C"), each = 8), 1:8)
varnames_s <- c("R", "I", "A", "S", "E", "C")
```

(penalized) Logistic regression

```
glmod_s <- glm(major ~ Real + Inve + Arti + Soci + Ente + Conv,
               data = data[train_ids , ], family = "binomial")
```

```r
#summary(glmod_s)
glm_preds_train_s <- predict(glmod_s, newdata = data[train_ids, ], type = "response")
glm_preds_test_s <- predict(glmod_s, newdata = data[test_ids, ], type = "response")
```

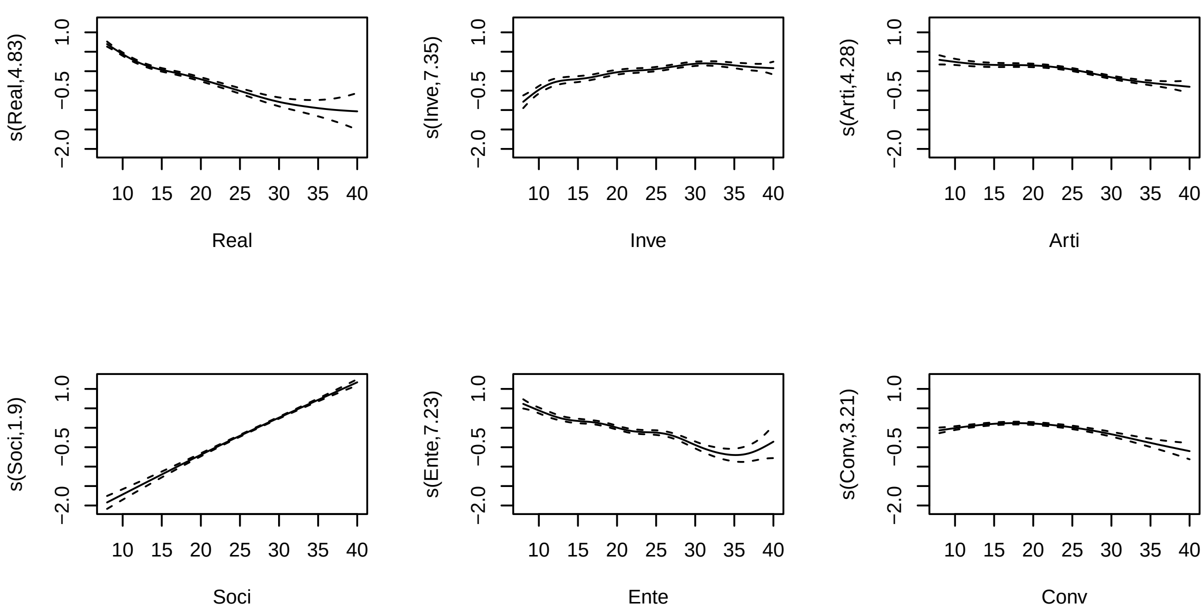
```r
library("glmnet")
X <- as.matrix(data[train_ids, varnames_i])
set.seed(42)
glmod_i <- glmnet(X, train_y, family = "binomial", alpha = 1, lambda = 0.0003568404)
glm_preds_train_i <- predict(glmod_i, newx = X, type = "response")
glm_preds_test_i <- predict(glmod_i, newx = as.matrix(data[test_ids, varnames_i]),
                            type = "response")
```

Generalized additive model

```r
library("mgcv")
gamod_s <- gam(major ~ s(Real) + s(Inve) + s(Arti) + s(Soci) + s(Ente) + s(Conv),
               data = data[train_ids , ], family = "binomial")
```

Figure 1

```r
par(mfrow = c(2, 3))
plot(gamod_s)
```




```r
gam_preds_train_s <- predict(gamod_s, newdata = data[train_ids, ], type = "response")
gam_preds_test_s <- predict(gamod_s, newdata = data[test_ids, ], type = "response")
```

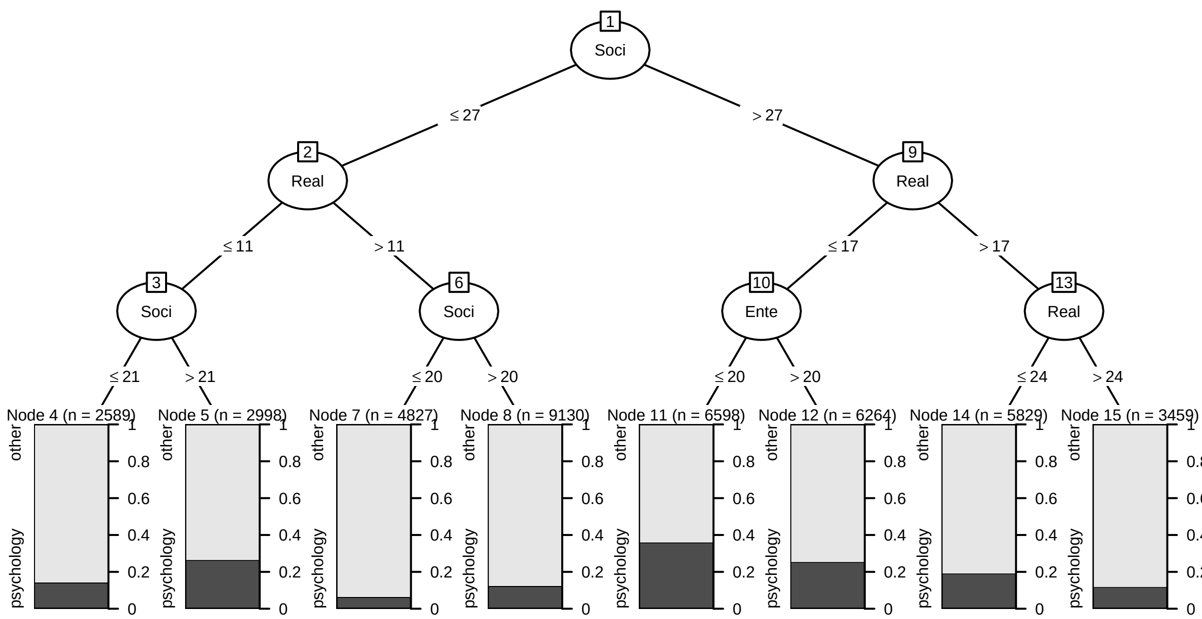
```
gamod_i <- gam(gam_form, data = data[train_ids , ], family = "binomial",
               select = TRUE)
```

```
gam_preds_train_i <- predict(gamod_i, newdata = data[train_ids, ], type = "response")
gam_preds_test_i <- predict(gamod_i, newdata = data[test_ids, ], type = "response")
```

Decision tree

Figure 2

```
library("partykit")
ct_s <- ctree(major ~ Real + Inve + Arti + Soci + Ente + Conv, data = data[train_ids , ],
              maxdepth = 7)
ct3 <- ctree(major ~ Real + Inve + Arti + Soci + Ente + Conv, data = data[train_ids , ],
             maxdepth = 3)
plot(ct3, gp = gpar(cex = .5), ip_args = list(pval = FALSE))
```




```
ct_preds_train_s <- predict(ct_s, type = "prob")[ , "psychology"]
ct_preds_test_s <- predict(ct_s, newdata = data[test_ids , ], type = "prob")[ , "psychology"]
```

```
ct_form <- formula(paste("major ~", paste(varnames_i, collapse = "+")))
ct <- ctree(ct_form, data = data[train_ids , ], maxdepth = 9)
ct_preds_train_i <- predict(ct, type = "prob")[ , "psychology"]
ct_preds_test_i <- predict(ct, newdata = data[test_ids , ], type = "prob")[ , "psychology"]
```

Gradient boosted tree ensemble

```
library("gbm")
set.seed(42)
gb_s <- gbm(I(as.numeric(major)-1) ~ Real + Inve + Arti + Soci + Ente + Conv,
            n.trees = 1100, interaction.depth = 3L, shrinkage = 0.01,
            data = data[train_ids , ])
```

```
gb_preds_train_s <- predict(gb_s, newdata = data[train_ids, ], type = "response")
```

```
## Using 1100 trees...
```

```
gb_preds_test_s <- predict(gb_s, newdata = data[test_ids, ], type = "response")
```

```
## Using 1100 trees...
```

```
library("gbm")
set.seed(42)
gbm_form <- formula(paste("I(as.numeric(major)-1) ~ ",
                          paste(paste0(rep(c("R", "I", "A", "S", "E", "C"),
                                           each = 8), 1:8), collapse = "+")))
gb_i <- gbm(gbm_form, n.trees = 3500, interaction.depth = 5L, shrinkage = 0.01,
            data = data[train_ids , ])
sum_i <- summary(gb_i, plotit = FALSE, method = permutation.test.gbm)
```

```
gb_preds_train_i <- predict(gb_i, newdata = data[train_ids, ], type = "response")
```

```
## Using 3500 trees...
```

```
gb_preds_test_i <- predict(gb_i, newdata = data[test_ids, ], type = "response")
```

```
## Using 3500 trees...
```

Random forest

```
library("ranger")
set.seed(42)
rf_s <- ranger(major ~ Real + Inve + Arti + Soci + Ente + Conv, data = data[train_ids , ],
                   probability = TRUE, mtry = 3L, min.node.size = 500,
             importance = "permutation")
```

```
rf_preds_train_s <- predict(rf_s, data = data[train_ids, ])$predictions[ , "psychology"]
rf_preds_test_s <- predict(rf_s, data = data[test_ids, ])$predictions[ , "psychology"]
```

```
set.seed(42)
varnames <- paste0(rep(c("R", "I", "A", "S", "E", "C"), each = 8), 1:8)
rf_form <- formula(paste("major ~", paste(varnames, collapse = "+")))
rf_i <- ranger(rf_form, data = data[train_ids , ], probability = TRUE,
            mtry = 10L, min.node.size = 500, importance = "permutation")
```

```r
rf_preds_train_i <- predict(rf_i, data = data[train_ids, ])$predictions[ , "psychology"]
rf_preds_test_i <- predict(rf_i, data = data[test_ids, ])$predictions[ , "psychology"]
```

Prediction rule ensembling

Table 1

```r
pr_preds_train_s <- predict(pr_s, type = "response")
pr_preds_test_s <- predict(pr_s, newdata = data[test_ids , ], type = "response")
imps <- pre::importance(pr_s, plot=FALSE)
varimps_pre <- imps$varimps
imps <- imps$baseimps[1:6, c("description", "coefficient")]
colnames(imps) <- c("Description", "Coefficient")
imps$Coefficient <- round(imps$Coefficient, digits = 3)
kable(imps, row.names = FALSE, align = c("l", "c"))
```

Description	Coefficient
Soci > 27 & Ente <= 31 & Conv <= 30	0.182
Soci > 23 & Ente <= 29 & Real <= 24	0.181
Real > 10 & Soci <= 35	-0.175
Real <= 22 & Soci > 19 & Inve > 18	0.138
Inve > 10 & Real <= 13	0.120
Conv <= 23 & Arti <= 29 & Soci > 21	0.112

```r
pr_preds_train_i <- predict(pr_i, type = "response")
pr_preds_test_i <- predict(pr_i, newdata = data[test_ids , ], type = "response")
```

k Nearest neighbours

```r
library("class")

## Model for training predictions
knn_mod <- knn(train = data[train_ids , varnames_s],
               test = data[train_ids , varnames_s],
               cl = as.factor(data[train_ids, "major"]),
               k = 300, use.all = TRUE, prob = TRUE)
## Need to obtain predicted probability for second class
knn_preds_train_s <- ifelse(
  knn_mod == "psychology", attr(knn_mod, "prob"), 1 - attr(knn_mod, "prob"))
```

```r
## Model for testing predictions
knn_mod <- knn(train = data[train_ids , varnames_s],
               test = data[test_ids , varnames_s],
```

```
               cl = as.factor(data[train_ids, "major"]),
               k = 300, use.all = TRUE, prob = TRUE)
knn_preds_test_s <- ifelse(
  knn_mod == "psychology", attr(knn_mod, "prob"), 1 - attr(knn_mod, "prob"))
```

```
## Model for training predictions
knn_mod <- knn(train = data[train_ids , varnames_i],
               test = data[train_ids , varnames_i],
               cl = as.factor(data[train_ids, "major"]),
               k = 100, use.all = TRUE, prob = TRUE)
## Need to obtain predicted probability for second class
knn_preds_train_i <- ifelse(
  knn_mod == "psychology", attr(knn_mod, "prob"), 1 - attr(knn_mod, "prob"))
```

```
## Model for testing predictions
knn_mod <- knn(train = data[train_ids , varnames_i],
               test = data[test_ids , varnames_i],
               cl = as.factor(data[train_ids, "major"]),
               k = 100, use.all = TRUE, prob = TRUE)
knn_preds_test <- ifelse(
  knn_mod == "psychology", attr(knn_mod, "prob"), 1 - attr(knn_mod, "prob"))
```

Model comparisons

Variable contributions

Figure 3

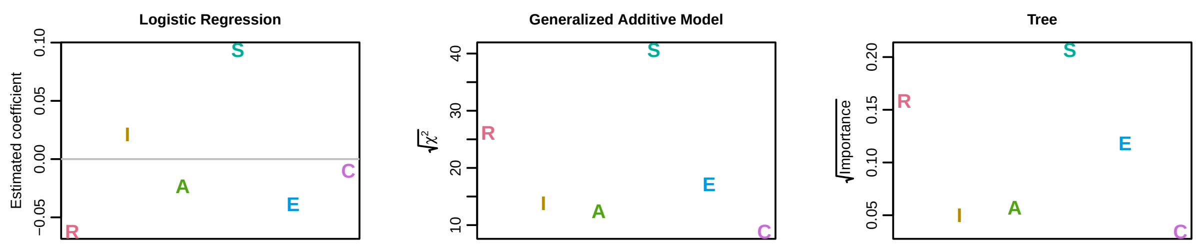
```
par(mfrow = c(1, 3))
par(mar = c(1, 4, 2, 1), mgp = c(1.5, .5, 0), tck = -0.05)
library("colorspace")

## Logistic regression
plot(coef(glmod_s)[-1], xaxt = "n", ylab = "Estimated coefficient",
     col = "white", cex.lab = .7, cex.axis = .7, xlab = " ",
     main = "Logistic Regression", cex.main = .7)
text(coef(glmod_s)[-1], labels = varnames_s, cex = 1,
     col = rep(qualitative_hcl(6)), font = 2)
abline(0, 0, col = "grey")

## Generalized additive model
sum <- summary(gamod_s)
plot(sqrt(sum$chi.sq), xaxt = "n", ylab = expression(sqrt(chi^2)),
     col = "white", cex.lab = .7, cex.axis = .7, xlab = " ",
     main = "Generalized Additive Model", cex.main = .7)
```

```r
text(sqrt(sum$chi.sq), labels = varnames_s, cex = 1,
     col = rep(qualitative_hcl(6)), font = 2)

## Conditional inference tree
ct6 <- cforest(major ~ Real + Inve + Arti + Soci + Ente + Conv,
               data = data[train_ids , ], ntree = 1L, mtry = 6,
               perturb = list(replace = FALSE, fraction = 1L),
               control = ctree_control(maxdepth = 6))
imps <- varimp(gettree(ct6), risk = "loglik")
imps <- imps[c("Real", "Inve", "Arti", "Soci", "Ente", "Conv")]
plot(sqrt(imps), xaxt = "n", ylab = expression(sqrt(Importance)),
     col = "white", cex.lab = .7, cex.axis = .7, xlab = " ",
     main = "Tree", cex.main = .7)
text(sqrt(imps), labels = varnames_s, cex = 1,
     col = rep(qualitative_hcl(6)), font = 2)
```




```r
## Gradient boosted ensemble
sum <- summary(gb_s, plotit = FALSE, method = permutation.test.gbm)
imps <- sum$rel.inf
names(imps) <- sum$var
imps <- imps[c("Real", "Inve", "Arti", "Soci", "Ente", "Conv")]
plot(sqrt(imps), xaxt = "n", ylab = expression(sqrt(Importance)),
     col = "white", cex.lab = .7, cex.axis = .7, xlab = " ",
     main = "Gradient Boosted Ensemble", cex.main = .7)
text(sqrt(imps), labels = varnames_s, cex = 1,
     col = rep(qualitative_hcl(6)), font = 2)

## Random forest
library("ranger")
load(file = "RF_subscales.Rda")
imps <- ranger::importance(rf_s)
plot(sqrt(imps), xaxt = "n", ylab = expression(sqrt(Importance)),
     col = "white", cex.lab = .7, cex.axis = .7, xlab = " ",
     main = "Random Forest", cex.main = .7)
text(sqrt(imps), labels = varnames_s, cex = 1,
     col = rep(qualitative_hcl(6)), font = 2)
```

```
## Prediction rule ensemble
imps <- varimps_pre$imp
names(imps) <- varimps_pre$varname
imps <- imps[c("Real", "Inve", "Arti", "Soci", "Ente", "Conv")]
plot(imps, xaxt = "n", ylab = "Importance",
     col = "white", cex.lab = .7, cex.axis = .7, xlab = " ",
     cex.main = .7, main = "Prediction Rule Ensemble")
text(imps, labels = varnames_s, cex = 1,
     col = rep(qualitative_hcl(6)), font = 2)
```

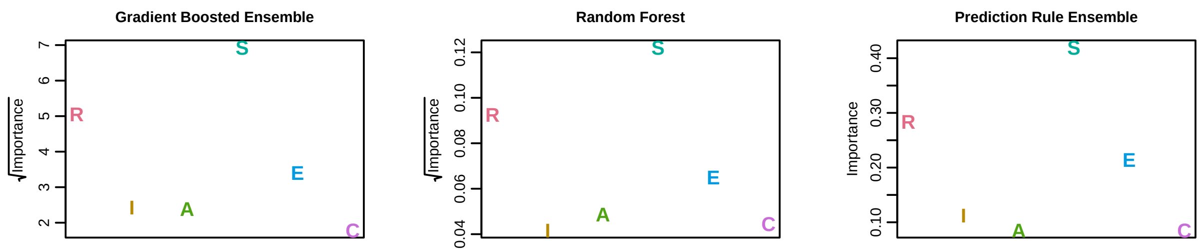


Figure 4

```
par(mar = c(1.5, 4, 0.2, 2), mgp = c(1.5, .5, 0), tck = -0.05)
par(mfrow = c(6, 1))

library("glmnet")
plot(coef(glmod_i)[-1], xaxt = "n", ylab = "Estimated coefficient",
     col = "white", cex.lab = .7, cex.axis = .7, xlab = " ",
     main = " ", cex.main = .7)
text(coef(glmod_i)[-1], labels = varnames_i, cex = .5,
     col = rep(qualitative_hcl(6), each = 8), font = 2)
abline(0, 0, col = "grey")
legend("topleft", legend = "Penalized logistic regression", cex = .7, bty = "n")
axis(1, 4.5 + c(0:5)*8, tick = FALSE, padj = -1.5,
     labels = c("Realistic", "Investigative", "Artistic", "Social" ,
                             "Enterprising", "Conventional"),
     cex.axis = .7)

library("mgcv")
load(file = "GAM_items.Rda")
sum <- summary(gamod_i)
plot(sqrt(sum$chi.sq), xaxt = "n", ylab = expression(sqrt(chi^2)),
     main = " ", cex.main = .7,
     col = "white", cex.lab = .7, cex.axis = .7, xlab = " ")
text(sqrt(sum$chi.sq), labels = varnames_i, cex = .5,
     col = rep(qualitative_hcl(6), each = 8), font = 2)
```

```
legend("topleft", legend = "Generalized Additive Model", cex = .7, bty = "n")
axis(1, 4.5 + c(0:5)*8, tick = FALSE, padj = -1.5,
     labels = c("Realistic", "Investigative", "Artistic", "Social" ,
                             "Enterprising", "Conventional"),
     cex.axis = .7)

ct6 <- cforest(ct_form,
               data = data[train_ids , ], ntree = 1L, mtry = 6,
               perturb = list(replace = FALSE, fraction = 1L),
               control = ctree_control(maxdepth = 6))
imps <- varimp(gettree(ct6), risk = "loglik")
imp_names <- names(imps)
imps <- c(imps, rep(0, times = 48 - length(imps)))
names(imps) <- c(imp_names, varnames_i[!varnames_i %in% imp_names])
plot(sqrt(imps[varnames_i]), xaxt = "n", ylab = expression(sqrt(Importance)),
     col = "white", cex.lab = .7, cex.axis = .7, xlab = " ",
     main = " ", cex.main = .7)
text(sqrt(imps[varnames_i]), labels = varnames_i, cex = .5,
     col = rep(qualitative_hcl(6), each = 8), font = 2)
legend("topleft", legend = "Tree", cex = .7, bty = "n")
axis(1, 4.5 + c(0:5)*8, tick = FALSE, padj = -1.5,
     labels = c("Realistic", "Investigative", "Artistic", "Social" ,
                             "Enterprising", "Conventional"),
     cex.axis = .7)

load(file = "gb_i_summary.Rda")
imps <- sum_i[match(varnames_i, sum_i$var), ]
plot(sqrt(imps$rel.inf), xaxt = "n", main = " ",
     ylab = expression(sqrt(Importance)), cex.main = .7,
     col = "white", cex.lab = .7, cex.axis = .7, xlab = " ")
text(sqrt(imps$rel.inf), labels = imps$var, cex = .5,
     col = rep(qualitative_hcl(6), each = 8), font = 2)
legend("topleft", legend = "Boosted ensemble", cex = .7, bty = "n")
axis(1, 4.5 + c(0:5)*8, tick = FALSE, padj = -1.5,
     labels = c("Realistic", "Investigative", "Artistic", "Social" ,
                             "Enterprising", "Conventional"),
     cex.axis = .7)

imps <- ranger::importance(rf_i)
plot(sqrt(imps), xaxt = "n", main = " ",
     col = "white", cex.lab = .7, cex.axis = .7, xlab = " ",
     ylab = expression(sqrt(Importance)), cex.main = .7)
text(sqrt(imps), labels = names(imps), cex = .5,
```

```r
     col = rep(qualitative_hcl(6), each = 8), font = 2)
legend("topleft", legend = "Random forest", cex = .7, bty = "n")
axis(1, 4.5 + c(0:5)*8, tick = FALSE, padj = -1.5,
     labels = c("Realistic", "Investigative", "Artistic", "Social" ,
                             "Enterprising", "Conventional"),
     cex.axis = .7)

imps <- pre::importance(pr_i, cex.axis = .7, plot = FALSE)$varimps
zero_vars <- varnames_i[!varnames_i %in% imps[ , 1]]
imps <- rbind(imps, data.frame(varname = zero_vars,
                               imp = rep(0, times = length(zero_vars))))
imps <- imps[match(varnames_i, imps$varname), ]
plot(imps$imp, xaxt = "n", main = " ",
     ylab = "Importance",
     col = "white", cex.lab = .7, cex.axis = .7, xlab = " ", cex.main = .7)
text(imps$imp, labels = imps$varname, cex = .5,
     col = rep(qualitative_hcl(6), each = 8), font = 2)
legend("topleft", legend = "Prediction rule ensemble", cex = .7, bty = "n")
axis(1, 4.5 + c(0:5)*8, tick = FALSE, padj = -1.5,
     labels = c("Realistic", "Investigative", "Artistic", "Social" ,
                             "Enterprising", "Conventional"),
     cex.axis = .7)
```

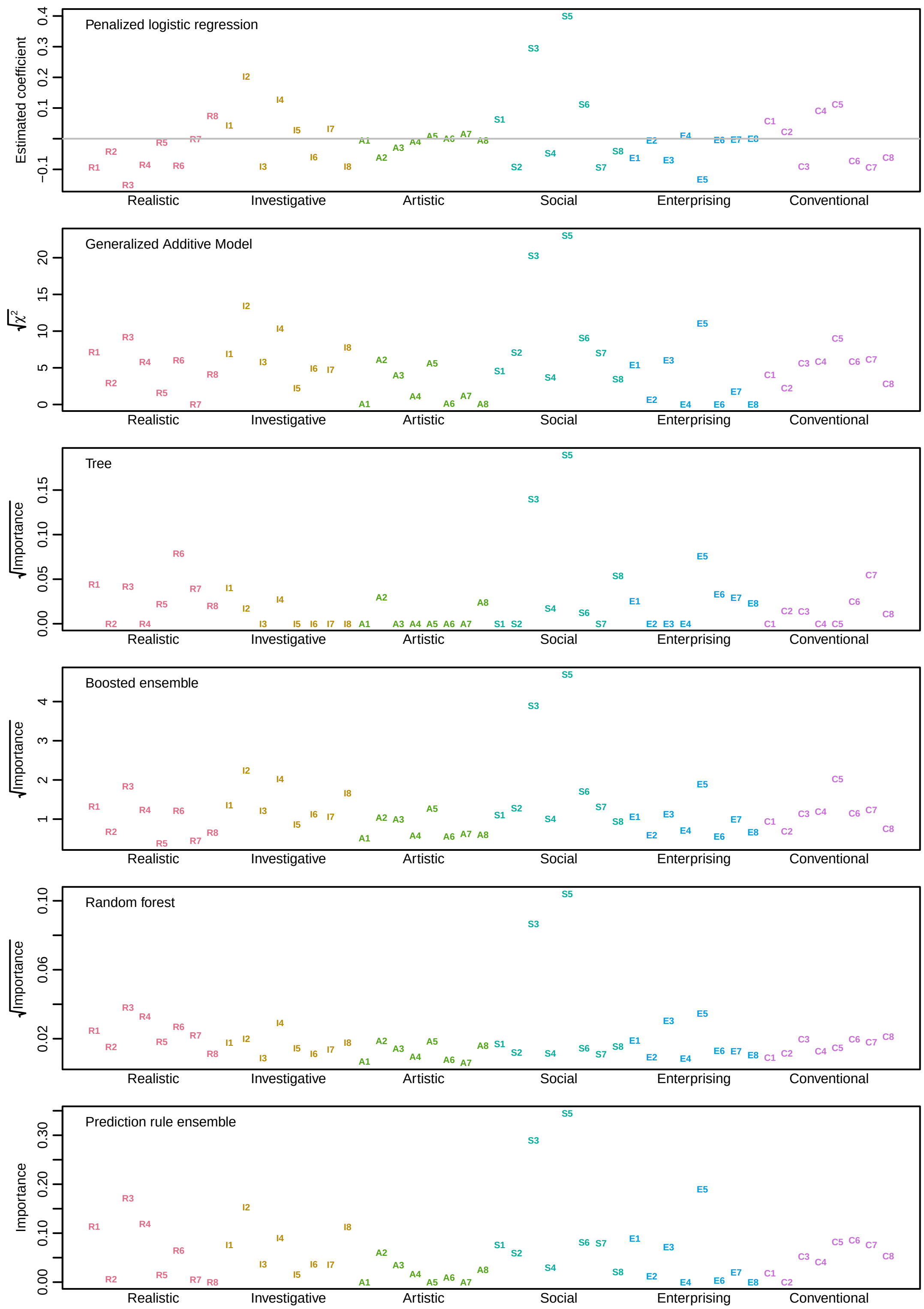
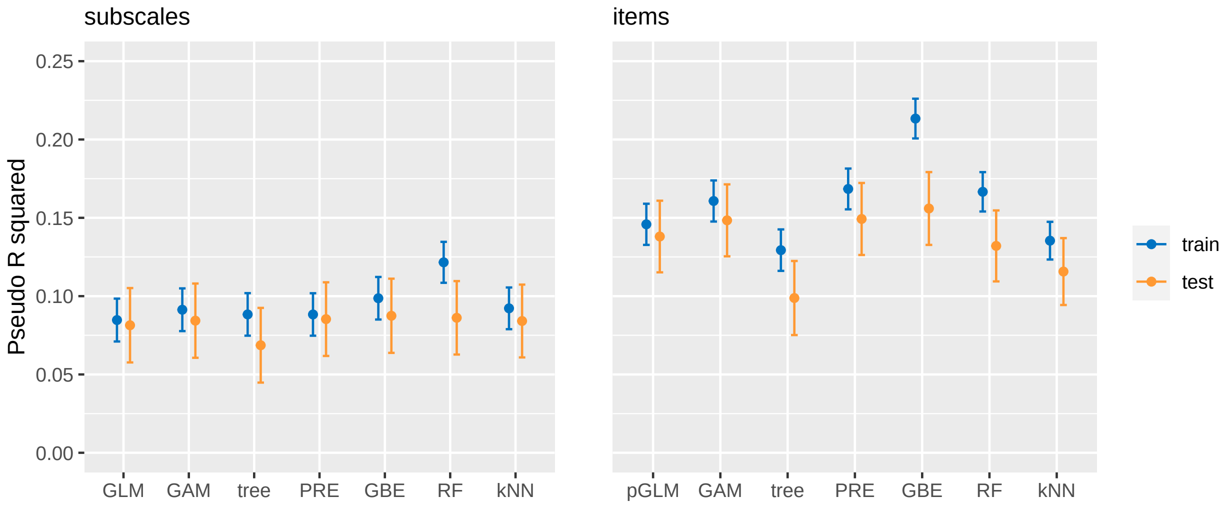

Penalized logistic regression
Estimated coefficient
Generalized Additive Model
$\sqrt{\chi^2}$
Tree
$\sqrt{\text{Importance}}$
Boosted ensemble
Random forest
Prediction rule ensemble
Importance
Realistic
Investigative
Artistic
Social
Enterprising
Conventional

Figure 5



subscales
items
Pseudo R squared
0.25
0.20
0.15
0.10
0.05
0.00
GLM
GAM
tree
PRE
GBE
RF
kNN
pGLM
GAM
tree
PRE
GBE
RF
kNN
train
test

Appendix B: Uni- and bivariate sample descriptives

```r
data$age[data$age > 103] <- NA
## some crazy ages were supplied
gender_tab <- prop.table(table(data$gender))
## 0=missing, 1=Male, 2=Female, 3=Other
marital_tab <- prop.table(table(data$married))
## 0=missing, 1=Never married, 2=Currently married, 3=Previously married
urban_tab <- prop.table(table(data$urban))
## 0=missing, 1=Rural (country side), 2=Suburban, 3=Urban (town, city)
race_tab <- prop.table(table(data$race))
## 0=missing, 1=Asian, 2=Arab, 3=Black,
## 4=Indigenous Australian / Native American / White, 5=Other
## (There was a coding mistake resulting in cat 4)
language_tab <- prop.table(table(data$engnat))
## 0=missing, 1=Yes, 2=No
```

```r
format(nrow(data), nsmall = 0, big.mark = ",")
```

```
## [1] "55,593"
```

```r
format(mean(data$age, na.rm = TRUE), digits = 2, nsmall = 2, big.mark = ",")
```

```
## [1] "33.06"
```

```r
format(sd(data$age, na.rm = TRUE), nsmall = 2, digits = 2, big.mark = ",")
```

```
## [1] "11.68"
```

```r
format(100*gender_tab[3], nsmall = 0, digits = 0, big.mark = ",")
```

```
##    2
## "67"
```

```r
format(100*gender_tab[2], nsmall = 0, digits = 0, big.mark = ",")
```

```
##    1
## "32"
```

```r
format(100*gender_tab[4], nsmall = 0, digits = 0, big.mark = ",")
```

```
##   3
## "1"
```

```r
format(100*marital_tab[2], nsmall = 0, digits = 0, big.mark = ",")
```

```
##    1
## "58"
```

```r
format(100*marital_tab[3], nsmall = 0, digits = 0, big.mark = ",")
```

```
##    2
## "33"
```

```r
format(100*marital_tab[4], nsmall = 0, digits = 0, big.mark = ",")
```

```
##   3
## "8"
```

```r
format(100*urban_tab[2], nsmall = 0, digits = 0, big.mark = ",")
```

```
##    1
## "22"
```

```r
format(100*urban_tab[3], nsmall = 0, digits = 0, big.mark = ",")
```

```
##    2
## "37"
```

```r
format(100*urban_tab[4], nsmall = 0, digits = 0, big.mark = ",")
```

```
##    3
## "40"
```

```r
format(100*language_tab[2], nsmall = 0, digits = 0, big.mark = ",")
```

```
##    1
## "66"
```

```r
format(100*language_tab[3], nsmall = 0, digits = 0, big.mark = ",")
```

```
##    2
## "34"
```

```r
format(100*race_tab[2], nsmall = 0, digits = 0, big.mark = ",")
```

```
##    1
## "20"
```

```r
format(100*race_tab[3], nsmall = 0, digits = 0, big.mark = ",")
```

```
##   2
## "1"
```

```r
format(100*race_tab[4], nsmall = 0, digits = 0, big.mark = ",")
```

```
##   3
## "7"
```

```r
format(100*race_tab[5], nsmall = 0, digits = 0, big.mark = ",")
```

```
##    4
```

```
## "61"
```

```r
format(100*race_tab[1], nsmall = 0, digits = 0, big.mark = ",")
```

```
##   0
## "1"
```

Figure 6

```r
par(mfrow = c(2, 3))
for (i in c("Real", "Inve", "Arti", "Soci" , "Ente", "Conv")) {
  dens <- density(data[ , i])
  plot(dens, main = "", xlab = i)
  polygon(dens, col="lightblue", border="black")
}
```

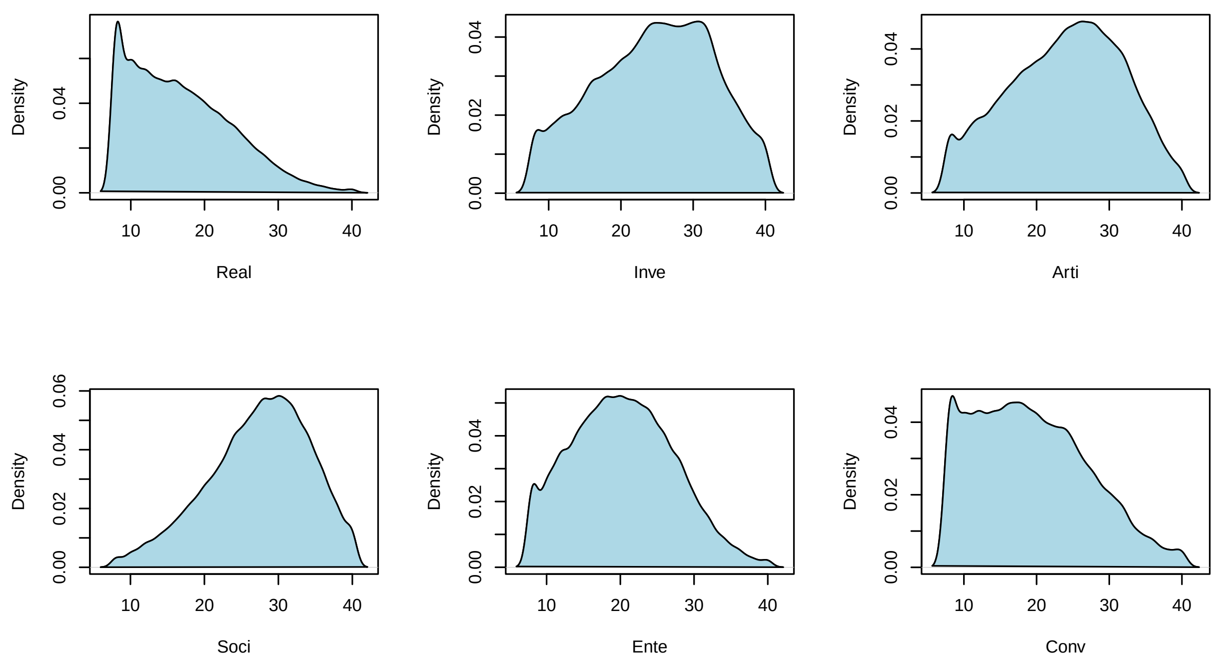


Table 2

```r
tab <- round(cor(data[ , c("Real", "Inve", "Arti", "Soci" , "Ente", "Conv")]),
             digits = 3L)
kable(tab)
```

	Real	Inve	Arti	Soci	Ente	Conv
Real	1.000	0.332	0.182	0.049	0.305	0.460
Inve	0.332	1.000	0.329	0.141	0.016	0.065
Arti	0.182	0.329	1.000	0.290	0.253	-0.056
Soci	0.049	0.141	0.290	1.000	0.356	0.124

	Real	Inve	Arti	Soci	Ente	Conv
Ente	0.305	0.016	0.253	0.356	1.000	0.464
Conv	0.460	0.065	-0.056	0.124	0.464	1.000

References

Fokkema, M. (2020). Fitting prediction rule ensembles with R package pre. *Journal of Statistical Software*, *92*(1), 1–30.

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, *33*(1), 1.

Greenwell, B., Boehmke, B., Cunningham, J., & Developers, G. (2020). *Gbm: Generalized boosted regression models*. https://CRAN.R-project.org/package=gbm

Hothorn, T., Hornik, K., & Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, *15*(3), 651–674. https://doi.org/10.1198/106186006X133933

Hothorn, T., & Zeileis, A. (2015). partykit: A modular toolkit for recursive partytioning in R. *Journal of Machine Learning Research*, *16*, 3905–3909. https://jmlr.org/papers/v16/hothorn15a.html

R Core Team. (2021). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Venables, W. N., & Ripley, B. D. (2002). *Modern applied statistics with S* (4th Edition). Springer. https://www.stats.ox.ac.uk/pub/MASS4/

Wood, S. N. (2017). *Generalized additive models: An introduction with R*. CRC press.

Wright, M. N., & Ziegler, A. (2017). ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, *77*(1), 1–17. https://doi.org/10.18637/jss.v077.i01