Homework3

Varun Ramanathan

2025-09-25

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.1     v stringr   1.5.2
## v ggplot2   4.0.0     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(broom)
library(estimatr)
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(colorfindr)
library(ggdist)
library(ggbeeswarm)
library(gghalves)
```

Background & Decision Rule

We need to decide whether to proceed with a full redesign. Finance requires a $\geq$ **$1.80** increase in **per-customer sales** to approve. We analyze 200 historical sales (Old vs New).

Primary decision rule: Approve only if the estimated effect (New − Old) is credibly $\geq$ **$1.80**.

Brand colors: Plots use a palette extracted from **SumerSports** via `colorfindr`. Categorical mappings (Old/New) use two brand colors; sequential mappings (NPS) use a white→brand ramp.

```
df <- read.csv("homework3_data.csv")
nrow(df)
```

```
## [1] 200
```

```
summary(df$sales)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   25.83   30.86   33.76   33.66   35.57   44.30
```

```
df <- df %>%
  mutate(
    design   = as.integer(design),
    design_f = factor(design, levels = c(0,1), labels = c("Old","New"))
  )

table(df$design_f)
```

```
##
## Old New
## 101  99
```

```
modern_theme <- theme_minimal(base_size = 12) +
  theme(
    panel.grid.minor = element_blank(),
    plot.title = element_text(face = "bold", size = 14),
    axis.title.y = element_text(margin = margin(r = 8))
  )
theme_set(modern_theme)

if(!requireNamespace("rvest", quietly=TRUE)) install.packages("rvest", quiet=TRUE)
if(!requireNamespace("urltools", quietly=TRUE)) install.packages("urltools", quiet=TRUE)
library(rvest); library(urltools)
```

```
##
## Attaching package: 'rvest'

## The following object is masked from 'package:readr':
##
##     guess_encoding
```

```
site <- "https://sumersports.com/"
html <- read_html(site)
imgs <- html_elements(html, "img") |> html_attr("src") |> na.omit() |> unique()
abs_urls <- vapply(imgs, function(u) if (grepl("^https?://", u)) u else url_absolute(u, site), characte
cand <- abs_urls[grepl("logo|brand|header|site|icon", abs_urls, ignore.case=TRUE)]
if(length(cand)==0) cand <- abs_urls
```

```
ord <- order(!grepl("\\.svg($|\\?)", cand), !grepl("\\.png($|\\?)", cand))
cand <- cand[ord]
ok <- FALSE
for (u in cand) {
  res <- try({ col_tbl <- colorfindr::get_colors(u, top_n=10); ok <- TRUE; logo_url <- u }, silent=TRUE)
  if (ok) break
}
if(!ok) stop("no colors found")

pal_raw <- colorfindr::make_palette(col_tbl, n=8, show=FALSE)
hex_to_rgb <- function(hex) col2rgb(hex)[,1]
lum <- function(hex){ rgb <- hex_to_rgb(hex)/255; sum(c(0.2126,0.7152,0.0722)*rgb) }
pal_df <- tibble(hex=pal_raw, lum=purrr::map_dbl(pal_raw, lum)) |> arrange(lum)

cat_cols <- pal_df$hex[c(2, nrow(pal_df)-1)]; names(cat_cols) <- c("Old","New")
primary <- pal_df$hex[round(nrow(pal_df)/2)]
seq_pal <- scales::colour_ramp(c("#FFFFFF", primary))
seq_map <- function(x) seq_pal(scales::rescale(x, to=c(0,1)))

cat_cols; primary; logo_url
```

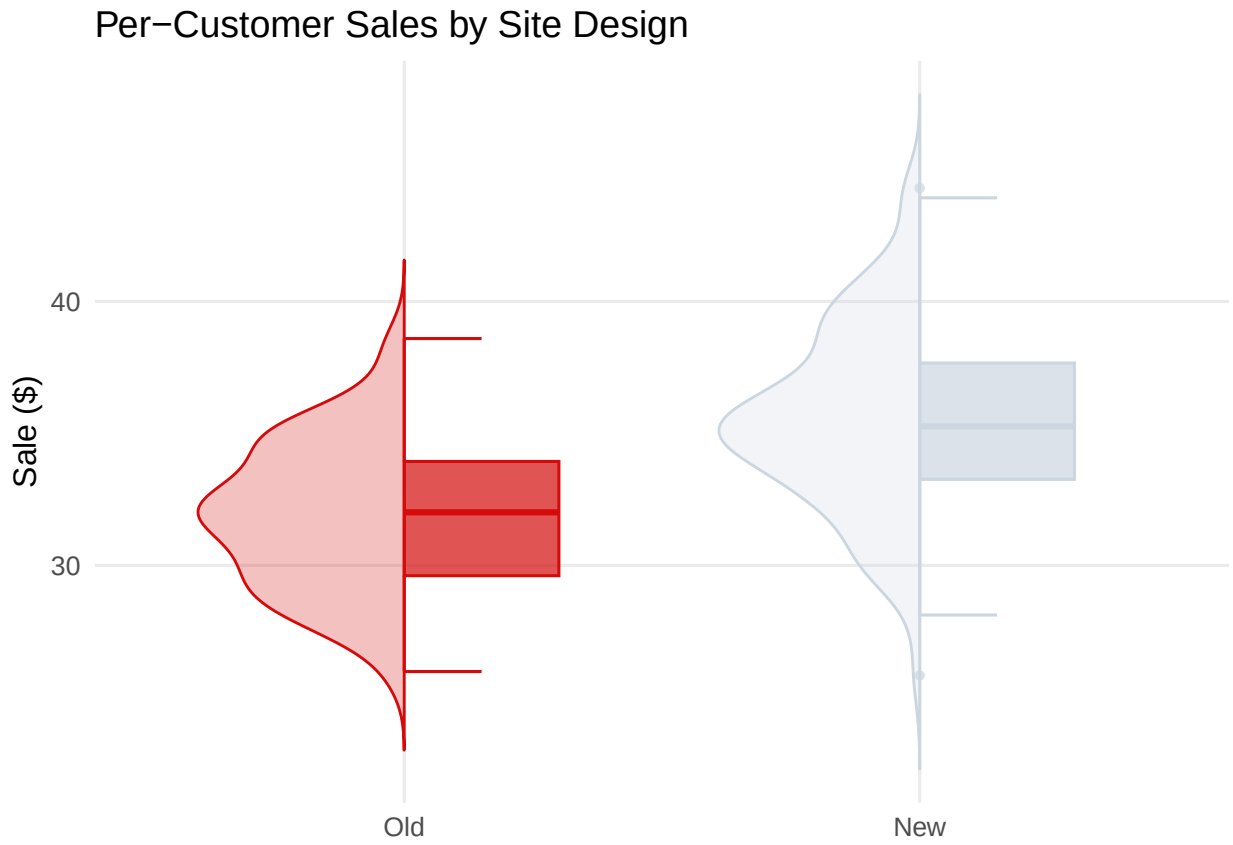
```
##       Old       New
## "#D50A0A" "#CCD6E1"
```

```
## [1] "#6E8AA9"
```

```
## [1] "https://cms-cdn.prd.sumersports.com/nfl-logo.svg"
```

```
library(gghalves)

ggplot(df, aes(design_f, sales, fill = design_f, color = design_f)) +
  geom_half_violin(side = "l", width = .8, alpha = .25, linewidth = .5, trim = FALSE) +
  geom_half_boxplot(side = "r", width = .6, alpha = .7, outlier.shape = 16) +
  scale_fill_manual(values = cat_cols, guide = "none") +
  scale_color_manual(values = cat_cols, guide = "none") +
  labs(x = NULL, y = "Sale ($)", title = "Per-Customer Sales by Site Design") +
  theme_minimal(base_size = 12) +
  theme(panel.grid.minor = element_blank())
```




```
group_summary <- df |>
  group_by(design_f) |>
  summarise(
    n = n(),
    mean_sales = mean(sales),
    sd_sales   = sd(sales),
    .groups = "drop"
  )

knitr::kable(
  group_summary |> mutate(across(mean_sales:sd_sales, ~round(.x, 2))),
  caption = "Sales summary by design (unadjusted)."
)
```

Table 1: Sales summary by design (unadjusted).

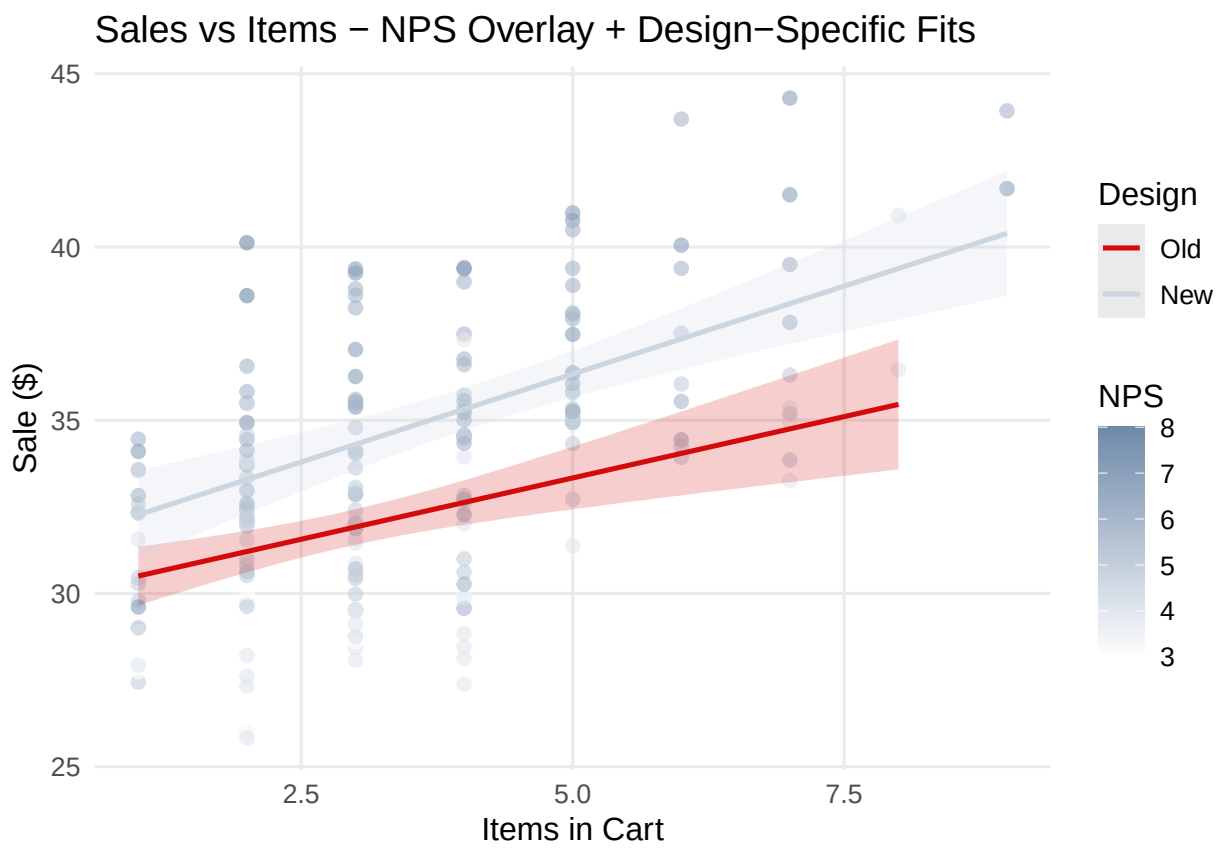
design_f	n	mean_sales	sd_sales
Old	101	31.85	2.78
New	99	35.51	3.53

```
if(!requireNamespace("ggnewscale", quietly=TRUE)) install.packages("ggnewscale", quiet=TRUE)
library(ggnewscale)

line_cols <- cat_cols
```

```r
ggplot(df, aes(items, sales)) +
  geom_point(aes(color = nps), alpha = 0.6, size = 2) +
  scale_color_gradientn(colors = seq_map(seq(0, 1, length.out = 7)), name = "NPS") +
  ggnewscale::new_scale_color() +
  ggnewscale::new_scale_fill() +
  geom_smooth(aes(color = design_f, fill = design_f), method = "lm", se = TRUE, linewidth = 0.9, alpha =
  scale_color_manual(values = line_cols, name = "Design") +
  scale_fill_manual(values = line_cols, guide = "none") +
  labs(x = "Items in Cart", y = "Sale ($)", title = "Sales vs Items -- NPS Overlay + Design-Specific Fi
  theme_minimal(base_size = 12) +
  theme(panel.grid.minor = element_blank())
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```r
tu <- broom::tidy(t.test(sales ~ design_f, data=df, alternative="greater"))
fa <- estimatr::lm_robust(sales ~ design + items, data=df, se_type="HC2")
ta <- broom::tidy(fa, conf.int=TRUE)

# Adjusted effect (design = New vs Old)
b   <- ta$estimate[ta$term=="design"]
se  <- ta$std.error[ta$term=="design"]
lwr <- ta$conf.low[ta$term=="design"]
upr <- ta$conf.high[ta$term=="design"]
```

```r
# Compute New−Old directly so it matches your table
mean_new <- with(df, mean(sales[design_f=="New"]))
mean_old <- with(df, mean(sales[design_f=="Old"]))
unadj_new_old <- mean_new - mean_old

# One-sided threshold test at $1.80 using adjusted estimate
thr <- 1.80
z   <- (b - thr)/se
p1  <- 1 - pnorm(z)
pass <- lwr >= thr

# Pretty CI printing for the one-sided t-test (upper bound may be Inf)
ci_hi <- if (is.finite(tu$conf.high)) sprintf("%.2f", tu$conf.high) else "Inf"

cat(sprintf("Unadjusted (New−Old): %.2f (95%% CI %.2f, %s), one-sided p=%.4f\n",
            unadj_new_old, tu$conf.low, ci_hi, tu$p.value))
```

```
## Unadjusted (New−Old): 3.66 (95% CI -4.41, Inf), one-sided p=1.0000
```

```r
cat(sprintf("Adjusted effect: %.2f (95%% CI %.2f, %.2f)\n", b, lwr, upr))
```

```
## Adjusted effect: 2.52 (95% CI 1.67, 3.37)
```

```r
cat(sprintf("Test vs $1.80: z=%.2f, p(one-sided)=%.4f; CI clears 1.80? %s\n",
            z, p1, ifelse(pass,"YES","NO")))
```

```
## Test vs $1.80: z=1.66, p(one-sided)=0.0480; CI clears 1.80? NO
```

```r
if ((b > thr) && (p1 < 0.05) && pass) {
  rec <- "Proceed with redesign"
} else if ((b > thr) && (p1 < 0.10)) {
  rec <- "Proceed conditionally (confirm with quick A/B)"
} else {
  rec <- "Hold on redesign"
}
cat(sprintf("Recommendation: %s\n", rec))
```

```
## Recommendation: Proceed conditionally (confirm with quick A/B)
```

Results & Recommendation

Unadjusted comparison. The raw difference (New − Old) is $3.66 * *with95$**-4.41 to** ∞ (one-sided p = 1.0000).

Adjusted effect. Controlling for items in cart, the redesign is associated with **$2.52 per customer** (95% CI **$1.67 to $3.37**).

Finance bar ($\geq$ $1.80). One-sided test vs $1.80 gives **p = 0.0480** and the 95% CI **does not** clear $1.80 from below.

Recommendation. Proceed conditionally (confirm with quick A/B).
This call uses the adjusted estimate, a direct test against $1.80, and interval uncertainty.

```r
monthly_customers <- 100000   # change as needed
est_monthly_rev   <- monthly_customers * b
knitr::kable(
  tibble(
    monthly_customers = scales::comma(monthly_customers),
    lift_per_customer = scales::dollar(round(b, 2)),
    est_monthly_lift  = scales::dollar(round(est_monthly_rev, 0))
  ),
  caption = "Back-of-envelope revenue impact."
)
```

Table 2: Back-of-envelope revenue impact.

monthly_customers	lift_per_customer	est_monthly_lift
100,000	$2.52	$251,896

Methods Notes

- Effects estimated via (i) **unadjusted difference in means** and (ii) **OLS**: `sales ~ design + items` with robust (HC2) SEs.
- We **do not** adjust for **NPS** because it is measured **after** the purchase and may lie on the causal pathway; controlling for post-treatment variables can bias effect estimates.

Alternative Statement & Fault-Tree Notes

Alternative statement: Even if the historical estimate is $< \$1.80$, a full redesign could still achieve $\geq$ **$1.80** because: 1) Full redesign scope may exceed the historical partial change, 2) Launch timing and marketing mix could raise revenue per customer, 3) The redesign might **increase items per cart**; our adjusted estimate holds items fixed and may be conservative for total impact.

Checks guided by the fault tree: - Compare unadjusted vs adjusted effects (cart-size confounding). - Inspect sales vs items by design with NPS overlay (Plot 2). - If proceeding, run a short **confirmatory A/B** with a stop rule requiring **95% CI lower bound $>$ $1.80**.