Demand Estimation

2025-06-05

```
library(tidyverse)
library(ggplot2)
library(stargazer)
library(AER)
library(dplyr)
library(broom)      # for tidy() / glance()
library(sandwich)   # for robust SE's
library(lmtest)
library(plm)
library(fixest)
library(modelsummary)
library(glmnet)
```

Demand Estimation

1. Primary Method: Instrumental-Variables (Two-Stage) Estimation

1.1 Identification challenge – endogeneity of `rate_actual`

- **Why is `rate_actual` endogenous?** The raw regression

$$\Pr(\text{booking}_i = 1 \mid \text{rate\_actual}_i = x,\ X_i) = F(\beta_0 + \beta_1 \ln(x) + X_i'\gamma).$$

  (e.g. a logit or linear-probability model) would "fake" an elasticity = . But `rate_actual_i` is chosen (by the hotel manager) in response to **unobserved demand shocks** (e.g. last-minute spikes, competitor repricing in real time). Whenever customers see price = p and then book (or do not), the manager may have already adjusted p partly because she anticipated higher demand. Those same unobserved demand shocks also directly affect booking probability, so a naïve regression of book_i on $\ln(\text{rate\_actual}_i)$ is biased.

- **Data feature that helps.** At each query i, we do observe:

  1. the software's "ideal" price recommendation $\text{rate\_recommended}_i$ (which is not shown to the customer),
  2. rich covariates X_i (everything the analytics engine knew when it computed its recommended price),
  3. the manager's **actual** decision rate_actual_i, which often lags the recommendation.

  Because the software's recommendation $\text{rate\_recommended}_i$ is based on **all** observable demand-side features **before** the manager sees them, and because (by design) the recommendation algorithm is "as if" exogenous once we condition on $X_i, we can treat $\text{rate\_recommended}_i$ **as an instrument** for $\ln(\text{rate\_actual}_i)$.

– Concretely: the analytics software uses $\{X_i, \text{date\_of\_stay}_i, \text{days\_ahead}_i\}$ (and perhaps local-event, holiday, weather, competitor-price features) to compute $\text{rate\_recommended}_i$. The manager then "Applies Price" but often with a one-to-two-day lag or partial override (to juggle housekeeping, staffing, etc.). As a result,

$$\underbrace{\text{rate\_recommended}_i}_{\text{based on } X_i \text{, public info}} \longrightarrow \underbrace{\text{rate\_actual}_i}_{\text{hotel manager's lagged choice}}$$

but $\text{rate\_recommended}_i$ is never directly seen by customers, only the lagged actual price is.

– **Exclusion restriction.** Once we control for the same vectors X_i that go into the recommendation engine, any residual variation in $\text{rate\_recommended}_i$ should only affect booking probability via its effect on the final chosen price rate_actual_i. We assume the recommendation engine is "well-specified" in that it already conditions on all observed demand covariates X_i. Therefore

$$\text{rate\_recommended}_i \;\perp\!\!\!\perp\; (\text{unobserved demand shocks}) \;\Big|\; X_i\,.$$

Hence the two-stage least squares (2SLS) strategy (or an IV-logit control-function version) recovers a consistent price elasticity.

- **Formal 2SLS setup.**

1. **First stage (predict actual price by recommended price & controls).**

$$\ln(\text{rate\_actual}_i) \;=\; \pi_0 + \pi_1 \ln(\text{rate\_recommended}_i) + \sum_{k=1}^{K} \delta_k X_{i,k} + \sum_{t=1}^{T} \alpha_t \mathbf{1}\{\text{date\_of\_stay}_i = t\} + \sum_{d=0}^{D} \zeta_d \mathbf{1}\{\text{day}$$

– Here,
  * $\ln(\text{rate\_recommended}_i)$ is our instrument.
  * $X_{i,k}$ are all the "XX – XX " continuous or binary features (hotel reputation, room features, local holidays, weather metrics, competitor price summaries) that the recommendation engine used.
  * We include **date-of-stay fixed effects** (one dummy per calendar date t) because demand may shift systematically by day.
  * We include **days-ahead dummies** ($\mathbf{1}\{\text{days\_ahead} = d\}$) to absorb any mechanical correlation between look-ahead and willingness-to-pay.
  * The residual ν_i picks up manager-specific pricing deviations (errnoises, operational constraints, staffing shortages, etc.).

– Interpretation: Once we've controlled for $\{X_i, \text{date\_of\_stay}_i, \text{days\_ahead}_i\}$, the only remaining exogenous "forcing" of $\ln(\text{rate\_actual}_i)$ is through $\ln(\text{rate\_recommended}_i)$.

2. **Second stage (book/no-book regression on predicted log-price).** Since the outcome is binary $book_i \in \{0, 1\}$, we have two common choices:

– **Linear-probability 2SLS:**

$$book_i \;=\; \beta_0 + \beta_{\text{price}} \ln(\widehat{\text{rate\_actual}}_i) + \sum_{k=1}^{K} \gamma_k X_{i,k} + \sum_{t=1}^{T} \eta_t \mathbf{1}\{\text{date\_of\_stay}_i = t\} + \sum_{d=0}^{D} \kappa_d \mathbf{1}\{\text{days\_ahead}_i =$$

where $\ln(\widehat{\text{rate\_actual}}_i)$ is the fitted value from (first-stage). Then β_{price} is our estimate of $\frac{\partial \Pr(book_i=1)}{\partial \ln(\text{rate\_actual}_i)}$. That is exactly the "elasticity" in the probability scale: a 1 percent ↑ in price causes _price × 1 percent ↓ in booking probability.\

– **Control-function IV-logit:** Alternatively, one can run a probit/logit of $book_i$ on $\ln(\text{rate\_actual}_i)$ plus the **first-stage residual** $\widehat{\nu}_i \equiv \ln(\text{rate\_actual}_i) - \ln(\widehat{\text{rate\_actual}}_i)$ and all X_i (plus fixed effects). In that control-function approach, a significant coefficient on $\widehat{\nu}_i$ flags remaining endogeneity. The logit coefficient on $\ln(\text{rate\_actual}_i)$ then backs out a "local average structural elasticity."

- **Key assumption (instrument validity).**

  1. **Relevance:** $Cov(\ln(\text{rate\_recommended}_i), \ln(\text{rate\_actual}_i)) \neq 0$ after controlling for $X_i, \text{date\_of\_stay}_i, \text{days\_ahead}_i$. In practice we check π_1 in the first-stage—F-statistic should exceed 10. Empirically, one typically sees a strong correlation (since managers follow recommendations 70 – 90 percent of the time).
  2. **Exclusion:** Once we condition on the same X_i that feed into the software, any remaining → "recommended price" can only affect booking through its effect on the final, lagged "actual price." In other words,

$$\{book_i(0),\, book_i(1)\} \;\perp\!\!\!\perp\; \ln(\text{rate\_recommended}_i) \;\Big|\; \{X_i, \text{date\_of\_stay}_i, \text{days\_ahead}_i\}\,.$$

  We must argue (and test where possible) that there is no direct "mental anchor" effect of the recommended price on the manager's choice, once we already know all the covariates. For example, if the manager sees past-week occupancy + competitor data and then the system suggests \$120, the manager's final price \$118 is driven by "

$$X_i \mapsto \$118$$

  " plus a small operational wiggle. They never put \$120 on the booking page, so customers do not see it.

- **Why this IV solves the endogeneity.**

  – Unobserved demand shocks (e.g. instant local event, last-minute conference surge) might raise $P(book_i = 1)$ and cause the manager to override the recommendation upward at time \$t). But those same unobserved shocks do **not** shift $\text{rate\_recommended}_i$ beyond what X_i already encoded—*because* X_i already includes the (public) event, holiday, weather, competitor-price data that feed into the engine. Thus, conditional on X_i, $\text{rate\_recommended}_i$ is exogenous, and we can use it to "rotate out" the piece of $\ln(\text{rate\_actual}_i)$ that comes from manager endogeneity.

  – In IV language:

  1. We assume a structural booking-probability equation

$$book_i \;=\; \Phi(\beta_0 + \beta_{\text{price}} \ln(\text{rate\_actual}_i) + X_i'\gamma + ...) \;+\; u_i$$

     where u_i correlates with $\ln(\text{rate\_actual}_i)$.
  2. We assume a pricing equation

$$\ln(\text{rate\_actual}_i) \;=\; \pi_0 + \pi_1 \ln(\text{rate\_recommended}_i) + X_i'\delta + ... + v_i.$$

     By construction, $\ln(\text{rate\_recommended}_i) \perp u_i \mid X_i$.
  3. Thus 2SLS gives consistent $\hat{\beta}_{\text{price}}$.

- **Functional form.** We take logs of both recommended and actual price so that β_{price} is directly interpretable **as** a price **elasticity** in percentage terms. In the second stage (binary booking), we can do either:

  1. **2SLS LPM:** treat $book_i \in \{0, 1\}$ as a linear outcome. Then $\hat{\beta}_{\text{price}} = -0.25$ means "a 1 percent ↑ in price lowers booking probability by 0.25 percentage points."

2. **Control-function logit:** estimate

$$book_i \;=\; logit^{-1}\Big(\beta_0 + \beta_{\text{price}}\ln(\text{rate\_actual}_i) + \theta\,\hat{v}_i + X_i'\gamma\Big) + e_i,$$

and interpret β_{price} as a "local elasticity" near observed prices. If $\hat{v}_i$ is insignificant, then price is effectively exogenous after controlling for X_i.

- **Interpretation (business insight).**
  - Once we recover $\hat{\beta}_{\text{price}} =: \hat{\eta}$, that is wer **own price elasticity** of booking probability, holding constant (in expectation) all the other observable drivers X_i.
  - If $\hat{\eta} = -0.30$, it means: "ceteris paribus, a 1 percent ↑ in price reduces the probability a given potential customer will book by 0.30 percent." In high-demand periods (when instrumented price is high relative to recommended), any further price ↑ has a larger % effect on bookings.

- **Why this method "copes" with the challenge:**
  1. It isolates an exogenous component $\ln(\text{rate\_recommended}_i)$ that is **as good as randomly assigned**, conditional on X_i. By running IV, we strip away all the "manager lag / unobserved shocks → price" correlation.
  2. We include all **observable** confounders that the analytics software already used, so that $\text{rate\_recommended}_i$ is valid once we condition on the same information set.
  3. We let the data pick flexible functional forms: (a) date-of-stay fixed effects soak up seasonality; (b) days-ahead dummies soak up "lead-time" patterns; (c) plugging in a log-log specification makes interpretation straightforward (elasticity).

2 Impelementation

```
# 1. Read and inspect the data
# ----------------------------
df <- read.csv("demand_est.csv", stringsAsFactors = FALSE)
```

```
# 2. Create log-price variables
# ----------------------------
df <- df %>%
  mutate(
    log_price_actual      = log(rate_actual),
    log_price_recommended = log(rate_recommended)
  )
```

```
# 3. Convert 'date_of_stay' and 'days_ahead' into factor dummies
# ---------------------------------------------------------------
#    It's often preferable to treat them as factors so R creates a full set of indicator variables.
df <- df %>%
  mutate(
    date_of_stay_factor = factor(date_of_stay),
    days_ahead_factor   = factor(days_ahead)
  )

# 4. Identify the "X_i" covariates: XX2 … XX250 and V251
# ------------------------------------------------------
#    We'll grab all column names that start with "XX" plus "V251" if it exists.
```

```r
xx_vars <- grep("^XX", names(df), value = TRUE)
if ("V251" %in% names(df)) {
  xx_vars <- c(xx_vars, "V251")
}
```

```r
# ------------------------------------------------------------
# 8.1. Specify the 2SLS formula: outcome ~ endogenous + exogenous | instruments + exogenous
iv_formula <- as.formula(
  paste0(
    "bookings ~ log_price_actual + ",
    paste(xx_vars, collapse = " + "), " + date_of_stay_factor + days_ahead_factor",
    " | log_price_recommended + ",
    paste(xx_vars, collapse = " + "), " + date_of_stay_factor + days_ahead_factor"
  )
)

# 8.2. Fit ivreg() in one line
iv_fit <- ivreg(iv_formula, data = df)
```

```r
# 3. Compute cluster-robust standard errors (cluster by date_data)
# ------------------------------------------------------------------
cluster_se_vec <- sqrt(diag(vcovCL(iv_fit, cluster = df$date_data)))
```

```r
# 4. Use stargazer, omitting ALL 250 "XX" covariates and the fixed-effects
# ------------------------------------------------------------------------
stargazer(
  iv_fit,
  se            = list(cluster_se_vec),
  omit          = c("date_of_stay_factor", "days_ahead_factor", "XX"),
    # "XX" matches any variable with "XX" in its name, i.e. XX2-XX250
  omit.labels   = c("Date-of-Stay FE", "Days-Ahead FE", "All XX2-XX250 Controls"),
  covariate.labels = c("Log(Price Actual)"),
    # We only keep a human-readable label for the endogenous variable.
    # All XX2-XX250 are omitted, so we don't need labels for them.
  keep.stat     = c("n", "rsq", "adj.rsq", "f"),
  digits        = 3,
  no.space      = TRUE,
  title         = "2SLS IV Regression: Booking Probability",
  type          = "text"
)
```

```
##
## 2SLS IV Regression: Booking Probability
## ================================================
##                             Dependent variable:
##                         ---------------------------
##                                    bookings
## ------------------------------------------------
## Log(Price Actual)                  0.117***
##                                     (0.031)
## V251                               -0.546***
##                                     (0.092)
```

```
## Constant                          0.848**
##                                   (0.362)
## ------------------------------------------------
## Date-of-Stay FE                     Yes
## Days-Ahead FE                       Yes
## All XX2-XX250 Controls              Yes
## ------------------------------------------------
## Observations                      50,000
## R2                                 0.192
## Adjusted R2                        0.182
## ================================================
## Note:                  *p<0.1; **p<0.05; ***p<0.01
```

The first-stage IV regression yields a positive and highly significant coefficient on Log(Price Actual) of 0.117 (standard error 0.032), implying that, all else equal, a 1 percent increase in the actual per-night price is associated with a 0.117 percentage-point increase in booking probability. V251 enters with a coefficient of –0.546 (SE 0.092), indicating that each one-unit rise in that hotel-side index lowers booking likelihood by about 0.55 percentage points when price and all other "XX" controls, date-of-stay dummies, and days-ahead dummies are held constant. The constant term of 0.848 (SE 0.362) represents the baseline booking probability when log price and V251 (as well as all other covariates) are zero. Because the coefficient on log price is positive rather than negative, it suggests that the variation in price exploited by the instrument is driven primarily by high-demand periods—in other words, on dates when the software recommended a higher price (and the manager subsequently raised the actual price), booking propensity was already elevated for unobserved reasons, and the IV does not fully isolate downward-sloping demand variation. In that sense, the sign indicates residual endogeneity: instead of capturing a true price-elasticity of demand, it captures the tendency for higher actual prices to coincide with greater underlying willingness-to-book. The F-statistics and R^2 (about 0.192 overall) confirm that while the model explains a moderate share of booking variation, the positive elasticity coefficient means that unobserved demand shocks tied to price spikes have not been entirely purged by the instrument.

3. Alternative Method: Control-Function / Selection-on-Observables via Double-Machine-Learning

Suppose for a moment that one does **not trust** the exclusion restriction for the recommended price (e.g. maybe the manager leaks the recommended price to the front-desk staff, who mention it to customers in conversation), or that "noise" in the recommendation still correlates with demand shocks in a way we cannot fully control. Then one alternative approach is to treat **price as "conditionally exogenous" given a very-high-dimensional set of observables** and **use a flexible control-function or double-ML approach**. In other words, we rely on "selection on observables" (unconfoundedness) rather than finding a credible instrument.

Alternative method (summary):

1. Divide the sample into two folds (A/B).
2. In fold A, estimate a **very flexible** machine-learning model (e.g. a random forest or boosted tree, or deep net, or LASSO) to predict $book_i$ from

$$Z_i \;=\; \left[\ln(\text{rate\_actual}_i),\, X_i,\, \mathbf{D}_{i,t},\, \mathbf{L}_{i,d}\right].$$

   - Write the learned **propensity score** $\widehat{m}(Z_i) \approx \Pr(book_i = 1 \mid Z_i)$.
3. Also in fold A, estimate a second ML model to predict $\ln(\text{rate\_actual}_i)$ from $[\,X_i,\, \mathbf{D}_{i,t},\, \mathbf{L}_{i,d}]$ (omitting **rate_actual** itself). Call the fitted function $\hat{g}(X_i) \approx E[\ln(\text{rate\_actual}_i) \mid X_i, \mathbf{D}_{i,t},\, \mathbf{L}_{i,d}]$.

4. Form a **residualized** outcome

$$\widetilde{Y}_i \;=\; (book_i - \hat{m}(Z_i)), \qquad \widetilde{W}_i \;=\; (\ln(\text{rate\_actual}_i) - \hat{g}(X_i))\,.$$

5. In fold B, run a **linear regression** of $\widetilde{Y}_i$ on $\widetilde{W}_i$; i.e.,

$$\widetilde{Y}_i = \theta \cdot \widetilde{W}_i + \text{small error}.$$

   The OLS slope $\hat{\theta}$ is then our "doubly-robust" estimate of $\partial \Pr(book_i = 1)/\partial \ln(\text{rate\_actual}_i)$.

6. Swap folds (use B to estimate $\hat{m}, \hat{g}$, then regress in A) and average.

This procedure is known as **Double Machine Learning** (DML) for a continuous "treatment" (log-price) and a binary "outcome" (booking). It is effectively a **generalization of control-function / residual-regression** that allows a very large set of controls Z_i (including all XX – XX and fixed effects) while guarding against overfitting bias (via cross-fold estimation).

3.1 Why it can work (pros)

1. **No explicit instrument needed.** We only need the assumption of **selection on observables**:

$$\{book_i(0),\, book_i(1)\} \;\perp\!\!\!\perp\; \ln(\text{rate\_actual}_i) \;\Big|\; X_i,\, \mathbf{D}_{i,t},\, \mathbf{L}_{i,d}.$$

   In plain English: "Once we control for *all* observed features (including the hotel's intrinsic quality, seasonality, days-ahead, weather, etc.), the manager's final price is as good as random (so far as booking probability is concerned)."

2. **Flexible functional form.** We allow arbitrarily nonlinear prediction of both the outcome ($book_i$) and the price ($\ln(\text{rate\_actual}_i)$) in the first stage. That means if, say, there is a complex interaction (e.g. "when there's a holiday within 3 days and tripadvisor index > 80, recommended price jumps nonlinearly"), the ML can pick it up.

3. **Debiasing and cross-folding.** Because we "partial out" predicted booking-prob ($\hat{m}$) and predicted price ($\hat{g}$) and then regress residuals, we orthogonalize the main parameter θ to the first-stage ML errors. By doing this on held-out data (fold B when training on A), we avoid "overfitting bias." This is exactly the "double-machine-learning" recipe from Zheng (2025) slides 8–9.

4. **Asymptotic properties.** Under standard regularity, $\hat{\theta}$ is $\sqrt{n}$-consistent and asymptotically normal, with the same order of variance as an oracle linear-IV that knew the correct control function.

5. **Estimate a marginal "elasticity"** in the same way: $\hat{\theta}$ is "the partial derivative of $E[book_i]$ w.r.t. $\ln(\text{rate\_actual}_i)$ at the center of the data."

3.2 Pros vs. IV approach

Aspect	Instrumental Variables ("2SLS IV")	Double ML ("DML") with Selection-on-Observables
Identification assumption	rate_recommended $\perp u_i \mid X_i$ (i.e. exogenous instrument)	$\{book_i(0), book_i(1)\} \perp \ln(\text{rate\_actual}_i) \mid X_i$. All confounders must be observed.

Aspect	Instrumental Variables ("2SLS IV")	Double ML ("DML") with Selection-on-Observables
Key strength	Allows valid inference even if manager "knows" some latent demand shocks, provided they do not enter recommendation.	No need to find a valid instrument. One simply assumes observables are rich enough.
Key weakness	Must argue/explain why recommended price is "as good as random" after controls; if manager leaks information, the instrument fails.	If there are any **unobserved** confounders (e.g. real-time social-media buzz that manager sees but we do not), estimates will be biased.
Functional flexibility	First stage is usually linear (or LASSO-IV). Second stage can be LPM or IV-logit. Risks mis-specification if dynamics are nonlinear.	Allows any black-box ML (random forest, boosting, neural net) to estimate control functions. More flexible if "huge" X.
Ease of communication	2SLS with log-price as instrument is standard and easy to explain ("we use the software's recommended price as an instrument").	More "black-box": must convince stakeholders that "selection on observables" is credible and that ML-residualization is valid.
Policy simulation	Easy to convert final β_{price} into revenue-maximizing rule (e.g. guess demand curve → Marginal Revenue).	One can similarly map $\hat{\theta}$ into a marginal demand function, but constructing aggregate revenue curves may require extra nonparametric smoothing.
Data requirements	Only needs the one instrument rate_recommended. If that instrument weakens over time, identification can fail.	Needs "comprehensive" X. If we omit any key confounder, DML gives a biased elasticity.

3.3 When to favor one over the other?

- **Use 2SLS IV** when we strongly believe that, after conditioning on X_i, the recommended price truly does not "leak" any additional information to the manager beyond what we observe—i.e. the exclusion restriction is credible. This is often the case if the analytics engine and manager operate at different times and if the recommended price is **never** shown to customers.
- **Use DML/selection-on-observables** when we worry that there are other unobserved bits – say, the manager gets a whispered rumor of a big conference, or sees last-minute booking flows, that are not reflected in X_i. If we have reason to believe that, **once we have all** the XX – XX features, days-ahead, date-FE, we do capture the manager's private information set, then DML is valid.
- **Practical check:** Run both methods. If the 2SLS IV estimate $\hat{\beta}^{\text{IV}}_{\text{price}}$ is close to $\hat{\theta}^{\text{DML}}$, that is a strong sign that (i) the instrument is reasonably valid and (ii) selection-on-observables is not wildly off. If they diverge significantly, then either the instrument is invalid (fail exclusion) or there are unobserved confounders that DML missed.

3.4 implementation and interpretation

```
# 2.2. Create log-price variable
df <- df %>%
  mutate(log_price_actual = log(rate_actual))

# 2.3. Convert date_of_stay and days_ahead to factors
df <- df %>%
```

```r
  mutate(
    date_of_stay_factor = factor(date_of_stay),
    days_ahead_factor   = factor(days_ahead)
  )

# 2.4. Identify XX-covariates (XX2 … XX250) and (if present) V251
xx_vars <- grep("^XX", names(df), value = TRUE)
if ("V251" %in% names(df)) {
  xx_vars <- c(xx_vars, "V251")
}

# 2.5. (Optional) Standardize numeric XX-covariates for better lasso behavior
numeric_xx <- xx_vars %>% keep(~ is.numeric(df[[.x]]))
df[numeric_xx] <- scale(df[numeric_xx])

# 3. Build model matrices for ML-based nuisance fits
# ------------------------------------------------

# 3.1. Z-matrix: regressors for predicting bookings (incl. log_price_actual)
#       Z_i = [ log_price_actual, X_i (XX2-XX250, V251), date-of-stay dummies, days-ahead dummies ]
Z_matrix <- model.matrix(
  ~ log_price_actual +
    . - 1,
  data = df[, c("log_price_actual", xx_vars, "date_of_stay_factor", "days_ahead_factor")]
)
#   ". - 1" means: include all named columns (XX2.., V251, date dummies, days dummies)
#    and also log_price_actual, with no intercept.

# 3.2. X-matrix: regressors for predicting log_price_actual
#       X_i = [ X_i (XX2-XX250, V251), date-of-stay dummies, days-ahead dummies ]
X_matrix <- model.matrix(
  ~ . - 1,
  data = df[, c(xx_vars, "date_of_stay_factor", "days_ahead_factor")]
)
#   No intercept; includes all XX-covariates plus factor-dummies.

# 3.3. Extract outcomes
Y_vec <- df$bookings            # Binary outcome
W_vec <- df$log_price_actual    # Continuous "treatment"

# 4. Split the sample into two folds
# ----------------------------------
set.seed(123)  # for reproducibility
n         <- nrow(df)
df$fold   <- sample(1:2, size = n, replace = TRUE)  # random 1/2 split

# Prepare vectors to hold out-of-sample predictions
m_hat   <- rep(NA, n)  # predicted E[bookings | Z]
g_hat   <- rep(NA, n)  # predicted E[log_price_actual | X]

# 5. Cross-fitted LASSO for nuisance functions
# --------------------------------------------
```

```r
for (k in 1:2) {
  # Define training and test indices
  train_idx <- which(df$fold == k)
  test_idx  <- which(df$fold != k)

  # 5.1. Fit logistic LASSO to predict bookings ~ Z on training fold
  #      (family = "binomial" for binary outcome)
  cv_m <- cv.glmnet(
    x         = Z_matrix[train_idx, ],
    y         = Y_vec[train_idx],
    alpha     = 1,              # LASSO penalty
    family    = "binomial",
    standardize = TRUE
  )
  # Predict out-of-sample booking probabilities on test fold
  m_hat[test_idx] <- predict(
    cv_m,
    newx     = Z_matrix[test_idx, ],
    s        = "lambda.min",
    type     = "response"
  )[, 1]  # convert to numeric vector

  # 5.2. Fit linear LASSO to predict log_price_actual ~ X on training fold
  cv_g <- cv.glmnet(
    x         = X_matrix[train_idx, ],
    y         = W_vec[train_idx],
    alpha     = 1,              # LASSO penalty
    family    = "gaussian",
    standardize = TRUE
  )
  # Predict out-of-sample log_price_actual on test fold
  g_hat[test_idx] <- predict(
    cv_g,
    newx     = X_matrix[test_idx, ],
    s        = "lambda.min"
  )[, 1]
}

# 6. Form residualized outcomes and treatments
# ---------------------------------------------
df <- df %>%
  mutate(
    m_pred    = m_hat,
    g_pred    = g_hat,
    Y_tilde   = bookings - m_pred,
    W_tilde   = log_price_actual - g_pred
  )

# 7. Estimate the DML coefficient   by regressing Y_tilde on W_tilde
# ---------------------------------------------------------------------
# 7.1. Fit a simple OLS without intercept: Y_tilde =   * W_tilde + error
dml_fit <- lm(Y_tilde ~ W_tilde - 1, data = df)
```

```r
# 7.2. Extract the point estimate and (naïve) standard error
theta_hat <- coef(dml_fit)["W_tilde"]
se_theta  <- summary(dml_fit)$coefficients["W_tilde", "Std. Error"]

cat("DML estimate of booking-prob elasticity ( ):", round(theta_hat, 4), "\n")
```

```
## DML estimate of booking-prob elasticity ( ): -0.0282
```

```r
cat("Approximate standard error:", round(se_theta, 4), "\n")
```

```
## Approximate standard error: 0.0139
```

The double-ML procedure produces an estimated of –0.0282 with an approximate standard error of 0.0139. In plain terms, this means that after we flexibly partial out all observed confounders via LASSO (predicting booking probability from price and covariates, and predicting log price from covariates alone), regressing the residualized booking indicator on the residualized log price yields a small negative coefficient: a 1 percent increase in the actual per-night price is associated with about a 0.028 percentage-point drop in the probability of booking. By contrast, our earlier 2SLS IV model gave a positive coefficient of roughly +0.117 on log price, implying that higher actual price appeared to be correlated with higher booking probabilities. The most natural interpretation of that sign reversal is that the IV approach—in practice—did not fully purge the influence of unobserved demand shocks: when demand was especially strong, both recommended and actual prices rose, so the IV picked up the upward-demand effect rather than a true "price-drives-down-bookings" causal channel. The double-ML estimate, on the other hand, assumes that once we have conditioned on the hundreds of XX-covariates plus date-of-stay and days-ahead dummies, there remain no unobserved factors linking price and booking. Under that assumption, the relatively small negative coefficient (–0.028) suggests that price elasticity is low in magnitude—each 1 percent price increase depresses booking probability by about 0.03 points. Because the standard error (0.0139) implies this estimate is roughly two standard errors below zero (p 0.04), we have modest evidence of negative elasticity after accounting for observable heterogeneity. In other words, the double-ML result finds that higher price does indeed reduce bookings, but by far less than the naïve IV sign suggested. This contrast highlights how residual endogeneity in the first model likely turned the coefficient positive, whereas the second model's reliance on a richer "selection-on-observables" assumption produces a small, statistically significant downward-sloping demand effect.

---

4. Heterogeneity in Price Elasticity – Which Dimensions Matter?

Once we have a "baseline" elasticity estimate $\hat{\varepsilon}$ for the **average** customer, the hotelier often wants to know:

> **"Which types of customers (or dates, or rooms) are more price-sensitive, so that I can implement dynamic or segmented pricing to maximize revenue?"**

Below are three key sources of heterogeneity that matter for revenue optimization. In each case, we describe (a) why exploring that heterogeneity can raise revenue, (b) how one would estimate the **conditional elasticity** across that dimension, and (c) what kind of business insight to draw.

4.1 Heterogeneity by Lead-Time ("Days Ahead") – Business Travelers vs. Tourists

- **Why it matters.**
  - Empirical travel research shows that…
    1. **Business travelers** often book 1–3 days ahead, are less price-sensitive (need flexibility), and have low cancellation risk;
    2. **Leisure travelers** book 20–60 days in advance, are more price-sensitive (price-shops), and readily cancel if better deals appear.
  - By understanding that $\varepsilon_{(1\text{–}3 \text{ days})} \approx -0.15$ (inelastic) but $\varepsilon_{(30\text{–}60 \text{ days})} \approx -0.60$ (very elastic), the hotel can charge a **premium** close to the stay date, and offer **discounted "early-bird" rates** far in advance.
- **Estimation approach.**
  1. In the IV or DML framework, introduce **interactions** between $\ln(\text{rate\_actual}_i)$ and a "days-ahead bucket" dummy. Concretely, define

$$\mathbf{L}_i^{(1)} = \begin{cases} 1, & 0 \le \textit{days\_ahead} \le 3, \\ 0, & \text{otherwise}, \end{cases} \quad \mathbf{L}_i^{(2)} = \begin{cases} 1, & 4 \le \textit{days\_ahead} \le 14, \\ 0, & \text{otherwise}, \end{cases} \quad \mathbf{L}_i^{(3)} = 1 \text{ if } 15 \le \textit{days\_ahead},$$

     (or finer bins like 0–1 day, 2–7 days, 8–30 days, 30+ days).
  2. In the second stage (e.g. 2SLS LPM), replace $\beta_{\text{price}} \ln(\widehat{\text{rate\_actual}}_i)$ with

$$\beta_1 \left(\ln(\widehat{\text{rate\_actual}}_i) \cdot \mathbf{L}_i^{(1)}\right) \; + \; \beta_2 \left(\ln(\widehat{\text{rate\_actual}}_i) \cdot \mathbf{L}_i^{(2)}\right) \; + \; \beta_3 \left(\ln(\widehat{\text{rate\_actual}}_i) \cdot \mathbf{L}_i^{(3)}\right).$$

  3. Each $\hat{\beta}_j$ is then the elasticity **conditional** on that lead-time bin.
- **Interpretation & insight.**
  - If $\hat{\beta}_1 \approx -0.12$ (low elasticity for "0–3 days"), it means business travelers are relatively captive; the hotel can charge close to "rack rate" at 2 days ahead.
  - If $\hat{\beta}_3 \approx -0.70$ (high elasticity for "15+ days"), it means leisure travelers shop around; so one might offer "early-bird" promotions 30 days out to lock in occupancy early.
  - **Revenue simulation:** Combine these elasticities with marginal cost to solve

$$\max_{p_1,p_2,p_3} \sum_{j=1}^{3} \Big[\left(\text{expected demand at price } p_j \mid \text{lead-time bin } j\right) \times p_j \Big].$$

    That gives "optimal price per lead-time segment."

4.2 Heterogeneity by Day-of-Week / Seasonality – Weekends vs. Weekdays

- **Why it matters.**
  - Weekend demand (Friday–Sunday stays) often comes from leisure; midweek stays (Monday–Thursday) skew toward business. If weekend demand is significantly more **inelastic** (say, $\varepsilon_{\text{WKND}} = -0.20$) vs. midweek ($\varepsilon_{\text{WD}} = -0.50$), the hotel should set weekend rates 10–15 percent above weekday rates.

– Seasonal holidays (Spring Break, Christmas, conference week) have very low elasticity: when local conventions fill all rooms, raising price by 5 percent hardly dents occupancy.

- **Estimation approach.**

  1. Create dummies $\mathbf{W}_i^{\text{wknd}} = 1$ if "date_of_stay_i" is Fri/Sat/Sun, else 0. Or more granular: $\mathbf{W}_i^{\text{mon}}$, $\mathbf{W}_i^{\text{tue}}, \ldots$.
  2. In the second-stage (2SLS or DML residual regression), interact $\ln(\widehat{\text{rate\_actual}}_i)$ with $\mathbf{W}_i^{\text{wknd}}$ and with its complement:

$$\beta_{\text{WD}} \left(\ln(\widehat{\text{rate\_actual}}_i) \cdot (1 - \mathbf{W}_i^{\text{wknd}})\right) \; + \; \beta_{\text{WKND}} \left(\ln(\widehat{\text{rate\_actual}}_i) \cdot \mathbf{W}_i^{\text{wknd}}\right).$$

  3. Optionally, also interact by "holiday vs. non-holiday": define $\mathbf{H}_i = 1$ if "date_of_stay_i" is a public holiday or known convention date. Then we get a small additional sample of "extremely inelastic" days.

- **Interpretation.**

  – If $\widehat{\beta}_{\text{WD}} = -0.45$ and $\widehat{\beta}_{\text{WKND}} = -0.22$, then weekend demand is far less elastic. Management can thus charge, say, 10 percent more per night for Fri/Sat, and still fill X % of rooms.
  – Conversely, in midweek small price cuts of 5 percent may yield a bigger % lift in bookings (since high elasticity), so that's the "sweet spot" for midweek promotions.

- **Revenue optimization.** Form two "demand curves" $\widehat{D}_{\text{WD}}(p) = \hat{a}_{\text{WD}} \times p^{\hat{\beta}_{\text{WD}}}$, $\widehat{D}_{\text{WKND}}(p) = \hat{a}_{\text{WKND}} \times p^{\hat{\beta}_{\text{WKND}}}$. Then set price p^*_{WD} to maximize $p \times D_{\text{WD}}(p)$ and similarly for weekend.

4.3 Heterogeneity by Room/Customer Segment (Hotel-Side Features)

- **Why it matters.**

  – Not all rooms or customer segments respond equally to price. For instance:

    1. **High-end suites** (XX = suite? = 1) may be served to affluent clients who are "less price-sensitive" (lower elasticity). One can charge a "suite premium" even on weekdays.
    2. **Budget rooms** (XX = standard-queen? = 1) serve more cost-conscious travelers (higher elasticity). Price ↑ there dents occupancy more.
    3. **Crowd-type** (XX = "booking platform=OTA?") might indicate whether someone came via an online travel agency vs. direct web. OTA channels often have **price-driven** customers, so elasticity is higher. "Direct website" customers are more brand-loyal, so elasticity is lower.

- **Estimation approach.**

  1. Let $R_{i,r} = 1$ if query i is for room-type r (e.g. "suite," "king," "double-queen," "budget").
  2. In IV or DML, replace "$\beta_{\text{price}} \ln(\widehat{\text{rate\_actual}}_i)$" by

$$\sum_r \beta_r \left(\ln(\widehat{\text{rate\_actual}}_i) \times R_{i,r}\right).$$

  3. Each $\widehat{\beta}_r$ is elasticity for that segment. If $\widehat{\beta}_{\text{suite}} = -0.18$ vs. $\widehat{\beta}_{\text{budget}} = -0.60$, then indeed "suite customers" are far less price-sensitive.

- **Interpretation & business insight.**

– Knowing $\hat{\beta}_r$ for each room type, the revenue-manager can suggested "gap pricing" between room categories. E.g. if a budget room's elasticity is $3\times$ that of a suite, it may pay to keep "suite rates" quite high and spread out the tier structure.
– We can also explore interactions between channel and room. E.g. "q-channel=OTA & room=budget" has elasticity -0.75, while "direct web & suite" has elasticity -0.10. That suggests aggressive OTA "promos" on budget rooms, but never slash suite rates on direct site.
