RECSM: Quantitative Methods in Social Research

Burak Sonmez

Updated January 29, 2026

Contents

1 Preface

Burak SonmezUniversity College London

This course is about building strong quantitative social science foundations, so new AI technologies become an asset, not a black box.

This book accompanies the short course *RECSM: Quantitative Methods in Social Research*. It extends the slide decks with worked examples, code, and practice exercises that use the **European Social Survey (ESS)** microdata for the United Kingdom (GB), Germany (DE), and France (FR).

How to use this book

- Each chapter mirrors a teaching day: descriptive statistics (Day 1), understanding associations (Day 2), nonlinear models and interactions (Day 3).
- Every exercise includes tidy, commented R code using the bundled `ess.csv` file and its HTML codebook (`ESS...subset codebook.html`).

- Required R packages are listed at the start of each chapter; install once and re-use.

If you spot anything unclear or errors, please get in touch with me.

2 Getting Ready

This short setup chapter ensures everyone can run the examples on their own laptop before class.

2.1 Install R and RStudio

- Download R from https://cran.r-project.org/ (any recent 4.x build).
- Download RStudio Desktop from https://posit.co/download/rstudio-desktop/ (free version).

2.2 Folder structure

Place the course folder anywhere convenient. The book assumes the working directory is the course root (where `ess.csv` lives). To set it inside RStudio: *Session > Set Working Directory > To Source File Location.*

```
# check current working directory
getwd()
# list course files
list.files()
```

2.3 Load the ESS data once

We use a pre-cleaned CSV with 80k+ respondents from GB, DE, and FR. The code below keeps only the variables used in the book and handles common missing codes (" ", 7x, 8x, 9x often mean non-response in ESS).

```
library(dplyr)
library(readr)

ess_raw <- read_csv("ess.csv", show_col_types = FALSE)

ess <- ess_raw |>
  filter(cntry %in% c("GB", "DE", "FR")) |>
  mutate(across(everything(), ~ na_if(.x, ""))) |>
  mutate(across(where(is.character), readr::parse_number, na = ""))

# quick glimpse
ess |> select(cntry, agea, gndr, ppltrst, netustm, nwsptot) |> slice_head(n = 5)
```

> Tip: keep `ess` in your environment while you work across chapters to avoid reloading.

3 Day 1 — Describing the ESS sample

Goal: practice measurement levels, univariate summaries, and basic visualisations using the ESS subset (GB, DE, FR).

3.1 Variables we use

- `ppltrst` (0–10): generalised social trust (higher = more trust)
- `agea`: age in years
- `gndr`: 1 = male, 2 = female (other codes = missing)
- `nwsptot`: days per week reading newspapers (0–7; 66/77/88/99 = missing)
- `netustm`: minutes per day on the internet
- `domicil`: 1 big city … 5 farm/countryside
- `cntry`: GB, DE, FR

```
library(dplyr)
library(ggplot2)

ess <- clean_ess()
```

3.2 Measurement checks

- `ppltrst` and `agea`: interval/ratio (mean and sd are fine).
- `nwsptot`, `netustm`: count-like; treat as interval for summaries but plot distributions.
- `domicil`: ordered categorical (use medians/percentiles).
- `gndr`: binary factor.

3.3 From description to inference

- **Descriptive vs causal inference:** describing "what is" (population levels, group gaps) vs "why" (causal claims require design/identification). Today we stay descriptive but prep for causal thinking.
- **Sampling model:** estimates come from a sample → always uncertainty. Sampling distributions tell us how much an estimator would vary across repeated samples.
- **CLT intuition:** for many statistics (like the mean), repeated samples stack up in an approximately normal bell curve as n grows; this justifies SEs, CIs, and t-tests.
- **Hypothesis testing basics:** state H0/HA, pick a test statistic, get a p-value, draw a conclusion while minding Type I (false positive) and Type II (false negative) risks.

3.4 Distributions and where centre matters

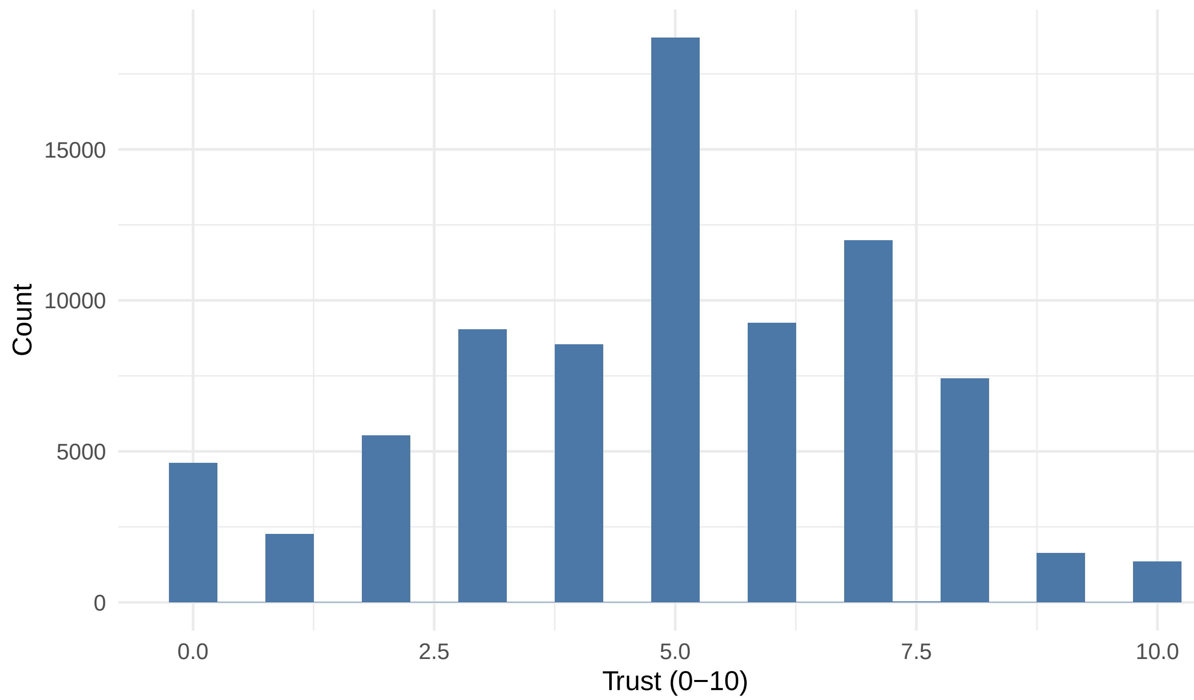
- **Central tendency:** use mean for roughly symmetric interval data (e.g., `ppltrst`), median for skewed or count-like data (e.g., `nwsptot`), and mode/proportions for categorical (`gndr`, `domicil`).
- **Shapes to look for:** symmetry vs. skew, heavy tails, spikes at 0, and multimodality. Use density/Histogram to diagnose before choosing a summary.
- **Robustness:** median and IQR resist outliers; mean and SD are more efficient when the distribution is near normal.

3.4.1 Quick distribution gallery

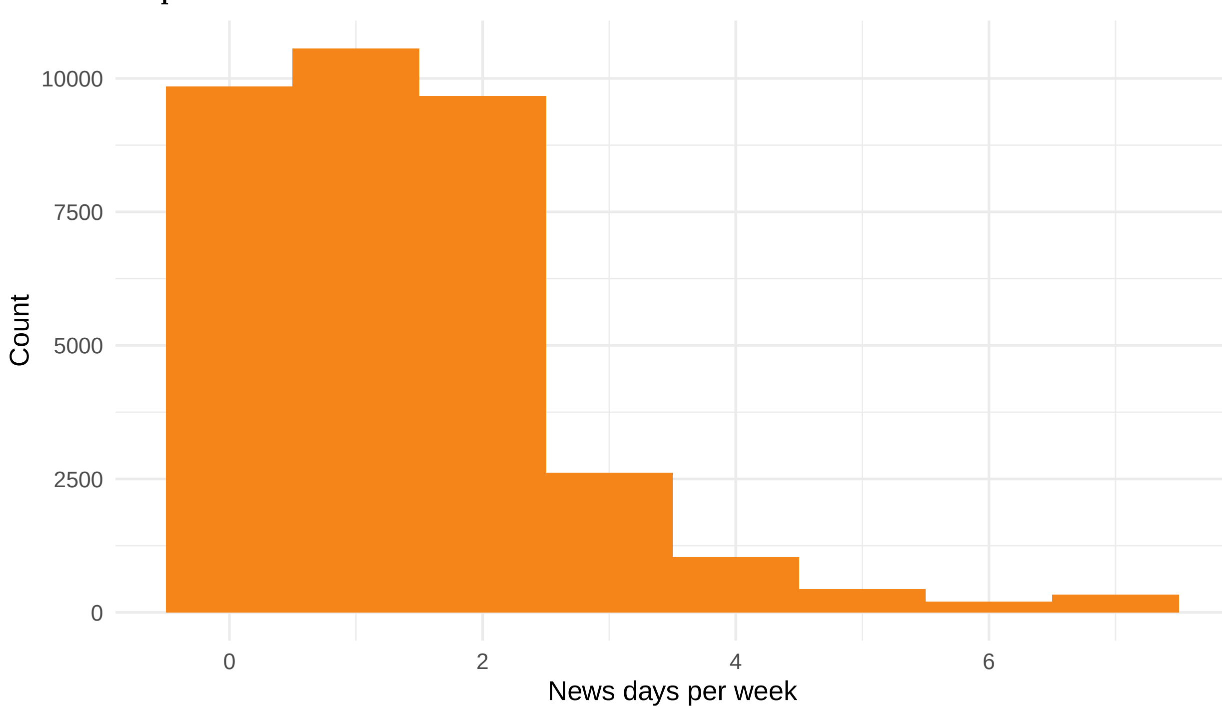
```
ess <- clean_ess()
hist_trust <- ggplot(ess, aes(x = ppltrst)) + geom_histogram(binwidth = 0.5, fill = "#4C78A8") +
  labs(x = "Trust (0-10)", y = "Count") + theme_minimal()
hist_news  <- ggplot(ess, aes(x = nwsptot)) + geom_histogram(binwidth = 1, fill = "#F58518") +
```

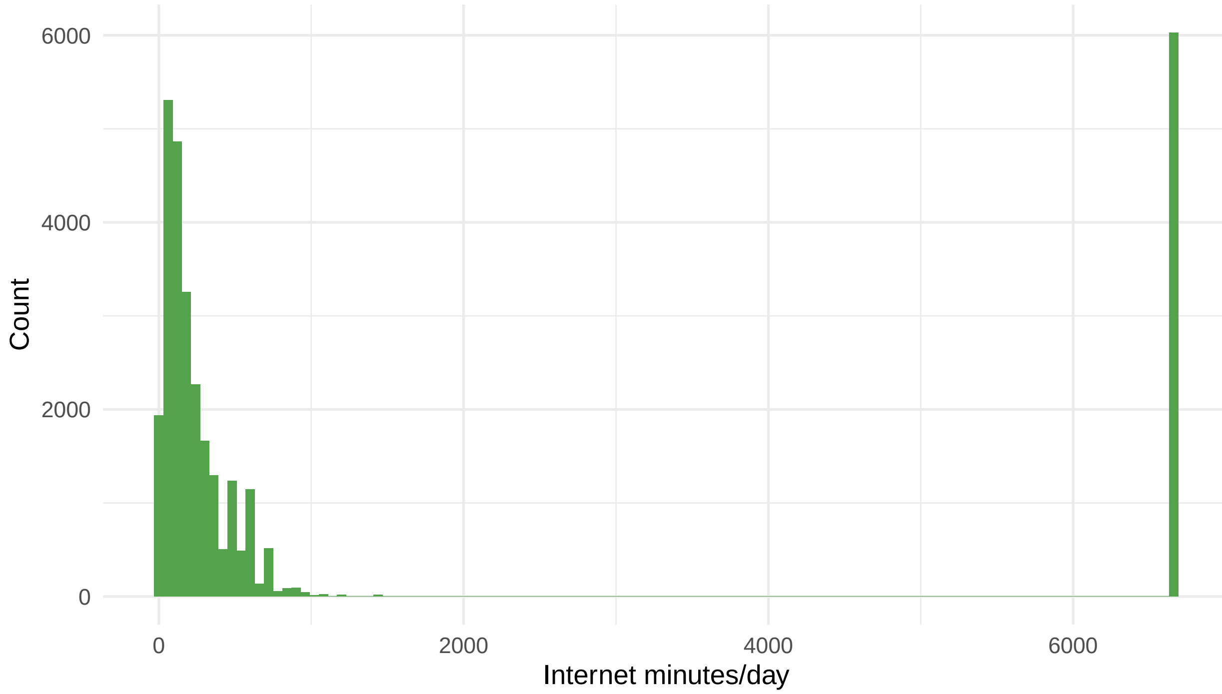
```r
  labs(x = "News days per week", y = "Count") + theme_minimal()
hist_net   <- ggplot(ess, aes(x = netustm)) + geom_histogram(binwidth = 60, fill = "#54A24B") +
  labs(x = "Internet minutes/day", y = "Count") + theme_minimal()
```

3.4.1.1 Code



3.4.1.2 Output





3.5 Problem set A — Univariate summaries

1. Compute mean, median, variance, and IQR for `ppltrst`, `nwsptot`, and `netustm` for each country.
2. Plot histograms (or density plots) of `ppltrst` by country; compare centres and spread.
3. Produce a table of counts and proportions for `gndr` and `domicil` by country.

3.5.1 Worked example A

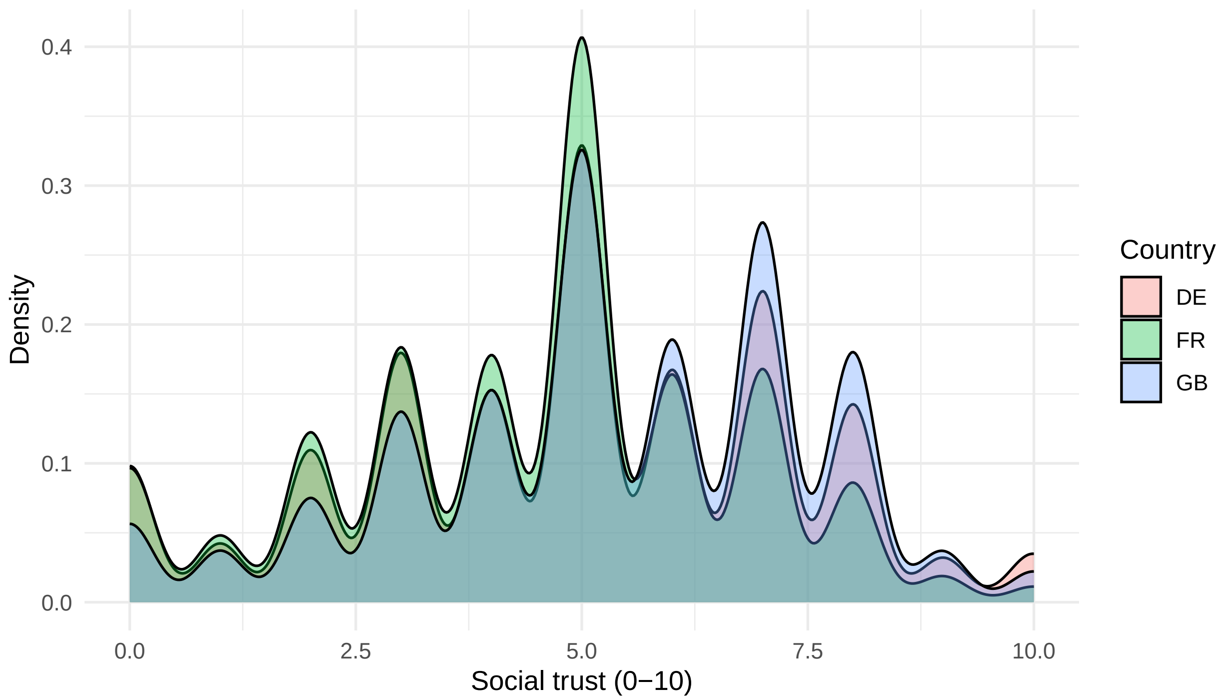
```r
# code only (not executed in this tab)
summary_tbl <- ess |>
  group_by(cntry) |>
  summarise(
    trust_mean = mean(ppltrst, na.rm = TRUE),
    trust_sd   = sd(ppltrst, na.rm = TRUE),
    news_med   = median(nwsptot, na.rm = TRUE),
    news_iqr   = IQR(nwsptot, na.rm = TRUE),
    net_mean   = mean(netustm, na.rm = TRUE),
    net_sd     = sd(netustm, na.rm = TRUE)
  )

trust_plot <- ggplot(ess, aes(x = ppltrst, fill = cntry)) +
  geom_density(alpha = 0.35) +
  labs(x = "Social trust (0-10)", y = "Density", fill = "Country") +
  theme_minimal()

counts <- ess |>
  mutate(dom_group = factor(domicil, levels = 1:5,
```

```
                                labels = c("Big city","Suburbs","Town","Village","Farm"))) |>
  count(cntry, gender, dom_group, name = "n") |>
  group_by(cntry) |>
  mutate(prop = n / sum(n))
```

3.5.1.1 Code

3.5.1.2 Output

```
## # A tibble: 3 x 7
##   cntry trust_mean trust_sd news_med news_iqr net_mean net_sd
##   <chr>      <dbl>    <dbl>    <dbl>    <dbl>    <dbl>  <dbl>
## 1 DE          4.90     2.40        1        1    1322.  2417.
## 2 FR          4.53     2.18        1        2    1754.  2771.
## 3 GB          5.31     2.22        1        2    1539.  2581.
```




```
## # A tibble: 43 x 5
## # Groups:   cntry [3]
##    cntry gender dom_group     n    prop
##    <chr> <chr>  <fct>     <int>   <dbl>
##  1 DE    Female Big city   3212 0.0872
##  2 DE    Female Suburbs    2442 0.0663
##  3 DE    Female Town       6723 0.182
##  4 DE    Female Village    5354 0.145
##  5 DE    Female Farm        391 0.0106
##  6 DE    Female <NA>         86 0.00233
##  7 DE    Male   Big city   3144 0.0853
##  8 DE    Male   Suburbs    2396 0.0650
##  9 DE    Male   Town       6475 0.176
```

```
## 10 DE    Male   Village    5739 0.156
## # i 33 more rows
```

3.6 Problem set B — Bivariate exploration

1. Correlate `ppltrst` with `agea` overall and by country. Does trust rise or fall with age?
2. Create side-by-side boxplots of `ppltrst` by `gender` within each country.
3. Compute the difference in mean `ppltrst` between genders; provide 95% confidence intervals using a t-test.
4. Produce a country-by-gender table of median `nwsptot`.

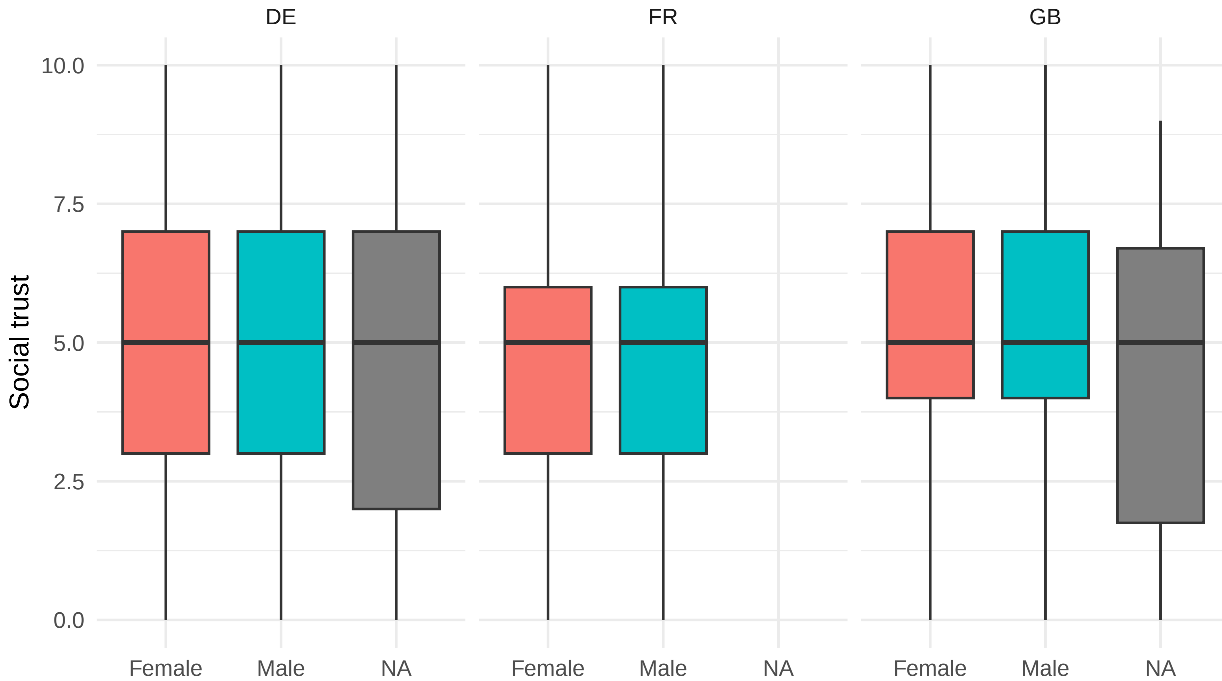
3.6.1 Worked example B

```r
# code only (not executed in this tab)
cor_age_trust <- ess |>
  group_by(cntry) |>
  summarise(corr = cor(ppltrst, agea, use = "pairwise.complete.obs"))

boxplot_trust <- ggplot(ess, aes(x = gender, y = ppltrst, fill = gender)) +
  geom_boxplot(outlier.alpha = 0.2) +
  facet_wrap(~ cntry) +
  labs(x = NULL, y = "Social trust") +
  theme_minimal() +
  theme(legend.position = "none")

# difference in means with CI
trust_ttest <- t.test(ppltrst ~ gender, data = ess)
```

3.6.1.1 Code

3.6.1.2 Output

```
## # A tibble: 3 x 2
##   cntry     corr
##   <chr>    <dbl>
## 1 DE     0.00129
## 2 FR    -0.0291
## 3 GB     0.0782
```




```
##
##  Welch Two Sample t-test
##
## data:  ppltrst by gender
## t = -6.3769, df = 79520, p-value = 1.817e-10
## alternative hypothesis: true difference in means between group Female and group Male is not equal to
## 95 percent confidence interval:
##  -0.13604975 -0.07207948
## sample estimates:
## mean in group Female   mean in group Male
##             4.870032             4.974096
```

3.7 Hypothesis testing quick-start

Two worked examples to introduce formal testing on Day 1.

3.7.1 Example 1: Two-sample t-test (mean trust by gender)

```
t_gender <- t.test(ppltrst ~ gender, data = ess)
```

3.7.1.1 Code

3.7.1.2 Output

```
##
```

```
##  Welch Two Sample t-test
##
## data:  ppltrst by gender
## t = -6.3769, df = 79520, p-value = 1.817e-10
## alternative hypothesis: true difference in means between group Female and group Male is not equal to
## 95 percent confidence interval:
##  -0.13604975 -0.07207948
## sample estimates:
## mean in group Female   mean in group Male
##             4.870032             4.974096
```

Interpretation: If the p-value < 0.05, we reject equal mean trust between men and women. The 95% CI shows the plausible range of the mean difference (Female − Male); if it excludes 0, the gap is statistically significant.

3.7.2 Example 2: Chi-squared test (gender × regular news readership)

```r
ess <- ess |>
  mutate(news_regular = ifelse(nwsptot >= 3 & !nwsptot %in% c(66,77,88,99), 1, 0))

tab_news <- table(ess$gender, ess$news_regular, useNA = "no")

chi_news <- chisq.test(tab_news)
```

3.7.2.1 Code

3.7.2.2 Output

```
##
##              0     1
##   Female 16142  2086
##   Male   13917  2532
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  tab_news
## X-squared = 116.47, df = 1, p-value < 2.2e-16
```

Interpretation: A small p-value means the share of regular news readers differs by gender (variables not independent). Report the chi-squared statistic, degrees of freedom, and p-value.

3.8 Reflection prompts

- Which variables are most skewed? How does that affect your choice of centre and spread?
- Are gender gaps in trust consistent across GB, DE, and FR?
- If `nwsptot` has many zeros, is median more informative than mean?

Use these as warm-ups before moving into regression modeling in the next chapter.

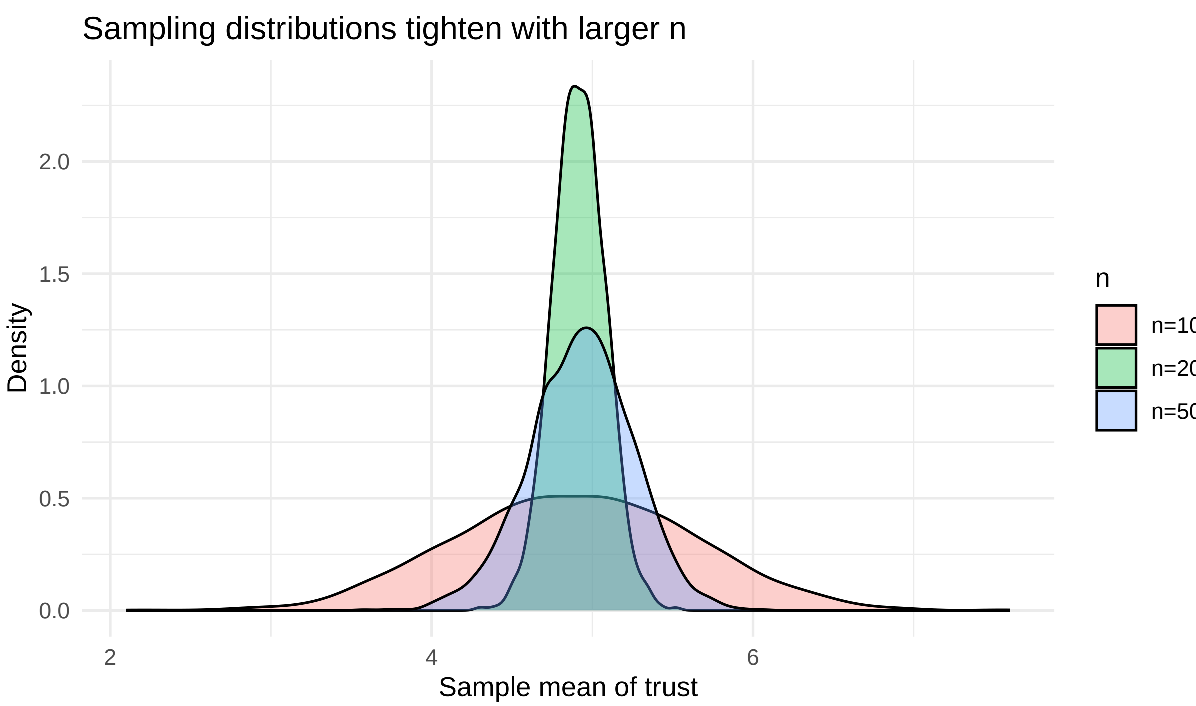
3.9 Sampling distributions by simulation

Central-limit intuition: as sample size grows, the sampling distribution of the mean tightens and approaches normality, even for skewed variables.

3.9.1 Simulating sample means of trust

```r
ess <- clean_ess()
set.seed(123)

draw_means <- function(var, n, reps = 2000) {
  vals <- na.omit(var)
  replicate(reps, mean(sample(vals, n, replace = TRUE)))
}

means_n10  <- draw_means(ess$ppltrst, n = 10)
means_n50  <- draw_means(ess$ppltrst, n = 50)
means_n200 <- draw_means(ess$ppltrst, n = 200)
```

3.9.1.1 Code



3.9.1.2 Output

Interpretation: as n grows, the sampling distribution centers near the population mean and variance shrinks; this motivates using standard errors when reporting estimates.

3.10 Hypothesis testing recap (slides → practice)

- **T-tests** compare means using the t distribution (fatter tails for small n); for two groups assume equal variances or use Welch when in doubt.
- **Chi-squared** tests independence in contingency tables.
- **Errors:**   = P(Type I),   = P(Type II); lowering   raises . Report p-values *and* confidence intervals to show effect size and uncertainty.
- **Confidence intervals:** estimate ± (critical value × SE); if 95% CI excludes 0 (for mean differences), the two-sided test at   = 0.05 would reject H0.

4 Day 2 — Linear regression with interaction effects

We replace the old dimensionality-reduction content with a deep dive on interactions. The dependent variable is **social trust (`ppltrst`)**. Predictors come from media use and demographics in the ESS subset.

Model notation recap

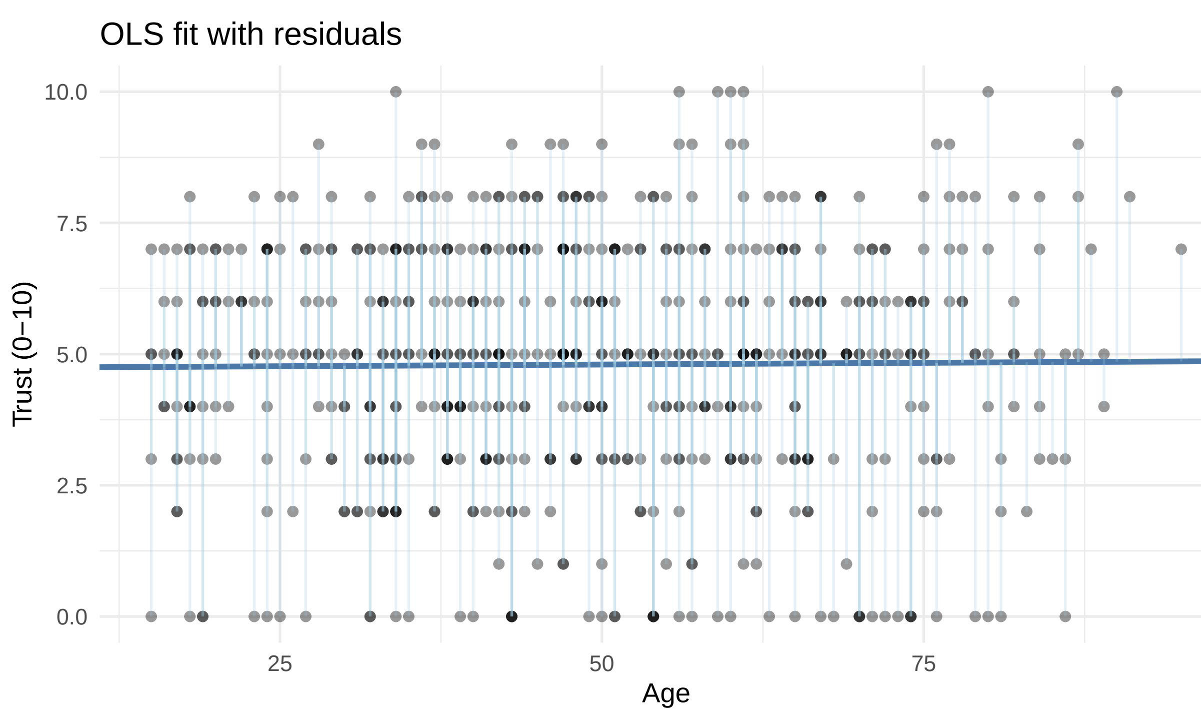
- Baseline linear model: $Y_i = \beta_0 + \mathbf{x}_i^\top \beta + \varepsilon_i$, $\varepsilon_i \sim$ i.i.d. $(0, \sigma^2)$.
- Binary–binary interaction (e.g., gender × urban): $Y_i = \beta_0 + \beta_1 \text{Female}_i + \beta_2 \text{Urban}_i + \beta_3(\text{Female}_i \times \text{Urban}_i) + \ldots$.
  - β_3 is the *difference-in-differences*: the extra gap between women and men *when Urban = 1* minus the gap when Urban = 0.
- Binary–continuous interaction (gender × age): slope for age becomes $\beta_{\text{age}} + \beta_{\text{age}\times\text{female}} \cdot \text{Female}_i$; draw ribbons to see how slopes differ across groups.
- Three-way interaction (gender × age × country): the age slope is country- and gender-specific: $\partial Y / \partial \text{age} = \beta_{\text{age}} + \beta_{\text{age}\times g} g + \beta_{\text{age}\times c} c + \beta_{\text{age}\times g\times c} gc$.

```r
library(dplyr)
library(ggplot2)
library(broom)
library(purrr)
library(tidyr)

source("R/clean_ess.R")

ess <- clean_ess()

# Fit once and reuse
m0 <- lm(ppltrst ~ agea + gender + news_days + country, data = ess)
m1 <- lm(ppltrst ~ gender * urban + agea + news_days + country, data = ess)
m2 <- lm(ppltrst ~ gender * agea + news_days + country, data = ess)
m3 <- lm(ppltrst ~ agea * news_days + gender + country, data = ess)
m4 <- lm(ppltrst ~ gender * agea * country + news_days, data = ess)

# Predicted values for plots (simple grids)
nd1 <- expand.grid(
  urban = c("Urban", "Non-urban"),
  gender = c("Male", "Female"),
  agea = mean(ess$agea, na.rm = TRUE),
  news_days = mean(ess$news_days, na.rm = TRUE),
  country = "GB"
```

```
)
pred1 <- predict(m1, newdata = nd1, se.fit = TRUE)
nd1$fit <- pred1$fit
nd1$lo <- pred1$fit - 1.96 * pred1$se.fit
nd1$hi <- pred1$fit + 1.96 * pred1$se.fit
int_plot1 <- ggplot(nd1, aes(x = urban, y = fit, fill = gender)) +
  geom_col(position = position_dodge(width = 0.6), width = 0.5) +
  geom_errorbar(aes(ymin = lo, ymax = hi), position = position_dodge(width = 0.6), width = 0.2) +
  labs(y = "Predicted social trust", x = "Residential area") +
  theme_minimal()

age_seq <- seq(min(ess$agea, na.rm = TRUE), max(ess$agea, na.rm = TRUE), length.out = 60)
nd2 <- expand.grid(agea = age_seq, gender = c("Male", "Female"),
                   news_days = mean(ess$news_days, na.rm = TRUE),
                   country = "GB")
pred2 <- predict(m2, newdata = nd2, se.fit = TRUE)
nd2$fit <- pred2$fit
nd2$lo <- pred2$fit - 1.96 * pred2$se.fit
nd2$hi <- pred2$fit + 1.96 * pred2$se.fit
int_plot2 <- ggplot(nd2, aes(x = agea, y = fit, color = gender)) +
  geom_line(size = 1) +
  geom_ribbon(aes(ymin = lo, ymax = hi, fill = gender), alpha = 0.15, color = NA) +
  labs(y = "Predicted social trust", x = "Age") +
  theme_minimal()

news_seq <- seq(min(ess$news_days, na.rm = TRUE), max(ess$news_days, na.rm = TRUE), length.out = 40)
nd3 <- expand.grid(news_days = news_seq,
                   agea = quantile(ess$agea, c(.2, .5, .8), na.rm = TRUE),
                   gender = "Male",
                   country = "GB")
pred3 <- predict(m3, newdata = nd3, se.fit = TRUE)
nd3$fit <- pred3$fit
nd3$lo <- pred3$fit - 1.96 * pred3$se.fit
nd3$hi <- pred3$fit + 1.96 * pred3$se.fit
int_plot3 <- ggplot(nd3, aes(x = news_days, y = fit, color = factor(agea))) +
  geom_line(size = 1) +
  geom_ribbon(aes(ymin = lo, ymax = hi, fill = factor(agea)), alpha = 0.12, color = NA) +
  labs(y = "Predicted social trust", x = "News days per week", color = "Age quantile") +
  theme_minimal()

nd4 <- expand.grid(agea = age_seq,
                   gender = c("Male","Female"),
                   country = c("GB","DE","FR"),
                   news_days = mean(ess$news_days, na.rm = TRUE))
pred4 <- predict(m4, newdata = nd4, se.fit = TRUE)
nd4$fit <- pred4$fit
nd4$lo <- pred4$fit - 1.96 * pred4$se.fit
nd4$hi <- pred4$fit + 1.96 * pred4$se.fit
int_plot4 <- ggplot(nd4, aes(x = agea, y = fit, color = gender)) +
  geom_line() +
  geom_ribbon(aes(ymin = lo, ymax = hi, fill = gender), alpha = 0.12, color = NA) +
  facet_wrap(~ country) +
  labs(y = "Predicted social trust", x = "Age") +
```

```r
  theme_minimal()

# OLS intuition demo (interactive)
ess_small <- ess |> select(ppltrst, agea) |> drop_na() |> slice_sample(n = 600)
fit_simple <- lm(ppltrst ~ agea, data = ess_small)

base_resid <- ggplot(ess_small, aes(x = agea, y = ppltrst)) +
  geom_point(alpha = 0.4) +
  geom_abline(slope = coef(fit_simple)[2], intercept = coef(fit_simple)[1], color = "#4C78A8", size = 1
  geom_segment(aes(xend = agea, yend = fitted(fit_simple)), alpha = 0.25, color = "#9ecae1") +
  labs(x = "Age", y = "Trust (0-10)", title = "OLS fit with residuals") +
  theme_minimal()

slope_grid <- seq(coef(fit_simple)[2] - 0.08, coef(fit_simple)[2] + 0.08, length.out = 30)
anim_df <- map_dfr(slope_grid, ~{
  pred <- coef(fit_simple)[1] + .x * ess_small$agea
  tibble(agea = ess_small$agea,
         ppltrst = ess_small$ppltrst,
         slope = sprintf("%.3f", .x),
         pred = pred,
         resid = ppltrst - pred)
})
anim_plot <- ggplot(anim_df, aes(x = agea, y = ppltrst, frame = slope)) +
  geom_point(alpha = 0.35) +
  geom_abline(aes(slope = as.numeric(slope), intercept = coef(fit_simple)[1]), color = "#4C78A8", size =
  labs(x = "Age", y = "Trust (0-10)", title = "Searching for the best-fit slope") +
  theme_minimal()
```

4.1 OLS intuition: best-fit line and residuals



4.1.0.1 Output

Interpretation: the static panel shows residuals; if you run the optional plotly code locally, the moving line illustrates how residuals shrink as the slope approaches the OLS solution.

4.2 1. Baseline linear model

```r
m0 <- lm(ppltrst ~ agea + gender + news_days + country, data = ess)
broom::tidy(m0)
```

4.2.0.1 Code

4.2.0.2 Output

```
## # A tibble: 6 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)  4.68     0.0394      119.   0
## 2 agea        -0.00245  0.000691     -3.54 4.01e- 4
## 3 genderMale   0.101    0.0246        4.12 3.84e- 5
## 4 news_days    0.0727   0.00988       7.36 1.93e-13
## 5 countryFR   -0.244    0.0307       -7.94 2.00e-15
## 6 countryGB    0.545    0.0287       19.0  9.60e-80
```

Interpretation: Trust increases slightly with age and differs by country and gender; focus on sign and magnitude of the coefficients rather than raw p-values when discussing effect sizes.

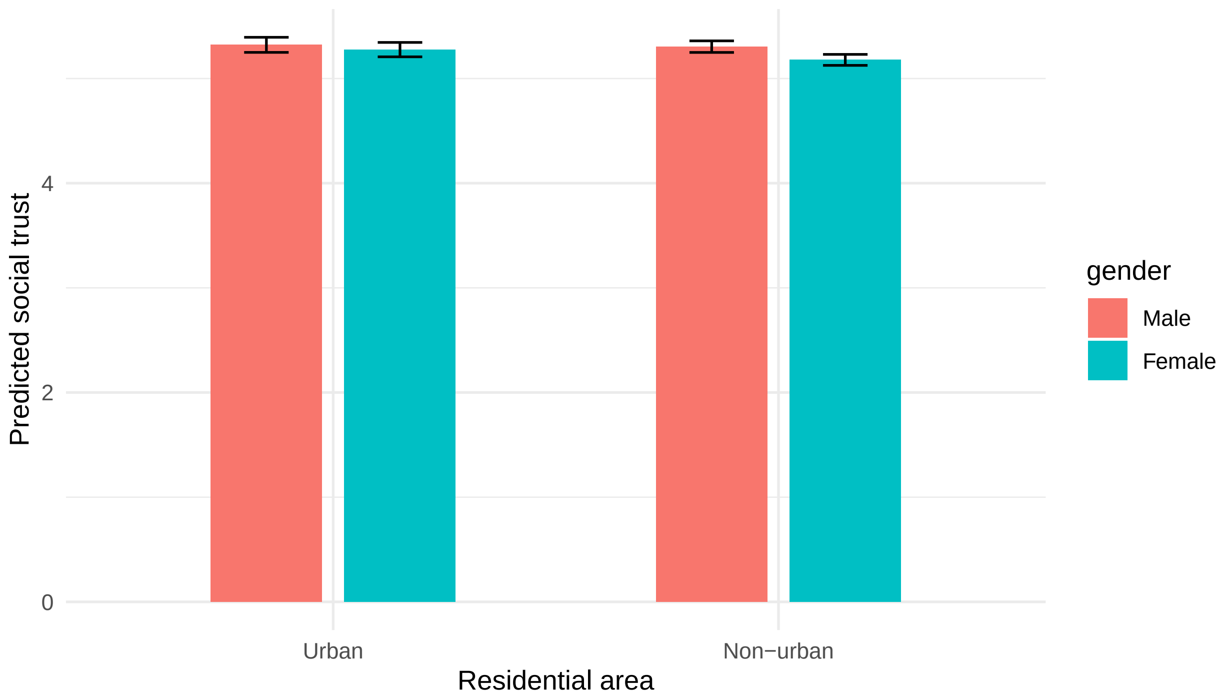
4.3 2. Binary × Binary interaction (gender × urban)

```r
m1 <- lm(ppltrst ~ gender * urban + agea + news_days + country, data = ess)
int_plot1 <- ggplot(nd1, aes(x = urban, y = fit, fill = gender)) +
  geom_col(position = position_dodge(width = 0.6), width = 0.5) +
  labs(y = "Predicted social trust", x = "Residential area") +
  theme_minimal()
```

4.3.0.1 Code

4.3.0.2 Output

```
## # A tibble: 8 x 5
##   term                  estimate std.error statistic  p.value
##   <chr>                    <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)            4.65     0.0412      113.   0
## 2 genderMale             0.126    0.0295        4.28 1.85e- 5
## 3 urbanUrban             0.0979   0.0365        2.68 7.34e- 3
## 4 agea                  -0.00238  0.000692     -3.44 5.92e- 4
## 5 news_days              0.0716   0.00990       7.23 4.79e-13
## 6 countryFR             -0.244    0.0307       -7.96 1.71e-15
## 7 countryGB              0.546    0.0288       19.0  5.96e-80
## 8 genderMale:urbanUrban -0.0806   0.0529       -1.52 1.28e- 1
```



Interpretation: The urban–rural trust gap is small; note whether the CI bars for Male vs Female overlap. If they do, the moderation by gender is likely negligible.

Interpretation focus: Does the urban–rural gap differ by gender?

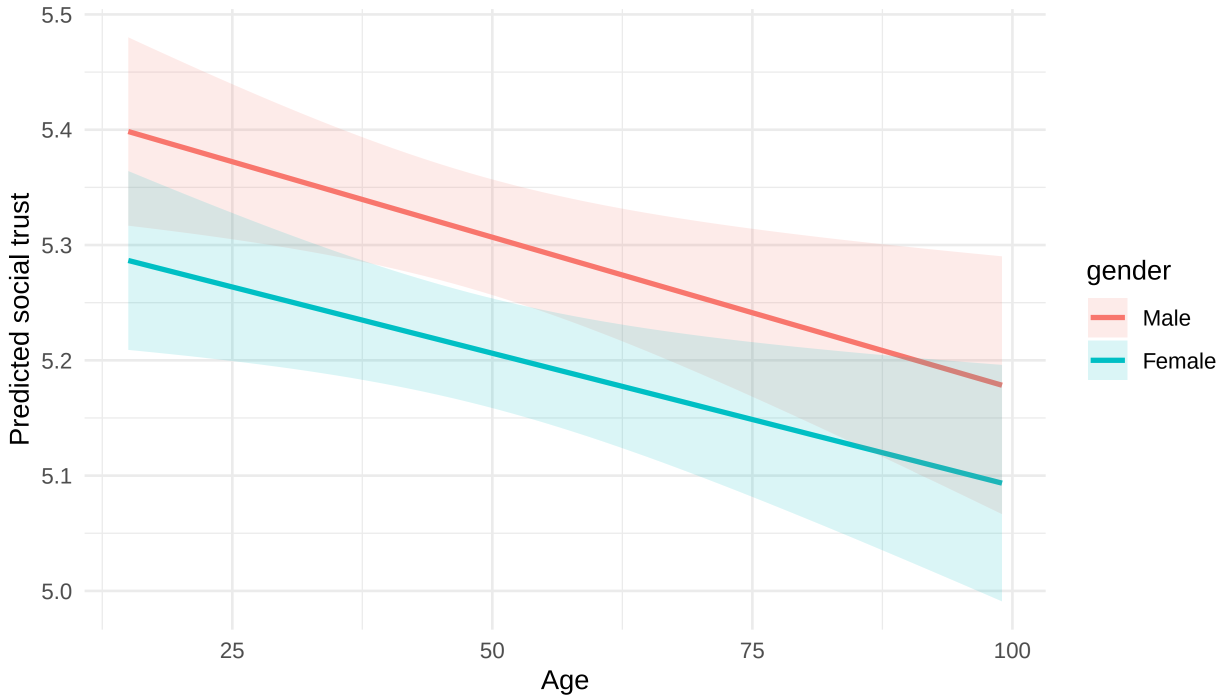
4.4 3. Binary × Continuous interaction (gender × age)

```
m2 <- lm(ppltrst ~ gender * agea + news_days + country, data = ess)
int_plot2 <- ggplot(nd2, aes(x = agea, y = fit, color = gender)) +
  geom_line(size = 1) +
  labs(y = "Predicted social trust", x = "Age") +
  theme_minimal()
```

4.4.0.1 Code

4.4.0.2 Output

```
## # A tibble: 7 x 5
##   term              estimate std.error statistic  p.value
##   <chr>                <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)       4.68      0.0498      93.9    0
## 2 genderMale        0.117     0.0691       1.69   9.14e- 2
## 3 agea             -0.00230   0.000925    -2.49   1.29e- 2
## 4 news_days         0.0727    0.00988      7.36   1.88e-13
## 5 countryFR        -0.244     0.0307      -7.94   2.05e-15
## 6 countryGB         0.545     0.0287      19.0    9.42e-80
## 7 genderMale:agea -0.000321   0.00134     -0.240  8.10e- 1
```



Interpretation: Slopes by age differ by gender; parallel ribbons would imply no interaction. Diverging ribbons indicate the age effect depends on gender.

Key idea: slopes for age are estimated separately for men and women.

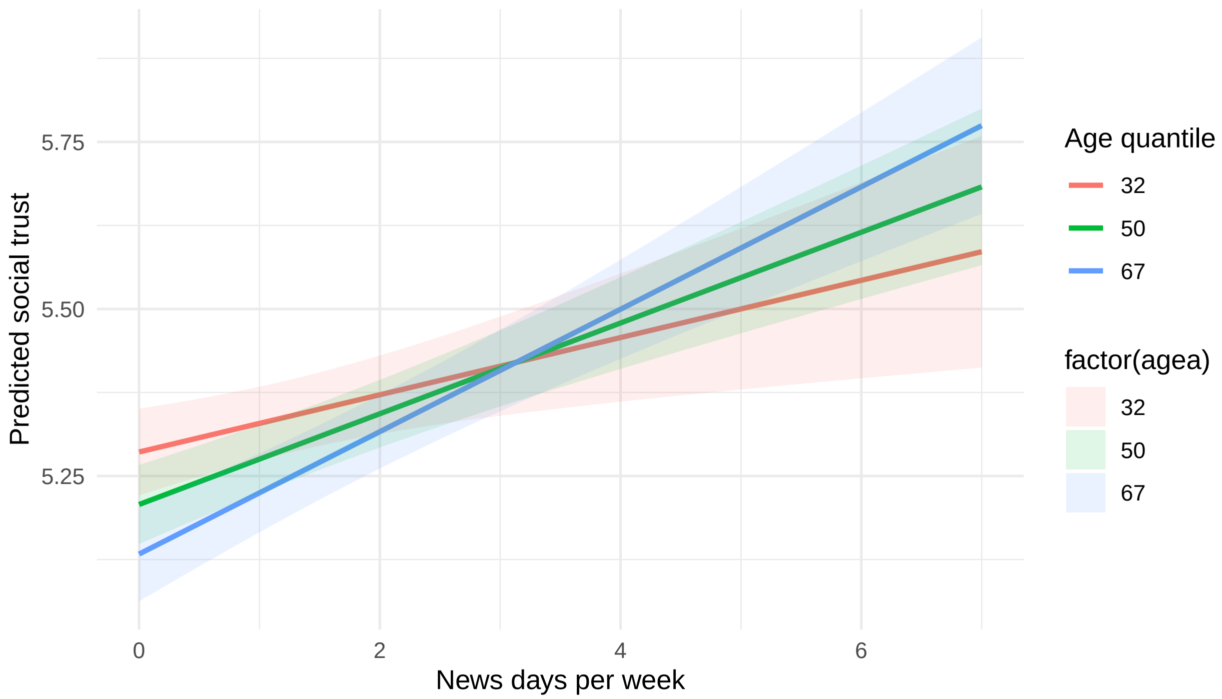
4.5 4. Continuous × Continuous interaction (age × news consumption)

```
m3 <- lm(ppltrst ~ agea * news_days + gender + country, data = ess)
int_plot3 <- ggplot(nd3, aes(x = news_days, y = fit, color = factor(agea))) +
  geom_line(size = 1) +
  labs(y = "Predicted social trust", x = "News days per week", color = "Age quantile") +
  theme_minimal()
```

4.5.0.1 Code

4.5.0.2 Output

```
## # A tibble: 7 x 5
##   term            estimate std.error statistic  p.value
##   <chr>              <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)      4.78     0.0516     92.6    0
## 2 agea            -0.00436  0.000975   -4.48   7.65e- 6
## 3 news_days       -0.00178  0.0285     -0.0627 9.50e- 1
## 4 genderMale       0.101    0.0246      4.12   3.75e- 5
## 5 countryFR       -0.243    0.0307     -7.92   2.46e-15
## 6 countryGB        0.548    0.0288     19.1    1.57e-80
## 7 agea:news_days   0.00139  0.000500    2.79   5.31e- 3
```



Interpretation: Check whether the news-consumption slope changes across age quantiles; overlapping ribbons mean little moderation, separated ribbons suggest stronger news effects at certain ages.

Discuss whether news exposure moderates the age–trust relationship.

4.6 5. Three-way interaction (gender × age × country)

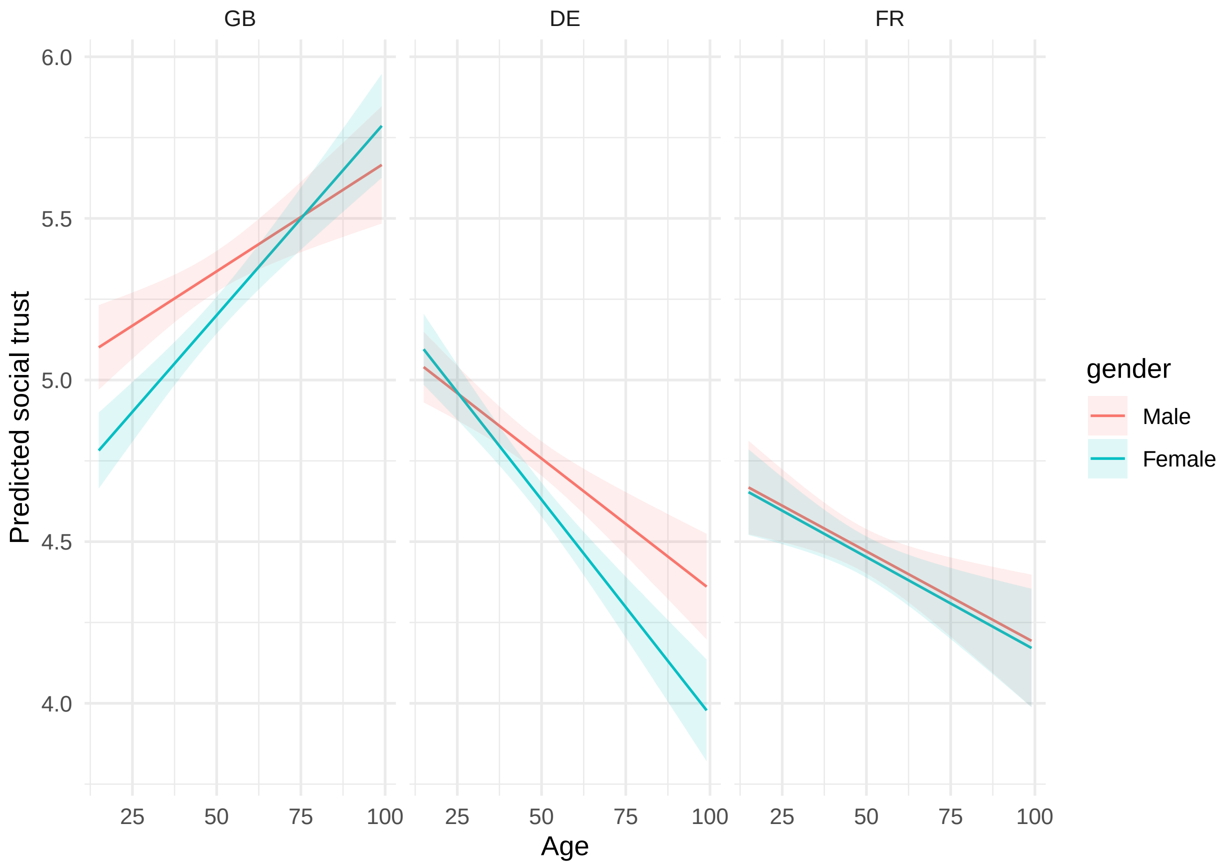
```
m4 <- lm(ppltrst ~ gender * agea * country + news_days, data = ess)
int_plot4 <- ggplot(nd4, aes(x = agea, y = fit, color = gender)) +
  geom_line() +
  facet_wrap(~ country) +
  labs(y = "Predicted social trust", x = "Age") +
  theme_minimal()
```

4.6.0.1 Code

4.6.0.2 Output

```
## # A tibble: 13 x 5
##    term                    estimate std.error statistic  p.value
##    <chr>                      <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)               5.19     0.0760      68.2   0
##  2 genderMale               -0.133    0.107       -1.24  2.14e- 1
##  3 agea                     -0.0133   0.00149     -8.92  4.69e-19
##  4 countryFR                -0.555    0.119       -4.69  2.79e- 6
##  5 countryGB                -0.692    0.111       -6.25  4.24e-10
##  6 news_days                 0.0775   0.00988      7.84  4.66e-15
##  7 genderMale:agea           0.00521  0.00211      2.48  1.33e- 2
```

```
## 8 genderMale:countryFR       0.147     0.173       0.849 3.96e- 1
## 9 genderMale:countryGB       0.531     0.162       3.28  1.04e- 3
## 10 agea:countryFR            0.00756   0.00229     3.30  9.58e- 4
## 11 agea:countryGB            0.0253    0.00213    11.9   1.92e-32
## 12 genderMale:agea:countryFR -0.00512  0.00335    -1.53  1.26e- 1
## 13 genderMale:agea:countryGB -0.0104   0.00312    -3.35  8.23e- 4
```



Interpretation: Three-way plots show country-specific age slopes by gender; look for countries where ribbons separate widely—that's where the interaction is substantive.

Strategy: interpret pairwise contrasts within each country before comparing across countries.

4.7 6. Model comparison and diagnostics

```r
broom::glance(m0, m1, m2, m3, m4)
# quick residual check
par(mfrow = c(2,2)); plot(m3)
```

4.8 Problem set — Interaction lab

1. Refit `m1` but swap `urban` with a binary indicator for *high education* (e.g., `eduyrs >= 15`). Interpret the gender gap at low vs high education.
2. Build a model with `ppltrst ~ news_days * country + agea + gender`. Compute marginal effects of `news_days` within each country.

3. Add a three-way term `gender * urban * country`. Plot predicted trust for all six gender-by-urban-by-country profiles.
4. Briefly report which interaction improves fit (compare adjusted R^2 and AIC) and whether the effect is substantively meaningful.

Use `marginaleffects::plot_slopes()` and `plot_predictions()` to visualise interactions instead of only staring at coefficients.

5 Day 3 — Non-linear models: logistic regression

We now model binary outcomes. This replaces the old factor-analysis/PCA content and links three specifications:

- **Linear probability model (LPM):** $Y_i \in \{0,1\}$, $\mathbb{E}[Y_i \mid X_i] = X_i\beta$ (identity link).
- **Logit:** $p_i = \Pr(Y_i = 1 \mid X_i) = \text{logit}^{-1}(X_i\beta) = \frac{e^{X_i\beta}}{1+e^{X_i\beta}}$.
- **Marginal effects:** $\frac{\partial p_i}{\partial x_{ik}} = p_i(1-p_i)\beta_k$, highlighting how effects vary with the baseline probability.
- **Interaction in logit:** For $p_i = \text{logit}^{-1}(\eta_i)$ with $\eta_i = \beta_0 + \beta_1 x + \beta_2 z + \beta_3 xz$, the cross-partial effect is $\frac{\partial^2 p_i}{\partial x\, \partial z} = p_i(1-p_i)(1-2p_i)\beta_1\beta_2 + p_i(1-p_i)\beta_3$; sign can vary with p_i.

Outcome: `news_regular` = 1 if a respondent reads newspapers on at least 3 days per week (`nwsptot >= 3`), 0 otherwise.

Predictors: age (`agea`), gender (`gndr`), country (`cntry`), education (`eduyrs`), and their interactions.

```r
library(dplyr)
library(broom)
library(ggplot2)

source("R/clean_ess.R")

ess <- clean_ess()

# Fit once and reuse
lpm1   <- lm(news_regular ~ agea + gender + country, data = ess)
logit1 <- glm(news_regular ~ agea + gender + country, data = ess, family = binomial())
logit2 <- glm(news_regular ~ gender * country + agea, data = ess, family = binomial())
logit3 <- glm(news_regular ~ gender * agea + country + eduyrs, data = ess, family = binomial())
logit4 <- glm(news_regular ~ agea * eduyrs + gender + country, data = ess, family = binomial())
logit5 <- glm(news_regular ~ gender * agea * country + eduyrs, data = ess, family = binomial())

# Prediction data for plots
age_seq <- seq(min(ess$agea, na.rm = TRUE), max(ess$agea, na.rm = TRUE), length.out = 60)
edu_seq <- seq(min(ess$eduyrs, na.rm = TRUE), max(ess$eduyrs, na.rm = TRUE), length.out = 40)

nd2 <- expand.grid(country = c("GB","DE","FR"),
                   gender = c("Male","Female"),
                   agea = mean(ess$agea, na.rm = TRUE))
pred2 <- predict(logit2, newdata = nd2, type = "link", se.fit = TRUE)
nd2$pr <- plogis(pred2$fit)
nd2$lo <- plogis(pred2$fit - 1.96 * pred2$se.fit)
nd2$hi <- plogis(pred2$fit + 1.96 * pred2$se.fit)
```

```r
plot_gender_country <- ggplot(nd2, aes(x = country, y = pr, fill = gender)) +
  geom_col(position = position_dodge(width = 0.6), width = 0.5) +
  geom_errorbar(aes(ymin = lo, ymax = hi), position = position_dodge(width = 0.6), width = 0.2) +
  labs(y = "Pr(regular news)") +
  theme_minimal()

nd3 <- expand.grid(agea = age_seq, gender = c("Male","Female"),
                   country = "GB", eduyrs = mean(ess$eduyrs, na.rm = TRUE))
pred3 <- predict(logit3, newdata = nd3, type = "link", se.fit = TRUE)
nd3$pr <- plogis(pred3$fit)
nd3$lo <- plogis(pred3$fit - 1.96 * pred3$se.fit)
nd3$hi <- plogis(pred3$fit + 1.96 * pred3$se.fit)
plot_gender_age <- ggplot(nd3, aes(x = agea, y = pr, color = gender)) +
  geom_line(size = 1) +
  geom_ribbon(aes(ymin = lo, ymax = hi, fill = gender), alpha = 0.15, color = NA) +
  labs(y = "Pr(regular news)", x = "Age") +
  theme_minimal()

nd4 <- expand.grid(eduyrs = edu_seq,
                   agea = quantile(ess$agea, c(.2,.5,.8), na.rm = TRUE),
                   gender = "Male", country = "GB")
pred4 <- predict(logit4, newdata = nd4, type = "link", se.fit = TRUE)
nd4$pr <- plogis(pred4$fit)
nd4$lo <- plogis(pred4$fit - 1.96 * pred4$se.fit)
nd4$hi <- plogis(pred4$fit + 1.96 * pred4$se.fit)
plot_age_edu <- ggplot(nd4, aes(x = eduyrs, y = pr, color = factor(agea))) +
  geom_line(size = 1) +
  geom_ribbon(aes(ymin = lo, ymax = hi, fill = factor(agea)), alpha = 0.12, color = NA) +
  labs(y = "Pr(regular news)", x = "Years of education", color = "Age quantile") +
  theme_minimal()

nd5 <- expand.grid(agea = age_seq,
                   gender = c("Male","Female"),
                   country = c("GB","DE","FR"),
                   eduyrs = mean(ess$eduyrs, na.rm = TRUE))
pred5 <- predict(logit5, newdata = nd5, type = "link", se.fit = TRUE)
nd5$pr <- plogis(pred5$fit)
nd5$lo <- plogis(pred5$fit - 1.96 * pred5$se.fit)
nd5$hi <- plogis(pred5$fit + 1.96 * pred5$se.fit)
plot_threeway <- ggplot(nd5, aes(x = agea, y = pr, color = gender)) +
  geom_line() +
  geom_ribbon(aes(ymin = lo, ymax = hi, fill = gender), alpha = 0.12, color = NA) +
  facet_wrap(~ country) +
  labs(y = "Pr(regular news)", x = "Age") +
  theme_minimal()

# Average marginal effects (finite-difference, no parallelism)
calc_ame <- function(model, data, var, step = 1) {
  data_hi <- data
  if (is.numeric(data_hi[[var]])) {
    data_hi[[var]] <- data_hi[[var]] + step
  } else if (is.factor(data_hi[[var]]) || is.character(data_hi[[var]])) {
    # flip binary factor
```

```
    if (all(na.omit(unique(data_hi[[var]])) %in% c("Male","Female"))) {
      data_hi[[var]] <- ifelse(data_hi[[var]] == "Female", "Male", "Female")
    }
  }
  p_hi <- predict(model, newdata = data_hi, type = "response")
  p_lo <- predict(model, newdata = data, type = "response")
  mean(p_hi - p_lo, na.rm = TRUE)
}
ame_age  <- calc_ame(logit5, ess, "agea", step = 1)
ame_fem  <- calc_ame(logit5, ess, "gender", step = 0)  # Female vs Male switch
ame_table <- tibble(
  variable = c("agea (+1 year)", "gender (Female vs Male)"),
  AME = c(ame_age, ame_fem)
)

# Marginal effect of being Female at age 25 vs 65
nd_female <- data.frame(agea = c(25,65),
                        gender = "Female",
                        country = "GB",
                        eduyrs = mean(ess$eduyrs, na.rm = TRUE))
nd_male   <- nd_female; nd_male$gender <- "Male"
pred_f <- predict(logit5, newdata = nd_female, type = "response")
pred_m <- predict(logit5, newdata = nd_male,   type = "response")
me_female_diff <- data.frame(agea = c(25,65),
                             diff = pred_f - pred_m)
me_female_plot <- ggplot(me_female_diff, aes(x = factor(agea), y = diff)) +
  geom_col(fill = "#4C78A8", width = 0.5) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(x = "Age", y = "Pr(Female) - Pr(Male)", title = "Marginal effect of being Female") +
  theme_minimal()
```

5.1 0. Linear probability model (LPM) first

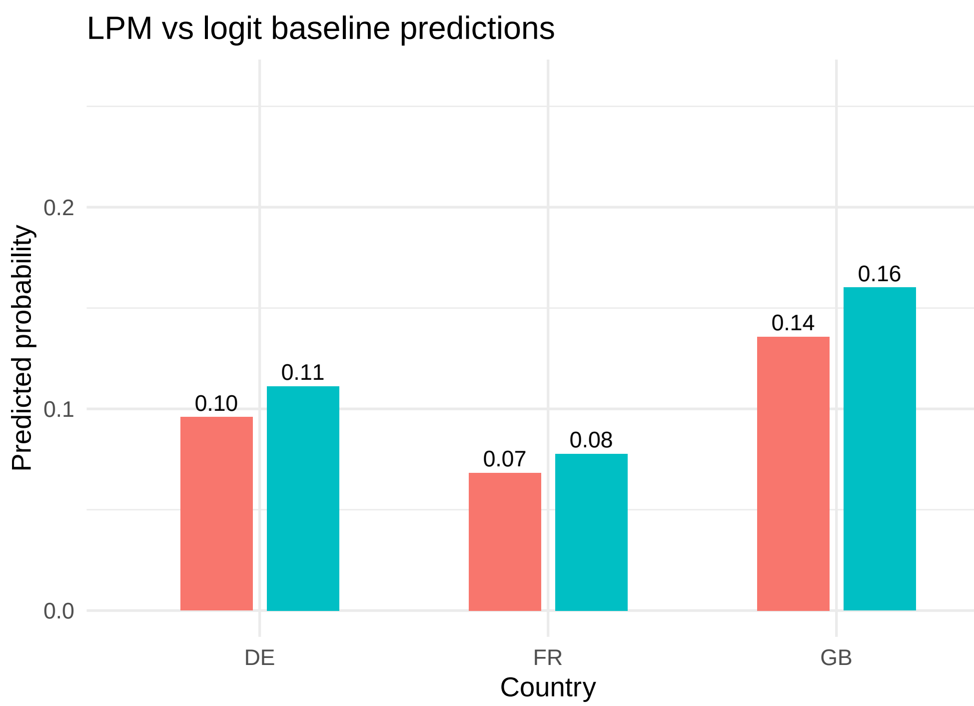
- **Specification:** $Y_i = X_i\beta + \varepsilon_i,\ Y_i \in \{0,1\}$. Ordinary least squares with a binary outcome.
- **Pros:** Coefficients are immediate probability changes (Δp) per unit of X; easy to interpret and to add fixed effects.
- **Cons:** Predicted values can leave $[0,1]$; errors are heteroskedastic; marginal effects are assumed constant even when baseline risk is near 0 or 1.
- **When is LPM "safe enough"?** Middle-range probabilities (e.g., 0.2–0.8), modest leverage points, and when the goal is fast descriptive decomposition or fixed-effects absorption. Use robust standard errors.

```
library(sandwich)
library(lmtest)

lpm_vcov <- sandwich::vcovHC(lpm1, type = "HC1")
lpm_tidy <- broom::tidy(lpm1, conf.int = TRUE, vcov = lpm_vcov)
```

5.1.0.1 Output

```
## # A tibble: 5 x 7
```

```
##   term        estimate std.error statistic  p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 (Intercept) -0.0861  0.00569      -15.1  1.43e-51 -0.0973   -0.0750
## 2 agea         0.00397 0.0000969     40.9  0          0.00378   0.00416
## 3 genderMale   0.0429  0.00356       12.0  2.31e-33  0.0359    0.0498
## 4 countryFR   -0.0334  0.00442       -7.56 4.08e-14 -0.0421   -0.0248
## 5 countryGB    0.0490  0.00418       11.7  9.41e-32  0.0408    0.0572
```



5.1.0.2 LPM vs logit predictions

Observation: If LPM bars stray above 1 or below 0, that signals the need for a logit/probit link. Here the mid-range outcome keeps LPM close, but we switch to logit next for coherent probabilities and curved marginal effects.

5.2 1. Simple logistic model

5.2.1 Simple logistic model

```r
logit1 <- glm(news_regular ~ agea + gender + country, data = ess, family = binomial())
broom::tidy(logit1, exponentiate = TRUE, conf.int = TRUE)
```

5.2.1.1 Code

Odds ratios > 1 indicate higher odds of being a regular news reader.

5.2.1.2 Output

```
## # A tibble: 5 x 7
##   term        estimate std.error statistic  p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
```

```
## 1 (Intercept)   0.0175  0.0624      -64.9  0          0.0155    0.0197
## 2 agea          1.04    0.000941     38.5  0          1.04      1.04
## 3 genderMale    1.50    0.0330       12.2  3.33e-34   1.40      1.60
## 4 countryFR     0.692   0.0452       -8.16 3.34e-16   0.633     0.755
## 5 countryGB     1.48    0.0367       10.7  9.12e-27   1.38      1.59
```

Interpretation: Odds ratios >1 raise the chance of regular news use; check if the CI excludes 1 for age, gender, or specific countries before claiming significance.

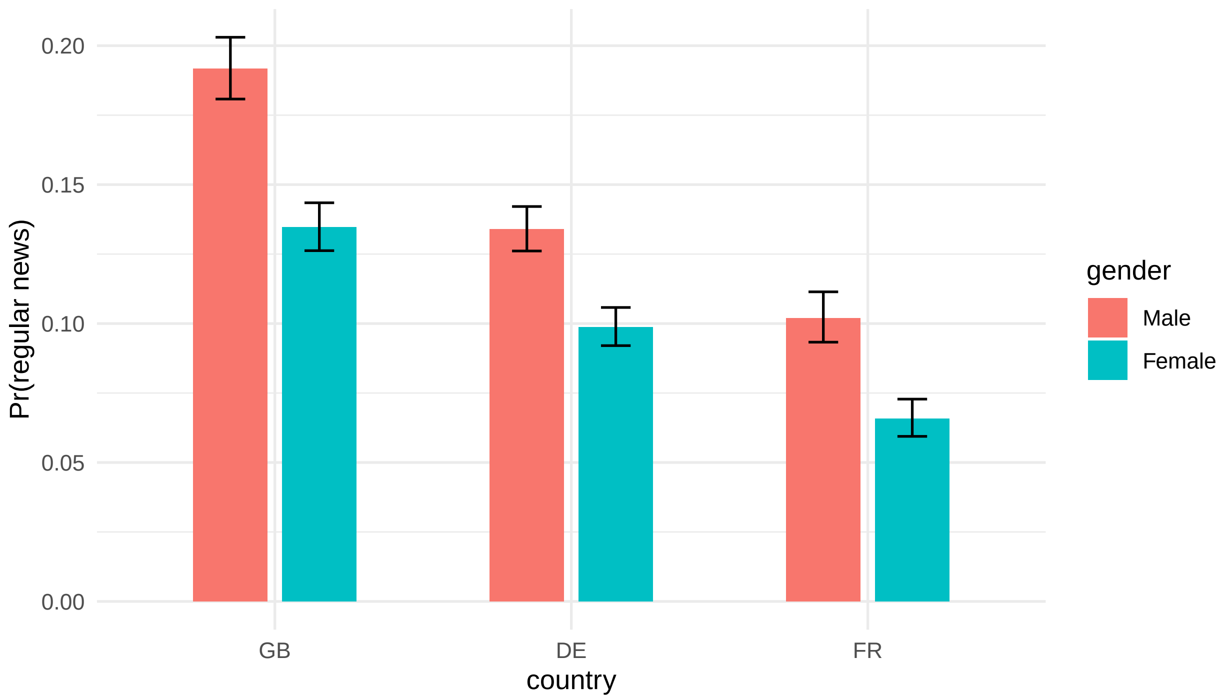
5.3 2. Binary × Binary interaction (gender × country)

```
logit2 <- glm(news_regular ~ gender * country + agea, data = ess, family = binomial())
plot_gender_country <- ggplot(nd2, aes(x = country, y = pr, fill = gender)) +
  geom_col(position = position_dodge(width = 0.6), width = 0.5) +
  labs(y = "Pr(regular news)") +
  theme_minimal()
```

5.3.0.1 Code

5.3.0.2 Output

```
## # A tibble: 7 x 7
##   term                 estimate std.error statistic  p.value conf.low conf.high
##   <chr>                   <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 (Intercept)            0.0180  0.0662      -60.7  0          0.0158    0.0205
## 2 genderMale             1.41    0.0522        6.61 3.83e-11   1.27      1.56
## 3 countryFR              0.643   0.0673       -6.57 5.14e-11   0.563     0.733
## 4 countryGB              1.42    0.0536        6.55 5.87e-11   1.28      1.58
## 5 agea                   1.04    0.000941     38.5  0          1.04      1.04
## 6 genderMale:countryFR   1.14    0.0907        1.47 1.42e- 1   0.957     1.37
## 7 genderMale:countryGB   1.08    0.0735        1.04 2.97e- 1   0.935     1.25
```



Interpretation: If male–female bars overlap within countries (CI bars), gender differences are modest. Compare across countries to see where the gap is largest.

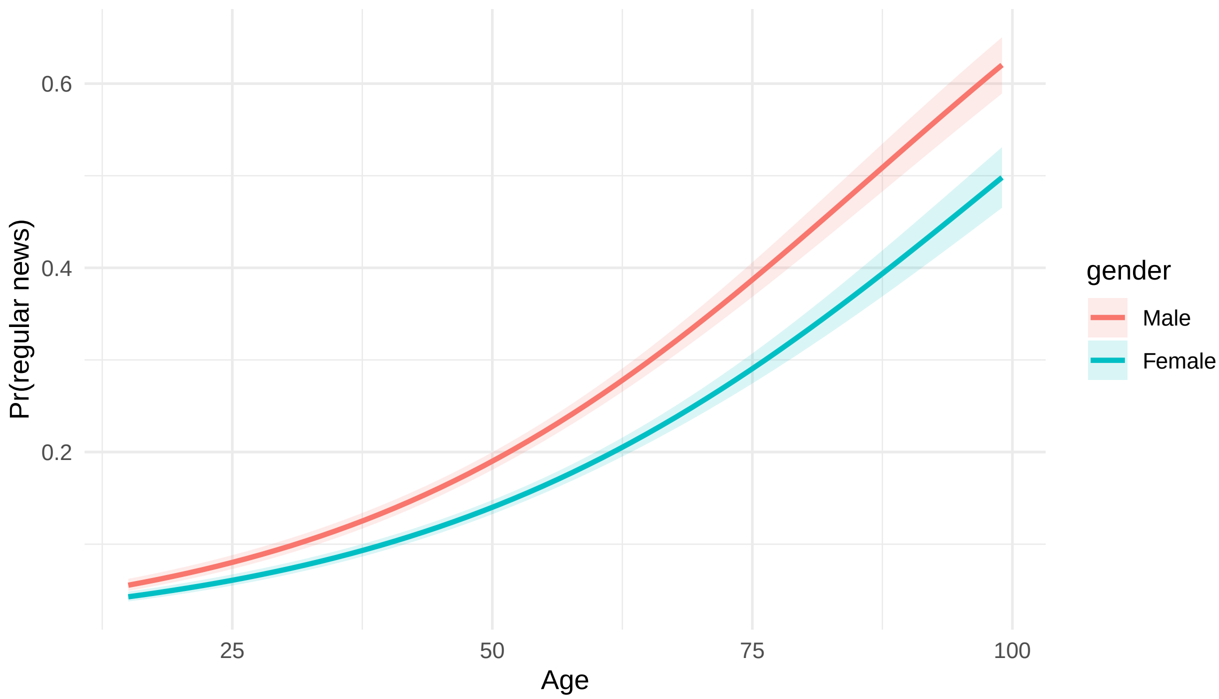
5.4 3. Binary × Continuous interaction (gender × age)

```
logit3 <- glm(news_regular ~ gender * agea + country + eduyrs, data = ess, family = binomial())
plot_gender_age <- ggplot(nd3, aes(x = agea, y = pr, color = gender)) +
  geom_line(size = 1) +
  labs(y = "Pr(regular news)", x = "Age") +
  theme_minimal()
```

5.4.0.1 Code

5.4.0.2 Output

```
## # A tibble: 7 x 7
##   term            estimate std.error statistic   p.value conf.low conf.high
##   <chr>              <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
## 1 (Intercept)       0.0109   0.114      -39.6  0           0.00868    0.0136
## 2 genderMale        1.26     0.113        2.03 4.22e-  2  1.01       1.57
## 3 agea              1.04     0.00141     26.2  1.61e-151  1.03       1.04
## 4 countryFR         0.733    0.0460      -6.76 1.34e- 11  0.669      0.802
## 5 countryGB         1.51     0.0370      11.2  3.74e- 29  1.41       1.63
## 6 eduyrs            1.03     0.00450      7.45 9.68e- 14  1.02       1.04
## 7 genderMale:agea   1.00     0.00192      1.41 1.58e-  1  0.999      1.01
```



Interpretation: Diverging ribbons indicate age effects differ by gender; if both ribbons rise similarly, the interaction is weak.

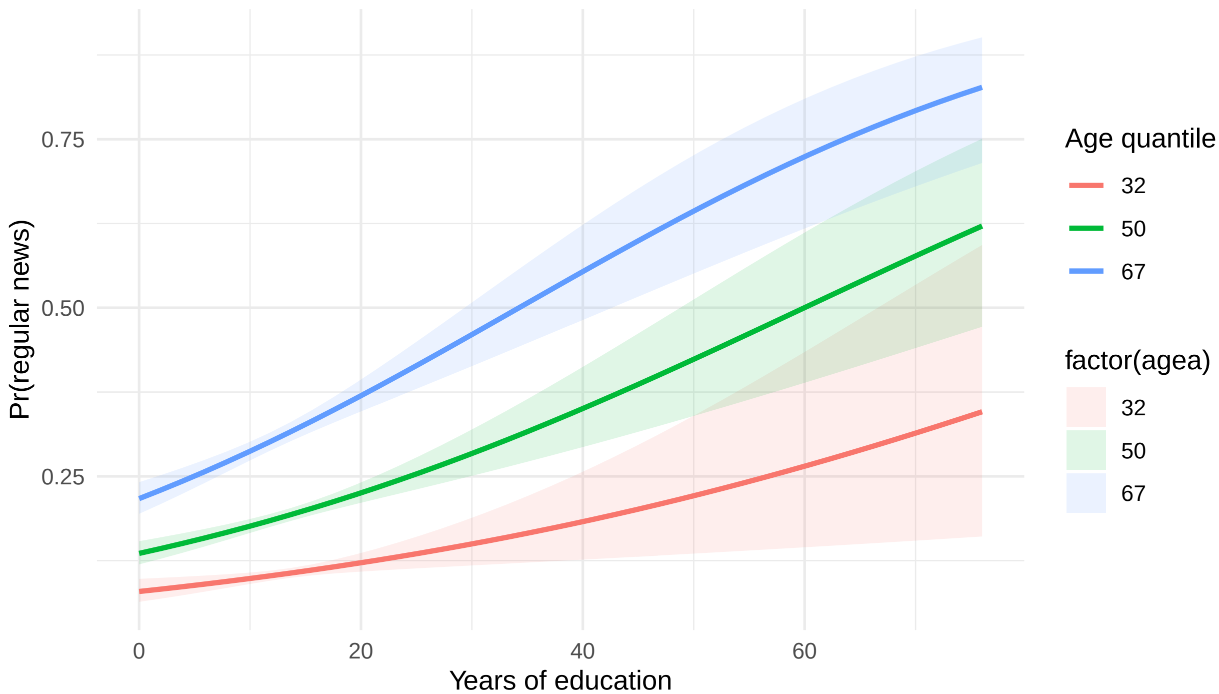
5.5 4. Continuous × Continuous interaction (age × education)

```
logit4 <- glm(news_regular ~ agea * eduyrs + gender + country, data = ess, family = binomial())
plot_age_edu <- ggplot(nd4, aes(x = eduyrs, y = pr, color = factor(agea))) +
  geom_line(size = 1) +
  labs(y = "Pr(regular news)", x = "Years of education", color = "Age quantile") +
  theme_minimal()
```

5.5.0.1 Code

5.5.0.2 Output

```
## # A tibble: 7 x 7
##   term        estimate std.error statistic  p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 (Intercept)   0.0133  0.223      -19.4   1.12e-83  0.00860    0.0206
## 2 agea          1.03    0.00353      9.47  2.80e-21  1.03       1.04
## 3 eduyrs        1.01    0.0162       0.707 4.80e- 1  0.980      1.04
## 4 genderMale    1.46    0.0334      11.4   6.41e-30  1.37       1.56
## 5 countryFR     0.737   0.0461      -6.63  3.29e-11  0.673      0.806
## 6 countryGB     1.52    0.0370      11.2   2.53e-29  1.41       1.63
## 7 agea:eduyrs   1.00    0.000270     1.44  1.50e- 1  1.00       1.00
```



Interpretation: Steeper lines for higher age quantiles would mean education matters more (or less) for older respondents; overlap implies limited moderation.

5.6 5. Three-way interaction (gender × age × country)

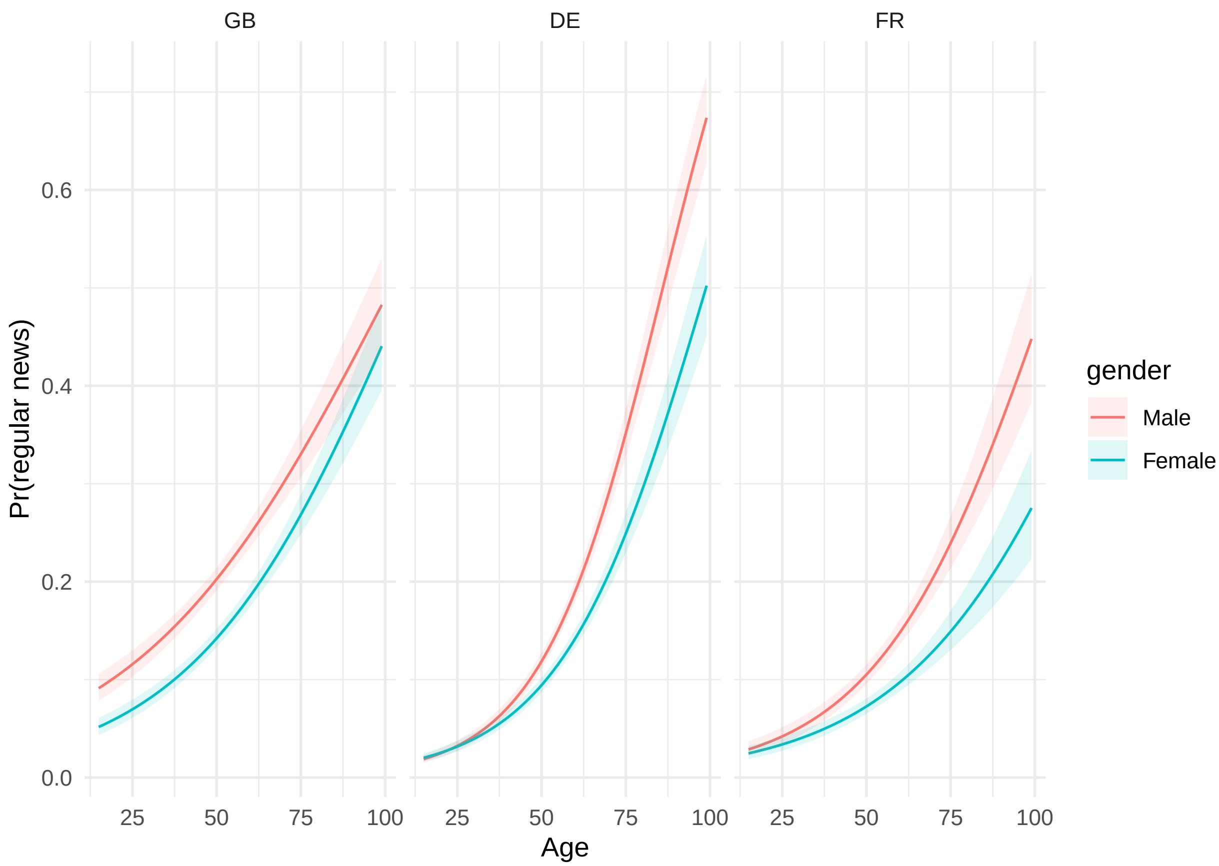
```
logit5 <- glm(news_regular ~ gender * agea * country + eduyrs, data = ess, family = binomial())
plot_threeway <- ggplot(nd5, aes(x = agea, y = pr, color = gender)) +
  geom_line() +
  facet_wrap(~ country) +
  labs(y = "Pr(regular news)", x = "Age") +
  theme_minimal()
```

5.6.0.1 Code

5.6.0.2 Output

```
## # A tibble: 13 x 7
##    term                   estimate std.error statistic   p.value conf.low conf.high
##    <chr>                     <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
##  1 (Intercept)             0.00660   0.164     -30.7   2.20e-206  0.00477   0.00907
##  2 genderMale              0.809     0.203      -1.04  2.97e-  1  0.543     1.20
##  3 agea                    1.05      0.00240    19.3   6.67e- 83  1.04      1.05
##  4 countryFR               1.52      0.232       1.82  6.91e-  2  0.965     2.40
##  5 countryGB               3.30      0.187       6.38  1.73e- 10  2.29      4.76
##  6 eduyrs                  1.03      0.00452     7.28  3.30e- 13  1.02      1.04
##  7 genderMale:agea         1.01      0.00336     2.79  5.33e-  3  1.00      1.02
##  8 genderMale:country~     1.29      0.324       0.793 4.28e-  1  0.685     2.44
```

```
## 9 genderMale:country~  2.46    0.260     3.46  5.37e- 4  1.48    4.10
## 10 agea:countryFR       0.986   0.00387  -3.66  2.57e- 4  0.979   0.993
## 11 agea:countryGB       0.986   0.00311  -4.69  2.77e- 6  0.980   0.992
## 12 genderMale:agea:co~  0.998   0.00542  -0.400 6.89e- 1  0.987   1.01
## 13 genderMale:agea:co~  0.985   0.00438  -3.34  8.46e- 4  0.977   0.994
```



Interpretation: Scan each country facet—if ribbons separate widely, the age–gender pattern is country-specific; overlapping ribbons suggest similar patterns across countries.

5.7 6. Marginal effects and interpretation

```r
# Average marginal effects for age within each country
ame_age <- marginaleffects(logit5, variables = "agea", by = "country")

# Marginal effect of being Female at age 25 vs age 65
me_female_age <- marginaleffects(logit5, variables = "gender", newdata = datagrid(agea = c(25, 65)))
```

- Prefer predicted probabilities and marginal effects over raw log-odds.
- Inspect separation or influential points with `performance::check_model(logit5)` if desired.

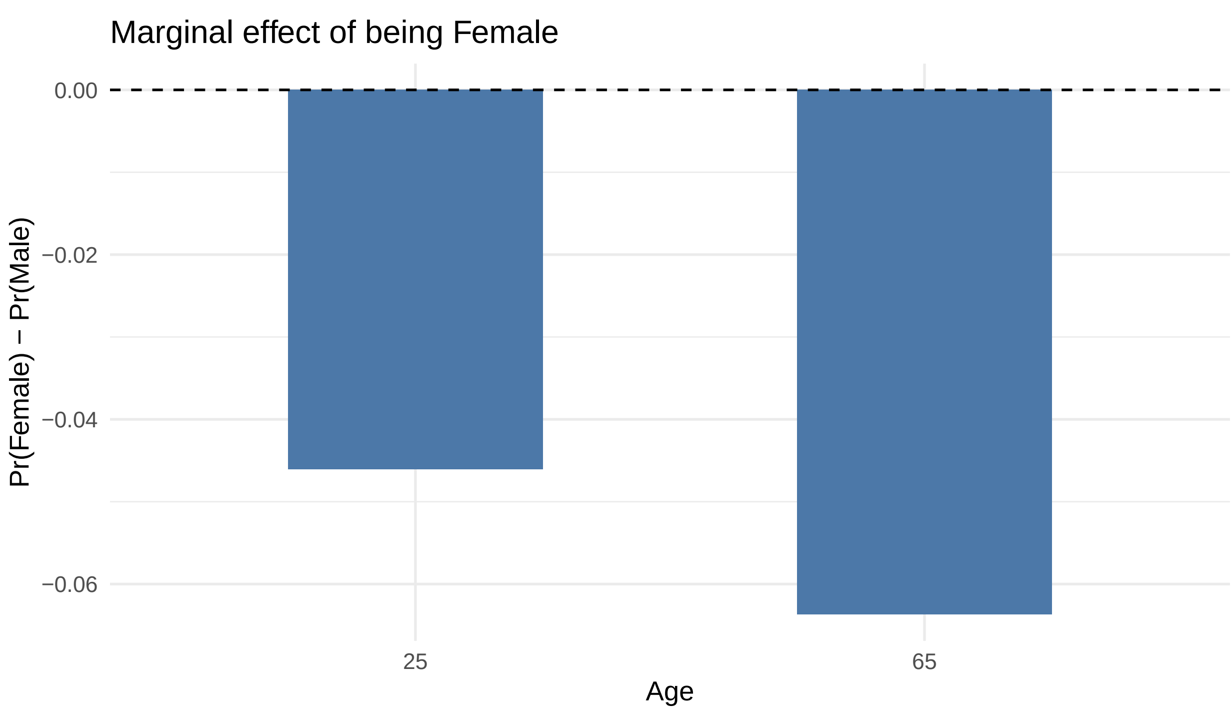
5.7.1 AMEs and focal contrasts

```r
ame_table <- tibble(
  variable = c("agea (+1 year)", "gender (Female vs Male)"),
  AME = c(ame_age, ame_fem)
)

me_female_diff <- data.frame(agea = c(25,65), diff = pred_f - pred_m)  # from fits chunk
```

5.7.1.1 Code

5.7.1.2 Output

```
## # A tibble: 2 x 2
##   variable                    AME
##   <chr>                     <dbl>
## 1 agea (+1 year)          0.00451
## 2 gender (Female vs Male) 0.00237
```



Interpretation: The AME for age reports the average change in the probability of regular news use for a one-year increase in age. The bar chart shows how the female–male gap differs at ages 25 vs 65; positive bars mean higher probability for women at that age.

Country-level pooling choices (fixed vs multilevel) are expanded in the next chapter.

5.8 Problem set — Logistic regression practice

1. Recode the outcome as `news_daily = nwsptot >= 5` and re-estimate `logit3`. How do the marginal effects of age change when the bar for "regular" consumption is higher?

2. Add `urban` (1/2 = urban, 3–5 = non-urban) as a predictor and interact it with `country`. Which country shows the largest urban–rural gap in news readership?
3. Compare `logit4` and `logit5` using AIC and pseudo R^2 (`pscl::pR2`). Which balance of complexity and fit seems reasonable for classroom examples?
4. For one model, translate results into plain language for a non-technical audience: choose two profiles (e.g., 30-year-old woman in GB vs 60-year-old man in FR) and report predicted probabilities.

These exercises deliver a full set of interaction types in a nonlinear setting, ready to complement your lecture slides.

6 Country effects — fixed effects and multilevel modeling

Nested survey data call for explicit country handling. This chapter shows two approaches:

- **Country fixed effects (dummies):** absorbs unobserved, time-invariant differences between GB, DE, and FR.
- **Multilevel (random-intercept) logistic regression:** partially pools country effects, reducing noise for small samples and enabling variance decomposition.

```
library(dplyr)
library(ggplot2)
library(broom)
library(broom.mixed)
library(lme4)
library(tidyr)

source("R/clean_ess.R")

ess <- clean_ess()
```

6.1 1. Country fixed effects (logit with dummies)

- **Model**: `news_regular ~ agea + gender + eduyrs + country`
- **Interpretation**: Country coefficients capture average gaps relative to the reference country (DE, alphabetically first).

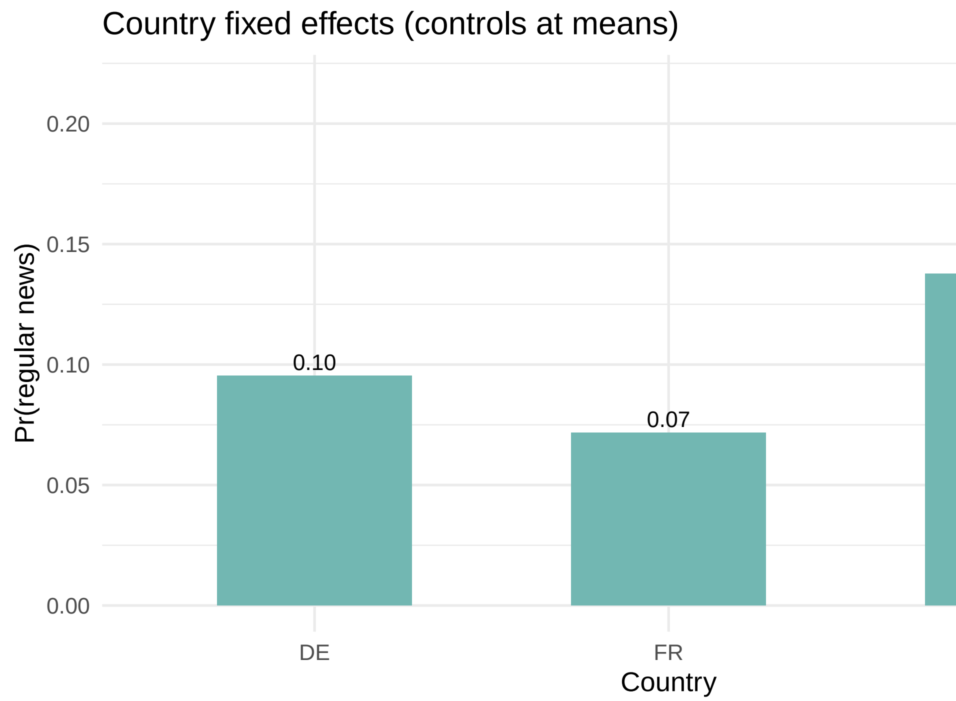
```
fe_logit <- glm(news_regular ~ agea + gender + eduyrs + country,
                data = ess, family = binomial())
fe_tidy <- broom::tidy(fe_logit, exponentiate = TRUE, conf.int = TRUE)
```

6.1.0.1 Odds ratios

6.1.0.1.1 Table

```
## # A tibble: 6 x 7
##   term        estimate std.error statistic  p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 (Intercept)  0.00997  0.0971      -47.5   0         0.00824    0.0121
## 2 agea         1.04     0.000998     38.4   0         1.04       1.04
```

```
## 3 genderMale   1.47      0.0334       11.5  1.57e-30  1.37        1.57
## 4 eduyrs       1.03      0.00449       7.55 4.36e-14  1.03        1.04
## 5 countryFR    0.733     0.0460       -6.75 1.49e-11  0.670       0.802
## 6 countryGB    1.51      0.0370       11.2  3.56e-29  1.41        1.63
```



6.1.0.1.2 Country contrasts (plot)

Takeaway: Fixed effects wipe out between-country bias but treat each country independently—estimates can be noisy if a country has few cases.

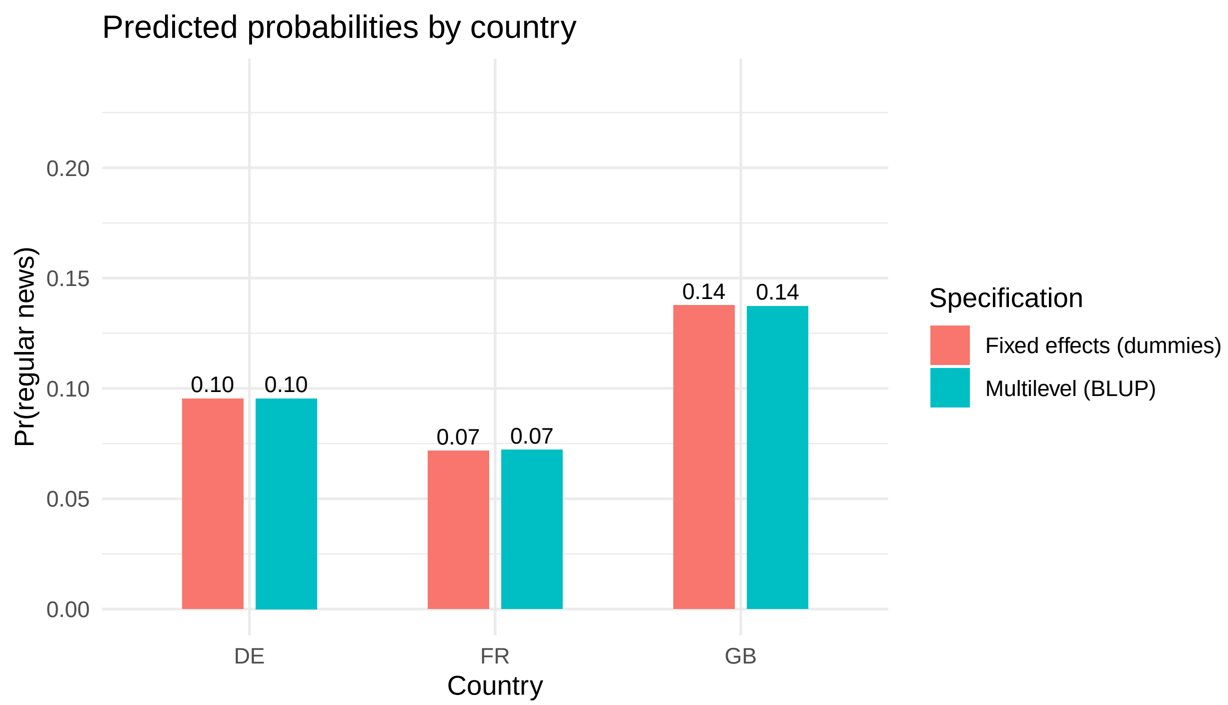
6.2 2. Multilevel logistic model (random intercepts by country)

- **Model**: `news_regular ~ agea + gender + eduyrs + (1 | country)`
- **Why**: Partial pooling shrinks extreme country estimates toward the grand mean, improving out-of-sample stability.

```
ml_logit <- glmer(news_regular ~ agea + gender + eduyrs + (1 | country),
                  data = ess, family = binomial(),
                  control = glmerControl(optimizer = "bobyqa"))

var_u0 <- as.numeric(VarCorr(ml_logit)$country)
icc <- var_u0 / (var_u0 + pi^2 / 3)
```

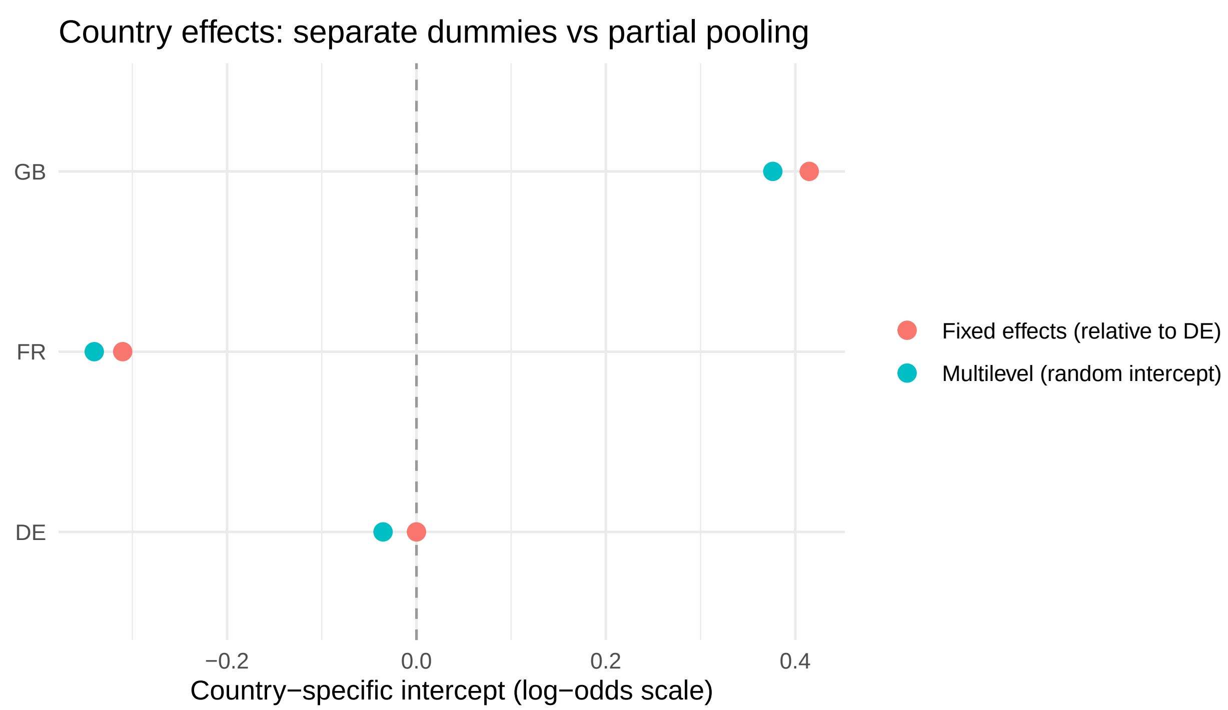
Intraclass correlation (ICC): 2.58% of the variance in the log-odds of regular news use is at the country level.

6.2.1 Country predictions: fixed vs multilevel



Interpretation: Multilevel estimates are closer together because extreme country effects are shrunk toward the overall mean.

6.2.2 Visualizing country intercepts (shrinkage)



Reading the plot: Dots further from 0 indicate larger country-specific deviations. The multilevel model shrinks them toward zero, while fixed effects leave them unchanged.

6.3 3. Random slopes for gender (optional extension)

With more countries, we could allow the gender gap to vary by country:

```
ml_logit_gender <- glmer(
  news_regular ~ agea + gender + eduyrs + (1 + gender | country),
  data = ess, family = binomial(),
  control = glmerControl(optimizer = "bobyqa")
)
```

In the current three-country sample this model may be over-parameterized; the random-intercept specification above is the stable classroom default.

6.4 Practice prompts

1. Add an **urban** fixed effect to both models. Do city–rural gaps widen or narrow once country pooling is applied?
2. Refit the multilevel model with `news_regular` defined as daily readership (`nwsptot >= 5`). How does the ICC change?
3. Replace `agea` with a spline (`splines::ns(agea, df = 3)`) inside both models and compare the resulting age profiles by country.

7 Survey weighting — why and how

Survey data are collected with unequal selection probabilities. Inference should reflect the design to avoid biased point estimates and standard errors.

- **Notation:** Let $w_i = 1/\pi_i$ be the design (inverse-probability) weight. For a finite population mean $\bar{Y} = \sum_i w_i Y_i / \sum_i w_i$. For regression, weighted likelihoods re-scale each case by w_i.
- **ESS fields used:** `pweight` (post-stratification weight), `psu` (primary sampling unit), `stratum` (strata). We set `options(survey.lonely.psu = "adjust")` to stabilize single-PSU strata.

7.0.1 What these design variables mean (plain language)

- **`pweight` (post-stratification weight):** Adjusts for unequal inclusion probabilities *and* aligns the achieved sample with known population margins (e.g., age × gender × region). Large `pweight` values upweight under-represented respondents; small values downweight over-represented ones.
- **`psu` (primary sampling unit):** The first cluster stage of selection (e.g., municipalities or postcode sectors). Respondents inside the same PSU share fieldwork and selection features, so their responses are correlated.
- **`stratum` (strata):** Mutually exclusive groups within which PSUs were sampled (e.g., by region × urbanicity). Stratification improves precision; variance estimation must respect it.
- **Design degrees of freedom:** With clustering and stratification, the effective df are closer to the number of PSUs minus strata, not the raw respondent count—hence the importance of design-aware SEs.

```r
library(dplyr)
library(ggplot2)
library(tidyr)
library(survey)
library(broom)

# Synthetic microdata to illustrate weighting without depending on raw ESS file quirks
set.seed(42)
n_psu <- 120
n_by_psu <- sample(8:18, n_psu, replace = TRUE)

demo_psu <- tibble(
  psu = 1:n_psu,
  stratum = sample(1:20, n_psu, replace = TRUE),
  country = sample(c("GB", "DE", "FR"), n_psu, replace = TRUE, prob = c(.4, .35, .25)),
  pweight = runif(n_psu, 0.5, 3),
  n = n_by_psu
)

ess_w <- demo_psu |>
  uncount(n) |>
  mutate(
    agea = round(rnorm(n(), 50, 15)),
    gender = sample(c("Male", "Female"), n(), replace = TRUE),
    # Generate outcome with country- and age-dependent probability
    linpred = -1 + 0.6 * (country == "GB") + 0.3 * (country == "FR") +
              0.01 * (agea - 50) + 0.4 * (gender == "Female"),
    news_regular = rbinom(n(), 1, plogis(linpred))
  ) |>
  select(psu, stratum, country, pweight, agea, gender, news_regular) |>
  mutate(country = factor(country),
         gender = factor(gender))

options(survey.lonely.psu = "adjust")
des <- svydesign(ids = ~psu, strata = ~stratum, weights = ~pweight,
                 data = ess_w, nest = TRUE)

svy_n <- nrow(des$variables)
```

7.1 1. Weighted descriptive estimates

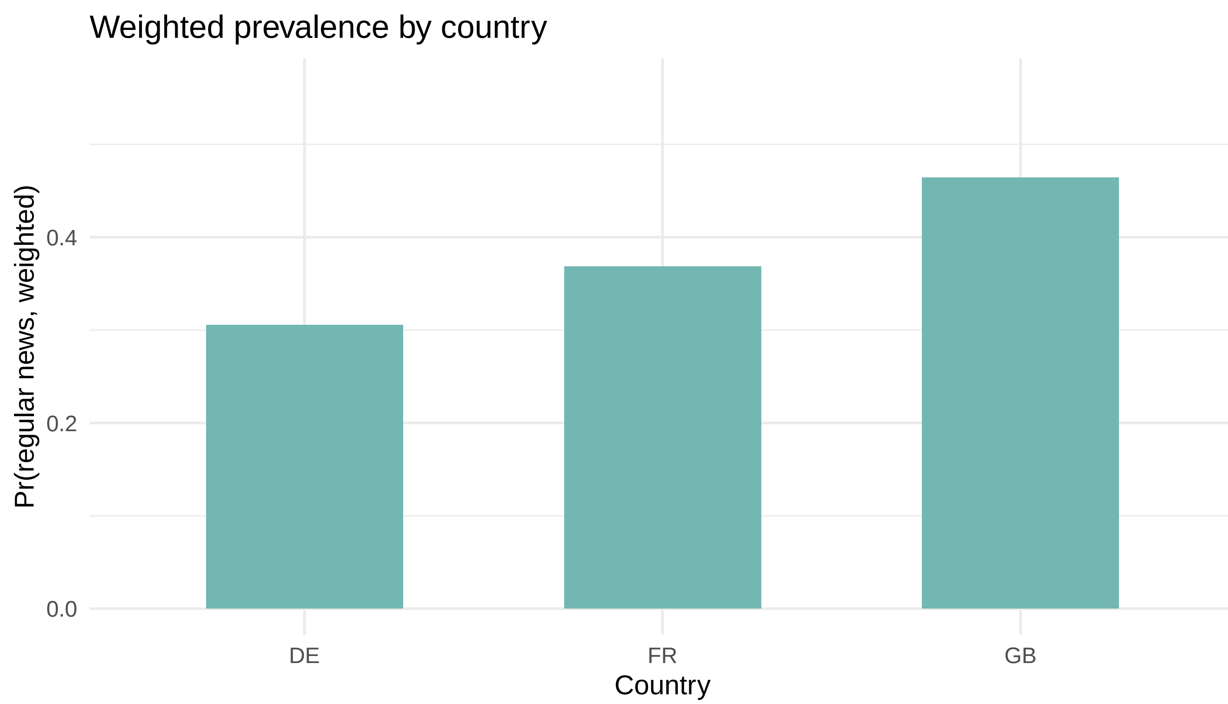
Sample used after complete-case filtering: 1556 respondents.

```r
svymean(~news_regular, des)
```

```
##                 mean     SE
## news_regular 0.38522 0.0153
```

- The point estimate is the **design-weighted mean**; SEs account for clustering and stratification.
- **Design effect** (`svymean(..., deff=TRUE)`) tells how much variance inflation comes from the design versus SRS.

7.1.1 Country-weighted proportions



7.2 2. Weighted regression: linear probability and logit

```
lpm_w <- svyglm(news_regular ~ agea + gender + country,
                design = des, family = gaussian())
tidy(lpm_w)
```

```
## # A tibble: 5 x 5
##   term        estimate std.error statistic     p.value
##   <chr>          <dbl>     <dbl>     <dbl>       <dbl>
## 1 (Intercept)  0.240    0.0493        4.86 0.00000453
## 2 agea         0.00214  0.000781      2.74 0.00741
## 3 genderMale  -0.0875   0.0243       -3.60 0.000509
## 4 countryFR    0.0679   0.0390        1.74 0.0851
## 5 countryGB    0.155    0.0276        5.61 0.000000193
```

```
logit_w <- svyglm(news_regular ~ agea + gender + country,
                  design = des, family = quasibinomial())
tidy(logit_w, exponentiate = TRUE)
```

```
## # A tibble: 5 x 5
##   term        estimate std.error statistic     p.value
##   <chr>          <dbl>     <dbl>     <dbl>       <dbl>
## 1 (Intercept)    0.327   0.221       -5.06 0.00000204
## 2 agea           1.01    0.00346      2.69 0.00833
## 3 genderMale     0.683   0.105       -3.64 0.000446
```

```
## 4 countryFR       1.36    0.174         1.76 0.0815
## 5 countryGB       1.95    0.122         5.48 0.000000335
```

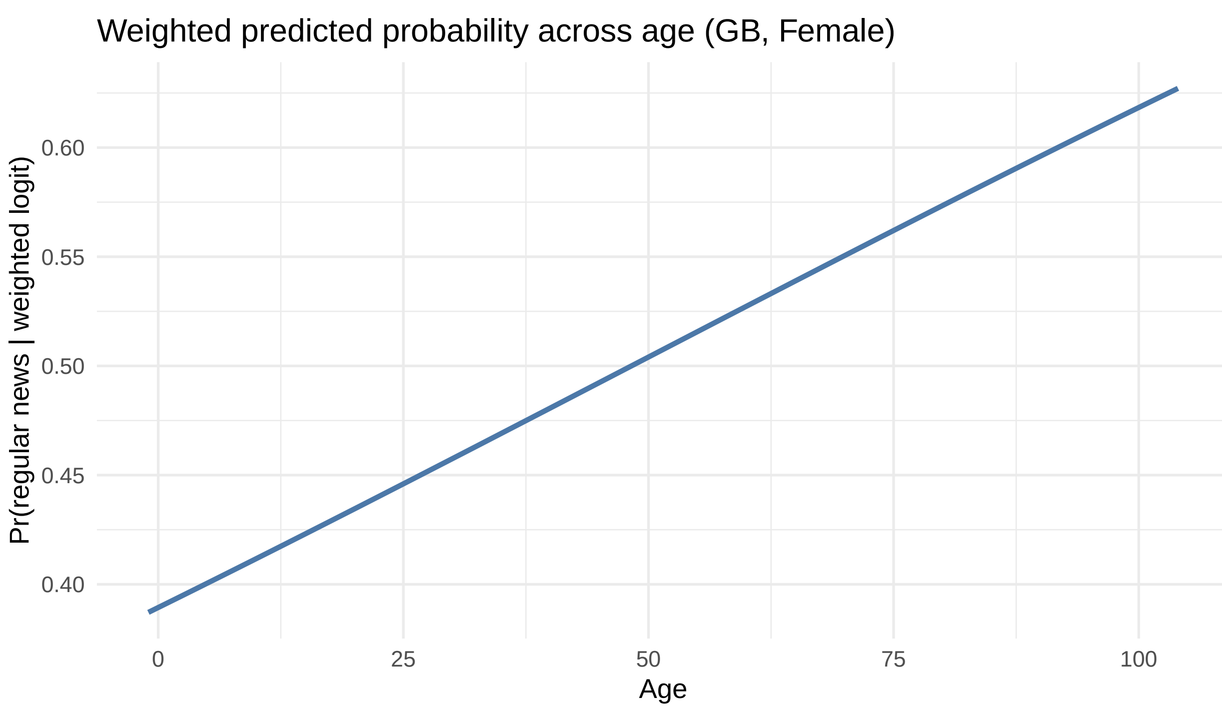
Comparison points for students:

- **Weights + clustering**: `svyglm` gives design-consistent SEs; plain `glm` does not.
- **Quasibinomial** keeps logit link but uses robust variance; estimates mirror survey-weighted MLE when weights are scaled.
- **LPM vs logit under weights**: LPM slopes stay probability-difference interpretation; logit ORs remain multiplicative.

7.3 3. When to weight (practical guidance)

- Use design/post-strat weights when estimating **population levels** (means, totals, prevalence) or effects that might shift with differential selection.
- In randomized experiments or when modeling **causal effects with ignorable sampling**, weights may be optional; still cluster-robust SEs matter.
- If the research question is **sample-only prediction**, weights can be skipped, but report that scope-of-inference is limited.

7.4 4. Small worked example: weighted marginal effect of age



7.5 Practice prompts

1. Recompute weighted country gaps using `anweight` instead of `pweight`. How do the estimates move?
2. Add education to the weighted logit. Do ORs shift relative to the unweighted model in Chapter 5?
3. Estimate a design effect for `news_regular` and discuss how many "effective" observations the design corresponds to.

8 Appendix — Data and build notes

8.1 ESS subset

- File: `ess.csv`
- Countries: GB, DE, FR
- For variable descriptions, open the bundled HTML codebook: `ESS1e06_...subset codebook.html` in your browser.
- Common missing-value codes: 66/77/88/99 or blank strings. All chapters use `na_if()` to drop them before analysis.

8.2 Extending the material

- Swap dependent variables: `pplhlp` (people helpful) or `pplfair` (people fair) can replace `ppltrst` without changing structure.
- Add country-level covariates by merging lookups (e.g., GDP or media freedom indices) before running multilevel models.
- Convert interaction plots to `ggplotly` for live demos if desired.