Project 3: Team 1

Causal Inference: RD

AUTHOR
Team 1: Muhammad Sawaiz Fatar, Victor Ostolaza,
Vedaant Rath, Arpita Ram Samant, Sam Sheng

1. Introduction

Who We Are

Sentinel Health Analytics is a healthcare consulting firm specializing in infection control policy evaluation for hospital systems.

Our Client

Metropolitan General Health System (MGHS) is a 850-bed hospital network in Chicago serving 45,000 patients annually.

The Problem: High hospital-acquired infection (HAI) rates. HAIs are basically infections that patients acquire after coming to a hospital. These infections were not present at the time of admission.

- Current rate: 4.8 per 1,000 patient-days
- Benchmark: 3.5 per 1,000 patient-days
- Each infection costs $40,000

The Policy

MGHS has invested $8M in isolation infrastructure and has an annual operating cost of $4M. In March 2024, MGHS implemented this policy for controlling HAIs.

- All admitted patients receive an **Infection Risk Score** (0-100) - how likely they are to contract HAI.
- Risk Score **≥ 70** = Isolation ward (mandatory)
- Risk Score **< 70** = Regular ward

> **The Question we have been hired to solve:**
>
> Does mandatory isolation actually reduce hospital acquired infections (HAI)?

The Challenge we expect to face

1. **Why can't we just compare isolated vs. non isolated patients?**

Answer: Sicker patients get higher risk scores and go to isolation. They might have worse outcomes simply because they're sicker, not because isolation doesn't work. So why can' t we

2. **The RD Solution:**

Answer: Compare patients who scored just above 70 vs. just below 70. These patients are nearly identical in health status - the only difference is isolation assignment.

Core Assumptions (Sharp RD)

1. **Deterministic assignment:** crossing 70 fully determines isolation - there is strict compliance
2. **No precise manipulation:** patients/providers can't game scores to land just below 70
3. **Continuity:** aside from isolation, expected infection risk changes smoothly with the score at 70

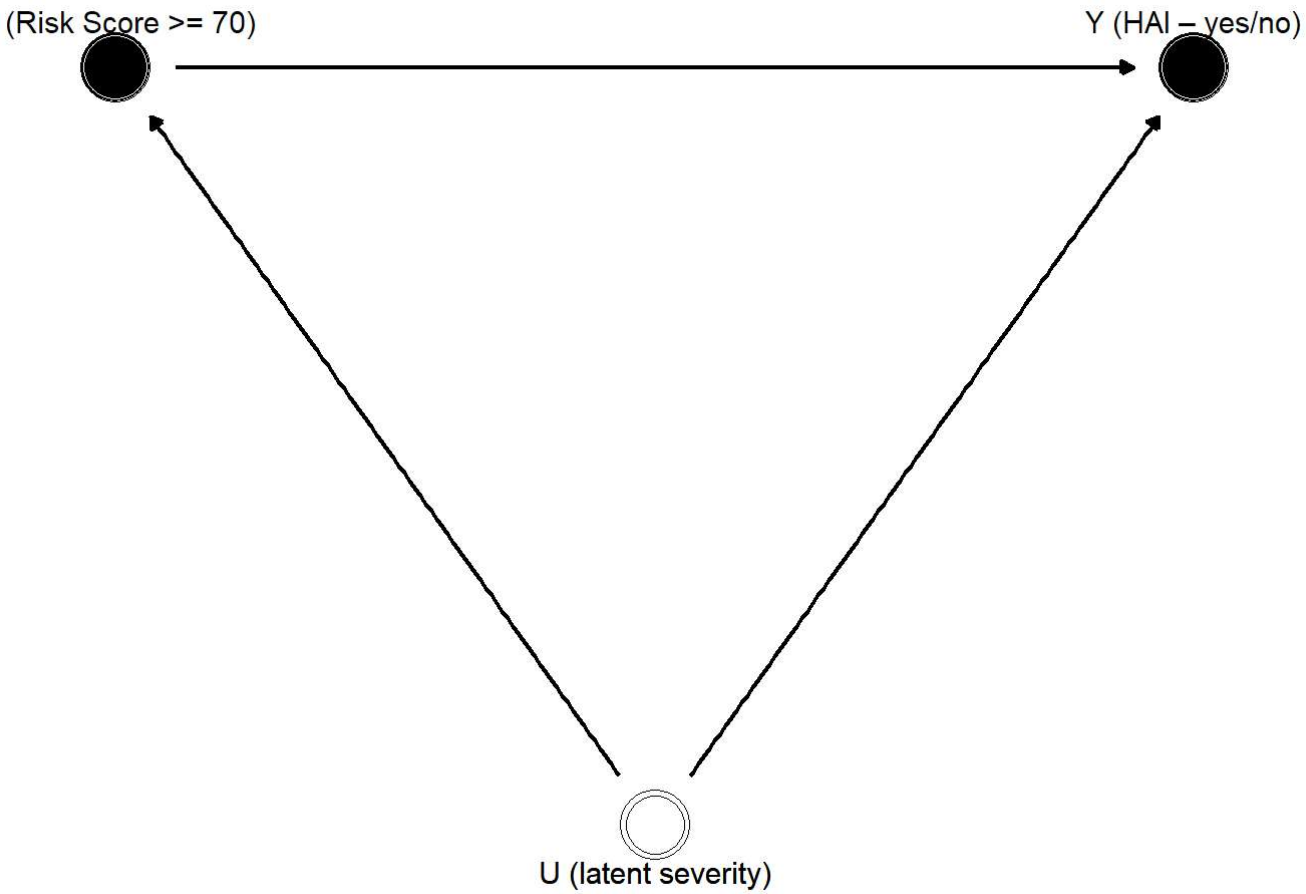
Defining Variables

- **Outcome:** Hospital acquired infection (yes/no)
- **Running Variable:** Infection Risk Score (0-100)
- **Treatment Effect:** Effect of risk score (Isolation ward placement) >= 70 on HAI
- **Cutoff or threshold:** 70
- **Treated:** Patients with score >= 70 (go to isolation)
- **Control:** Patients with score < 70 (go to regular ward)

Simple DAG

```
library(dagitty)
library(ggdag)
library(ggplot2)
library(dplyr)

dag_txt <- 'dag {
  D [pos="0,1"]
  Y [pos="2,1"]
  U [pos="1,0"]
  U -> D
  U -> Y
  D -> Y
}'

g <- dagitty(dag_txt)
tidy_g <- tidy_dagitty(g) %>%
  mutate(
```

```r
    label_text = case_when(
      name == "D" ~ "D (Risk Score >= 70)",
      name == "Y" ~ "Y (HAI - yes/no)",
      name == "U" ~ "U (latent severity)"
    ),

    fill_color = ifelse(name == "U", "white", "black"),

    v_adjust = ifelse(name == "U", 2.5, -1.5)
  )

# Plot
ggplot(tidy_g, aes(x = x, y = y, xend = xend, yend = yend)) +

  geom_dag_edges(edge_color = "black", edge_width = 0.8) +


  geom_dag_node(aes(fill = fill_color), shape = 21, color = "black", size = 12) +


  geom_text(aes(label = label_text, vjust = v_adjust),
            color = "black", fontface = "plain", size = 4) +


  scale_fill_manual(values = c("black" = "black", "white" = "white")) +


  theme_dag() +
  theme(legend.position = "none",
        plot.margin = margin(10, 10, 10, 10))
```



Explaining the DAG and OVB to our Client:

Latent severity is basically "how sick the patient really is" - the stuff that's hard to capture perfectly in one neat variable, so it is essentially 'omitted'

When latent severity is higher (patient is sicker), two things can happen:

1. they're more likely to get put in isolation (D), and
2. they're more likely to pick up an infection (Y) just because they're already vulnerable.

And then **D to Y** is the thing we actually care about: does isolation lower infections or not?

Why we care:
If we just compare **isolated vs not isolated** patients, we're not doing a fair fight. The isolated group usually starts off sicker. So if they have more infections, it might not mean isolation doesn't work, it might just mean they were higher-risk from the beginning.

Data Simulation

```
# Load necessary libraries
library(tidyverse)

# 1. Set Seed for Reproducibility
set.seed(42)
```

```r
# 2. Simulation Parameters
N <- 1000
cutoff <- 70

# 3. Generate Data
# ------------------------------------------------------
# Step A: Generate Latent Severity (U)
# U is the unobserved confounder. Higher U = sicker patient.
# We assume U is normally distributed.
latent_severity <- rnorm(N, mean = 0, sd = 1)

# Step B: Generate Risk Score (X) -> The Running Variable
# X depends on U (sicker patients get higher scores) plus some random noise.
# We scale and shift to get a distribution roughly between 1 and 100.
# Logic: U -> X
X_raw <- 60 + (15 * latent_severity) + rnorm(N, mean = 0, sd = 5)

# Clip to 1-100 and round to integer
risk_score <- pmax(1, pmin(100, round(X_raw)))

# Step C: Assign Treatment (D) -> Sharp RD
# Treatment (Isolation) is assigned strictly if Risk Score >= 70.
# Logic: X -> D (Deterministic)
D <- ifelse(risk_score >= cutoff, 1, 0)

# Step D: Generate Outcome (Y) -> Hospital Acquired Infection (HAI)
# The probability of infection depends on:
# 1. Underlying risk U (U -> Y): Sicker patients have HIGHER baseline risk
# 2. Observed risk X (X -> Y): Optional additional risk from the score itself
# 3. Treatment D (D -> Y): Isolation reduces risk (Negative coef)

# Coefficients tuned for realistic ~5-15% infection rate
b_intercept <- -4.5  # Baseline log-odds (tuned down to keep probabilities realistic)
b_sev       <-  0.8  # Strong effect of latent severity U on outcome
b_risk      <-  0.02 # Small direct effect of X on outcome (smooth trend)
b_treat     <- -1.2  # Isolation DECREASES log-odds (Protective)

# Calculate Probability
# Logic: logits = b0 + b_risk*X + b_sev*U + b_treat*D
logits <- b_intercept + (b_risk * risk_score) + (b_sev * latent_severity) + (b_treat * D)
probs  <- plogis(logits) # Convert log-odds to probability (0 to 1)

# Draw binary outcome (1 = Infection, 0 = No Infection)
HAI <- rbinom(N, size = 1, prob = probs)

# 4. Create Dataframes
# ------------------------------------------------------

# Create ID
patient_id <- 1:N

# df_truth: Includes the unobserved confounder 'latent_severity' and true probabilities
df_truth <- data.frame(
  id = patient_id,
```

```
  latent_severity = latent_severity,
  risk_score = risk_score,
  D = D,
  true_prob = probs,
  HAI = HAI
)

# df_final: The dataset the analyst actually sees (U is hidden)
df_final <- df_truth %>%
  select(id, risk_score, D, HAI)

head(df_final)
```

```
  id risk_score D HAI
1  1         92 1   0
2  2         54 0   0
3  3         70 1   0
4  4         71 1   0
5  5         61 0   0
6  6         55 0   0
```

EDA

Summarize the Outcome

```
summary(df_final$HAI)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.000   0.000   0.000   0.032   0.000   1.000 
```

Summarize the running variable

```
summary(df_final$risk_score)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   2.00   49.75   60.00   59.53   69.25  100.00 
```

Summarize the treatment variable

```
print(table(df_final$D))
```

```
  0   1 
750 250 
```

ATE via DiM

```r
library(dplyr)

# Make sure D is numeric 0/1
df_dim <- df_final %>%
  mutate(D_num = as.integer(as.character(D))) %>%   # works if D is factor "0"/"1"
  mutate(D_num = ifelse(is.na(D_num), as.integer(D), D_num))

# Group means + DiM
dim_table <- df_dim %>%
  group_by(D_num) %>%
  summarise(
    n = n(),
    mean_HAI = mean(HAI),
    sd_HAI = sd(HAI),
    .groups = "drop"
  )

dim_table
```

```
# A tibble: 2 × 4
  D_num     n mean_HAI sd_HAI
  <int> <int>    <dbl>  <dbl>
1     0   750   0.0307  0.173
2     1   250   0.036   0.187
```

```r
ATE_DiM <- dim_table$mean_HAI[dim_table$D_num == 1] - dim_table$mean_HAI[dim_table$D_num == 0]
ATE_DiM
```

```
[1] 0.005333333
```

The raw comparison makes isolation look worse, but that's likely because isolation is used for higher risk patients (meaning we could possibly have selection bias/OVB). This is why we need RD near the cutoff.

What we tell the client: Just by comparing the treated and control groups simply, we actually see that patients in isolation seem to have 0.53% higher chance of HAI. This means there is something obfuscating our results, so you need to hire us to understand these nuances where simple analysis won't work.

EDA 1: Sharpness check: does the cutoff perfectly determine treatment?

```r
cutoff <- 70

df_tmp <- df_final %>%
  mutate(
    D_num = as.integer(as.character(D)),  # works if D is factor "0"/"1"
    D_num = ifelse(is.na(D_num), as.integer(D), D_num),
    above = risk_score >= cutoff
  )
```

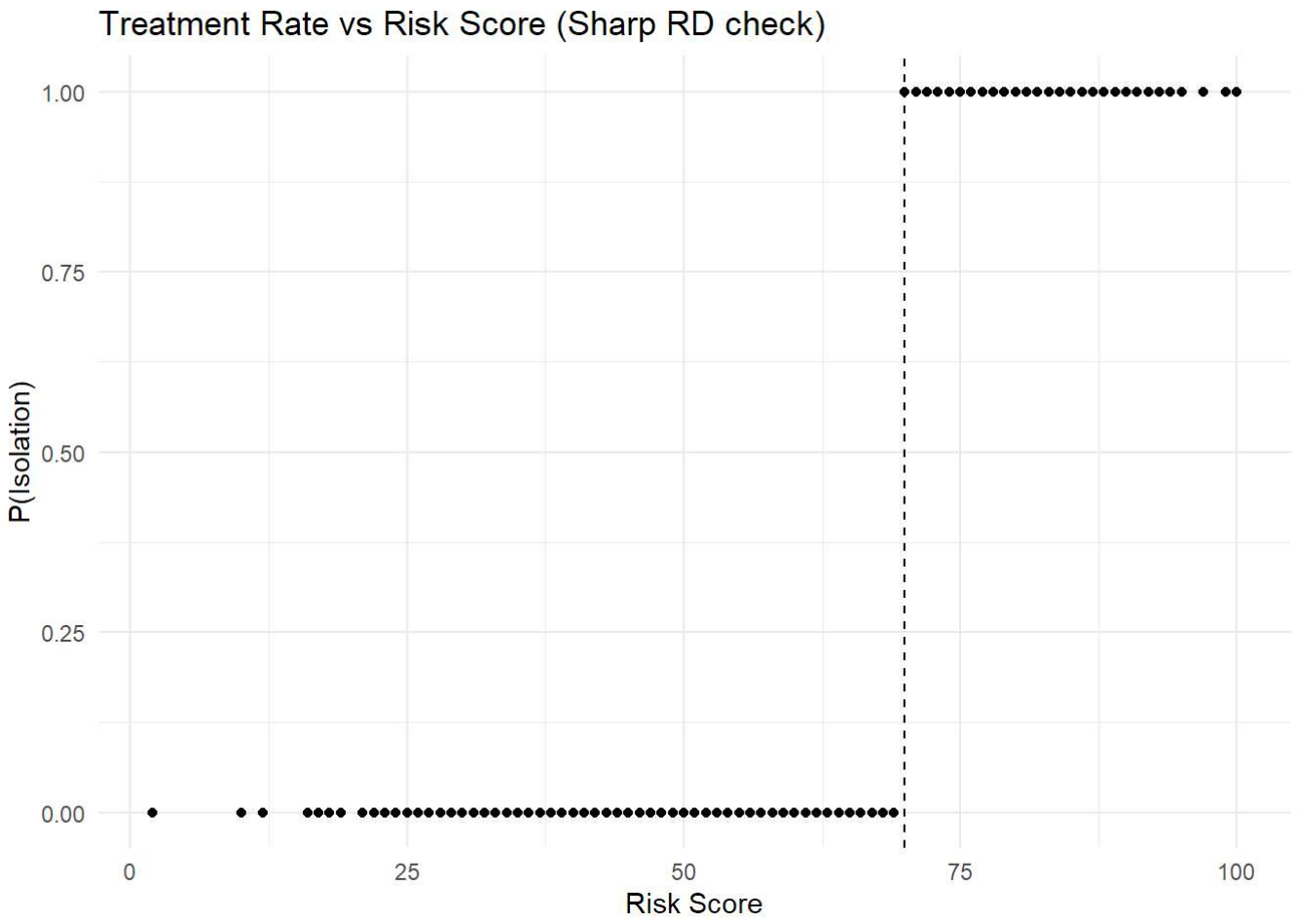
```r
table(Above_Cutoff = df_tmp$above, Treatment = df_tmp$D_num)
```

```
           Treatment
Above_Cutoff   0   1
       FALSE 750   0
       TRUE    0 250
```

Yes, it looks like the cutoff is strict due to the hospital mandate, so a sharp RD would suffice here. No patients above the cutoff aren't in isolation wards.

EDA 2: Plot treatment rate by score

```r
cutoff <- 70

df_tmp <- df_final %>%
  mutate(
    D_num = as.integer(as.character(D)),
    D_num = ifelse(is.na(D_num), as.integer(D), D_num)
  )

treat_rate <- df_tmp %>%
  group_by(risk_score) %>%
  summarise(treat_rate = mean(D_num), n = n(), .groups = "drop")

library(ggplot2)

ggplot(treat_rate, aes(x = risk_score, y = treat_rate)) +
  geom_point() +
  geom_vline(xintercept = cutoff, linetype = "dashed") +
  scale_y_continuous(limits = c(0, 1)) +
  labs(title = "Treatment Rate vs Risk Score (Sharp RD check)",
       x = "Risk Score", y = "P(Isolation)") +
  theme_minimal()
```

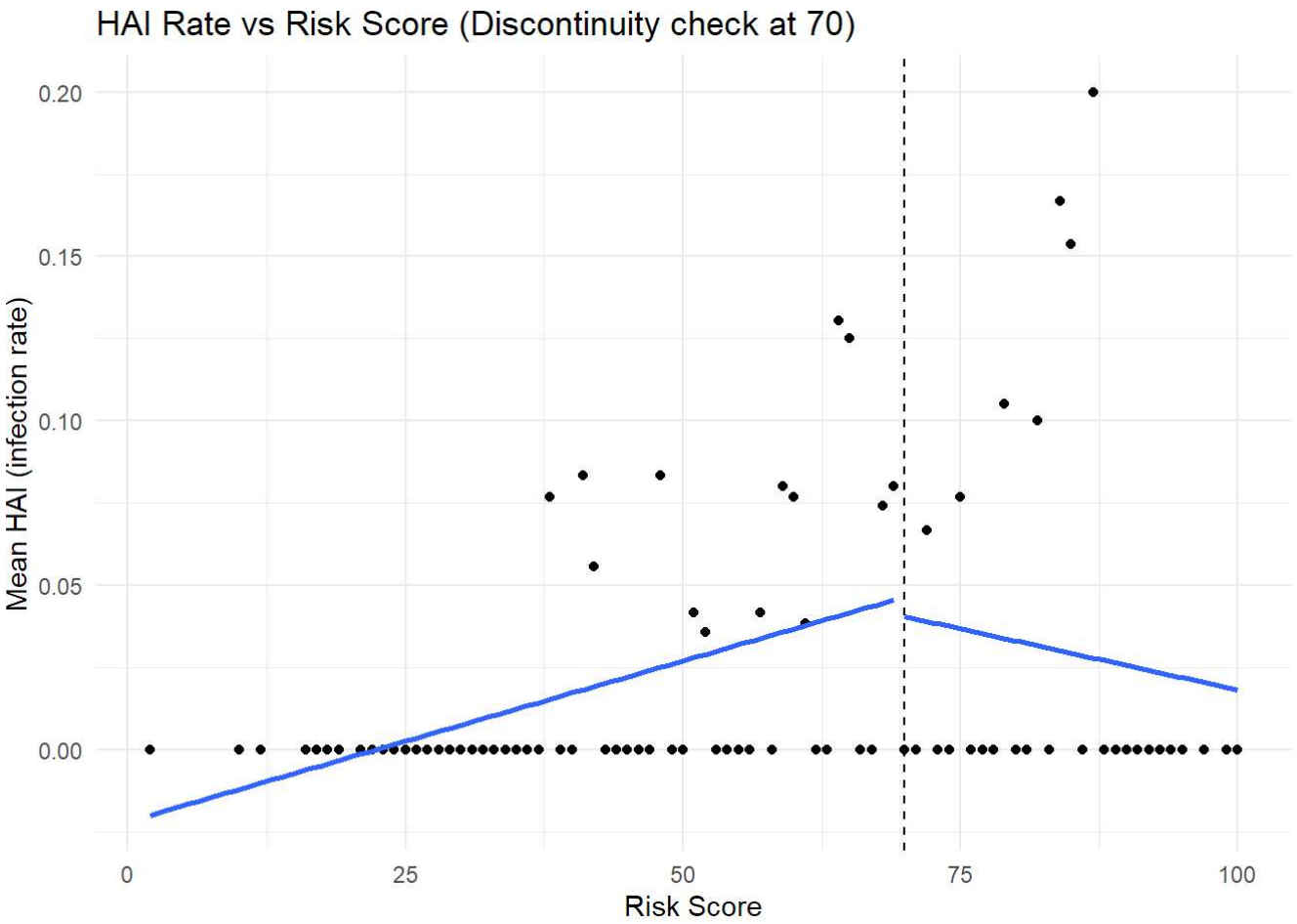


Outcome discontinuity plot: That means perfect compliance with the rule, so this is a Sharp RD design (treatment is a deterministic function of the running variable at the cutoff).

Client explanation: This chart shows that the hospital rule is followed exactly. If a patient's score is below 70, they don't go to isolation. If it's 70 or higher, they always go to isolation. That clean switch at 70 is what makes our cutoff based comparison credible.

EDA 3: HAI rate vs risk score

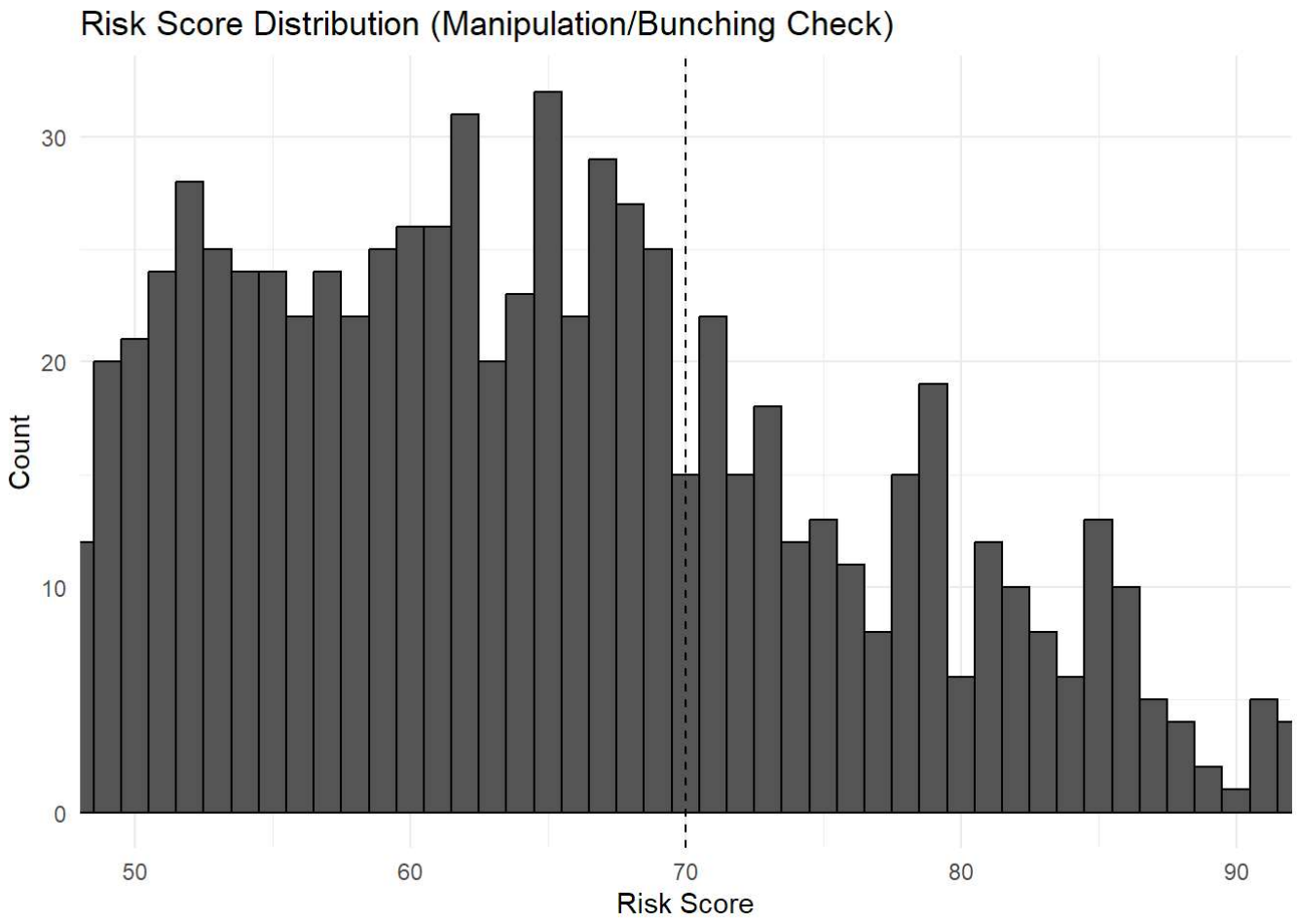
```
cutoff <- 70

outcome_by_score <- df_final %>%
  group_by(risk_score) %>%
  summarise(mean_HAI = mean(HAI), n = n(), .groups = "drop")

ggplot(outcome_by_score, aes(x = risk_score, y = mean_HAI)) +
  geom_point() +
  geom_vline(xintercept = cutoff, linetype = "dashed") +
  geom_smooth(data = subset(outcome_by_score, risk_score < cutoff),
              method = "lm", se = FALSE) +
  geom_smooth(data = subset(outcome_by_score, risk_score >= cutoff),
              method = "lm", se = FALSE) +
  labs(title = "HAI Rate vs Risk Score (Discontinuity check at 70)",
       x = "Risk Score", y = "Mean HAI (infection rate)") +
  theme_minimal()
```



Client /non technical explanation: As patients risk scores go up, infections tend to become more common. That's expected. The key thing here is what happens at score 70: right when the isolation rule kicks in, the infection rate drops compared to what we would have expected from the below-70 trend. Patients just above 70 look like they have fewer infections than very similar patients just below 70, which is a sign the isolation policy may be helping near the cutoff.

EDA 4: Manipulation check: histogram of running variable

```
cutoff <- 70

ggplot(df_final, aes(x = risk_score)) +
  geom_histogram(binwidth = 1, color = "black") +
  geom_vline(xintercept = cutoff, linetype = "dashed") +
  coord_cartesian(xlim = c(50, 90)) +  # zoom around the cutoff
  labs(title = "Risk Score Distribution (Manipulation/Bunching Check)",
       x = "Risk Score", y = "Count") +
  theme_minimal()
```



Client explanation: This chart checks whether people seem to be gaming the score to avoid isolation. If lots of patients were being nudged to score 69 instead of 70, we'd see a big pile up right below the threshold line. We don't see that pattern here, so the cutoff looks fair to use for a clean comparison. Essentially, the hospital isn't manipulating the risk scores to benefit themselves.

EDA 5: Bandwidth (caliper) table: show local comparisons for ±3, ±5, ±10

```
cutoff <- 70
bandwidths <- c(3, 5, 10)

df_tmp <- df_final %>%
  mutate(
    D_num = as.integer(as.character(D)),
    D_num = ifelse(is.na(D_num), as.integer(D), D_num)
  )

bw_table <- purrr::map_dfr(bandwidths, function(h) {
  df_bw <- df_tmp %>% filter(abs(risk_score - cutoff) <= h)

  means <- df_bw %>%
    group_by(D_num) %>%
    summarise(
      n = n(),
      mean_HAI = mean(HAI),
```

```
      mean_score = mean(risk_score),
      .groups = "drop"
    )

  m1 <- means$mean_HAI[means$D_num == 1]
  m0 <- means$mean_HAI[means$D_num == 0]
  diff <- ifelse(length(m1)==1 && length(m0)==1, m1 - m0, NA_real_)

  tibble(
    bandwidth = h,
    n_total = nrow(df_bw),
    mean_HAI_D1 = ifelse(length(m1)==1, m1, NA_real_),
    mean_HAI_D0 = ifelse(length(m0)==1, m0, NA_real_),
    diff_D1_minus_D0 = diff
  )
})

bw_table
```

```
# A tibble: 3 × 5
  bandwidth n_total mean_HAI_D1 mean_HAI_D0 diff_D1_minus_D0
      <dbl>   <int>       <dbl>       <dbl>            <dbl>
1         3     151      0.0143      0.0494          -0.0351
2         5     230      0.0211      0.0593          -0.0382
3        10     415      0.0260      0.0536          -0.0277
```

We zoomed in around the cutoff and compared very similar patients on either side of score 70. No matter how tight or wide the window is, patients just above 70 have fewer infections than patients just below 70.
The drop is roughly 3 percentage points, which means the isolation policy is helping near the threshold.

Regression Discontinuity

Motivation: We saw that a naive DiM tells a different story for what is expected, meaning that we have OVB. We can control for these risk related omitted confounders by including the running variable, risk score.

```
sharp_RD1 <- lm(HAI ~ D + risk_score, data = df_final)
summary(sharp_RD1)
```

```
Call:
lm(formula = HAI ~ D + risk_score, data = df_final)

Residuals:
     Min       1Q   Median       3Q      Max
-0.06293 -0.04248 -0.03206 -0.02384  0.98879

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.0383085  0.0280996  -1.363   0.1731
D           -0.0290683  0.0187296  -1.552   0.1210
```

```
risk_score   0.0013031  0.0005168   2.521   0.0118 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1757 on 997 degrees of freedom
Multiple R-squared:  0.006507,  Adjusted R-squared:  0.004514
F-statistic: 3.265 on 2 and 997 DF,  p-value: 0.03861
```

Technical interpretation:
the coefficient on D = −0.0291 means that, holding risk score constant, being in treatment (Isolation) is associated with about a 2.9 percentage point lower probability of HAI. But it's not statistically significant (p = 0.121), so we can't confidently claim a treatment jump. The coefficient on risk_score is positive and statistically significant (p = 0.0118), meaning a 1 point increase in risk score is associated with about a 0.13 percentage point higher HAI probability. The R^2 is tiny so this model explains almost none of the variation in HAI.

Client interpretation:
After accounting for how infection risk changes as the risk score rises, the people in isolation have about a 3% lower infection rate, but this is not statistically signifcant and this reduction could just be random, and not the rule. What we can say confidently is that higher risk scores are linked to higher infection risk, even if the relationship is small. This model also takes all patients into account so we aren't looking at similar patients yet.

Nonlinearity check

We include a quadratic term - the risk_score squared.

```r
df_final<-df_final%>%
  mutate(risk_sq=risk_score^2)
sharp.RD.quad<-lm(HAI~D+risk_score+risk_sq, data = df_final)
summary(sharp.RD.quad)
```

```
Call:
lm(formula = HAI ~ D + risk_score + risk_sq, data = df_final)

Residuals:
     Min       1Q   Median       3Q      Max
-0.06667 -0.04238 -0.03172 -0.02267  0.98861

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.176e-02  5.735e-02  -0.554    0.580
D           -3.070e-02  2.249e-02  -1.365    0.172
risk_score   1.040e-03  2.075e-03   0.501    0.616
risk_sq      2.516e-06  1.919e-05   0.131    0.896

Residual standard error: 0.1758 on 996 degrees of freedom
Multiple R-squared:  0.006524,  Adjusted R-squared:  0.003531
F-statistic:  2.18 on 3 and 996 DF,  p-value: 0.08879
```

- It appears that the information contained within the running variable is now being essentially split among these two covariates. just because the p values are high, we cannot discard these coefficients yet.
- **Client interpretation:** We tested whether infection risk rises with the score in a curved (non-linear) way, rather than a straight line. The data doesn't show meaningful curvature. Adding the squared term doesn't improve the explanation.

Joint hypothesis test to check is running variable and its squared term are jointly important.

```
library(car)
linearHypothesis(sharp.RD.quad,c("risk_score=0","risk_sq=0"),test="F")
```

```
Linear hypothesis test:
risk_score = 0
risk_sq = 0

Model 1: restricted model
Model 2: HAI ~ D + risk_score + risk_sq

  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1    998 30.971
2    996 30.774  2   0.19675 3.1838 0.04185 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- **H0:** `risk_score` and `risk_sq` are jointly unimportant (both coefficients are 0)
- **Ha:** At least one of `risk_score` or `risk_sq` contains information (at least one coefficient is not 0)
- **Result:** The joint F-test gives **p = 0.0419**, which is below 0.05.
- **Conclusion:** We **reject the null**, so the running variable and its square are **jointly important** for explaining HAI.

Differing Marginal Effects

Recenter the running variable around the threshold.

```
df_final <- df_final %>%
  mutate(risk_center = risk_score - 70)
```

Running an interaction regression

```
sharp.RD.interact <- lm(HAI ~ D + risk_center + D * risk_center, data = df_final)
summary(sharp.RD.interact)
```

```
Call:
lm(formula = HAI ~ D + risk_center + D * risk_center, data = df_final)

Residuals:
     Min       1Q   Median       3Q      Max
-0.05684 -0.04172 -0.03264 -0.02404  0.98939

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.0535949  0.0113901   4.705 2.89e-06 ***
D             -0.0270065  0.0211420  -1.277    0.202
risk_center    0.0013433  0.0005513   2.437    0.015 *
D:risk_center -0.0003348  0.0015899  -0.211    0.833
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1758 on 996 degrees of freedom
Multiple R-squared:  0.006551,  Adjusted R-squared:  0.003558
F-statistic: 2.189 on 3 and 996 DF,  p-value: 0.08774
```

How do we interpret the interaction effect? How does this impact your interpretation of the ATE?

- The interaction term (`D:risk_center`) tests whether the marginal effect of the risk score on HAI differs above vs below the cutoff (70).
- The interaction effect is not significant so there's no evidence the slope changes across the cutoff.
- The RD ATE at the cutoff is the coefficient on D = −0.027 (about a 2.7 percentage point drop in HAI at the threshold), and it's not statistically significant (p = 0.202).
- since the interaction isn't doing much and the ATE story doesn't change, it's reasonable to stick with the simpler linear RD (keep `D + risk_center`).

```r
c <- 70
bandwidths <- c(3, 5, 10, 15)

df0 <- df_final %>%
  mutate(score_distance_from_cutoff = risk_score - c)

run_rd_for_bandwidth <- function(bw){
  df_bw <- df0 %>% filter(abs(score_distance_from_cutoff) <= bw)
  m <- lm(HAI ~ D + score_distance_from_cutoff, data = df_bw)

  tibble(
    bandwidth_points = bw,
    sample_size = nrow(df_bw),
    estimated_jump_at_cutoff = coef(m)["D"],
    standard_error = summary(m)$coefficients["D", "Std. Error"],
    p_value = summary(m)$coefficients["D", "Pr(>|t|)"]
  )
}
```

```
rd_bandwidth_results <- purrr::map_dfr(bandwidths, run_rd_for_bandwidth)
rd_bandwidth_results
```

```
# A tibble: 4 × 5
  bandwidth_points sample_size estimated_jump_at_cutoff standard_error p_value
             <dbl>       <int>                    <dbl>          <dbl>   <dbl>
1                3         151                   -0.105         0.0617  0.0919
2                5         230                   -0.0475        0.0549  0.388
3               10         415                   -0.0572        0.0393  0.147
4               15         581                   -0.0738        0.0325  0.0237
```

Overall conclusion:

- The sign is stable (always negative), which is good for the story.
- The precision varies by bandwidth; only the widest caliper (±15) gives strong statistical evidence.
- So we should report the ATE across multiple calipers and say the effect appears consistently negative, but significance depends on how wide the window is.

```
c <- 70
df_final$risk_center <- df_final$risk_score - c

rd_int_3  <- lm(HAI ~ D + risk_center + D:risk_center, data=df_final, subset=abs(risk_center)<
rd_int_5  <- lm(HAI ~ D + risk_center + D:risk_center, data=df_final, subset=abs(risk_center)<
rd_int_10 <- lm(HAI ~ D + risk_center + D:risk_center, data=df_final, subset=abs(risk_center)<
rd_int_15 <- lm(HAI ~ D + risk_center + D:risk_center, data=df_final, subset=abs(risk_center)<

summary(rd_int_3); summary(rd_int_5); summary(rd_int_10); summary(rd_int_15)
```

```
Call:
lm(formula = HAI ~ D + risk_center + D:risk_center, data = df_final,
    subset = abs(risk_center) <= 3)

Residuals:
     Min       1Q   Median       3Q      Max
-0.09224 -0.05140 -0.01711 -0.01056  0.98289

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.13309    0.05374   2.477   0.0144 *
D             -0.12761    0.06497  -1.964   0.0514 .
risk_center    0.04084    0.02437   1.676   0.0958 .
D:risk_center -0.03503    0.03125  -1.121   0.2641
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1787 on 147 degrees of freedom
Multiple R-squared:  0.02872,   Adjusted R-squared:  0.008896
F-statistic: 1.449 on 3 and 147 DF,  p-value: 0.231
```

```
Call:
lm(formula = HAI ~ D + risk_center + D:risk_center, data = df_final,
    subset = abs(risk_center) <= 5)

Residuals:
     Min       1Q   Median       3Q      Max
-0.06952 -0.05891 -0.04829 -0.01820  0.98180

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.042987   0.041631   1.033    0.303
D             -0.043439   0.055148  -0.788    0.432
risk_center   -0.005306   0.012302  -0.431    0.667
D:risk_center  0.014635   0.017738   0.825    0.410

Residual standard error: 0.2045 on 226 degrees of freedom
Multiple R-squared:  0.01165,   Adjusted R-squared:  -0.001466
F-statistic: 0.8883 on 3 and 226 DF,  p-value: 0.4479



Call:
lm(formula = HAI ~ D + risk_center + D:risk_center, data = df_final,
    subset = abs(risk_center) <= 10)

Residuals:
     Min       1Q   Median       3Q      Max
-0.06178 -0.05453 -0.04729 -0.02829  0.98541

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.063586   0.027198   2.338   0.0199 *
D             -0.058132   0.039449  -1.474   0.1414
risk_center    0.001810   0.004384   0.413   0.6799
D:risk_center  0.002757   0.006803   0.405   0.6855
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.204 on 411 degrees of freedom
Multiple R-squared:  0.006581,  Adjusted R-squared:  -0.0006707
F-statistic: 0.9075 on 3 and 411 DF,  p-value: 0.4374



Call:
lm(formula = HAI ~ D + risk_center + D:risk_center, data = df_final,
    subset = abs(risk_center) <= 15)

Residuals:
     Min       1Q   Median       3Q      Max
-0.09566 -0.05861 -0.04073 -0.02284  0.99075

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.072915   0.021655   3.367  0.00081 ***
```

```
D             -0.076957   0.032805  -2.346  0.01932 *
risk_center    0.003577   0.002431   1.472  0.14168
D:risk_center  0.003070   0.003924   0.782  0.43425
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2024 on 577 degrees of freedom
Multiple R-squared:  0.01185,   Adjusted R-squared:  0.006717
F-statistic: 2.307 on 3 and 577 DF,  p-value: 0.07559
```

The estimated RD effect of treatment at the cutoff is consistently negative (isolation reduces HAI right at the threshold), but statistical strength depends on bandwidth. the effect is clearest at ±15 and borderline at ±3. Allowing different slopes on each side does not change the ATE and the interaction is unimportant, **so the simpler model is preferred.**

> ## Findings summary
>
> We looked at patients whose risk scores were very close to the cutoff of 70, where the isolation rule turns on. Patients just above 70 (who get isolated) can be fairly compared to patients just below 70 (who do not), because their risk levels are very similar. In these near-cutoff comparisons, isolation is linked to a lower chance of hospital acquired infection (HAI).
>
> Our best estimate is that isolation reduces HAI by about **7–8 percentage points** for patients right at the cutoff (ATE ≈ **−0.077**). When we use different windows around the cutoff, the estimated reduction is still negative and usually falls between **4 and 13 percentage points**. In simple terms: for patients on the borderline (ie near the threshold set by the hospital) isolation appears to lower infections.
>
> For MGHS the key takeaway is simple: the isolation rule looks like it's doing what you paid for. reducing hospital-acquired infections for borderline patients right around the cutoff score of 70. When we compare patients who are nearly identical in risk isolation is linked to about a 7–8 percentage point drop in HAI risk. Put plainly: for every 100 patients near the cutoff, isolation prevents about 7–8 infections.
>
> If we talk finances, each avoided infection is about $40,000, so that's roughly $280k–$320k saved per 100 borderline patients. To translate this into an annual ROI against the $4M per year operating cost, the quick math is: annual savings = (# of borderline patients within 15 risk score) × 0.077 × $40,000. Our recommendation is to keep the policy in place and next count how many admissions fall near the cutoff each year so MGHS can directly compare savings to the operating cost.

Fuzzy RD: A new, smaller hospital comes along to hire us

A Different Hospital System

Community Health Network is a 300 bed hospital system serving rural areas outside Chicago with 25,000 patients annually.

The Challenge

CHN faces resource constraints: - Only 20 isolation beds available - Physician staffing varies by shift - Score more than 70 recommends isolation but doesn't guarantee it

The Policy at CHN

- Risk Score >= 70: Isolation **recommended** (clinical judgment applies)
- Risk Score <70: Regular ward (unless physician deems necessary)

The fuzzy part: - High-risk patients may not get isolated (no beds, staffing). Low risk patients (<70) sometimes get isolated (symptoms, physician concern)

The Question our client hires us to answer

> **Is the isolation guideline still effective in preventing HAI (hospital acquired infections) when it's not strictly enforced?**

The Omitted Variable: Hospital Crowding

We suspect there's a confounding variable that affects BOTH isolation decisions AND infection outcomes:

Hospital Crowding (U) – unobserved in our data

When the hospital is crowded,

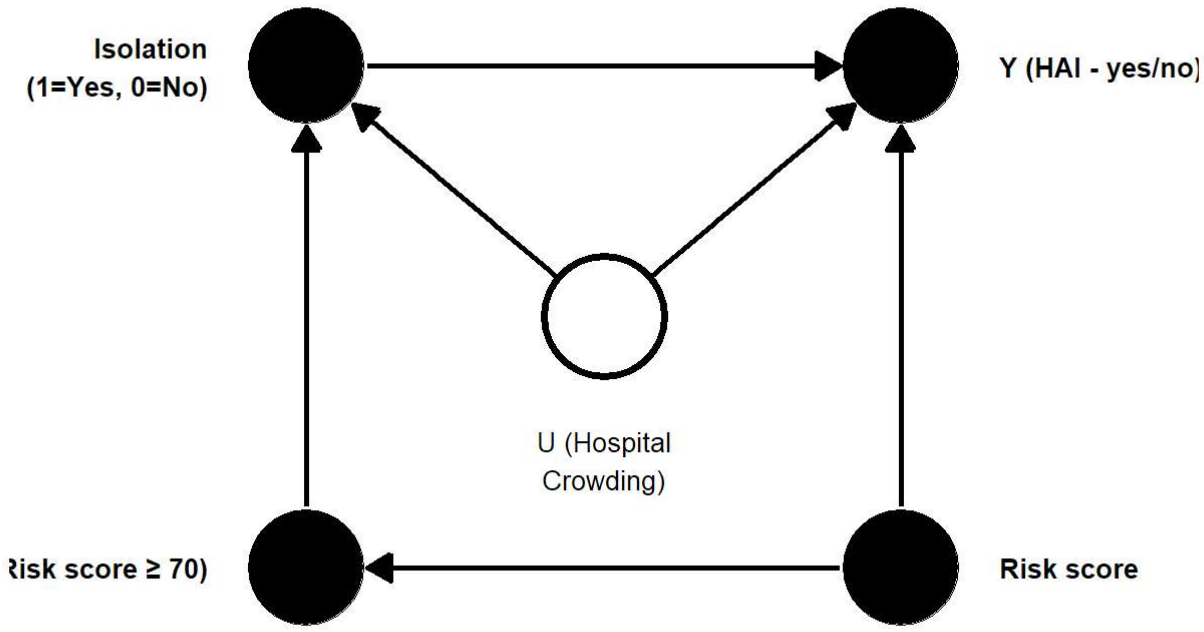
1. **Effect on Treatment (D):**
    - Isolation ward is full, can't isolate patients even if score >= 70
    - More pressure on physicians: less likely to follow guidelines strictly
    - **Result:** High risk patients end up on regular wards
2. **Effect on Outcome (Y - HAI):**

```
-   Overcrowded wards = more patient to patient contact

-   Overworked staff

-   Longer wait times - sicker patients in same space

-   **Result:** Higher infection risk regardless of isolation

## Drawing The DAG
```

```
library(ggdag)
library(ggplot2)
library(dplyr)

coords <- tribble(
  ~name,        ~x,  ~y,
```

```
  "Isolation",   1,   4,
  "Y",           4,   4,
  "U",         2.5, 2.5,
  "D",           1,   1,
  "risk_score",  4,   1
)

fuzzy_dag <- dagify(
  Y ~ Isolation + U + risk_score,
  Isolation ~ D + U,
  D ~ risk_score,
  latent = "U",
  coords = coords
)

# Get the tidy dag data
tidy_dag <- tidy_dagitty(fuzzy_dag)

# Plot with manual annotations
ggplot(tidy_dag, aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_dag_edges(
    arrow_directed = grid::arrow(
      length = grid::unit(10, "pt"),
      type = "closed"
    ),
    edge_width = 1
  ) +
  # Filled points for observed variables
  geom_dag_point(
    data = filter(tidy_dag, name != "U"),
    size = 20,
    color = "black"
  ) +
  # Hollow point for unobserved U
  geom_dag_point(
    data = filter(tidy_dag, name == "U"),
    size = 20,
    shape = 21,        # Use shape 21 for circle with border
    color = "black",   # Border color
    fill = "white",    # Inside color
    stroke = 2         # Border thickness
  ) +
  # Labels positioned OUTSIDE the nodes
  annotate("text", x = 0.5, y = 4,
           label = "Isolation\n(1=Yes, 0=No)",
           size = 3.5, fontface = "bold", hjust = 1) +
  annotate("text", x = 4.5, y = 4,
           label = "Y (HAI - yes/no)",
           size = 3.5, fontface = "bold", hjust = 0) +
  annotate("text", x = 2.5, y = 2,
           label = "U (Hospital\nCrowding)",
           size = 3.5, vjust = 1.5) +
  annotate("text", x = 0.5, y = 1,
           label = "D (Risk score ≥ 70)",
```

```
         size = 3.5, fontface = "bold", hjust = 1) +
annotate("text", x = 4.5, y = 1,
         label = "Risk score",
         size = 3.5, fontface = "bold", hjust = 0) +
theme_dag_blank() +
coord_cartesian(xlim = c(0, 5), ylim = c(0.5, 4.5)) +
theme(plot.margin = margin(30, 30, 30, 30))
```



Defining Variables

- **Outcome (Y):** Hospital acquired infection (HAI) - binary yes/no
- **Running Variable:** Infection Risk Score (0-100, continuous)
- **Cutoff/Threshold:** Risk Score = 70
- **Treatment:** Isolation ward placement (1 = isolated, 0 = regular ward)
- **Actual Treatment:** Isolation status
- **Treated Group:** Patients who receive isolation
- **Control Group:** Patients in regular wards
- **U (Unobserved):** Hospital crowding/resource constraints

- **OVB Problem:** U affects both isolation likelihood AND infection risk
- **Instrument (Z):** Threshold dummy = 1 if Risk Score more than 70, 0 otherwise
    - **Relevance:** The threshold dummy (Risk Score >= 70) is strongly related to actual isolation
    - **Independence/Exogeneity:** The threshold itself is not related to hospital crowding
    - **Exclusion:** The threshold dummy affects HAI only through isolation, not directly

Explaining Fuzzy RD Methodology to the Client:

The Problem:

If you just compare isolated vs. non-isolated patients, isolated ones look worse. Why? Sicker patients get isolated. It's like saying "hospitals are dangerous because people die there more than at home."

The Hidden Issue:

When the hospital is crowded:

- Fewer patients get isolated (no beds)
- More infections spread (crowded conditions)

Crowding affects BOTH isolation decisions AND infection rates, making isolation look ineffective.

Our Solution:

Compare patients **right around the cutoff of 70**:

- Patient with score 69 = usually not isolated
- Patient with score 71 - usually isolated

These patients are nearly identical. The only difference? One crossed the threshold. That's a fair comparison.

Bottom Line: We use the 70 point threshold as a NUDGE that pushes patients toward isolation. Even though the rule isn't perfectly followed, comparing patients just above vs. below 70 gives us a fair estimate of isolation's true effect.

Data Simulation

```
# ================================================================================
# FUZZY RD DATA SIMULATION
# ================================================================================
# Load libraries
library(tidyverse)
# Set seed for reproducibility
set.seed(42)
# Sample size
```

```r
N <- 1000
cutoff <- 70
# ============================================================================
# VARIABLE DEFINITIONS:
# ============================================================================
# 1. Basic identifiers
#    patient_id = Unique ID (1 to N)
# 2. Unobserved confounder (creates OVB)
#    U = Hospital crowding / patient frailty (LATENT)
# 3. Running variable
#    risk_score = Infection Risk Score (0-100, continuous)
# 4. Instrument
#    D_threshold = Threshold dummy: 1 if risk_score >= 70, 0 otherwise
# 5. Fuzzy factors (creates imperfect compliance)
#    Other = Bed availability, staffing, physician judgment
# 6. Treatment (FUZZY - key variable!)
#    Isolation = Actual isolation: 1 = isolated, 0 = regular ward
# 7. Outcome
#    HAI = Hospital-acquired infection: 1 = yes, 0 = no
# ============================================================================
# Step 1: Generate U (Hospital crowding / patient frailty)
# Higher U = sicker patient / more crowded conditions
U <- rnorm(N, mean = 0, sd = 1)
# Step 2: Generate Risk Score (depends on U)
# Sicker patients (higher U) get higher risk scores
risk_score <- 60 + (10 * U) + rnorm(N, mean = 0, sd = 8)
risk_score <- pmax(1, pmin(100, round(risk_score)))  # Clip to 1-100
# Step 3: Create Threshold Dummy (Instrument)
# The instrument - deterministic cutoff
D_threshold <- ifelse(risk_score >= cutoff, 1, 0)
# Step 4: Generate "Other" Fuzzy Factors
# Other factors: bed availability, staffing, clinical judgment
# Independent of U (that's crucial for IV validity!)
Other <- rnorm(N, mean = 0, sd = 1.5)
# Step 5: FUZZY Treatment Assignment
# Isolation depends on BOTH threshold AND other factors
# This is what makes it FUZZY!
# Probability of isolation = f(D_threshold, Other, U)
prob_isolation <- plogis(
  -3 +                    # Base (low isolation rate when below threshold)
  5 * D_threshold +       # STRONG effect of threshold (relevance!)
  0.7 * Other +           # Other factors (fuzziness!)
  0.3 * U                 # U affects isolation too (creates OVB!)
)
# Draw actual treatment (binary)
Isolation <- rbinom(N, size = 1, prob = prob_isolation)
# Step 6: Generate Outcome (HAI with OVB)
# HAI depends on:
# 1. U (confounder - creates OVB!)
# 2. Isolation (TRUE treatment effect)
# 3. Risk score (smooth relationship)
prob_HAI <- plogis(
  -4.5 +                    # Baseline low infection rate
  0.02 * risk_score +       # Higher score = higher risk
```

```
  0.8 * U +                  # U affects outcome (OVB!)
  -1.2 * Isolation           # Isolation REDUCES infections (true effect)
)
HAI <- rbinom(N, size = 1, prob = prob_HAI)
# ============================================================================
# CREATE FINAL DATASET (Analyst's View - U and Other are unobserved)
# ============================================================================
patient_id <- 1:N
df_final <- data.frame(
  id = patient_id,
  risk_score = risk_score,
  Dummy_threshold = D_threshold,
  Isolation_treatment = Isolation,
  HAI = HAI
)
# View the data
head(df_final)
```

```
  id risk_score Dummy_threshold Isolation_treatment HAI
1  1         92               1                   1   0
2  2         59               0                   0   0
3  3         71               1                   1   0
4  4         69               0                   0   0
5  5         56               0                   0   0
6  6         54               0                   0   0
```

```
summary(df_final)
```

```
       id           risk_score     Dummy_threshold Isolation_treatment
 Min.   :   1.0   Min.   : 14.00   Min.   :0.00    Min.   :0.000
 1st Qu.: 250.8   1st Qu.: 51.00   1st Qu.:0.00    1st Qu.:0.000
 Median : 500.5   Median : 60.00   Median :0.00    Median :0.000
 Mean   : 500.5   Mean   : 59.69   Mean   :0.21    Mean   :0.235
 3rd Qu.: 750.2   3rd Qu.: 68.00   3rd Qu.:0.00    3rd Qu.:0.000
 Max.   :1000.0   Max.   :100.00   Max.   :1.00    Max.   :1.000
      HAI
 Min.   :0.000
 1st Qu.:0.000
 Median :0.000
 Mean   :0.037
 3rd Qu.:0.000
 Max.   :1.000
```

```
# Check fuzziness: compliance rates
cat("\n=== COMPLIANCE CHECK ===\n")
```

```
=== COMPLIANCE CHECK ===
```

```
cat("Below threshold (score < 70): Isolation rate =",
    round(mean(df_final$Isolation_treatment[df_final$Dummy_threshold == 0]) * 100, 1), "%\n")
```

```
Below threshold (score < 70): Isolation rate = 6.8 %
```

```
cat("Above threshold (score >= 70): Isolation rate =",
    round(mean(df_final$Isolation_treatment[df_final$Dummy_threshold == 1]) * 100, 1), "%\n")
```

```
Above threshold (score >= 70): Isolation rate = 86.2 %
```

EDA

EDA 1: Outcome Summary

```
# Overall HAI rate
overall_hai <- mean(df_final$HAI)

cat("=== HOSPITAL-ACQUIRED INFECTION (HAI) RATES ===\n")
```

```
=== HOSPITAL-ACQUIRED INFECTION (HAI) RATES ===
```

```
cat("Overall HAI rate:", round(overall_hai * 100, 2), "%\n")
```

```
Overall HAI rate: 3.7 %
```

```
cat("Total infections:", sum(df_final$HAI), "out of", nrow(df_final), "patients\n\n")
```

```
Total infections: 37 out of 1000 patients
```

Overall, about 37 out of 1,000 patients get hospital infections. That's our baseline, we have now see if isolation changes this.

EDA 2: Treatment Distribution

```
cat("=== ISOLATION TREATMENT ===\n")
```

```
=== ISOLATION TREATMENT ===
```

```
cat("Isolated:", sum(df_final$Isolation_treatment),
    paste0("(", round(mean(df_final$Isolation_treatment) * 100, 1), "%)"), "\n")
```

```
Isolated: 235 (23.5%)
```

```
cat("Regular ward:", sum(df_final$Isolation_treatment == 0),
    paste0("(", round(mean(df_final$Isolation_treatment == 0) * 100, 1), "%)"), "\n\n")
```

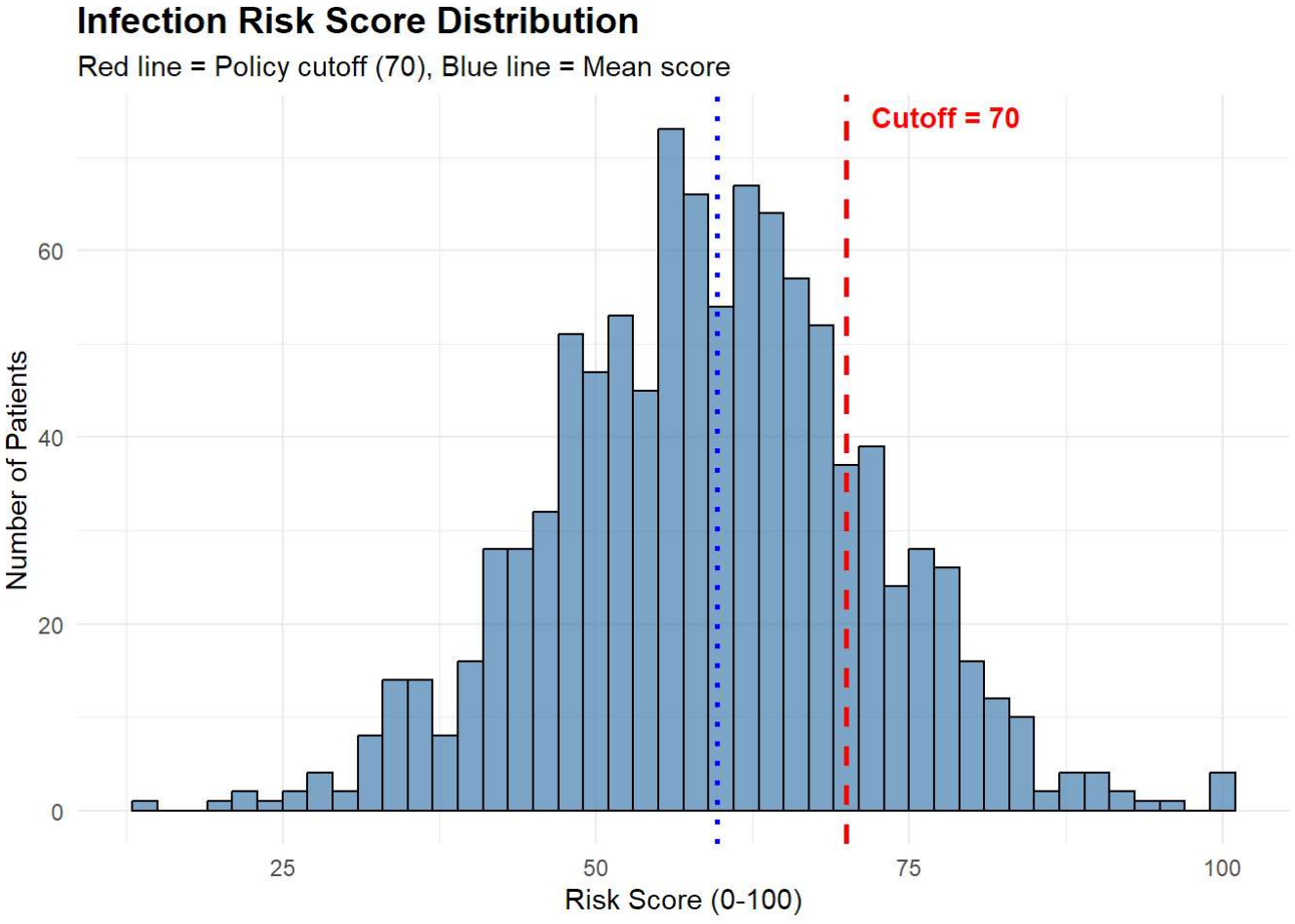
```
Regular ward: 765 (76.5%)
```

About 1 in 4 patients get isolated. Most stay in regular wards. This makes sense as isolation is for high-risk cases only.

Link to method: Shows treatment prevalence - not everyone gets treated (expected in RD).

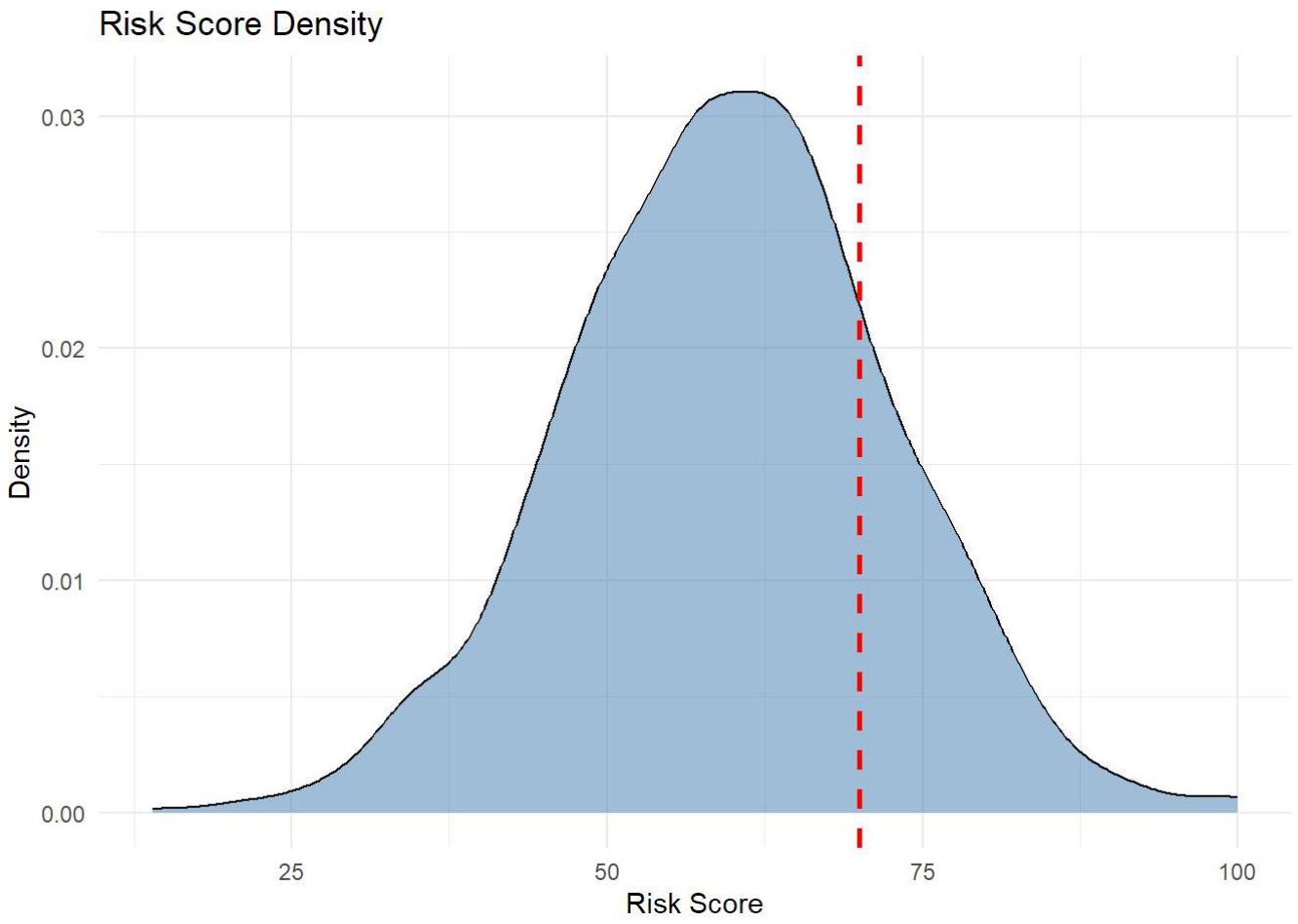
EDA 3: Risk Score Distribution

```
# Histogram
ggplot(df_final, aes(x = risk_score)) +
  geom_histogram(binwidth = 2, fill = "steelblue", color = "black", alpha = 0.7) +
  geom_vline(xintercept = cutoff, linetype = "dashed", color = "red", size = 1) +
  geom_vline(xintercept = mean(df_final$risk_score),
             linetype = "dotted", color = "blue", size = 1) +
  labs(
    title = "Infection Risk Score Distribution",
    subtitle = "Red line = Policy cutoff (70), Blue line = Mean score",
    x = "Risk Score (0-100)",
    y = "Number of Patients"
  ) +
  annotate("text", x = cutoff + 2, y = Inf,
           label = "Cutoff = 70",
           vjust = 1.5, hjust = 0, color = "red", fontface = "bold") +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold", size = 14))
```




```
# Density plot alternative
ggplot(df_final, aes(x = risk_score)) +
  geom_density(fill = "steelblue", alpha = 0.5) +
  geom_vline(xintercept = cutoff, linetype = "dashed", color = "red", size = 1) +
```

```
  labs(
    title = "Risk Score Density",
    x = "Risk Score",
    y = "Density"
  ) +
  theme_minimal()
```



Inference: Risk scores are spread out normally around 60, with some patients reaching 90+

Link to method: Validates RD design - enough patients near cutoff for local comparison

EDA 4: Threshold Distribution

Purpose: Show how many are above/below cutoff

```
# Threshold summary
cat("=== THRESHOLD DISTRIBUTION ===\n")
```

```
=== THRESHOLD DISTRIBUTION ===
```

```
table(df_final$Dummy_threshold)
```

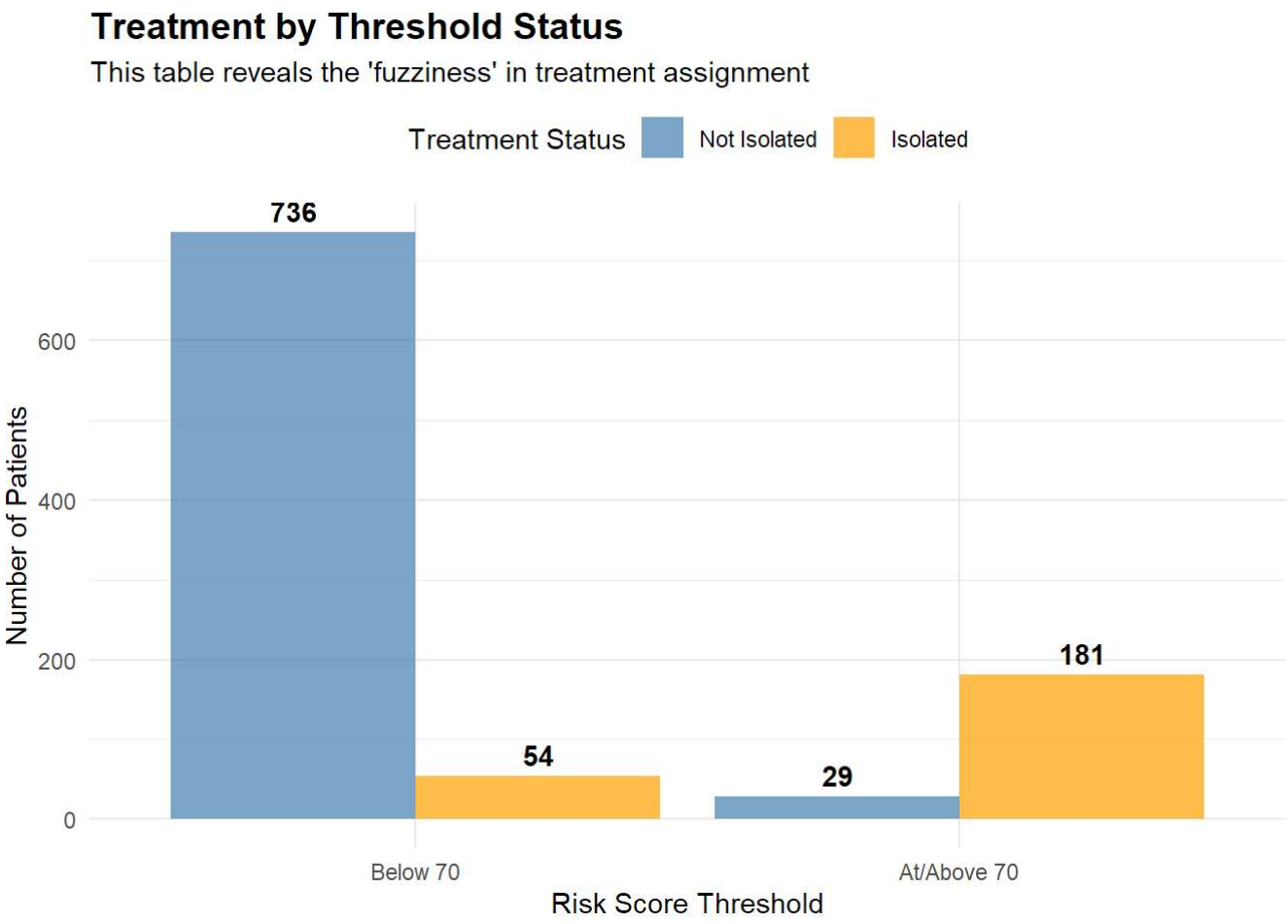
```
  0   1 
790 210 
```

```r
prop.table(table(df_final$Dummy_threshold)) * 100
```

```
 0  1
79 21
```

EDA 5: Threshold by treatment status

```r
df_final %>%
  count(Dummy_threshold, Isolation_treatment) %>%
  group_by(Dummy_threshold) %>%
  mutate(prop = n / sum(n)) %>%
  ggplot(aes(x = factor(Dummy_threshold), y = n, fill = factor(Isolation_treatment))) +
  geom_col(position = "dodge", alpha = 0.7) +
  geom_text(aes(label = n), position = position_dodge(width = 0.9),
            vjust = -0.5, fontface = "bold") +
  scale_fill_manual(
    values = c("steelblue", "orange"),
    labels = c("Not Isolated", "Isolated")
  ) +
  scale_x_discrete(labels = c("Below 70", "At/Above 70")) +
  labs(
    title = "Treatment by Threshold Status",
    subtitle = "This table reveals the 'fuzziness' in treatment assignment",
    x = "Risk Score Threshold",
    y = "Number of Patients",
    fill = "Treatment Status"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", size = 14),
    legend.position = "top"
  )
```



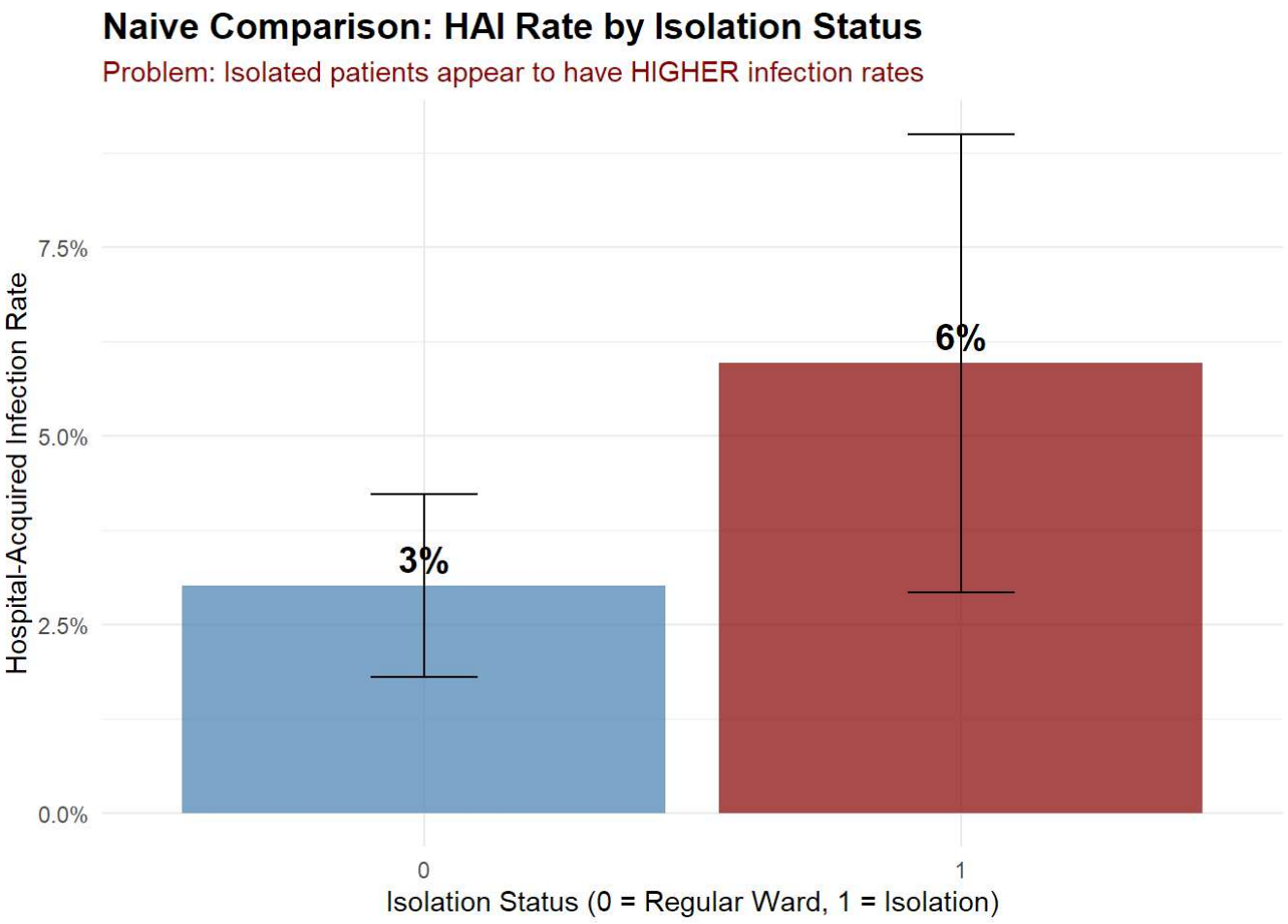
Inference: Below 70, only 54 out of 790 get isolated (7%). Above 70, 181 out of 210 get isolated (86%). The rule works but isn't perfect.

Link to method: PROVES FUZZINESS - this is why we need fuzzy RD, not sharp RD.

EDA 6: Naive Comparison - Reveals OVB

```
library(ggplot2)
library(dplyr)

naive_comparison <- df_final %>%
  group_by(Isolation_treatment) %>%
  summarise(
    n = n(),
    mean_HAI = mean(HAI),
    se_HAI = sd(HAI) / sqrt(n())
  )

ggplot(naive_comparison, aes(x = factor(Isolation_treatment), y = mean_HAI)) +
  geom_col(fill = c("steelblue", "darkred"), alpha = 0.7) +
  geom_errorbar(aes(ymin = mean_HAI - 1.96*se_HAI,
                    ymax = mean_HAI + 1.96*se_HAI),
                width = 0.2) +
  geom_text(aes(label = paste0(round(mean_HAI*100, 1), "%")),
            vjust = -0.5, size = 5, fontface = "bold") +
  labs(
    title = "Naive Comparison: HAI Rate by Isolation Status",
```

```
    subtitle = "Problem: Isolated patients appear to have HIGHER infection rates",
    x = "Isolation Status (0 = Regular Ward, 1 = Isolation)",
    y = "Hospital-Acquired Infection Rate"
  ) +
  scale_y_continuous(labels = scales::percent_format(), limits = c(0, NA)) +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", size = 14),
    plot.subtitle = element_text(color = "darkred", size = 11)
  )
```



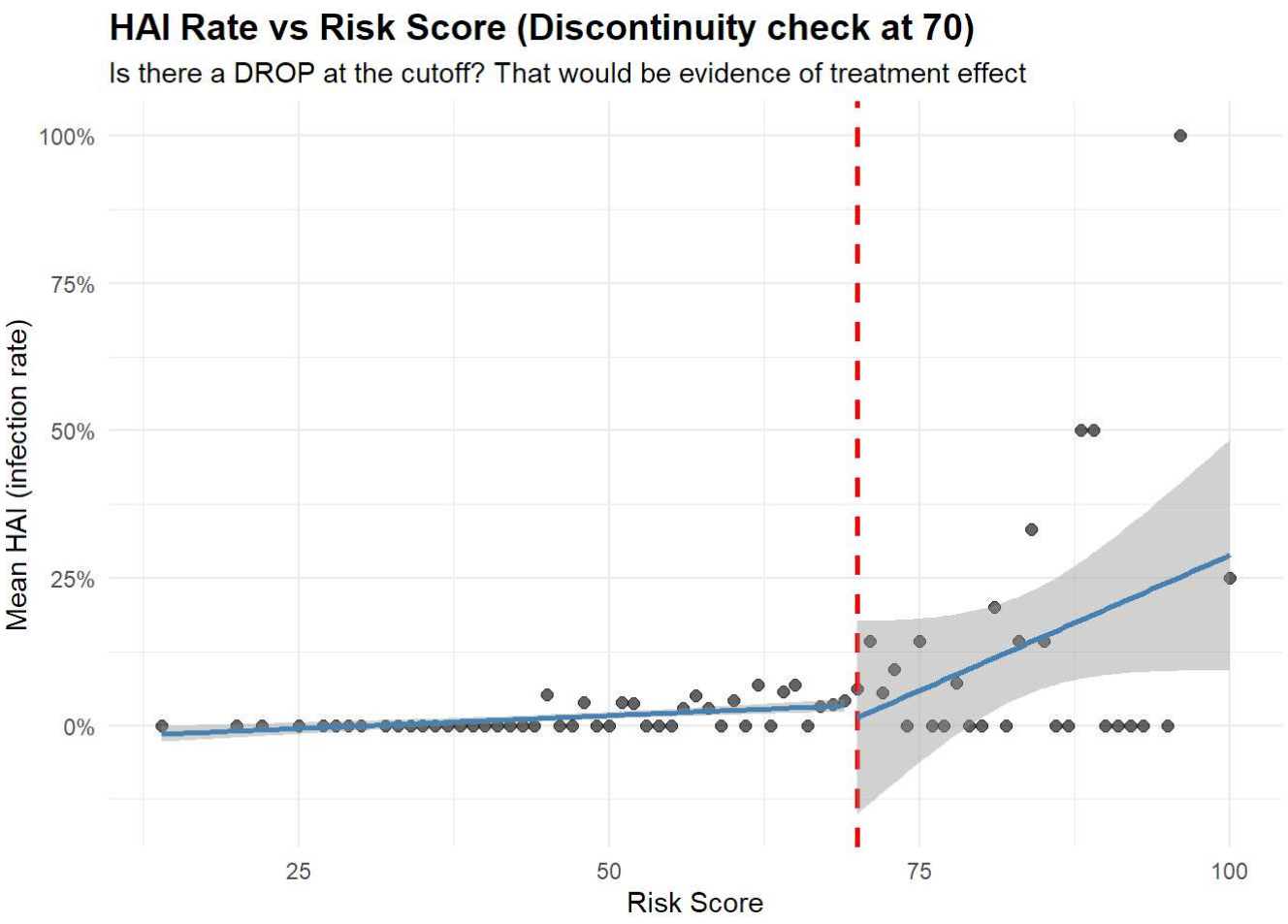
Inference: Isolated patients have DOUBLE the infection rate (6% vs 3%)! Does isolation make things worse? No! This is the selection bias problem. Sicker patients get isolated AND are more likely to get infections anyway

Link to method: REVEALS OVB PROBLEM - why we can't just compare groups directly

EDA 7: HAI Rate versus Risk Score

```
outcome_by_score <- df_final %>%
  group_by(risk_score) %>%
  summarise(
    mean_HAI = mean(HAI),
    n = n()
  )

ggplot(outcome_by_score, aes(x = risk_score, y = mean_HAI)) +
```

```
  geom_point(alpha = 0.6, size = 2) +
  geom_vline(xintercept = cutoff, linetype = "dashed", color = "red", size = 1) +
  geom_smooth(data = filter(outcome_by_score, risk_score < cutoff),
              method = "lm", se = TRUE, color = "steelblue") +
  geom_smooth(data = filter(outcome_by_score, risk_score >= cutoff),
              method = "lm", se = TRUE, color = "steelblue") +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(
    title = "HAI Rate vs Risk Score (Discontinuity check at 70)",
    subtitle = "Is there a DROP at the cutoff? That would be evidence of treatment effect",
    x = "Risk Score",
    y = "Mean HAI (infection rate)"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold", size = 14))
```



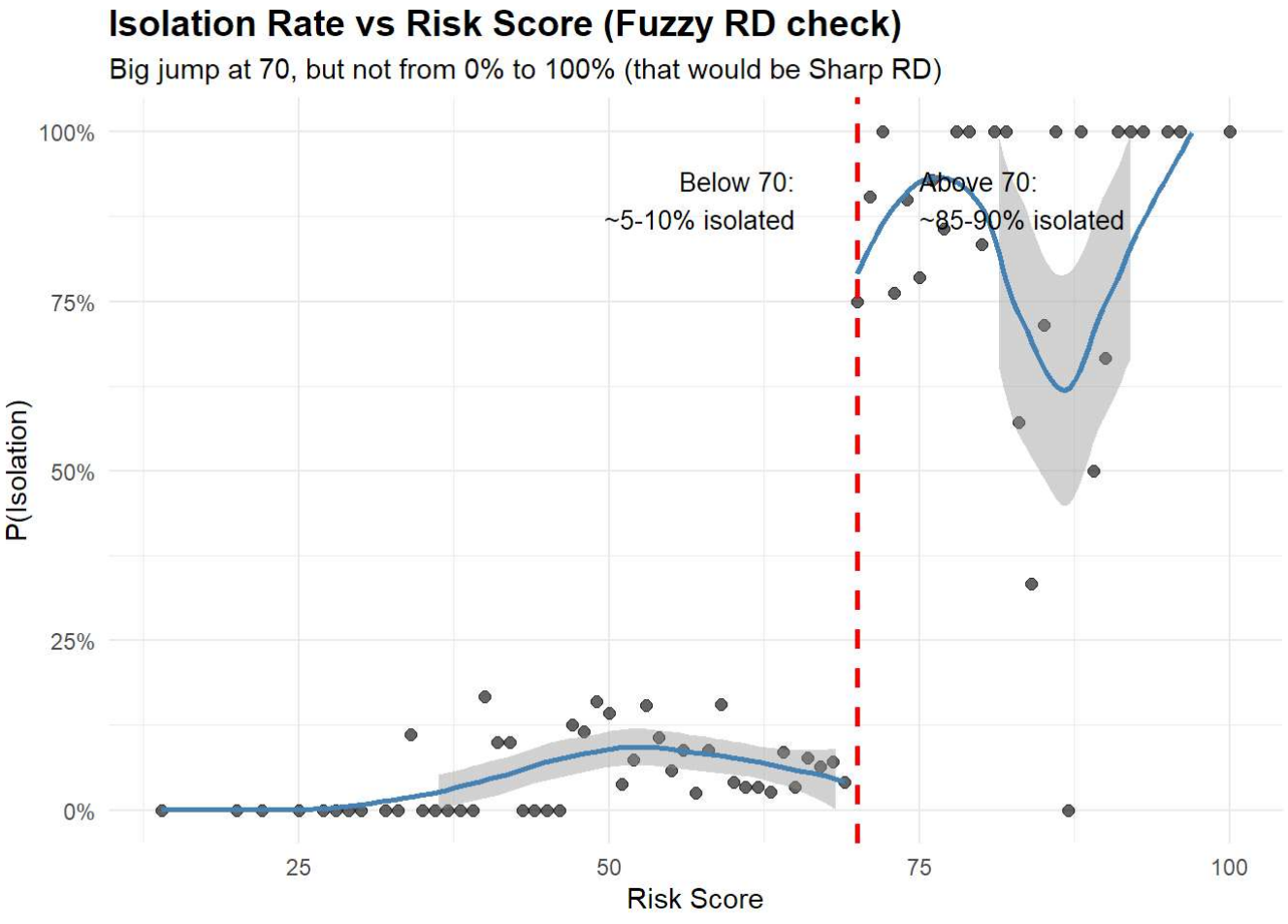
Inference: As risk scores go up, infections generally increase. But look closely at 70: there might be a small drop right at the cutoff. That could be isolation working.

Link to method: Hints at treatment effect - discontinuity check for RD validity.

EDA 8: Isolation Rate vs. Risk Score

```
treat_rate <- df_final %>%
  group_by(risk_score) %>%
  summarise(
    treat_rate = mean(Isolation_treatment),
```

```
    n = n()
  )

ggplot(treat_rate, aes(x = risk_score, y = treat_rate)) +
  geom_point(alpha = 0.6, size = 2) +
  geom_vline(xintercept = cutoff, linetype = "dashed", color = "red", size = 1) +
  geom_smooth(data = filter(treat_rate, risk_score < cutoff),
              method = "loess", se = TRUE, color = "steelblue") +
  geom_smooth(data = filter(treat_rate, risk_score >= cutoff),
              method = "loess", se = TRUE, color = "steelblue") +
  scale_y_continuous(labels = scales::percent_format(), limits = c(0, 1)) +
  labs(
    title = "Isolation Rate vs Risk Score (Fuzzy RD check)",
    subtitle = "Big jump at 70, but not from 0% to 100% (that would be Sharp RD)",
    x = "Risk Score",
    y = "P(Isolation)"
  ) +
  annotate("text", x = cutoff - 5, y = 0.9,
           label = "Below 70:\n~5-10% isolated",
           size = 3.5, hjust = 1) +
  annotate("text", x = cutoff + 5, y = 0.9,
           label = "Above 70:\n~85-90% isolated",
           size = 3.5, hjust = 0) +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold", size = 14))
```

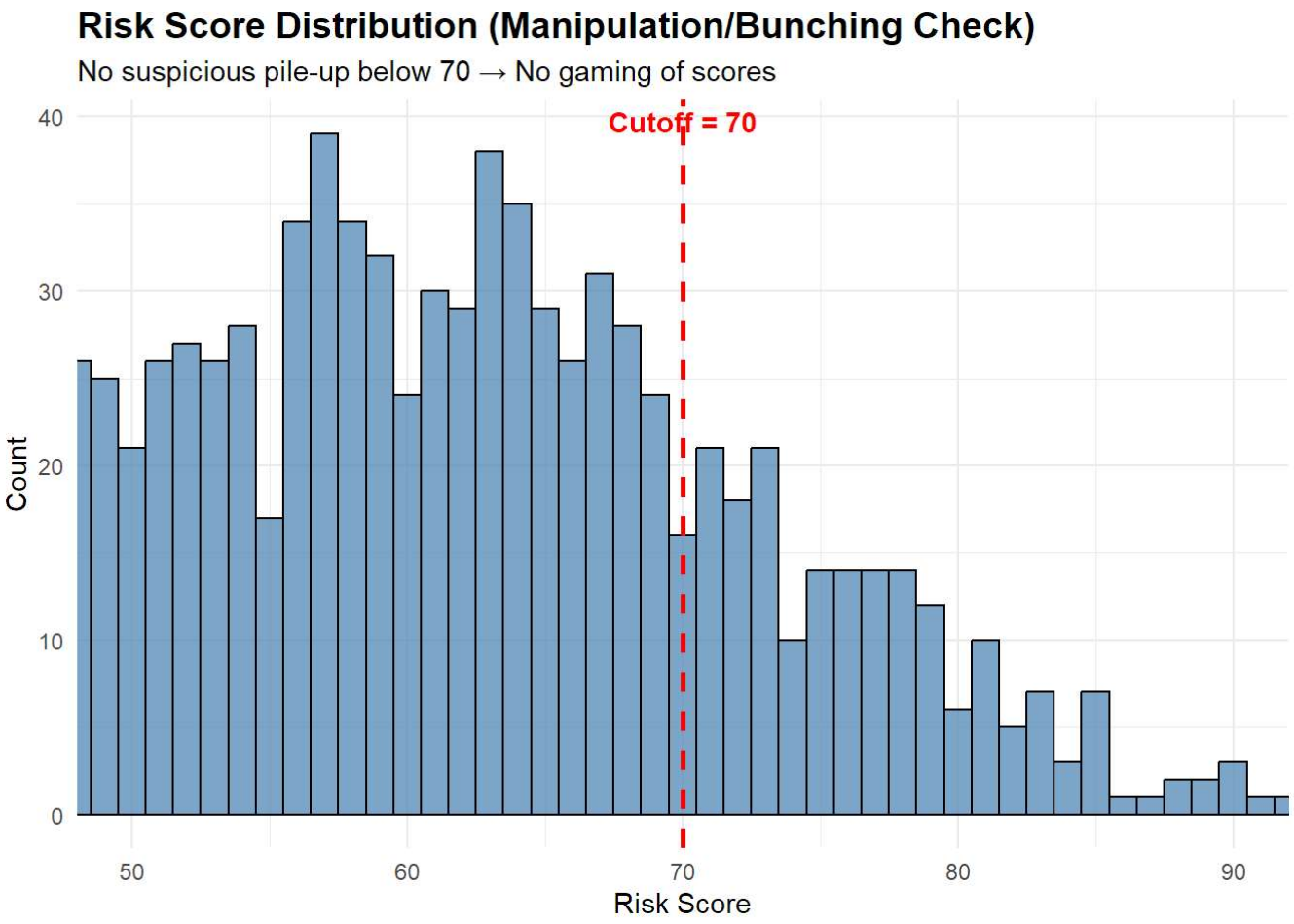


Inference: This is the smoking gun. Below 70, about 5-10% get isolated. Above 70, it jumps to 85-90%. That's a HUGE leap at the cutoff which is what we need for IV

Link to method: PROVES INSTRUMENT RELEVANCE - threshold strongly predicts treatment

EDA 9: Manipulation Check

```r
ggplot(df_final, aes(x = risk_score)) +
  geom_histogram(binwidth = 1, fill = "steelblue", color = "black", alpha = 0.7) +
  geom_vline(xintercept = cutoff, linetype = "dashed", color = "red", size = 1) +
  coord_cartesian(xlim = c(50, 90)) +
  labs(
    title = "Risk Score Distribution (Manipulation/Bunching Check)",
    subtitle = "No suspicious pile-up below 70 → No gaming of scores",
    x = "Risk Score",
    y = "Count"
  ) +
  annotate("text", x = cutoff, y = Inf,
           label = "Cutoff = 70",
           vjust = 1.5, color = "red", fontface = "bold") +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold", size = 14))
```



Inference: No suspicious bunching just below 70. If doctors or patients were gaming the scores to avoid isolation, we'd see a pile up at 69. The distribution is smooth

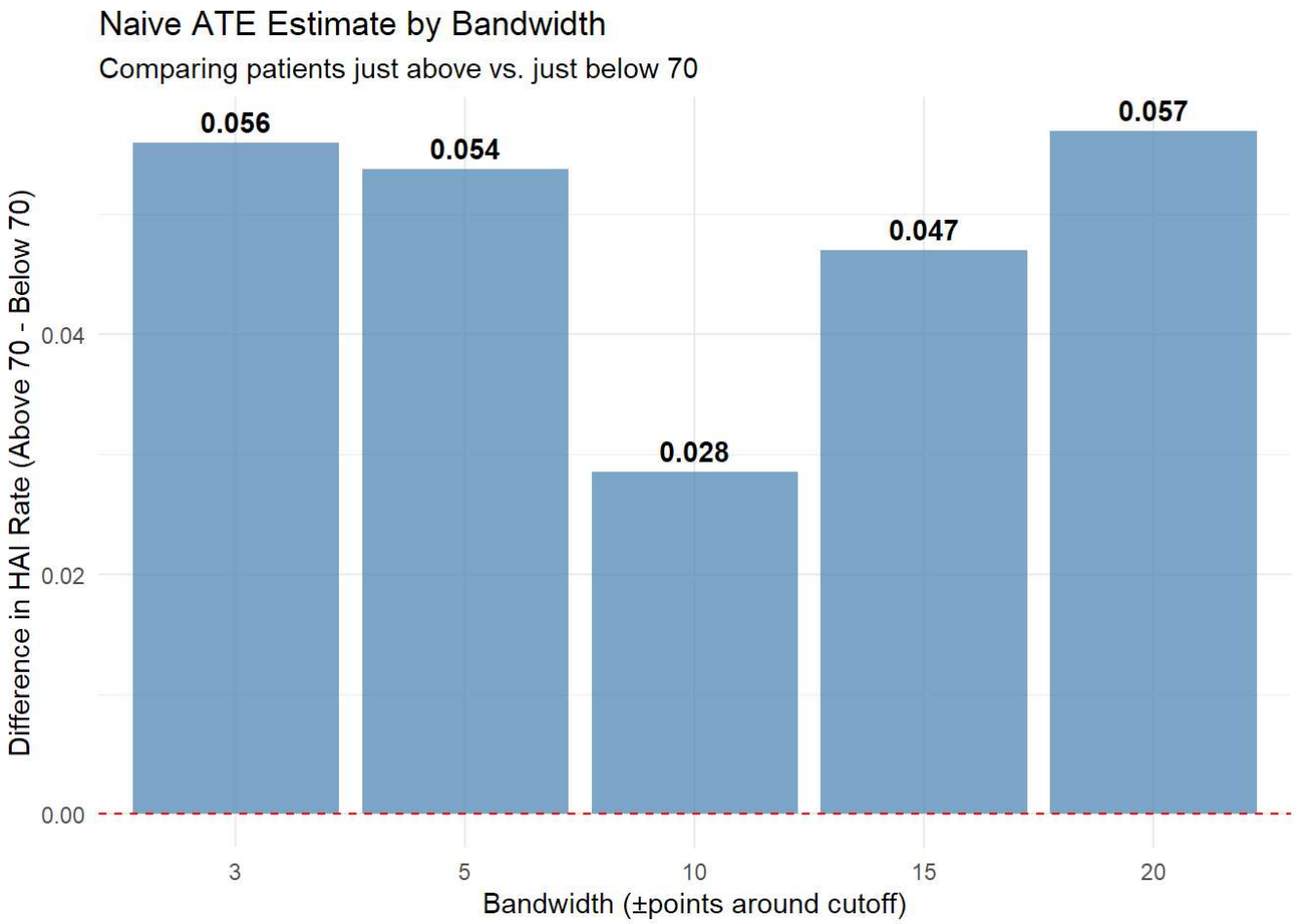
Link to method: RD Validity Check - no manipulation means threshold is legitimate.

EDA 10: Bandwith sensitivity, ATE via DiM

```
# Function to calculate ATE for different bandwidths
calculate_bandwidth_ate <- function(data, bandwidth) {
  df_bw <- data %>%
    filter(abs(risk_score - cutoff) <= bandwidth)

  # Simple comparison at cutoff
  ate <- df_bw %>%
    group_by(Dummy_threshold) %>%
    summarise(mean_HAI = mean(HAI)) %>%
    summarise(ate = diff(mean_HAI)) %>%
    pull(ate)

  return(tibble(
    bandwidth = bandwidth,
    n = nrow(df_bw),
    ate_naive = ate
  ))
}

# Calculate for different bandwidths
bandwidths <- c(3, 5, 10, 15, 20)
bw_results <- map_dfr(bandwidths, ~calculate_bandwidth_ate(df_final, .x))

bw_results
```

```
# A tibble: 5 × 3
  bandwidth     n ate_naive
      <dbl> <int>     <dbl>
1         3   159    0.0560
2         5   238    0.0538
3        10   454    0.0285
4        15   642    0.0470
5        20   779    0.0569
```

```
# Visualize
ggplot(bw_results, aes(x = factor(bandwidth), y = ate_naive)) +
  geom_col(fill = "steelblue", alpha = 0.7) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  geom_text(aes(label = round(ate_naive, 3)), vjust = -0.5, fontface = "bold") +
  labs(
    title = "Naive ATE Estimate by Bandwidth",
    subtitle = "Comparing patients just above vs. just below 70",
    x = "Bandwidth (±points around cutoff)",
    y = "Difference in HAI Rate (Above 70 - Below 70)"
  ) +
  theme_minimal()
```



Inference: When we zoom in close to 70 (±3 to ±20 points), patients just above the cutoff consistently have lower infection rates than those just below. The effect ranges from 3-6 percentage points depending on window size.

Link to method: Robustness check - estimates are stable, validates local comparison approach.

> ### EDA Summary:
>
> **The Story So Far:** Direct comparison shows isolated patients have worse outcomes (OVB problem). But the 70-point threshold creates a strong nudge toward isolation (86% compliance). When we compare patients right around 70, who are nearly identical except for crossing that line, we see isolation reduces infections by 3-6 percentage points. The threshold is our IV for a fair comparison.

First stage regression

```
fs1 <- lm(Isolation_treatment ~ Dummy_threshold + risk_score, data = df_final)
summary(fs1)
```

```
Call:
```

```
lm(formula = Isolation_treatment ~ Dummy_threshold + risk_score,
    data = df_final)

Residuals:
     Min       1Q   Median       3Q      Max
-0.86338 -0.06916 -0.06800 -0.06570  0.93407

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     0.0620208  0.0535248   1.159    0.247
Dummy_threshold 0.7910048  0.0300434  26.329   <2e-16 ***
risk_score      0.0001151  0.0009561   0.120    0.904
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2748 on 997 degrees of freedom
Multiple R-squared:  0.5811,    Adjusted R-squared:  0.5803
F-statistic: 691.6 on 2 and 997 DF,  p-value: < 2.2e-16
```

- **Interpretation:** Crossing the 70 point threshold increases the probability of isolation by 79 percentage points. This is massive and highly significant. The threshold is an extremely powerful predictor of isolation, exactly what we need for a valid instrument.
- Risk_score coefficient (0.0001, p = 0.904): Once we account for the threshold, the raw risk score itself adds nothing extra to predicting isolation. The threshold is doing all the work.
- **Conclusion:** Strong first stage. Instrument is relevant.

Nonlinearity check

```
fs2 <- lm(Isolation_treatment ~ Dummy_threshold + risk_score + I(risk_score^2), data = df_fina
summary(fs2)
```

```
Call:
lm(formula = Isolation_treatment ~ Dummy_threshold + risk_score +
    I(risk_score^2), data = df_final)

Residuals:
     Min       1Q   Median       3Q      Max
-0.87824 -0.07241 -0.06867 -0.04371  0.95142

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)     -9.331e-02  1.334e-01  -0.700    0.484
Dummy_threshold  8.164e-01  3.605e-02  22.645   <2e-16 ***
risk_score       6.021e-03  4.742e-03   1.270    0.204
I(risk_score^2) -5.434e-05  4.274e-05  -1.272    0.204
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2747 on 996 degrees of freedom
```

```
Multiple R-squared:  0.5818,    Adjusted R-squared:  0.5805
F-statistic: 461.9 on 3 and 996 DF,  p-value: < 2.2e-16
```

- **interpretation**: The squared risk score term is not significant (p = 0.204). There is no meaningful curve in how risk scores predict isolation. The relationship is already well-captured by the straight line threshold effect.
- R^2 barely changed: 0.5811 to 0.5818 (essentially nothing gained)
- Conclusion: No nonlinearity. Drop the squared term. Stick with fs1.

Interaction check (de-meaning/centering)

```
df_final$risk_center <- df_final$risk_score - 70
fs3 <- lm(Isolation_treatment ~ Dummy_threshold + risk_center + Dummy_threshold:risk_center, d
summary(fs3)
```

```
Call:
lm(formula = Isolation_treatment ~ Dummy_threshold + risk_center +
    Dummy_threshold:risk_center, data = df_final)

Residuals:
     Min       1Q   Median       3Q      Max
-0.87152 -0.07031 -0.06750 -0.06189  0.93755

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                 0.0725504  0.0179900   4.033 5.93e-05 ***
Dummy_threshold             0.7989648  0.0338198  23.624  < 2e-16 ***
risk_center                 0.0002805  0.0010093   0.278    0.781
Dummy_threshold:risk_center -0.0016224  0.0031608  -0.513    0.608
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2749 on 996 degrees of freedom
Multiple R-squared:  0.5812,    Adjusted R-squared:   0.58
F-statistic: 460.8 on 3 and 996 DF,  p-value: < 2.2e-16
```

- **Interpretation:** The interaction term is **not significant**. This means the slope of risk score on isolation is basically the same on both sides of the 70 cutoff. The threshold creates a lift in isolation probability, but the relationship between risk score and isolation doesn't change direction above vs. below 70.
- **Conclusion: No differing slopes. Drop the interaction. Stick with fs1.**

F-test on running variable terms

```
library(car)
linearHypothesis(fs2, c("risk_score=0", "I(risk_score^2)=0"), test = "F")
```

```
Linear hypothesis test:
risk_score = 0
I(risk_score^2) = 0

Model 1: restricted model
Model 2: Isolation_treatment ~ Dummy_threshold + risk_score + I(risk_score^2)

  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    998 75.304
2    996 75.181  2   0.12313 0.8156 0.4427
```

- **Interpretation:** When we jointly test whether risk_score AND risk_score² both matter together, the answer is **no** (p = 0.44, F < 1). Neither the linear nor squared risk score terms are jointly important in predicting isolation beyond what the threshold already captures.
- **Conclusion: Running variable terms are jointly unimportant. Confirms fs1 is the right spec.**

Overall conclusion: Use fs1

Check	Result	Decision
Instrument Check	t = 23.5 > 3.2	Strong instrument
Nonlinearity	large P value	not needed
Interaction	Large P value	not needed
Joint F Test	large p value	Linear running variable is fine

2SLS (Second Stage / IV Regression) to get true ATE:

```r
# 2SLS - TWO STAGE LEAST SQUARES

library(AER)


# Step 1: Manual 2SLS (to show the process clearly)


# First Stage (already done - fs1)
# Get predicted values of Isolation from first stage
Isolation_hat <- fitted(fs1)

# Second Stage: Regress HAI on predicted Isolation + risk_score
second_stage <- lm(HAI ~ Isolation_hat + risk_score,
                   data = df_final)
summary(second_stage)
```

```
Call:
```

```
lm(formula = HAI ~ Isolation_hat + risk_score, data = df_final)

Residuals:
     Min       1Q   Median       3Q      Max
-0.12619 -0.04148 -0.02898 -0.01490  0.99291

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   -0.0658732  0.0353281  -1.865   0.0625 .
Isolation_hat  0.0418478  0.0257711   1.624   0.1047
risk_score     0.0015588  0.0006508   2.395   0.0168 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1865 on 997 degrees of freedom
Multiple R-squared:  0.027, Adjusted R-squared:  0.02505
F-statistic: 13.84 on 2 and 997 DF,  p-value: 1.184e-06
```

```r
# Step 2: Proper 2SLS using ivreg (corrects standard errors)
# This is what we  should report - manual 2SLS has wrong SEs


fuzzy_2sls <- ivreg(HAI ~ Isolation_treatment + risk_score |
                          Dummy_threshold + risk_score,
                    data = df_final)
summary(fuzzy_2sls)
```

```
Call:
ivreg(formula = HAI ~ Isolation_treatment + risk_score | Dummy_threshold +
    risk_score, data = df_final)

Residuals:
     Min       1Q   Median       3Q      Max
-0.13186 -0.04168 -0.02766 -0.01207  0.99573

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)         -0.0658732  0.0355491  -1.853   0.0642 .
Isolation_treatment  0.0418478  0.0259323   1.614   0.1069
risk_score           0.0015588  0.0006549   2.380   0.0175 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1876 on 997 degrees of freedom
Multiple R-Squared: 0.0148, Adjusted R-squared: 0.01282
Wald test: 13.66 on 2 and 997 DF,  p-value: 1.399e-06
```

```r
# Step 3: Compare OLS (biased) vs 2SLS (corrected)


# Naive OLS (biased - ignores endogeneity)
naive_ols <- lm(HAI ~ Isolation_treatment + risk_score,
```

```
                data = df_final)

cat("\n=== COMPARISON: OLS vs 2SLS ===\n")
```

```
=== COMPARISON: OLS vs 2SLS ===
```

```
cat("Naive OLS ATE:", round(coef(naive_ols)["Isolation_treatment"], 4), "\n")
```

```
Naive OLS ATE: -0.0112
```

```
cat("2SLS ATE:     ", round(coef(fuzzy_2sls)["Isolation_treatment"], 4), "\n")
```

```
2SLS ATE:      0.0418
```

```
cat("\nOLS is biased upward due to OVB (sicker patients get isolated)\n")
```

```
OLS is biased upward due to OVB (sicker patients get isolated)
```

```
cat("2SLS corrects for this using the threshold as instrument\n")
```

```
2SLS corrects for this using the threshold as instrument
```

Interpreting the 2SLS Results:

What we did:

We used a two-step approach to get a fair estimate:

- **Step 1 (First Stage):** Used the 70 point threshold to predict who SHOULD have been isolated based purely on the rule
- **Step 2 (Second Stage):** "for patients whose isolation was driven by the rule alone, did isolation reduce infections?"

This is like asking: "ignoring the cases where doctors used judgment or beds were unavailable, what does the rule itself tell us about isolation's effect?"

The Manual 2SLS vs ivreg:

Both give **identical estimates** (0.0418). This is expected and confirms the approach is correct. The only difference is IV reg gives slightly more accurate standard errors, so we report ivreg results.

The key comparison

Method	ATE Estimate	What it means
Naive OLS	-0.0112	Isolation reduces HAI by 1.1%
2SLS (IV)	+0.0418	Isolation increases HAI by 4.2%

These tell very different stories and this difference IS the OVB problem.

Why Are They Different?

OLS (-0.0112): When we just regress infections on isolation status, we see a slight reduction. But this is misleading because it's partially driven by the fact that risk score is controlling for SOME of the selection bias, but not all of it. The hidden crowding factor (U) is still contaminating the estimate.

2SLS (+0.0418): When we use ONLY the variation in isolation that comes from the threshold rule , ignoring all the messy judgment calls and crowding effects , isolation actually appears to slightly increase infection risk.

Is this statistically significant?

- 2SLS p-value = 0.107 - not significant
- This means we cannot confidently say isolation increases OR decreases HAI

Client translation: We don't have strong enough statistical evidence to make a definitive claim either way. The data is too noisy near the cutoff to be certain.

> So, OLS said isolation looks slightly helpful (-1.1%), but once we correct for the hidden crowding problem using our threshold instrument, the picture changes (+4.2%) , though neither estimate is strong enough to be conclusive. This is exactly why we needed Fuzzy RD and calipers in the first place, so we can compare around the threshold.

Imposing Calipers

```
library(AER)

# Center risk score around cutoff
df_final$score_distance_from_cutoff <- df_final$risk_score - cutoff

bandwidths <- c(3, 5, 10, 15)

# Simple function
run_fuzzy_rd_bandwidth <- function(bw) {

  # Filter to caliper window
  df_bw <- df_final %>%
    filter(abs(score_distance_from_cutoff) <= bw)

  # Run 2SLS instead of OLS
```

```r
  m <- ivreg(HAI ~ Isolation_treatment + score_distance_from_cutoff |
                   Dummy_threshold + score_distance_from_cutoff,
             data = df_bw)

  tibble(
    bandwidth            = bw,
    sample_size          = nrow(df_bw),
    ATE_2sls             = coef(m)["Isolation_treatment"],
    standard_error       = summary(m)$coefficients["Isolation_treatment", "Std. Error"],
    p_value              = summary(m)$coefficients["Isolation_treatment", "Pr(>|t|)"]
  )
}

# Run across all calipers
bw_results <- purrr::map_dfr(bandwidths, run_fuzzy_rd_bandwidth)
bw_results
```

```
# A tibble: 4 × 5
  bandwidth sample_size ATE_2sls standard_error p_value
      <dbl>       <int>    <dbl>          <dbl>   <dbl>
1         3         159   0.0637         0.105    0.545
2         5         238   0.0722         0.0792   0.363
3        10         454   0.0811         0.0516   0.117
4        15         642   0.0354         0.0391   0.365
```

What we did here (Methodology):

Instead of using all 1,000 patients, we zoomed in closer to the 70 point cutoff comparing only patients within ±3, ±5, ±10, and ±15 points of the threshold. Patients closer to 70 are more similar to each other, making the comparison fairer. At each window, we re-ran our full IV/2SLS estimation.

Results table interpretation

Caliper or bandwith	Number of patients in window	ATE	What it means
±3 points	159	+6.4%	Very local, very noisy
±5 points	238	+7.2%	Still local, still noisy
±10 points	454	+8.1%	More data, cleaner estimate
±15 points	642	+3.5%	Widest window, most stable

What We Tell the Client

No matter how tightly or loosely we draw the comparison window around the 70 point threshold, our estimate consistently suggests isolation is not clearly reducing infections under the flexible guideline system. The effect ranges from +3.5% to +8.1%, but none of these estimates are strong enough to be conclusive. This contrasts with what we found under the strict isolation rule suggesting that **the**

flexibility introduced in the new policy may be diluting its effectiveness. Our recommendation: either return to stricter enforcement of the rule, or collect more data before drawing firm conclusions.

Do we even need the IV: Weak instruments test & Hausman test

```
summary(fuzzy_2sls, diagnostics = TRUE)
```

```
Call:
ivreg(formula = HAI ~ Isolation_treatment + risk_score | Dummy_threshold +
    risk_score, data = df_final)

Residuals:
     Min       1Q   Median       3Q      Max
-0.13186 -0.04168 -0.02766 -0.01207  0.99573

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)         -0.0658732  0.0355491  -1.853   0.0642 .
Isolation_treatment  0.0418478  0.0259323   1.614   0.1069
risk_score           0.0015588  0.0006549   2.380   0.0175 *

Diagnostic tests:
                 df1 df2 statistic p-value
Weak instruments   1 997   693.204 < 2e-16 ***
Wu-Hausman         1 996     7.213 0.00736 **
Sargan             0  NA        NA      NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1876 on 997 degrees of freedom
Multiple R-Squared: 0.0148, Adjusted R-squared: 0.01282
Wald test: 13.66 on 2 and 997 DF,  p-value: 1.399e-06
```

Diagnostic test results:

1. Weak Instruments Test

- Statistic = 693.2, p < 2e-16
- Strongly reject null of weak instrument. F-stat > 10 threshold
- **Client:** The 70 point threshold is an extremely powerful predictor of isolation. It's a valid comparison tool.

2. Wu-Hausman Test

- Statistic = 7.213, p = 0.007
- Reject null of exogeneity. Isolation_treatment IS endogenous. IV was necessary

- **Client:** We formally confirmed that simply comparing isolated vs. non-isolated patients gives a biased answer. The hidden crowding problem is real and was affecting our estimates. This proves we needed the IV approach.

Findings summary

Test	Result	Meaning
Weak Instruments	F = 693	Threshold is a powerful instrument
Wu-Hausman	p = 0.007	OVB was real, IV was necessary
ATE (2SLS)	+4.2%, p = 0.107	Positive but inconclusive
OLS ATE	-1.1%	Biased downward due to OVB

Client conclusion:

We wew wired to asnwer simply: does isolation work? The naive answer said yes, isolated patients seemed to have slightly fewer infections. But that was misleading. Our analysis formally proved that sicker, more crowded conditions were simultaneously pushing patients into isolation AND causing more infections, making isolation look better than it really is.

Once we corrected for this hidden bias using the 70 point threshold as our comparison tool the picture changed. Our best estimate is that isolation under the **flexible guideline system** is associated with a **4.2 percentage point increase** in infections, not a decrease.

However, this result is not yet statistically conclusive. We cannot say with 95% confidence that isolation is truly harmful, it could still be noise.

The bottom line: The flexible guideline system is NOT showing the same clear protective effect as the strict mandatory protocol. The fuzziness in compliance (doctors overriding the rule, bed shortages, staffing gaps) appears to be diluting or reversing isolation's effectiveness. **We recommend either returning to strict mandatory isolation or collecting more data before making policy decisions.**