

STAT 230A Final Project Code

Andrew Du

2024-05-09

```
library(devtools)
```

```
## Loading required package: usethis
```

```
load_all("mfx")
```

```
## i Loading mfx
```

Probit Marginal Effects

Table 1 Redoing analyses from table 1, but with logit regression and robust standard error for all

Original table:

```
summary_volunteer_data <- read.dta("data/summary_volunteer.dta")
summary_volunteer_data$children <- ifelse(summary_volunteer_data$children == "yes", 1.0, 0.0)
summary_volunteer_data$bedroom <- ifelse(summary_volunteer_data$bedroom == "yes", 1.0, 0.0)
```

```
table_1_formulas <- list(
  volunteer ~ children,
  volunteer ~ married,
  volunteer ~ children + married + commute + bedroom,
  volunteer ~ children + married + commute + bedroom + high_educ + tenure,
  volunteer ~ children + married + commute + bedroom + high_educ + tenure + grosswage,
  volunteer ~ grosswage,
  volunteer ~ children + married + commute + bedroom + high_educ + tenure + grosswage + age + men
)
```

```
table_1_regressors <- c("age", "tenure", "grosswage", "children", "bedroom", "commute", "men", "married")
```

```
compare_logit_probit_fit <- function(formula, data) {
  model_probit <- glm(formula, data=data, family=binomial(link="probit"))
  model_logit <- glm(formula, data=data, family=binomial(link="logit"))
```

```
  cat("probit model summary:", "\n")
  print(summary(model_probit))
  cat("logit model summary:", "\n")
  print(summary(model_logit))
}
```

```
compare_logit_probit <- function(formula, data, robust=FALSE, ehw_type="HC1", parse=FALSE, regressors=table_1_regressors) {
  model_probit <- probitmfx(formula, data=data, robust=robust, robust_type=ehw_type)
  model_logit <- logitmfx(formula, data=data, robust=robust, robust_type=ehw_type)
```

```
  if (robust) {
```

```

    cat("probit model average marginal effect with", ehwt_type, "error:", "\n")
    print(model_probit$mfcest)
    cat("logit model average marginal effect with", ehwt_type, "error:", "\n")
    print(model_logit$mfcest)
  } else {
    cat("probit model average marginal effect with standard error:", "\n")
    print(model_probit$mfcest)
    cat("logit model average marginal effect with standard error:", "\n")
    print(model_logit$mfcest)
  }

  if (parse && !is.null(regressors)) {
    results <- format_logit_probit_results(model_probit$mfcest,
                                           model_logit$mfcest,
                                           regressors)

    return(results)
  }
}

format_logit_probit_results <- function(mfcest_log, mfcest_prob, regressors) {
  df_mfcest_log <- data.frame(mfcest_log)
  df_mfcest_prob <- data.frame(mfcest_prob)

  formatted <- data.frame(Info = vector("character", length(regressors)),
                           row.names=regressors,
                           stringsAsFactors=FALSE)

  for (feat in regressors) {
    if (feat %in% rownames(df_mfcest_log)) {
      se_log <- df_mfcest_log[feat, "Std..Err."]
      se_prob <- df_mfcest_prob[feat, "Std..Err."]
      ame_log <- df_mfcest_log[feat, "dF.dx"]
      ame_prob <- df_mfcest_prob[feat, "dF.dx"]

      formatted[feat, "Info"] <- sprintf("$\\substack{%.3f / %.3f \\\\ (%.4f / %.4f)}$",
                                          ame_log, ame_prob, se_log, se_prob)

    } else {
      formatted[feat, "Info"] <- ""
    }
  }

  return(formatted)
}

parse_significance <- function(p_val) {
  significance <- ifelse(p_val < 0.01, "***",
                          ifelse(p_val < 0.05, "**",
                                  ifelse(p_val < 0.1, "*", "")))

  return(significance)
}

format_hci_results <- function(mfcest_se, mfcest_hc0, mfcest_hc1, regressors) {

```

```

df_mfxest_se <- data.frame(mfxest_se)
df_mfxest_hc0 <- data.frame(mfxest_hc0)
df_mfxest_hc1 <- data.frame(mfxest_hc1)

formatted <- data.frame(Info = vector("character", length(regressors)),
                        row.names=regressors,
                        stringsAsFactors=FALSE)

for (feat in regressors) {
  if (feat %in% rownames(df_mfxest_se)) {
    signif_se <- parse_significance(df_mfxest_se[feat, "P..z."])
    signif_hc0 <- parse_significance(df_mfxest_hc0[feat, "P..z."])
    signif_hc1 <- parse_significance(df_mfxest_hc1[feat, "P..z."])

    assert_that(signif_se == signif_hc0 && signif_se == signif_hc1, msg="significance values are not

    se <- df_mfxest_se[feat, "Std..Err."]
    hc0 <- df_mfxest_hc0[feat, "Std..Err."]
    hc1 <- df_mfxest_hc1[feat, "Std..Err."]

    ame <- df_mfxest_se[feat, "dF.dx"]

    formatted[feat, "Info"] <- sprintf("$\\substack{%.3f %s \\\\ (%.4f / %.4f / %.4f)}$",
                                       ame, signif_se, se, hc0, hc1)

  } else {
    formatted[feat, "Info"] <- ""
  }
}

return(formatted)
}

dprobit <- function(formula,
                    data,
                    parse=FALSE,
                    regressors=NULL,
                    robust=TRUE,
                    robust_type="HC1") {
  model_probit <- probitmfx(formula, data=data, robust=robust, robust_type=robust_type)
  print(model_probit$mfxest)

  if (parse && !is.null(regressors)) {
    results <- format_dprobit_results(model_probit$mfxest,
                                      regressors)

    return(results)
  }
}

format_dprobit_results <- function(mfxest, regressors) {
  df_mfxest <- data.frame(mfxest)

  formatted <- data.frame(Info = vector("character", length(regressors)),

```

```

        row.names=regressors,
        stringsAsFactors=FALSE)

for (feat in regressors) {
  if (feat %in% rownames(df_mfxest)) {
    signif <- parse_significance(df_mfxest[feat, "P..z."])

    se <- df_mfxest[feat, "Std..Err."]
    ame <- df_mfxest[feat, "dF.dx"]

    formatted[feat, "Info"] <- sprintf("$\\substack{ %.3f %s \\\\ (%.4f) }$",
                                         ame, signif, se)

  } else {
    formatted[feat, "Info"] <- ""
  }
}

return(formatted)
}

compare_hci_errors <- function(formula, data, parse=FALSE, regressors=NULL) {
  model_probit_se <- probitmfx(formula, data=data, robust=FALSE)
  model_probit_hc0 <- probitmfx(formula, data=data, robust=TRUE, robust_type="HC0")
  model_probit_hc1 <- probitmfx(formula, data=data, robust=TRUE, robust_type="HC1")

  cat("probit model with standard errors:", "\n")
  print(model_probit_se$mfxest)
  cat("probit model with hc0 correction:", "\n")
  print(model_probit_hc0$mfxest)
  cat("probit model with hc1 correction:", "\n")
  print(model_probit_hc1$mfxest)

  if (parse && !is.null(regressors)) {
    results <- format_hci_results(model_probit_se$mfxest,
                                   model_probit_hc0$mfxest,
                                   model_probit_hc1$mfxest,
                                   regressors)

    return(results)
  }
}

```

```

child_reg <- glm(children ~ married + bedroom - 1, data=summary_volunteer_data, family=binomial(link="probit"),
summary(child_reg)

```

```

##
## Call:
## glm(formula = children ~ married + bedroom - 1, family = binomial(link = "probit"),
##      data = summary_volunteer_data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## married      2.1573      0.1809   11.92  <2e-16 ***
## bedroom     -2.1825      0.1486  -14.69  <2e-16 ***

```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1377.98  on 994  degrees of freedom
## Residual deviance:  789.65  on 992  degrees of freedom
## AIC: 793.65
##
## Number of Fisher Scoring iterations: 7
cor(summary_volunteer_data$married, summary_volunteer_data$bedroom)

## [1] 0.2661777
cor(summary_volunteer_data$married, summary_volunteer_data$children)

## [1] 0.7206834
cor(summary_volunteer_data$bedroom, summary_volunteer_data$children)

## [1] 0.2012989
other_formula <- married ~ bedroom
compare_hci_errors(other_formula, data=summary_volunteer_data)

## probit model with standard errors:
##           dF/dx   Std. Err.         z       P>|z|
## bedroom 0.1948745 0.01934402 10.07415 7.188268e-24
## probit model with hc0 correction:
##           dF/dx   Std. Err.         z       P>|z|
## bedroom 0.1948745 0.01934465 10.07382 7.212249e-24
## probit model with hc1 correction:
##           dF/dx   Std. Err.         z       P>|z|
## bedroom 0.1948745 0.01936414 10.06368 7.995376e-24

#other_formula <- volunteer ~ children + married + commute + bedroom
other_formula <- volunteer ~ married + commute
compare_hci_errors(other_formula, data=summary_volunteer_data)

## probit model with standard errors:
##           dF/dx   Std. Err.         z       P>|z|
## married 0.0678426226 0.0452026619 1.500855 0.133393175
## commute 0.0008716001 0.0002906053 2.999257 0.002706386
## probit model with hc0 correction:
##           dF/dx   Std. Err.         z       P>|z|
## married 0.0678426226 0.0452567727 1.499060 0.133858039
## commute 0.0008716001 0.0002871793 3.035038 0.002405058
## probit model with hc1 correction:
##           dF/dx   Std. Err.         z       P>|z|
## married 0.0678426226 0.0453252226 1.496796 0.134446286
## commute 0.0008716001 0.0002876137 3.030454 0.002441864
```

Make table 1

```
df_table <- data.frame(row.names=table_1_regressors)
for (f in table_1_formulas) {
  #compare_hci_errors(f, data=summary_volunteer_data,
  #                   parse=TRUE,
```

```

#               regressors=table_1_regressors)
col <- compare_hci_errors(f, data=summary_volunteer_data, parse=TRUE,
                        regressors=table_1_regressors)
df_table <- cbind(df_table, col)
#print(compare_hci_errors(f, data=summary_volunteer_data, parse=TRUE,
#                        regressors=table_1_regressors))
#cat("\n")
}

```

```

## probit model with standard errors:
##           dF/dx   Std. Err.         z       P>|z|
## children 0.1234432 0.05551546 2.223583 0.02617654
## probit model with hc0 correction:
##           dF/dx   Std. Err.         z       P>|z|
## children 0.1234432 0.05551735 2.223507 0.02618163
## probit model with hc1 correction:
##           dF/dx   Std. Err.         z       P>|z|
## children 0.1234432 0.05557329 2.221269 0.02633275
## probit model with standard errors:
##           dF/dx   Std. Err.         z       P>|z|
## married 0.0949605 0.04373423 2.171308 0.02990786
## probit model with hc0 correction:
##           dF/dx   Std. Err.         z       P>|z|
## married 0.0949605 0.04373465 2.171288 0.02990944
## probit model with hc1 correction:
##           dF/dx   Std. Err.         z       P>|z|
## married 0.0949605 0.04377872 2.169102 0.03007493
## probit model with standard errors:
##           dF/dx   Std. Err.         z       P>|z|
## children 0.0534996329 0.0828888974 0.6454379 0.518643473
## [ reached getOption("max.print") -- omitted 3 rows ]
## probit model with hc0 correction:
##           dF/dx   Std. Err.         z       P>|z|
## children 0.0534996329 0.0823888335 0.6493554 0.51610869
## [ reached getOption("max.print") -- omitted 3 rows ]
## probit model with hc1 correction:
##           dF/dx   Std. Err.         z       P>|z|
## children 0.0534996329 0.0825968339 0.6477201 0.517165975
## [ reached getOption("max.print") -- omitted 3 rows ]
## probit model with standard errors:
##           dF/dx   Std. Err.         z       P>|z|
## children 0.0747107340 0.0830494198 0.8995937 0.3683365166
## [ reached getOption("max.print") -- omitted 5 rows ]
## probit model with hc0 correction:
##           dF/dx   Std. Err.         z       P>|z|
## children 0.0747107340 0.0834068261 0.8957388 0.3703922593
## [ reached getOption("max.print") -- omitted 5 rows ]
## probit model with hc1 correction:
##           dF/dx   Std. Err.         z       P>|z|
## children 0.0747107340 0.0837020724 0.8925793 0.3720825292
## [ reached getOption("max.print") -- omitted 5 rows ]
## probit model with standard errors:
##           dF/dx   Std. Err.         z       P>|z|
## children 0.0813928116 0.0831738628 0.9785864 0.3277843776

```

```

## [ reached getOption("max.print") -- omitted 6 rows ]
## probit model with hc0 correction:
##          dF/dx      Std. Err.          z          P>|z|
## children    0.0813928116 0.0832636469  0.9775312 3.283062e-01
## [ reached getOption("max.print") -- omitted 6 rows ]
## probit model with hc1 correction:
##          dF/dx      Std. Err.          z          P>|z|
## children    0.0813928116 0.0836007480  0.9735895 0.3302603951
## [ reached getOption("max.print") -- omitted 6 rows ]
## probit model with standard errors:
##          dF/dx      Std. Err.          z          P>|z|
## grosswage -0.01916174 0.01702246 -1.125674 0.2603036
## probit model with hc0 correction:
##          dF/dx      Std. Err.          z          P>|z|
## grosswage -0.01916174 0.01710443 -1.120279 0.2625949
## probit model with hc1 correction:
##          dF/dx      Std. Err.          z          P>|z|
## grosswage -0.01916174 0.01712167 -1.119151 0.2630755
## probit model with standard errors:
##          dF/dx      Std. Err.          z          P>|z|
## children    0.0843345593 0.0838445433  1.0058443 0.3144904541
## [ reached getOption("max.print") -- omitted 8 rows ]
## probit model with hc0 correction:
##          dF/dx      Std. Err.          z          P>|z|
## children    0.0843345593 0.084162784  1.0020410 0.316323799
## [ reached getOption("max.print") -- omitted 8 rows ]
## probit model with hc1 correction:
##          dF/dx      Std. Err.          z          P>|z|
## children    0.0843345593 0.0845893597  0.9969878 0.3187704338
## [ reached getOption("max.print") -- omitted 8 rows ]

colnames(df_table) <- paste0("(", seq_along(df_table), ")")
latex_table <- xtable(df_table)
print(latex_table, type="latex", sanitize.text.function = function(x) {x}, tabular.environment = "longtable")

## Warning in print.xtable(latex_table, type = "latex", sanitize.text.function =
## function(x) {: Attempt to use "longtable" with floating = TRUE. Changing to
## FALSE.

## % latex table generated in R 4.3.2 by xtable 1.8-4 package
## % Thu May  9 23:42:49 2024
## \begin{longtable}{rllllllll}
##   \hline
##   & (1) & (2) & (3) & (4) & (5) & (6) & (7) & \\
##   \hline
##   age & & & & & & $\substack{-0.002}{\phantom{0}}$ & $\substack{(0.0073}{\phantom{0}}$ / 0.0073 / 0.0073)}$ & \\
##   tenure & & & & & & $\substack{-0.003}{\phantom{0}}$ & *** & $\substack{(0.0008}{\phantom{0}}$ / 0.0008 / 0.0008)}$ & & $\substack{-0.004}{\phantom{0}}$ & *** & \\
##   grosswage & & & & & & $\substack{0.048}{\phantom{0}}$ & ** & $\substack{(0.0239}{\phantom{0}}$ / 0.0231 / 0.0232)}$ & & $\substack{-0.019}{\phantom{0}}$ & \\
##   children & & & & & & $\substack{0.123}{\phantom{0}}$ & ** & $\substack{(0.0555}{\phantom{0}}$ / 0.0555 / 0.0556)}$ & & & $\substack{0.053}{\phantom{0}}$ & \\
##   bedroom & & & & & & $\substack{0.095}{\phantom{0}}$ & *** & $\substack{(0.0351}{\phantom{0}}$ / 0.0351 / 0.0352)}$ & & $\substack{0.088}{\phantom{0}}$ & ** & \\
##   commute & & & & & & $\substack{0.001}{\phantom{0}}$ & ** & $\substack{(0.0003}{\phantom{0}}$ / 0.0003 / 0.0003)}$ & & $\substack{0.001}{\phantom{0}}$ & ** & \\
##   men & & & & & & $\substack{0.010}{\phantom{0}}$ & & $\substack{(0.0359}{\phantom{0}}$ / 0.0361 / 0.0363)}$ & & \\
##   married & & & & & & $\substack{0.095}{\phantom{0}}$ & ** & $\substack{(0.0437}{\phantom{0}}$ / 0.0437 / 0.0438)}$ & & $\substack{0.012}{\phantom{0}}$ & \\
##   high_educ & & & & & & $\substack{-0.080}{\phantom{0}}$ & ** & $\substack{(0.0328}{\phantom{0}}$ / 0.0329 / 0.0330)}$ & & $\substack{-0.088}{\phantom{0}}$ & *** & \\
##   \hline

```

```
## \hline
## \end{longtable}
```

```
cat("\n")
```

```
#print(df_table)
```

Generate dprobit versus dlogit:

```
df_table <- data.frame(row.names=table_1_regressors)
for (f in table_1_formulas) {
  #compare_hci_errors(f, data=summary_volunteer_data,
  #                  parse=TRUE,
  #                  regressors=table_1_regressors)
  col <- compare_logit_probit(f, data=summary_volunteer_data,
                             robust=TRUE,
                             ehv_type="HC1",
                             parse=TRUE,
                             regressors=table_1_regressors)
  df_table <- cbind(df_table, col)
  #print(compare_hci_errors(f, data=summary_volunteer_data, parse=TRUE,
  #                          regressors=table_1_regressors))
  #cat("\n")
}
```

```
## probit model average marginal effect with HC1 error:
##          dF/dx  Std. Err.      z    P>|z|
## children 0.1234432 0.05557329 2.221269 0.02633275
## logit model average marginal effect with HC1 error:
##          dF/dx  Std. Err.      z    P>|z|
## children 0.1234432 0.05557329 2.221269 0.02633274
## probit model average marginal effect with HC1 error:
##          dF/dx  Std. Err.      z    P>|z|
## married 0.0949605 0.04377872 2.169102 0.03007493
## logit model average marginal effect with HC1 error:
##          dF/dx  Std. Err.      z    P>|z|
## married 0.0949605 0.04377872 2.169102 0.03007494
## probit model average marginal effect with HC1 error:
##          dF/dx  Std. Err.      z    P>|z|
## children 0.0534996329 0.0825968339 0.6477201 0.517165975
## [ reached getOption("max.print") -- omitted 3 rows ]
## logit model average marginal effect with HC1 error:
##          dF/dx  Std. Err.      z    P>|z|
## children 0.0543867670 0.0830714319 0.6546988 0.512661697
## [ reached getOption("max.print") -- omitted 3 rows ]
## probit model average marginal effect with HC1 error:
##          dF/dx  Std. Err.      z    P>|z|
## children 0.0747107340 0.0837020724 0.8925793 0.3720825292
## [ reached getOption("max.print") -- omitted 5 rows ]
## logit model average marginal effect with HC1 error:
##          dF/dx  Std. Err.      z    P>|z|
## children 0.0770554931 0.0843047411 0.9140114 0.36071086
## [ reached getOption("max.print") -- omitted 5 rows ]
## probit model average marginal effect with HC1 error:
##          dF/dx  Std. Err.      z    P>|z|
## children 0.0813928116 0.0836007480 0.9735895 0.3302603951
```



```
## [ reached getOption("max.print") -- omitted 6 rows ]
## logit model average marginal effect with HC1 error:
##          dF/dx      Std. Err.          z      P>|z|
## children  0.0835057626 0.0843924485  0.9894933 0.322421845
## [ reached getOption("max.print") -- omitted 6 rows ]
## probit model average marginal effect with HC1 error:
##          dF/dx      Std. Err.          z      P>|z|
## grosswage -0.01916174 0.01712167 -1.119151 0.2630755
## logit model average marginal effect with HC1 error:
##          dF/dx      Std. Err.          z      P>|z|
## grosswage -0.01918743 0.01713736 -1.119626 0.2628733
## probit model average marginal effect with HC1 error:
##          dF/dx      Std. Err.          z      P>|z|
## children  0.0843345593 0.0845893597  0.9969878 0.3187704338
## [ reached getOption("max.print") -- omitted 8 rows ]
## logit model average marginal effect with HC1 error:
##          dF/dx      Std. Err.          z      P>|z|
## children  0.0864253178 0.0852941353  1.0132621 0.3109349673
## [ reached getOption("max.print") -- omitted 8 rows ]

colnames(df_table) <- paste0("(", seq_along(df_table), ")")
latex_table_row1 <- xtable(df_table[, c("(1)", "(2)", "(3)", "(4)")]
latex_table_row2 <- xtable(df_table[, c("(5)", "(6)", "(7)")]
#print(latex_table_row1, type="latex", sanitize.text.function = function(x) {x}, tabular.environment="t
#print(latex_table_row2, type="latex", sanitize.text.function = function(x) {x}, tabular.environment="t
#latex_table <- xtable(df_table)
#print(latex_table, type="latex", sanitize.text.function = function(x) {x}, tabular.environment = "tabu
# cat("\n")
#print(subset(df_table[, c("(1)", "(2)", "(3)")])))

switch_return_data <- read.dta("data/switch_return.dta")
#switch_return_data
switch_return_data$performance_dif <- switch_return_data$perform11
- switch_return_data$perform10
```

Table 6

```
## [1] -0.0845730081 -0.0004498826 -0.9185097814 0.3774434924 -0.3699803054
## [6] -0.4509830475 -0.6507793069 0.0317654535 0.2581991553 -0.2486563027
## [11] -1.0376454592 -0.6090658307 -0.1598459482 -0.0464893170 0.2111089975
## [16] -0.5949486494 0.9680397511 -0.2427495569 -0.2511793375 1.0820868015
## [21] -0.5713105202 0.2363075912 -1.1186299324 1.0573406219 -0.4628151953
## [26] 0.7814708948 0.0617674813 0.0718049183 -0.6204665303 0.4670314193
## [31] -0.6298297644 -0.1280111521 -0.8753935099 -1.3078856468 0.4048437476
## [36] -1.1921101809 0.6225458384 0.7656851411 0.4459251761 -0.7029303312
## [41] -0.0444566086 0.7591874003 -0.5240535736 0.2395724058 1.2317928076
## [46] 0.7808128595 0.9718297720 0.5532929301 0.9237177372 -0.4030235112
## [51] 0.2256805152 -0.1617872864 -0.8232656717 0.6790611744 -0.1555806547
## [56] -0.7668246031 -0.8375025392 0.2822687328 0.5998853445 0.9045217037
## [61] -0.4364964366 0.0589389205 0.6215265989 -0.3656853735 0.0376971699
## [66] -0.3778011501 -0.1278285086 -1.2408034801 0.0626866892 -0.1584193408
## [71] 0.4900640547 0.1790713519 0.0876360461 -0.2288022488 -0.8202182651
## [76] -0.1043635160 -0.6349496245 -0.1211163998 -0.2523199022 0.0974182487
## [81] -1.5389206409 0.2920782268 -0.5695837736 -0.2456163615 -0.2292380780
```

```
## [86] 1.1420956850 -0.1489395350 -0.9487728477 -0.6681518555 -0.4824496806
## [91] 0.0946361050 0.0252965800 1.2596043348 0.1213412061 -0.3354152739
## [96] 0.1156113893 -0.0420185141 -0.3237470388 0.3795528710 1.1921372414
## [101] 0.9864637256 0.3989112973 0.0660753176 0.6394618154 0.6409057975
## [106] 1.1966077089 0.3330287933 0.1317722350 0.7438176274 0.0441721380
```

```
switch_home_data <- read.dta("data/switch_gohome.dta")
#switch_home_data
switch_home_data$performance_dif <- switch_home_data$perform11
- switch_home_data$perform10
```

```
## [1] -0.455909640 -0.655318975 -1.409042358 -0.270698756 -0.001532828
## [6] 0.687097669 -0.239500150 -0.320678741 0.718863904 0.845071018
## [11] -0.876527071 0.890312910 0.696573436 -0.347946048 -0.342568547
## [16] -0.797329903 0.050977934 -0.662377357 -0.032493778 -1.071249843
## [21] -0.088463545 -0.050123475 -0.713614941 -0.461394757 -0.401106209
## [26] 0.190205932 0.500678182 -0.011098285 0.147362828 0.578033030
## [31] -0.610499859 -1.181747317 0.666700721 0.274410397 0.136786744
## [36] -0.532272518 -0.428260326 -0.085549563 -0.105935223 -0.634947002
## [41] 0.814952672 0.266597152 0.104429536 -0.974455774 0.251256585
## [46] 0.315242410 -0.371437520 -0.413139373 -0.514282942 -0.014250415
## [51] 0.047367483 -0.409759969 0.377791524 0.379412979 -0.206144333
## [56] 0.217729464 0.312125683 0.151331916 -0.590668797 -0.413749725
## [61] -0.142845348 0.559072912 -0.267997116 0.070734940 0.041309230
## [66] -0.178749159 -0.122516744 0.811179399 -0.279303551 0.879110396
## [71] 0.317860454 0.361350089 1.101897836 0.993134797
```

```
#colnames(switch_home_data)
```

```
table_6a_formulas <- c(
  return ~ perform11,
  return ~ perform10,
  return ~ perform11 + perform10,
  return ~ perform11 + perform10 + married + livewparents + costofcommute,
  return ~ performance_dif,
  return ~ performance_dif + married + livewparents + costofcommute
)
```

```
table_6b_formulas <- c(
  join ~ perform11,
  join ~ perform10,
  join ~ perform11 + perform10,
  join ~ perform11 + perform10 + married + livewparents + costofcommute,
  join ~ performance_dif,
  join ~ performance_dif + married + livewparents + costofcommute
)
```

```
table_6_regressors <- c(
  "perform11",
  "perform10",
  "performance_dif",
  "married",
  "livewparents",
  "costofcommute"
)
```

```
df_table <- data.frame(row.names=table_6_regressors)
for (f in table_6a_formulas) {
  df_table <- cbind(df_table, dprobit(f, data=switch_return_data, robust=TRUE, parse=TRUE, regressors=t
  cat("\n")
}
```

```
##           dF/dx Std. Err.           z      P>|z|
## perform11 -0.07467966 0.05653193 -1.321017 0.1864956
##
##           dF/dx Std. Err.           z      P>|z|
## perform10 0.009213696 0.066916 0.1376905 0.890485
##
##           dF/dx Std. Err.           z      P>|z|
## perform11 -0.1684312 0.07656210 -2.199929 0.0278119
## perform10 0.1434162 0.09460363 1.515969 0.1295272
##
##           dF/dx Std. Err.           z      P>|z|
## perform11 -0.228647512 0.083128631 -2.750527 0.005949956
## perform10 0.214250906 0.096069470 2.230166 0.025736396
## married -0.213995493 0.076041583 -2.814190 0.004890026
## livewparents -0.186217304 0.105402065 -1.766733 0.077272958
## costofcommute -0.009734421 0.008552525 -1.138193 0.255040060
##
##           dF/dx Std. Err.           z      P>|z|
## performance_dif -0.07467966 0.05653193 -1.321017 0.1864956
##
##           dF/dx Std. Err.           z      P>|z|
## performance_dif -0.087649549 0.054779862 -1.6000323 0.10959141
## married -0.186073379 0.083966143 -2.2160525 0.02668791
## livewparents -0.189381893 0.103779920 -1.8248414 0.06802497
## costofcommute -0.008380183 0.009175566 -0.9133151 0.36107687
```

```
colnames(df_table) <- paste0("(", seq_along(df_table), ")")
latex_table <- xtable(df_table)
print(latex_table, type="latex", sanitize.text.function = function(x) {x})
```

```
## % latex table generated in R 4.3.2 by xtable 1.8-4 package
## % Thu May 9 23:42:49 2024
## \begin{table}[ht]
## \centering
## \begin{tabular}{rllllll}
## \hline
## & (1) & (2) & (3) & (4) & (5) & (6) \\
## \hline
## perform11 & & $\substack{-0.075 \\ \text{perform10}}$ & & $\substack{0.0565 \\ \text{performance\_dif}}$ & & $\substack{-0.168 \\ \text{married}}$ & & $\substack{0.0766 \\ \text{livewparents}}$ & & $\substack{0.0548 \\ \text{costofcommute}}$ \\
## & & $\substack{0.009 \\ \text{performance\_dif}}$ & & $\substack{0.143 \\ \text{married}}$ & & $\substack{0.0946 \\ \text{livewparents}}$ & & $\substack{0.0840 \\ \text{costofcommute}}$ \\
## & & $\substack{0.009 \\ \text{performance\_dif}}$ & & $\substack{0.143 \\ \text{married}}$ & & $\substack{0.0946 \\ \text{livewparents}}$ & & $\substack{0.0840 \\ \text{costofcommute}}$ \\
## & & $\substack{0.009 \\ \text{performance\_dif}}$ & & $\substack{0.143 \\ \text{married}}$ & & $\substack{0.0946 \\ \text{livewparents}}$ & & $\substack{0.0840 \\ \text{costofcommute}}$ \\
## & & $\substack{0.009 \\ \text{performance\_dif}}$ & & $\substack{0.143 \\ \text{married}}$ & & $\substack{0.0946 \\ \text{livewparents}}$ & & $\substack{0.0840 \\ \text{costofcommute}}$ \\
## \hline
## \end{tabular}
## \end{table}
```

```
cat("\n")
```

```
print(df_table)
```

```
## (1)
## perform11      $\backslash\backslash\backslash\backslash$ { -0.075  \\\ (0.0565) }$
## perform10
## performance_dif
## married
## livewparents
## costofcommute
## (2)
## perform11
## perform10      $\backslash\backslash\backslash\backslash$ { 0.009  \\\ (0.0669) }$
## performance_dif
## married
## livewparents
## costofcommute
## (3)
## perform11      $\backslash\backslash\backslash\backslash$ { -0.168 ** \\\ (0.0766) }$
## perform10      $\backslash\backslash\backslash\backslash$ { 0.143  \\\ (0.0946) }$
## performance_dif
## married
## livewparents
## costofcommute
## (4)
## perform11      $\backslash\backslash\backslash\backslash$ { -0.229 *** \\\ (0.0831) }$
## perform10      $\backslash\backslash\backslash\backslash$ { 0.214 ** \\\ (0.0961) }$
## performance_dif
## married      $\backslash\backslash\backslash\backslash$ { -0.214 *** \\\ (0.0760) }$
## livewparents  $\backslash\backslash\backslash\backslash$ { -0.186 * \\\ (0.1054) }$
## costofcommute $\backslash\backslash\backslash\backslash$ { -0.010 \\\ (0.0086) }$
## (5)
## perform11
## perform10
## performance_dif $\backslash\backslash\backslash\backslash$ { -0.075  \\\ (0.0565) }$
## married
## livewparents
## costofcommute
## (6)
## perform11
## perform10
## performance_dif $\backslash\backslash\backslash\backslash$ { -0.088 \\\ (0.0548) }$
## married      $\backslash\backslash\backslash\backslash$ { -0.186 ** \\\ (0.0840) }$
## livewparents  $\backslash\backslash\backslash\backslash$ { -0.189 * \\\ (0.1038) }$
## costofcommute $\backslash\backslash\backslash\backslash$ { -0.008 \\\ (0.0092) }$
```

```
df_table <- data.frame(row.names=table_6_regressors)
```

```
for (f in table_6b_formulas) {
```

```
  df_table <- cbind(df_table, dprobit(f, data=switch_home_data, robust=TRUE, parse=TRUE, regressors=tab
```

```
  cat("\n")
```

```
}
```

```
##          dF/dx  Std. Err.          z    P>|z|
## perform11 0.05662425 0.08202323 0.6903441 0.4899778
```

```
##
##          dF/dx Std. Err.          z      P>|z|
## perform10 0.03865825 0.1074493 0.3597813 0.7190107
##
##          dF/dx Std. Err.          z      P>|z|
## perform11 0.06285987 0.1067845 0.58866081 0.5560888
## perform10 -0.01103753 0.1441952 -0.07654575 0.9389849
##
##          dF/dx Std. Err.          z      P>|z|
## perform11 0.06335255 0.11076521 0.5719535 0.5673535
## perform10 -0.02129916 0.15319430 -0.1390336 0.8894236
## married 0.10138049 0.13551045 0.7481378 0.4543770
## livewparents 0.05630027 0.13996857 0.4022351 0.6875110
## costofcommute 0.01088855 0.01131784 0.9620700 0.3360144
##
##          dF/dx Std. Err.          z      P>|z|
## performance_dif 0.05662425 0.08202323 0.6903441 0.4899778
##
##          dF/dx Std. Err.          z      P>|z|
## performance_dif 0.05251605 0.08719543 0.6022798 0.5469879
## married 0.09914129 0.13474675 0.7357602 0.4618767
## livewparents 0.05966161 0.13428857 0.4442792 0.6568408
## costofcommute 0.01081508 0.01128594 0.9582790 0.3379221
```

```
colnames(df_table) <- paste0("(", seq_along(df_table), ")")
latex_table <- xtable(df_table)
#print(latex_table, type="latex", sanitize.text.function = function(x) {x})
#cat("\n")
#print(df_table)
```

Fixed Effects Regressions

```
exper_perf_data <- read_dta("data/performance_during_exper.dta")
exper_perf_data$during_experiment <- as.integer(exper_perf_data$year_week >= 201049)
exper_perf_data$experiment_treatment_during <- exper_perf_data$during_experiment * exper_perf_data$exper
#exper_perf_data
```

```
table_2_targets <- c(
  "perform1"
)
table_2_regressors <- c(
  c("experiment_treatment", "year_week", "personid")
)
```

```
cols_used <- c(table_2_regressors[1], table_2_targets[1])
table_2_regressors[1]
```

```
## [1] "experiment_treatment"
```

```
#cols_used
```

```
col_1_regression <- feelm(perform1 ~ experiment_treatment * + experiment_control + as.factor(year_week)
  | personid | 0 | personid,
  data=exper_perf_data,
  subset=(year_week != 201049 & (expgroup == 1 | expgroup == 0)))
```

```
## Warning in chol.default(mat, pivot = TRUE, tol = tol): the matrix is either
## rank-deficient or not positive definite

summary(col_1_regression)

## Warning in chol.default(mat, pivot = TRUE, tol = tol): the matrix is either
## rank-deficient or not positive definite

##
## Call:
##      felm(formula = perform1 ~ experiment_treatment * +experiment_control +      as.factor(year_week)
##
## Residuals:
##      Min        1Q    Median        3Q        Max
## -4.2689 -0.4187  0.0424  0.4687  3.7219
##
## Coefficients:
##                                     Estimate Cluster s.e. t value Pr(>|t|)
## experiment_treatment                2.12965      0.09312  22.871 < 2e-16
## experiment_control                 1.89729      0.09556  19.854 < 2e-16
## as.factor(year_week)201002          1.62942      0.08519  19.127 < 2e-16
## as.factor(year_week)201003          1.76952      0.08865  19.961 < 2e-16
## as.factor(year_week)201004          1.87433      0.09033  20.750 < 2e-16
## as.factor(year_week)201005          1.77772      0.08988  19.779 < 2e-16
## as.factor(year_week)201006          1.76600      0.08496  20.787 < 2e-16
## as.factor(year_week)201007          1.53386      0.07249  21.160 < 2e-16
##
## experiment_treatment                ***
## experiment_control                  ***
## as.factor(year_week)201002          ***
## as.factor(year_week)201003          ***
## as.factor(year_week)201004          ***
## as.factor(year_week)201005          ***
## as.factor(year_week)201006          ***
## as.factor(year_week)201007          ***
## [ reached getOption("max.print") -- omitted 79 rows ]
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7826 on 17472 degrees of freedom
## (705 observations deleted due to missingness)
## Multiple R-squared(full model): 0.4018   Adjusted R-squared: 0.3904
## Multiple R-squared(proj model): 0.1227   Adjusted R-squared: 0.106
## F-statistic(full model, *iid*):35.24 on 333 and 17472 DF, p-value: < 2.2e-16
## F-statistic(proj model): 22.69 on 87 and 248 DF, p-value: < 2.2e-16

col_1_regression_absorb_time <- felm(perform1 ~ experiment_treatment + experiment_control
| personid + year_week | 0 | personid,
data=exper_perf_data,
subset=(year_week != 201049 & (expgroup == 1 | expgroup == 0)))

## Warning in chol.default(mat, pivot = TRUE, tol = tol): the matrix is either
## rank-deficient or not positive definite

summary(col_1_regression_absorb_time)

## Warning in chol.default(mat, pivot = TRUE, tol = tol): the matrix is either
```

```
## rank-deficient or not positive definite

##
## Call:
##   felm(formula = perform1 ~ experiment_treatment + experiment_control |      personid + year_week |
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.2689 -0.4187  0.0424  0.4687  3.7219
##
## Coefficients:
##              Estimate Cluster s.e. t value Pr(>|t|)
## experiment_treatment    0.2324      0.0622   3.735 0.000233 ***
## experiment_control         NaN      0.0000      NaN      NaN
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7826 on 17472 degrees of freedom
## (705 observations deleted due to missingness)
## Multiple R-squared(full model): 0.4018   Adjusted R-squared: 0.3904
## Multiple R-squared(proj model): 0.005065   Adjusted R-squared: -0.0139
## F-statistic(full model, *iid*):35.24 on 333 and 17472 DF, p-value: < 2.2e-16
## F-statistic(proj model): 6.977 on 2 and 248 DF, p-value: 0.001128

yw_cluster_regression <- felm(perform1 ~ experiment_treatment + as.factor(year_week)
                             | personid | 0 | year_week + experiment_treatment,
                             data=exper_perf_data,
                             subset=(year_week != 201049 & (expgroup == 1 | expgroup == 0)))

## Warning in newols(mm, nostats = nostats[1], exactDOF = exactDOF, onlyse =
## onlyse, : Negative eigenvalues set to zero in multiway clustered variance
## matrix. See felm(...,psdef=FALSE)

summary(yw_cluster_regression)

## Warning in chol.default(mat, pivot = TRUE, tol = tol): the matrix is either
## rank-deficient or not positive definite

##
## Call:
##   felm(formula = perform1 ~ experiment_treatment + as.factor(year_week) |      personid | 0 | year_w
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.2689 -0.4187  0.0424  0.4687  3.7219
##
## Coefficients:
##              Estimate Cluster s.e. t value Pr(>|t|)
## experiment_treatment    0.232359      0.020917  11.109 <2e-16 ***
## as.factor(year_week)201002 1.629415      0.011471 142.049 <2e-16 ***
## as.factor(year_week)201003 1.769518      0.010299 171.807 <2e-16 ***
## as.factor(year_week)201004 1.874329      0.007855 238.625 <2e-16 ***
## as.factor(year_week)201005 1.777720      0.004540 391.548 <2e-16 ***
## as.factor(year_week)201006 1.766002      0.005304 332.942 <2e-16 ***
## as.factor(year_week)201007 1.533860      0.005002 306.643 <2e-16 ***
## as.factor(year_week)201008 1.677859      0.004554 368.414 <2e-16 ***
```

```
## [ reached getOption("max.print") -- omitted 77 rows ]
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7826 on 17472 degrees of freedom
## (705 observations deleted due to missingness)
## Multiple R-squared(full model): 0.4018    Adjusted R-squared: 0.3904
## Multiple R-squared(proj model): 0.1227    Adjusted R-squared: 0.106
## F-statistic(full model, *iid*):35.24 on 333 and 17472 DF, p-value: < 2.2e-16
## F-statistic(proj model): 2.714e+04 on 85 and 1 DF, p-value: 0.004829
```

Table 4

```
colnames(exper_perf_data)
```

```
## [1] "personid"          "year_week"
## [3] "expgroup"          "perform1"
## [5] "phonecall"         "phonecallraw"
## [7] "treatment"         "experiment_treatment"
## [9] "experiment_control" "experiment_nantong"
## [11] "homethatweek"      "experiment_home"
## [13] "logphonecall"      "logcallpersec"
## [15] "logcalllength"     "logcall_dayworked"
## [17] "logdaysworked"     "wage_month"
## [19] "experiment_treatment_commute120" "date"
## [21] "during_experiment" "experiment_treatment_during"
```

```
#exper_perf_data$experiment_control <- as.integer(exper_perf_data$expgroup == 0)
#exper_perf_data$experiment_treatment <- as.integer(exper_perf_data$expgroup == 1 & exper_perf_data$year_week == 201002)
#exper_perf_data$experiment_nantong <- as.integer(exper_perf_data$expgroup == 3)
#exper_perf_data
```

```
col_1_regression <- felm(perform1 ~ experiment_treatment_during
+ experiment_control
+ as.factor(year_week)
| personid | 0 | personid,
data=exper_perf_data,
subset=(year_week != 201049 & (expgroup != 2)))
summary(col_1_regression)
```

```
##
## Call:
## felm(formula = perform1 ~ experiment_treatment_during + experiment_control + as.factor(year_week),
## data = exper_perf_data, subset = (year_week != 201049 & (expgroup != 2)))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.8705 -0.4650  0.0404  0.5166  3.9519
##
## Coefficients:
##              Estimate Cluster s.e. t value Pr(>|t|)
## experiment_treatment_during  0.19396      0.04637   4.183 3.02e-05 ***
## experiment_control          -0.03467      0.04720  -0.734   0.463
## as.factor(year_week)201002   1.80792      0.04288  42.161 < 2e-16 ***
## as.factor(year_week)201003   2.06448      0.04329  47.684 < 2e-16 ***
## as.factor(year_week)201004   2.15205      0.04512  47.698 < 2e-16 ***
## as.factor(year_week)201005   2.21217      0.04481  49.371 < 2e-16 ***
```



```
## as.factor(year_week)201006 2.11384 0.04329 48.828 < 2e-16 ***
## as.factor(year_week)201007 1.38370 0.04088 33.851 < 2e-16 ***
## [ reached getOption("max.print") -- omitted 78 rows ]
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8041 on 97926 degrees of freedom
## (1071 observations deleted due to missingness)
## Multiple R-squared(full model): 0.3801 Adjusted R-squared: 0.3685
## Multiple R-squared(proj model): 0.1506 Adjusted R-squared: 0.1348
## F-statistic(full model, *iid*):32.88 on 1826 and 97926 DF, p-value: < 2.2e-16
## F-statistic(proj model): 106.2 on 86 and 1740 DF, p-value: < 2.2e-16
```

```
col_2_regression <- felm(phonecall ~ experiment_treatment_during
+ experiment_control
+ as.factor(year_week)
| personid | 0 | personid,
data=exper_perf_data,
subset=(year_week != 201049 & (expgroup != 2)))
summary(col_2_regression)
```

```
##
## Call:
## felm(formula = phonecall ~ experiment_treatment_during + experiment_control + as.factor(year_week),
## data = exper_perf_data, subset = (year_week != 201049 & (expgroup != 2)))
## Residuals:
## Min 1Q Median 3Q Max
## -4.8708 -0.4535 0.0516 0.5169 4.4633
##
## Coefficients:
## Estimate Cluster s.e. t value Pr(>|t|)
## experiment_treatment_during 0.28073 0.04730 5.936 3.67e-09 ***
## experiment_control -0.01091 0.04256 -0.256 0.798
## as.factor(year_week)201002 1.95374 0.04261 45.850 < 2e-16 ***
## as.factor(year_week)201003 2.21211 0.04314 51.275 < 2e-16 ***
## as.factor(year_week)201004 2.31134 0.04510 51.252 < 2e-16 ***
## as.factor(year_week)201005 2.36769 0.04529 52.274 < 2e-16 ***
## as.factor(year_week)201006 2.27925 0.04277 53.286 < 2e-16 ***
## as.factor(year_week)201007 1.55329 0.04512 34.424 < 2e-16 ***
## [ reached getOption("max.print") -- omitted 78 rows ]
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.799 on 85086 degrees of freedom
## (14235 observations deleted due to missingness)
## Multiple R-squared(full model): 0.365 Adjusted R-squared: 0.3538
## Multiple R-squared(proj model): 0.1809 Adjusted R-squared: 0.1664
## F-statistic(full model, *iid*):32.56 on 1502 and 85086 DF, p-value: < 2.2e-16
## F-statistic(proj model): 165.5 on 86 and 1416 DF, p-value: < 2.2e-16
```

```
col_3_regression <- felm(perform1 ~ experiment_treatment_during
+ experiment_control
+ as.factor(year_week)
| personid | 0 | personid,
data=exper_perf_data,
```

```
subset=(year_week != 201049 & (expgroup != 3)))
summary(col_3_regression)
```

```
##
## Call:
##   felm(formula = perform1 ~ experiment_treatment_during + experiment_control +      as.factor(year_
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.3001 -0.4253  0.0384  0.4730  3.7797
##
## Coefficients:
##              Estimate Cluster s.e. t value Pr(>|t|)
## experiment_treatment_during  0.30153      0.05914   5.099  5.1e-07 ***
## experiment_control          0.06634      0.06047   1.097   0.273
## as.factor(year_week)201002   1.68219      0.07553  22.271 < 2e-16 ***
## as.factor(year_week)201003   1.79349      0.07771  23.080 < 2e-16 ***
## as.factor(year_week)201004   1.88997      0.07855  24.061 < 2e-16 ***
## as.factor(year_week)201005   1.83249      0.07764  23.604 < 2e-16 ***
## as.factor(year_week)201006   1.83239      0.07498  24.438 < 2e-16 ***
## as.factor(year_week)201007   1.50369      0.06862  21.915 < 2e-16 ***
## [ reached getOption("max.print") -- omitted 78 rows ]
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7702 on 27298 degrees of freedom
## (705 observations deleted due to missingness)
## Multiple R-squared(full model): 0.4281 Adjusted R-squared: 0.4171
## Multiple R-squared(proj model): 0.1122 Adjusted R-squared: 0.09513
## F-statistic(full model, *iid*): 39 on 524 and 27298 DF, p-value: < 2.2e-16
## F-statistic(proj model): 23.38 on 86 and 438 DF, p-value: < 2.2e-16
```

```
col_4_regression <- felm(phonecall ~ experiment_treatment_during
+ experiment_control
+ as.factor(year_week)
| personid | 0 | personid,
data=exper_perf_data,
subset=(year_week != 201049 & (expgroup != 3)))
summary(col_4_regression)
```

```
##
## Call:
##   felm(formula = phonecall ~ experiment_treatment_during + experiment_control +      as.factor(year_
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.2909 -0.3690  0.0782  0.4828  4.6398
##
## Coefficients:
##              Estimate Cluster s.e. t value Pr(>|t|)
## experiment_treatment_during  0.31206      0.06393   4.882 1.97e-06 ***
## experiment_control          0.01919      0.06066   0.316   0.752
## as.factor(year_week)201002   2.05485      0.09486  21.662 < 2e-16 ***
## as.factor(year_week)201003   2.13822      0.09927  21.539 < 2e-16 ***
```

```

## as.factor(year_week)201004  2.33876      0.09967  23.466 < 2e-16 ***
## as.factor(year_week)201005  2.19691      0.10003  21.962 < 2e-16 ***
## as.factor(year_week)201006  2.25662      0.09505  23.741 < 2e-16 ***
## as.factor(year_week)201007  1.77515      0.09198  19.299 < 2e-16 ***
## [ reached getOption("max.print") -- omitted 78 rows ]
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.787 on 14945 degrees of freedom
## (13267 observations deleted due to missingness)
## Multiple R-squared(full model): 0.4242    Adjusted R-squared: 0.412
## Multiple R-squared(proj model): 0.1513    Adjusted R-squared: 0.1334
## F-statistic(full model, *iid*):34.95 on 315 and 14945 DF, p-value: < 2.2e-16
## F-statistic(proj model): 28.69 on 86 and 229 DF, p-value: < 2.2e-16

```