MY457: Reappraisal of Factorial Survey Experiments to Predict Real-World Behavior: A Cautionary Tale from Hiring Studies

49187

Tue/27/May

1 Main results replication

The authors' main argument is that Factorial Survey Experiments (FS) are not suitable for measuring real-world behavior but only attitudes and judgments. To test their argument they focus on a case study from hiring studies, namely employer hiring decisions. More specifically, focusing on entry-level apprenticeship positions in Germany, they conduct two experiments; a Field Experiment (FE) and a Factorial Survey Experiment (FS). Their intention is to use the results from the FE experiment as a benchmark for comparison, assuming it has measured the "true" effect, and then compare the FS one to that in order to verify or reject their claim. The outcome of interest is the likelihood of a job applicant receiving an invitation to a job interview. For this reason, they create fictitious job applications where they randomly vary some applicant attributes to estimate their effects. More specifically, the main treatments they are interested in are the applicant's ethnic background (German or Turkish) and the applicant's education (on the one hand, the applicant having upper secondary high school degree -"abitur"- and having attended some college but without degree, and on the other hand him only having "abitur"), since it has been extensively argued in the literature that these two attributes tend to influence employers' decisions in Germany.

For the estimation of the Average Treatment Effect (ATE) they used regressions (also controlling for some additional factors like experimental wave, occupation etc, in order to increase precision). Their main hypothesis (H1) is that the FS experiment could more easily mirror the "true" FE effect of higher education non-completion on the likelihood of receiving an invitation for a job, than the effect of the applicant's ethnic background. Figure 1 and 2 replicates the two effects (the difference in predicted probability, holding the other predictors at their mean), by experiment. The authors conclude that, in contrast to their expectations, FS cannot replicate FE in neither experimental condition, leading them to reject H1.

The authors argue that there are some psychological factors in the respondents, like social desirability bias (SDB) and the effort dedicated for the completion of the survey, that distort the estimation of the "true" effect in FS survey. For this reason, they examine these factors in their follow-up hypotheses.

Their second hypothesis is that the two treatments may be better replicated in the FS for respondents who are less prone to give socially desirable answers (i.e. they are on the lower quantile of the social desirability index). But again, as it is evident from Figure 3 they fail to confirm their hypothesis since the ethnic background as well as the higher education non-completion effects are insignificant across all the groups in the FS survey, compared with the significant FE effects.

Finally, the authors also reject their third hypothesis (H3) which states that both the treatment effects should be more easily replicated in FS for those respondents who show a higher level of effort when completing the survey (where effort is measured as time spent completing the survey). This is evident from Figure 4 which shows that for all the response-time levels, both of the effects cannot approximate the FE effects, since they exhibit different direction and pretty much null statistical significance.

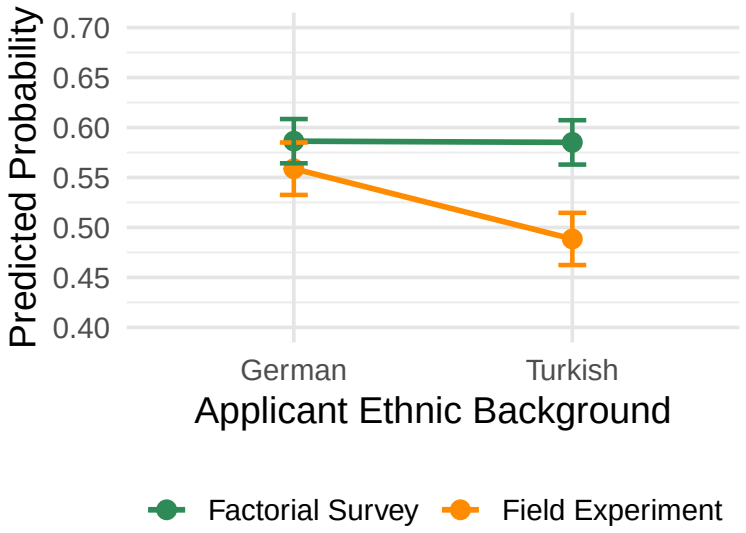

Figure 1: Predicted probabilities of invitation for job interview by ethnic background

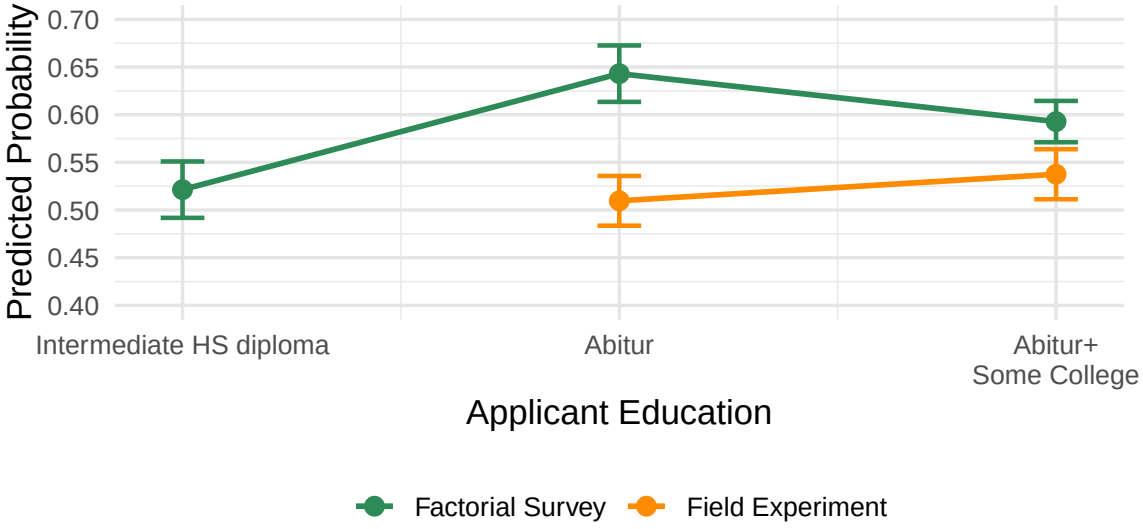

Figure 2: Predicted probabilities of invitation for job interview by higher education non-completion

2 Critical re-evaluation

In general terms the authors deserve a credit for including a wide variety of robustness checks to further amplify their arguments. Since the FS experiment is by design different than the FE experiment, there is a possibility that several factors may compromise the accurate comparison of the two. Most importantly measurement error in the FS experiment may come from several factors, among which are the slightly different measurement of the outcome compared with the FE experiment, the fact that the recruiters participating in the FS survey may not be the same ones that participated in the FE, and the fact that since the FS experiment is a survey it may inevitable encounter the selection bias problem. The supplementary material addresses, in much detail, those and many other important issues.

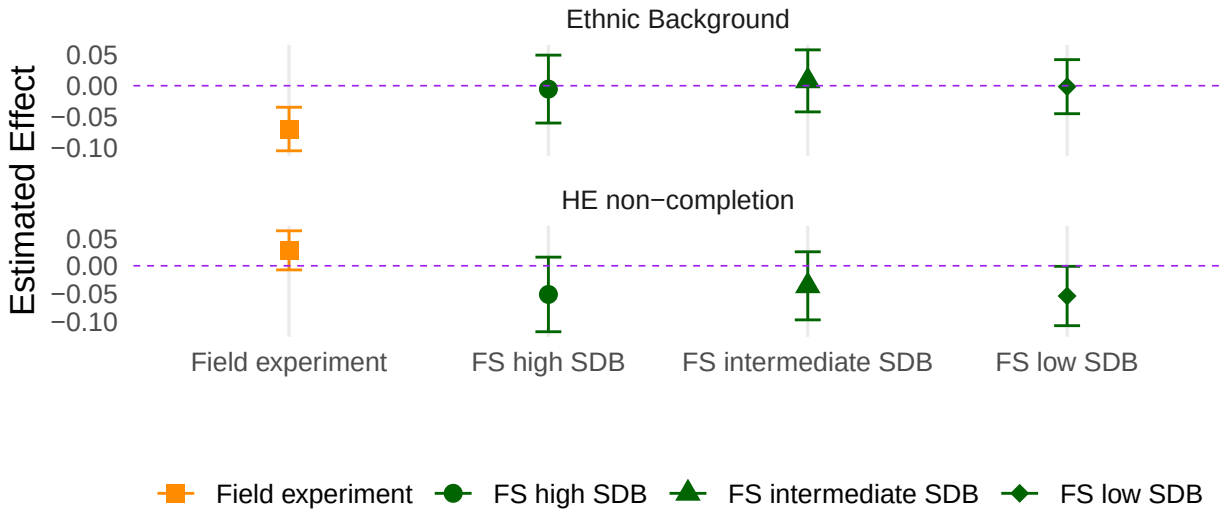

Figure 3: Effects of Ethnic Background and HE Non-completion by Social Desirability, in FE and FS experiments

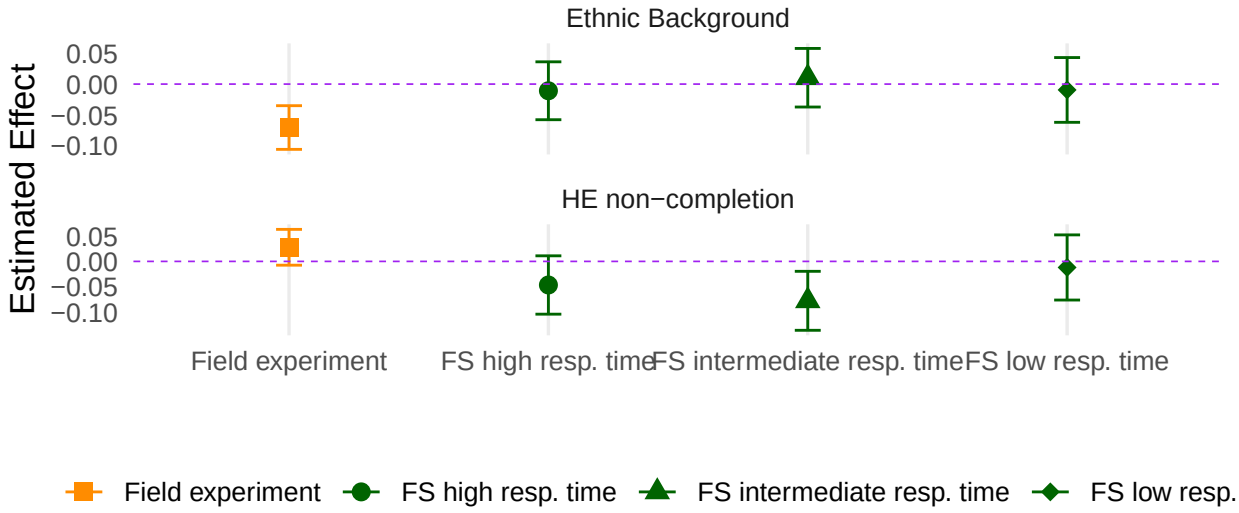

Figure 4: Effects of Ethnic Background and HE Non-completion by Response Time, in FE and FS experiments

However, the authors do not thoroughly discuss potential design issues, rather they solely focus on measurement problems. For example, since randomized experiments were implemented, one informative thing for someone to check could be if randomization worked properly. To that extent, I created balance tables for both experiments. After inspection of tables 1 and 2, it seems that both experiments achieve the balancing property, since covariates in both cases are balanced between treatment and control groups and all t-statistics are insignificant, thus indicating that the unconfoundedness assumption is successfully met.

Table 1: FE Experiment: Balance Table (Percentages)

variable	German	Turkish	Abitur	Abitur+Some College	t-statistic	p-value
Gender: Female	50.6	49.9	50.3	50.2	0.37	0.71
Gender: Male	49.4	50.1	49.7	49.8	-0.37	0.71
Occupation: Electronics Technician	21.9	21.8	22.3	21.3	0.09	0.93
Occupation: Laboratory Technician	13.2	13.3	13.6	12.9	-0.12	0.91
Occupation: Administration Clerk	42.0	42.0	41.3	42.7	-0.04	0.97
Occupation: Media Clerk	23.0	22.9	22.8	23.0	0.05	0.96
wave = 1	52.9	52.9	52.9	52.9	0.05	0.96
wave = 2	47.1	47.1	47.1	47.1	-0.05	0.96

Table 2: FS Experiment: Balance Table (Percentages)

variable	German	Turkish	Intermediate HS	Abitur	Abitur+Some College	t-statistic	p-value
Gender: Female	50.0	50.0	50.3	50.8	49.4	0.00	1.00
Gender: Male	50.0	50.0	49.7	49.2	50.6	0.00	1.00
Occupation: Electronics Technician	13.3	13.3	13.3	13.3	13.3	0.00	1.00
Occupation: Laboratory Technician	14.6	14.6	14.6	14.6	14.6	0.00	1.00
Occupation: Administration Clerk	50.6	50.6	50.6	50.6	50.6	0.00	1.00
Occupation: Media Clerk	21.5	21.5	21.5	21.5	21.5	0.00	1.00
wave = 1	44.8	44.8	44.8	44.8	44.8	0.00	1.00
wave = 2	55.2	55.2	55.2	55.2	55.2	0.00	1.00
Intermediate SES	33.6	33.9	34.7	33.8	33.2	-0.17	0.86
Low SES	32.8	33.4	33.4	33.0	33.0	-0.38	0.71
High SES	33.6	32.8	31.9	33.2	33.8	0.55	0.58
Low Grades	33.6	33.4	33.3	33.3	33.7	0.17	0.86
High Grades	33.4	33.2	33.8	33.6	32.9	0.10	0.92
Intermediate Grades	33.0	33.4	32.9	33.0	33.4	-0.27	0.78

However, in the FS experiment, according to the authors, randomization involved two steps. In the first step, it was implemented in order to design the vignettes (random assignment of the vignette attributes). In the second step, it was implemented for assigning a set of vignettes to the respondents. According to the authors, they created a pool of 18 sets of 8 vignettes each and randomly assigned one set to each respondent. For this reason, we need to also check balance of the assigned vignettes across the participants.

Although simple balance tables show almost perfect randomization of ethnic background and education across each vignette attribute, balance on a single margin does not guarantee that the full joint distribution of the six experimental conditions is balanced as well. Put differently, some specific vignettes may be shown far more (or far less), which in practice erodes the intended orthogonality of the design.

To account for this imbalance I estimated a measure of deviation of each vignette as follows:

$$d = 100(\frac{n_c}{N/cells}) - 100 \quad (1)$$

, where d stands for deviation, n_c is the realized count of a specific vignette and $N/cells$ is the count expected under perfect balance, where N indicates the total number of vignette ratings and $cells$ the total number of distinct vignettes. By multiplying with 100 it is converted to a percentage and by substracting 100 it is centered around zero so that it can be interpreted as percentage deviation from zero, where zero indicates perfect balance.

Interestingly, only 71 vignettes lay within +- 20% deviation; extremes ranged from –80 % to + 315%. As it is evident from Figure 5 most of the vignettes were over and under-represented. Unfortunately, we do not have the data to accurately explain why that happened, but the most probable reason is sampling error.It is very possible that this happened because some vignettes were included in more sets and others in less sets. The authors mention that the original vignettes "were allocated in 18 sets of eight vignettes following principles of d-efficiency", but they do not mention more on this process, neither in the main text nor in the supplementary material. So there may be the case that an error happened during that process and led to unequal distribution of the vignettes across sets.

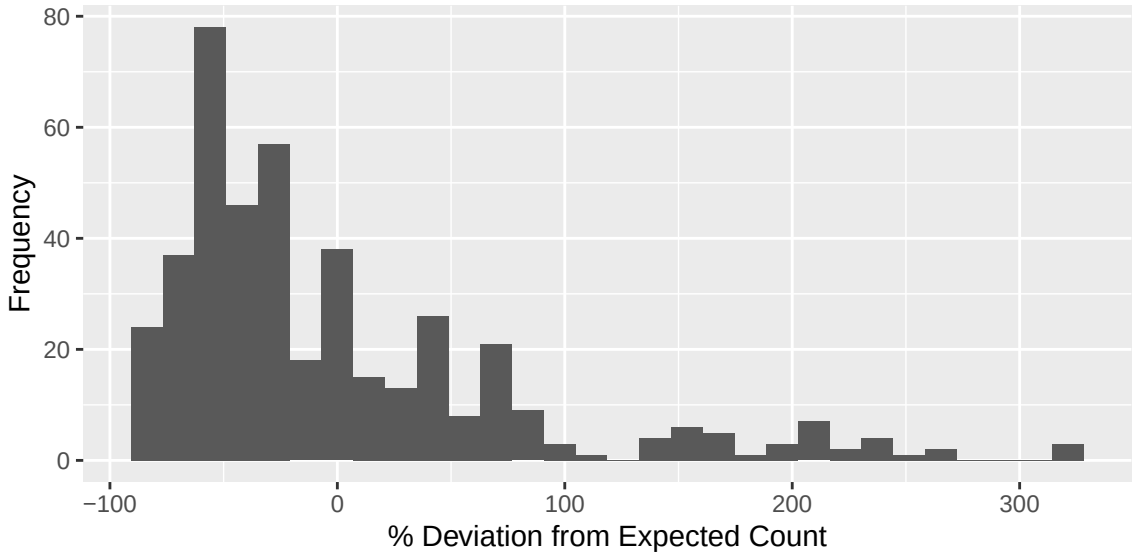

Figure 5: Distribution of vignette deviations

3 A new perspective

Since the existence of imbalance is implied, we can assume that the orthogonality assumption of the factorial survey design is violated. This could lead to estimates of the effects that are biased or imprecise. For this reason I restricted the data only for vignettes that are in the +- 20% deviation range, and then I rerun the paper's regressions to estimate the ethnic background and higher education non-completion effects, and, accordingly, check whether the FS survey can replicate the FE results. It should be noted however that after trimming, the remaining number of observations is 636. For this reason this can lead to higher standard errors, which can lead to smaller t-statistic and thus non-significant p-values. However, we can at least approximate the direction and magnitude of the effects and make comparisons with the FE experiment. In figures 6 and 7, I plot the effects of ethnic background and higher education non-completion as predicted probabilities for each level (holding the rest of the predictors at their means) in order to facilitate comparison with the author's original plots (Figure 1 and 2).

Interestingly, we can support exactly the opposite statement that the authors suggested. The FS now is closely replicating the FE results for both effects. The ethnic background effect is positive and of approximately the same magnitude in both the FE and FS experiments. The same applies for the higher education non-completion effect. In contrast, the authors found an almost zero ethnic background effect on the FS

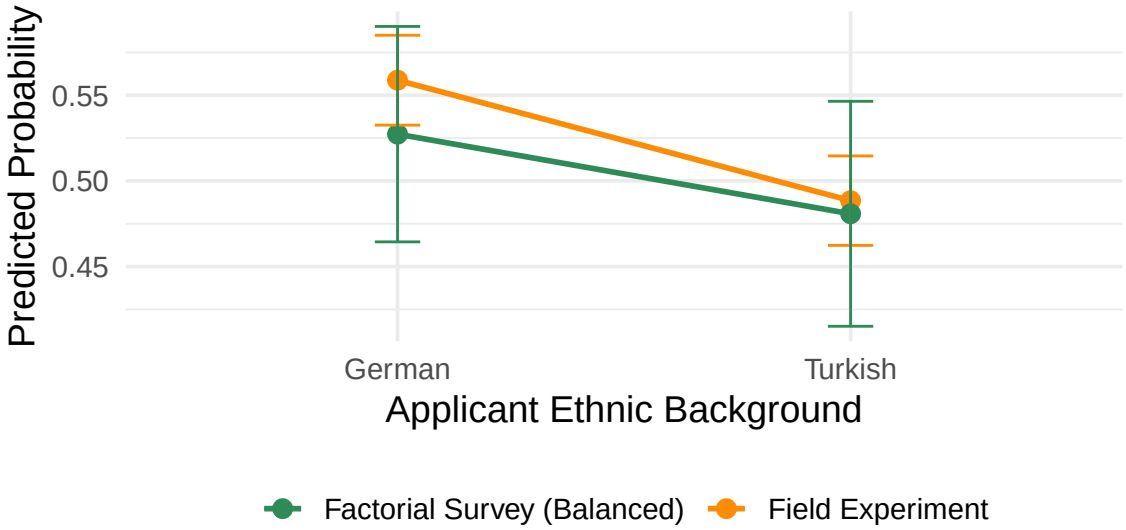

Figure 6: Predicted Probability of invitation for job interview by ethnic background

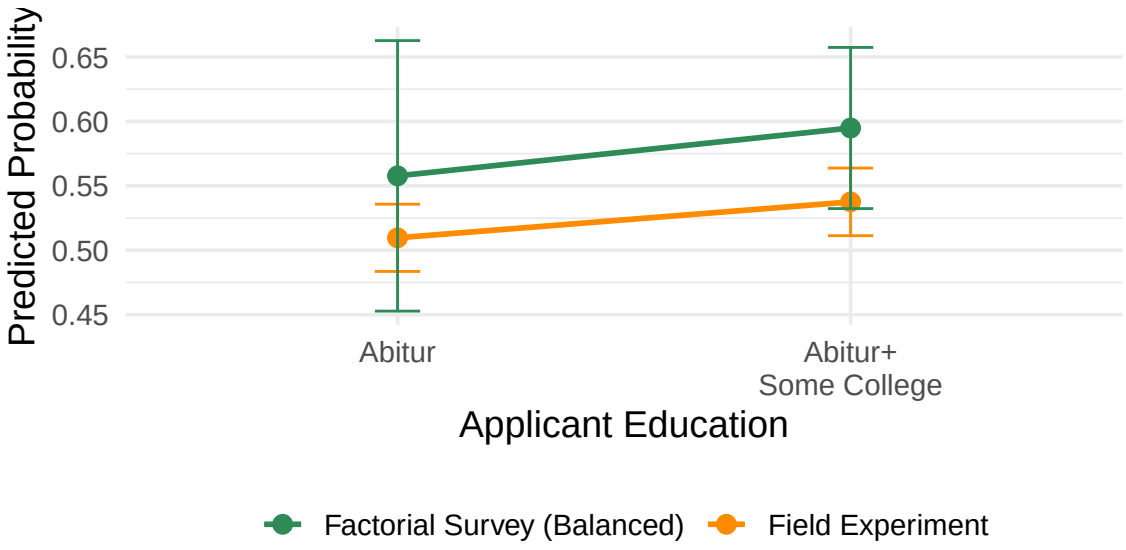

Figure 7: Predicted Probability of invitation for job interview by HE non-completion

experiment and a negative higher education non-completion effect on the same experiment.

These results suggest that the failure of the FS experiment to replicate the FE effect in the original analysis may stem from imbalance in the FS design rather than the psychological mechanisms (e.g., social desirability or survey effort) tested in H2 and H3, or other measurement errors that the authors very extensively addressed and tackled. Moreover, for this reason, I will not rerun the regressions for H2 and H3 on the trimmed dataset, since the discrepancy between FS and FE effects no longer exists when design imbalance is addressed. Running subgroup analyses (by social desirability or response time) under these conditions would aim to explain a gap that is no longer observed and would only introduce unnecessary complexity without advancing the core argument.

4 Conclusion

This reappraisal highlights a critical limitation in the original analysis of the factorial survey (FS) experiment: the apparent failure of FS to replicate field experiment (FE) effects was likely due not to inherent weaknesses in the FS methodology, but rather to imbalances in the vignette design and assignment. By restricting the analysis to the subset of vignettes that were most evenly distributed across the participants (in other words shown more or less equally), the FS results align much more closely with those of the FE, but of course with limitations in statistical power due to fewer observations included in the regression. This finding directly challenges the original authors' conclusion that FS experiments are fundamentally unreliable for capturing real-world behavioral effects.

Moreover, once this imbalance is accounted for, the need to explore psychological moderators—such as social desirability or survey effort—becomes moot, since the discrepancy between FS and FE outcomes disappears. This reappraisal underscores the importance of careful design validation and balance diagnostics in experimental work. It also suggests that, when properly implemented, factorial survey experiments may offer valid and informative approximations of real-world behavior.

5 Code appendix

```
# this chunk contains code that sets global options for the entire .Rmd.
# we use include=FALSE to suppress it from the top of the document, but it will still appear in the app
knitr::opts_chunk$set(echo=FALSE, warning=FALSE, message=FALSE, linewidth=60)

library(tidyverse)
library(haven)
library(sandwich)
library(lmtest)
library(margins)
library(clubSandwich)
library(broom)
library(tableone)
library(emmeans)

# and any other options in R:
options(scipen=999)

# Run this once  if needed to install and load all required packages

packages_needed <- c(
  "tidyverse", "haven", "sandwich", "lmtest", "margins",
  "clubSandwich", "broom", "tableone", "emmeans"
)

# Install any missing packages
installed <- rownames(installed.packages())
to_install <- setdiff(packages_needed, installed)

if (length(to_install)) {
  install.packages(to_install)
}

# Load all required packages
invisible(lapply(packages_needed, library, character.only = TRUE))
fe = read_dta('replication_package_incl_data/00_data/validation_fe.dta')
fs = read_dta('replication_package_incl_data/00_data/validation_fs.dta')

# FE MODEL

# Load and prepare FE data
fe_data <- fe %>%
  rename(applicant_migration = fe_applicant_migration,
         applicant_female = fe_applicant_female,
         applicant_dropout = fe_applicant_dropout)

fe_data <- fe_data %>%
  mutate(
    applicant_dropout = factor(applicant_dropout),
    applicant_female = factor(applicant_female),
    applicant_migration = factor(applicant_migration),
    occupational_field = factor(occupational_field),
    wave = factor(wave)
```

```
  )

# Fit linear probability model
model_fe <- lm(callback_strict ~ applicant_dropout + applicant_female +
                            applicant_migration + occupational_field + wave,
               data = fe_data)


# Predicted probabilities
fe_probs <- emmeans(model_fe, ~ applicant_migration) %>%
  tidy(conf.int = TRUE) %>%
  transmute(source        = "Field Experiment",
            migration     = applicant_migration,
            predicted_prob = estimate,
            lower         = conf.low,
            upper         = conf.high)




# Load and prepare FS data
fs_data <- fs %>%
  mutate(ID = as.integer(ID)) %>%
  rename(applicant_migration = fs_applicant_migration,
         applicant_female    = fs_applicant_female,
         applicant_education = fs_applicant_education) %>%
  mutate(

    applicant_education   = relevel(factor(applicant_education), ref = "2"),
    fs_achievement        = relevel(factor(fs_achievement),        ref = "2"),
    fs_ses                = relevel(factor(fs_ses),               ref = "2"),
    occupational_field    = relevel(factor(occupational_field),   ref = "1"),
    applicant_migration   = factor(applicant_migration),
    applicant_female      = factor(applicant_female),
    wave                  = factor(wave)
  )

# Fit linear probability model
model_fs <- lm(invitation_dich ~ applicant_education + applicant_female +
                                 fs_achievement    + fs_ses +
                                 occupational_field + applicant_migration + wave,
               data = fs_data)

# Cluster-robust SEs
cr_se <- vcovCR(model_fs, cluster = fs_data$ID, type = "CR2")

# Predicted probabilities
fs_probs <- emmeans(model_fs,
                    ~ applicant_migration,
                    cov.reduce = mean) %>%
  tidy(conf.int = TRUE) %>%
  transmute(source         = "Factorial Survey",
            migration      = applicant_migration,
```

```
            predicted_prob = estimate,
            lower          = conf.low,
            upper          = conf.high)




# Plotting Ethnic Background
ethnic_plot_data <- bind_rows(fe_probs, fs_probs)
ethnic_plot_data$migration <- factor(ethnic_plot_data$migration,
                                     levels = c("0", "1"),
                                     labels = c("German", "Turkish"))




ggplot(ethnic_plot_data, aes(x = migration, y = predicted_prob, group = source, color = source)) +
  geom_line(size = 1) +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.1, size = 0.8) +
  theme_minimal(base_size = 14) +
  scale_color_manual(values = c("Field Experiment" = "darkorange", "Factorial Survey" = "seagreen4")) +
  scale_y_continuous(limits = c(0.4, 0.7), breaks = seq(0.4, 0.7, 0.05)) +
  ylab("Predicted Probability") +
  xlab("Applicant Ethnic Background") +
  theme(
    legend.title = element_blank(),
    legend.position = "bottom",
    panel.grid.major = element_line(color = "gray90")
  )

# Plotting Dropout
fe_probs_dropout <- emmeans(model_fe, ~ applicant_dropout) %>%
  tidy(conf.int = TRUE) %>%
  transmute(source        = "Field Experiment",
            dropout       = applicant_dropout,
            predicted_prob = estimate,
            lower          = conf.low,
            upper          = conf.high)



fs_probs_dropout <- emmeans(model_fs,
                    ~ applicant_education,
                    cov.reduce = mean) %>%
  tidy(conf.int = TRUE) %>%
  transmute(source         = "Factorial Survey",
            education      = applicant_education,
            predicted_prob = estimate,
            lower          = conf.low,
            upper          = conf.high)
```

```r
education_plot_data <- bind_rows(fe_probs_dropout, fs_probs_dropout) %>%
  mutate(dropout = if_else(dropout == 0,2,3))


education_plot_data$dropout[is.na(education_plot_data$dropout)] <- as.numeric(education_plot_data$educa


ggplot(education_plot_data, aes(x = dropout, y = predicted_prob, group = source, color = source)) +
  geom_line(size = 1) +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.1, size = 0.8) +
  theme_minimal(base_size = 13) +
  scale_color_manual(values = c("Field Experiment" = "darkorange", "Factorial Survey" = "seagreen4")) +
  scale_y_continuous(limits = c(0.4, 0.7), breaks = seq(0.4, 0.7, 0.05)) +
  scale_x_continuous(
    breaks = c(1, 2, 3),
    labels = c("Intermediate HS diploma", "Abitur", "Abitur+\nSome College")
  )+
  ylab("Predicted Probability") +
  xlab("Applicant Education") +
  theme(
    legend.title = element_blank(),
    legend.position = "bottom",
    panel.grid.major = element_line(color = "gray90")
  )

# Prepare data for plotting Figure 3


# Drop missing
fs_data_h2 <- fs_data %>%
  filter(!is.na(socdesire_std))

# Re-standardize
fs_data_h2 <- fs_data_h2 %>%
  mutate(socdesire_std = as.numeric(scale(socdesire_std)))

# Compute cutoffs
quantiles <- quantile(fs_data_h2$socdesire_std, probs = c(0.33, 0.66), na.rm = TRUE)
p33 <- quantiles[[1]]
p66 <- quantiles[[2]]

# Categorize
fs_data_h2 <- fs_data_h2 %>%
  mutate(level_socdesire = case_when(
    socdesire_std <= p33 ~ "Low",
    socdesire_std > p33 & socdesire_std <= p66 ~ "Medium",
    socdesire_std > p66 ~ "High"
  )) %>%
  mutate(level_socdesire = factor(level_socdesire, levels = c("Low", "Medium", "High")))

# Run a model for each group
fs_coef_table_h2 <- list()
```

```r
for (lvl in levels(fs_data_h2$level_socdesire)) {

  df <- fs_data_h2 %>% filter(level_socdesire == lvl)

  model <- lm(invitation_dich ~ applicant_education +
                                applicant_female +
                                fs_achievement +
                                fs_ses +
                                occupational_field +
                                applicant_migration +
                                wave,
              data = df)

  se <- vcovCR(model, cluster = df$ID, type = "CR2")

  tidy_out <- tidy(model, conf.int = TRUE, conf.level = 0.95, vcov = se) %>%
    mutate(group = lvl)

  fs_coef_table_h2[[lvl]] <- tidy_out
}

fs_coef_df_h2 <- bind_rows(fs_coef_table_h2)

# Extract ethnic background effect from FE model
fe_coef <- tidy(model_fe, conf.int = TRUE, conf.level = 0.95) %>%
  filter(term == "applicant_migration1") %>%
  transmute(
    source = "Field experiment",
    estimate = estimate,
    lower = conf.low,
    upper = conf.high,
    shape = 15,
    color = "darkorange",
    outcome_label = "Ethnic Background"
  )

# Filter ethnic background estimates
fs_plot_data_h2 <- fs_coef_df_h2 %>%
  filter(term == "applicant_migration1") %>%
  mutate(
    source = case_when(
      group == "Low" ~ "FS low SDB",
      group == "Medium" ~ "FS intermediate SDB",
      group == "High" ~ "FS high SDB"
    ),
    shape = case_when(
      group == "Low" ~ 18,
      group == "Medium" ~ 17,
      group == "High" ~ 16
    ),
    color = "darkgreen",
    outcome_label = "Ethnic Background"
  ) %>%
```

```r
  select(source, estimate, lower = conf.low, upper = conf.high, shape, color, outcome_label)

# Extract dropout effect from FE model
fe_coef_dropout <- tidy(model_fe, conf.int = TRUE, conf.level = 0.95) %>%
  filter(term == "applicant_dropout1") %>%
  transmute(
    source = "Field experiment",
    estimate = estimate,
    lower = conf.low,
    upper = conf.high,
    shape = 15,
    color = "darkorange",
    outcome_label = "HE non-completion"
  )

# Filter dropout estimates
fs_plot_data_dropout_h2 <- fs_coef_df_h2 %>%
  filter(term == "applicant_education3") %>%
  mutate(
    source = case_when(
      group == "Low" ~ "FS low SDB",
      group == "Medium" ~ "FS intermediate SDB",
      group == "High" ~ "FS high SDB"
    ),
    shape = case_when(
      group == "Low" ~ 18,
      group == "Medium" ~ 17,
      group == "High" ~ 16
    ),
    color = "darkgreen",
    outcome_label = "HE non-completion"
  ) %>%
  select(source, estimate, lower = conf.low, upper = conf.high, shape, color, outcome_label)

# Combine both effects
plot_combined <- bind_rows(fe_coef, fs_plot_data_h2, fe_coef_dropout, fs_plot_data_dropout_h2)

# Faceted plot
ggplot(plot_combined, aes(x = source, y = estimate, color = source, shape = source)) +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.1) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "purple", linewidth = 0.3) +
  scale_shape_manual(values = setNames(plot_combined$shape, plot_combined$source)) +
  scale_color_manual(values = setNames(plot_combined$color, plot_combined$source)) +
  labs(
    y = "Estimated Effect",
    x = ""
  ) +
  facet_wrap(~ outcome_label, scales = "free_y", ncol = 1) +
  theme_minimal(base_size = 13) +
  theme(
    legend.position = "bottom",
    legend.title = element_blank(),
```

```
    panel.grid.minor = element_blank(),
    panel.grid.major.y = element_blank()
  )


# Prepare data for plotting Figure 4

# Drop missing
fs_data_h3 <- fs_data %>%
  filter(!is.na(time_use))

# Compute cutoffs
quantiles <- quantile(fs_data_h3$time_use, probs = c(0.33, 0.66), na.rm = TRUE)
p33 <- quantiles[[1]]
p66 <- quantiles[[2]]

# Categorize
fs_data_h3 <- fs_data_h3 %>%
  mutate(level_timeuse = case_when(
    time_use <= p33 ~ "1",
    time_use > p33 & time_use <= p66 ~ "2",
    time_use > p66 ~ "3"
  )) %>%
  mutate(level_timeuse = factor(level_timeuse, levels = c("1", "2", "3")))

# Run a model for each group
fs_coef_table_h3 <- list()

for (lvl in levels(fs_data_h3$level_timeuse)) {

  df <- fs_data_h3 %>% filter(level_timeuse == lvl)

  model <- lm(invitation_dich ~ applicant_education +
                                applicant_female +
                                fs_achievement +
                                fs_ses +
                                occupational_field +
                                applicant_migration +
                                wave,
              data = df)

  se <- vcovCR(model, cluster = df$ID, type = "CR2")

  tidy_out <- tidy(model, conf.int = TRUE, conf.level = 0.95, vcov = se) %>%
    mutate(group = lvl)

  fs_coef_table_h3[[lvl]] <- tidy_out
}

fs_coef_df_h3 <- bind_rows(fs_coef_table_h3)

# Process ethnic background effects
fs_plot_data_h3 <- fs_coef_df_h3 %>%
```

```
  filter(term == "applicant_migration1") %>%
  mutate(
    source = case_when(
      group == "1" ~ "FS low resp. time",
      group == "2" ~ "FS intermediate resp. time",
      group == "3" ~ "FS high resp. time"
    ),
    shape = case_when(
      group == "1" ~ 18,
      group == "2" ~ 17,
      group == "3" ~ 16
    ),
    color = "darkgreen",
    outcome_label = "Ethnic Background"
  ) %>%
  select(source, estimate, lower = conf.low, upper = conf.high, shape, color, outcome_label)

# Process HE non-completion effects
fs_plot_data_dropout_h3 <- fs_coef_df_h3 %>%
  filter(term == "applicant_education3") %>%
  mutate(
    source = case_when(
      group == "1" ~ "FS low resp. time",
      group == "2" ~ "FS intermediate resp. time",
      group == "3" ~ "FS high resp. time"
    ),
    shape = case_when(
      group == "1" ~ 18,
      group == "2" ~ 17,
      group == "3" ~ 16
    ),
    color = "darkgreen",
    outcome_label = "HE non-completion"
  ) %>%
  select(source, estimate, lower = conf.low, upper = conf.high, shape, color, outcome_label)

# Field experiment effects - add outcome labels
fe_coef_labeled <- fe_coef %>%
  mutate(outcome_label = "Ethnic Background")

fe_coef_dropout_labeled <- fe_coef_dropout %>%
  mutate(outcome_label = "HE non-completion")

# Combine all into one plot dataset
plot_combined_h3 <- bind_rows(
  fe_coef_labeled,
  fs_plot_data_h3,
  fe_coef_dropout_labeled,
  fs_plot_data_dropout_h3
)

# Faceted plot
ggplot(plot_combined_h3, aes(x = source, y = estimate, color = source, shape = source)) +
```

```r
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.1) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "purple", linewidth = 0.3) +
  scale_shape_manual(values = setNames(plot_combined_h3$shape, plot_combined_h3$source)) +
  scale_color_manual(values = setNames(plot_combined_h3$color, plot_combined_h3$source)) +
  labs(
    y = "Estimated Effect",
    x = ""  ) +
  facet_wrap(~ outcome_label, scales = "free_y", ncol = 1) +
  theme_minimal(base_size = 13) +
  theme(
    legend.position = "bottom",
    legend.title = element_blank(),
    panel.grid.minor = element_blank(),
    panel.grid.major.y = element_blank()
  )




# Variables to analyze
fe_vars <- c("applicant_female", "occupational_field", "wave")

# Initialize empty dataframe
fe_results <- data.frame()

# Vector for label mapping
label_map <- c(
  "applicant_female = 1" = "Gender: Female",
  "applicant_female = 0" = "Gender: Male",
  "occupational_field = 1" = "Occupation: Electronics Technician",
  "occupational_field = 2" = "Occupation: Laboratory Technician",
  "occupational_field = 3" = "Occupation: Administration Clerk",
  "occupational_field = 4" = "Occupation: Media Clerk"
)

# Loop for creating table
for (var in fe_vars) {

  var_levels <- unique(na.omit(fe_data[[var]]))

  for (lvl in var_levels) {

    # Binary indicator for level
    temp_data <- fe_data %>%
      mutate(temp = .data[[var]] == lvl)

    # Proportions by migration
    mig_props <- temp_data %>%
      group_by(applicant_migration) %>%
      summarise(prop = mean(temp, na.rm = TRUE)) %>%
      pivot_wider(names_from = applicant_migration, values_from = prop, names_prefix = "migration_")
```

```
    # Proportions by dropout
    drop_props <- temp_data %>%
      group_by(applicant_dropout) %>%
      summarise(prop = mean(temp, na.rm = TRUE)) %>%
      pivot_wider(names_from = applicant_dropout, values_from = prop, names_prefix = "dropout_")

    # t-test
    t_mig <- t.test(temp ~ applicant_migration, data = temp_data)

    # Add to result
    row <- tibble(
      variable = paste0(var, " = ", lvl),
      migration_0 = mig_props$migration_0,
      migration_1 = mig_props$migration_1,
      dropout_0 = drop_props$dropout_0,
      dropout_1 = drop_props$dropout_1,
      t_statistic = t_mig$statistic,
      p_value = t_mig$p.value
    )

    # Replace with label if in label_map
    if (row$variable %in% names(label_map)) {
    row$variable <- label_map[[row$variable]]}

    fe_results <- bind_rows(fe_results, row)
  }
}


# Rename column names
fe_results <- fe_results %>%
  rename(
    German = migration_0,
    Turkish = migration_1,
    Abitur = dropout_0,
    `Abitur+Some College` = dropout_1,
    `t-statistic` = t_statistic,
    `p-value` = p_value
  )


# Multiply columns by 100 and round
fe_results_percent <- fe_results %>%
  mutate(
    German = round(German * 100, 1),
    Turkish = round(Turkish * 100, 1),
    Abitur   = round(Abitur   * 100, 1),
    `Abitur+Some College`   = round( `Abitur+Some College`   * 100, 1),
    `t-statistic`   = round(`t-statistic`, 2),
    `p-value`   = round(`p-value`, 2)

  )
```

```r
# Use format to center-align (by padding with spaces)
fe_results_percent <- fe_results_percent %>%
  mutate(across(where(is.numeric), ~ format(.x, justify = "centre", width = 5)))


# Show the table
knitr::kable(
  fe_results_percent,
  caption = "FE Experiment: Balance Table (Percentages)",
  align = c("l", rep("c", ncol(fe_results_percent) - 1)),
  booktabs = TRUE,
  position = 'H'
)




fs_vars <- c("applicant_female", "occupational_field", "wave","fs_ses","fs_achievement")
fs_results <- data.frame()

# Vector for label mapping
label_map <- c(
  "applicant_female = 1" = "Gender: Female",
  "applicant_female = 0" = "Gender: Male",
  "occupational_field = 1" = "Occupation: Electronics Technician",
  "occupational_field = 2" = "Occupation: Laboratory Technician",
  "occupational_field = 3" = "Occupation: Administration Clerk",
  "occupational_field = 4" = "Occupation: Media Clerk",
  "fs_achievement = 1" = "Low Grades",
  "fs_achievement = 2" = "Intermediate Grades",
  "fs_achievement = 3" = "High Grades",
  "fs_ses = 1" = "Low SES",
  "fs_ses = 2" = "Intermediate SES",
  "fs_ses = 3" = "High SES"


)

# Loop for creating table
for (var in fs_vars) {

  var_levels <- unique(na.omit(fs_data[[var]]))

  for (lvl in var_levels) {

    # Create a temporary dataset with the binary indicator
    temp_data <- fs_data %>%
      mutate(temp = .data[[var]] == lvl)

    # Proportions by migration
    mig_props <- temp_data %>%
      group_by(applicant_migration) %>%
      summarise(prop = mean(temp, na.rm = TRUE), .groups = "drop") %>%
```

```
      pivot_wider(names_from = applicant_migration, values_from = prop, names_prefix = "migration_")

    # Proportions by education
    edu_props <- temp_data %>%
      group_by(applicant_education) %>%
      summarise(prop = mean(temp, na.rm = TRUE), .groups = "drop") %>%
      pivot_wider(names_from = applicant_education, values_from = prop, names_prefix = "education_")

    # t-test
    t_mig <- t.test(temp ~ applicant_migration, data = temp_data)

    # Combine into row
    row <- tibble(
      variable = paste0(var, " = ", lvl),
      migration_0 = mig_props$migration_0,
      migration_1 = mig_props$migration_1,
      education_1 = edu_props$education_1,
      education_2 = edu_props$education_2,
      education_3 = edu_props$education_3,
      t_statistic = t_mig$statistic,
      p_value = t_mig$p.value
    )

    # Replace with label if in label_map
    if (row$variable %in% names(label_map)) {
    row$variable <- label_map[[row$variable]]
}

    fs_results <- bind_rows(fs_results, row)
  }

}


# Rename column names
fs_results <- fs_results %>%
  rename(
    German = migration_0,
    Turkish = migration_1,
    `Intermediate HS` = education_1,
    Abitur = education_2,
    `Abitur+Some College` = education_3,
    `t-statistic` = t_statistic,
    `p-value` = p_value
  )


# Multiply columns by 100 and round
fs_results_percent <- fs_results %>%
  mutate(
    German = round(German * 100, 1),
    Turkish = round(Turkish * 100, 1),
    Abitur   = round(Abitur   * 100, 1),
```

```
    `Abitur+Some College`    = round( `Abitur+Some College`    * 100, 1),
    `Intermediate HS`    = round( `Intermediate HS`    * 100, 1),
    `t-statistic`    = round(`t-statistic`, 2),
    `p-value`    = round(`p-value`, 2)
  )

# Use format to center-align (by padding with spaces)
fs_results_percent <- fs_results_percent %>%
  mutate(across(where(is.numeric), ~ format(.x, justify = "centre", width = 5)))



# Print final FS table
cat("\\begin{table}[H]
\\centering
\\caption{FS Experiment: Balance Table (Percentages)}
\\resizebox{1.08\\textwidth}{!}{%")

print(knitr::kable(
  fs_results_percent,
  booktabs = TRUE,
  align = c("l", rep("c", ncol(fs_results_percent) - 1)),
  format = "latex"
))

cat("} % end-resizebox
\\end{table}")



covars <- c("applicant_female", "applicant_education","applicant_migration", "occupational_field",
            "fs_achievement","fs_ses")


# Distinct vignettes count
cell_counts = fs_data %>%
  count(across(covars)) %>%
  arrange(desc(n))



# Total number of vignette ratings
N      <- nrow(fs_data)

# Total number of distinct vignettes
cells  <- n_distinct(cell_counts %>% select(-n))

# Expected count if perfectly balanced
exp_n  <- N / cells


# Deviation measure
imbalance_summary <- cell_counts %>%
```

```r
  mutate(pct  = round(100 * (n / exp_n) - 100, 1)) %>%
  arrange(desc(pct))



#  A lot of over or under  represented vignettes (71 are the only ones more or less equally represented
normal_vignettes = nrow(imbalance_summary %>% filter(pct<=20 & pct>=-20))




# Create deviation measure on FS data
imbalance <- fs_data %>%
  group_by(applicant_education, applicant_migration, applicant_female,
           fs_achievement, fs_ses, occupational_field) %>%
  summarise(
    n = n()
  ) %>%
  mutate(
    pct_deviation = 100 * (n / exp_n) - 100
  )

ggplot(imbalance, aes(x = pct_deviation)) +
  geom_histogram() +

  labs(x = "% Deviation from Expected Count",
       y = "Frequency")



# KEEPING ONLY THE MOST BALANCED VIGNETTES

# Flag balanced vignettes
balanced_cells <- imbalance_summary %>%
  filter(abs(pct) <= 20) %>%
  select(applicant_female:fs_ses)

# Keep only rows with balanced vignettes
fs_trim_balance <- fs_data %>%
  semi_join(balanced_cells,
            by = c("applicant_education","applicant_migration","applicant_female",
                   "fs_achievement","fs_ses","occupational_field"))

# Re-estimate main FS model
model_trim_balance <- lm(invitation_dich ~ applicant_education + applicant_female +
                                   fs_achievement   + fs_ses +
                                   occupational_field + applicant_migration + wave,
                 data = fs_trim_balance)


# Predicted probabilities for ethnic background
fs_trim_probs_migration <- emmeans(model_trim_balance,
                                   ~ applicant_migration,
```

```
                                    cov.reduce = mean) %>%
  tidy(conf.int = TRUE) %>%
  transmute(source         = "Factorial Survey (Balanced)",
            group          = applicant_migration,
            predicted_prob = estimate,
            lower          = conf.low,
            upper          = conf.high)

# Predicted probabilities for higher education non-completion
fs_trim_probs_education <- emmeans(model_trim_balance,
                                    ~ applicant_education,
                                    cov.reduce = mean) %>%
  tidy(conf.int = TRUE) %>%
  filter(applicant_education %in% c("2", "3")) %>%
  transmute(source         = "Factorial Survey (Balanced)",
            group          = applicant_education,
            predicted_prob = estimate,
            lower          = conf.low,
            upper          = conf.high)

# FE probabilities: ethnic background
fe_probs_mig <- fe_probs %>%
  transmute(source = "Field Experiment",
            group = migration,
            predicted_prob = predicted_prob,
            lower = lower,
            upper = upper)

# FE probabilities: HE non-completion
fe_probs_edu_fixed <- emmeans(model_fe, ~ applicant_dropout) %>%
  tidy(conf.int = TRUE) %>%
  filter(applicant_dropout %in% c("0", "1")) %>%
  transmute(source         = "Field Experiment",
            group          = applicant_dropout,
            predicted_prob = estimate,
            lower          = conf.low,
            upper          = conf.high)


# Combine
ethnic_plot_data <- bind_rows(fe_probs_mig, fs_trim_probs_migration)

# Plot Ethnic Background
ggplot(ethnic_plot_data, aes(x = group, y = predicted_prob, color = source, group = source)) +
  geom_point(size = 3) +
  geom_line(size = 1) +
  geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.1) +
  scale_color_manual(values = c("Field Experiment" = "darkorange", "Factorial Survey (Balanced)" = "sea
  scale_x_discrete(labels = c("0" = "German", "1" = "Turkish")) +
  labs(
    x = "Applicant Ethnic Background",
    y = "Predicted Probability",
    color = NULL
```

```
  ) +
  theme_minimal(base_size = 14) +
  theme(legend.position = "bottom")


# Combine
education_plot_data_clean <- bind_rows(
  fe_probs_edu_fixed,
  fs_trim_probs_education
) %>%
  mutate(
    group_clean = case_when(
      group %in% c("0", "2") ~ "Abitur",
      group %in% c("1", "3") ~ "Abitur+\nSome College"
    )
  )

# Plot HE non-completion
ggplot(education_plot_data_clean, aes(x = group_clean, y = predicted_prob, color = source, group = sour
  geom_point(size = 3) +
  geom_line(size = 1) +
  geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.1) +
  scale_color_manual(values = c("Field Experiment" = "darkorange",
                                "Factorial Survey (Balanced)" = "seagreen4")) +
  labs(
    x = "Applicant Education",
    y = "Predicted Probability",
    color = NULL
  ) +
  theme_minimal(base_size = 14) +
  theme(legend.position = "bottom")



# this chunk generates the complete code appendix.
# eval=FALSE tells R not to re-run (``evaluate'') the code here.
```