Problem Set 1 Code

2025-11-13

Task 1

In the following a descriptive analysis of the *detailed_fish_market_data*, regarding Whiting is conducted. At first the required packages and the dataset are loaded.

```
## Load packages and dataset
library(readr)
library(dplyr)
library(ggplot2)
library(tidyr)
library(lmtest)

detailed_data <- read_tsv("../data/detailed_fish_market_data.txt")
```

Data preparation

The focus of the descriptive analysis will be on price and the quantities received and sold throughout the days in April and May 1992. Therefore observations that are *'NA'* for those variables are removed via the package 'dplyr'. Additionally the dataset is filtered for Whiting, as required for Task 1 and arranged by date.

```
# delete those rows that have NA for
# "price","quan", "totr", "tots" and
# filter for whiting (no king)
detailed_whiting <- detailed_data %>%
  filter(!is.na(pric),
         !is.na(totr),
         !is.na(tots),
         type == "w") %>%
  arrange(date)

print(paste("Obs:", nrow(detailed_whiting),
            "- Vars:", ncol(detailed_whiting)))
```

```
## [1] "Obs: 478 - Vars: 17"
```

The cleaned whiting dataset now has 478 observations for 17 variables.

If one inspects the dataset via *'View(detailed_whiting)'* it gets clear, that there are multiple observations for each observed day in the period of April to Mai 1992. Each of them depicts a transaction between a customer and the fish-dealer. That is particularly important for understanding the next step.

Out of the 478 observations there are two that do not fit in. The total quantity that the dealers sold (variable *tots*) is equal for all rows (i.e. transactions) of the same dealer at a given day.

What immediately stands out is that only one dealer is observed on almost all days. For this dealer, multiple transactions are always recorded. Only on two days are two dealers observed, as indicated by multiple distinct observations for 'tots'. However, these additional dealers are unrepresentative, as there is only one observation for each of them. The remaining observations all appear to come from a single, larger dealer. These two observations are therefore deleted using 'dplyr'.

```
## There seem to be two entries in the dataset, where there are two dealer per day.
# Since this is the case only for two out of all days in April and May:
# drop those two observations
detailed_whiting <- detailed_whiting %>%
  # frequency of the same tots value for different days
  group_by(date, dayw, tots) %>%
  mutate(n_same_tots = n()) %>%

  # number of distinct tots days
  group_by(date, dayw) %>%
  mutate(n_tots_values = n_distinct(tots)) %>%
  ungroup() %>%

  # delete rows for which (there are multiple different tots values
  #                                   AND
  #                         for which the tot value only appears once)
  filter(!(n_tots_values > 1 & n_same_tots == 1)) %>%

  # delete rows that are not longer needed
  select(-n_same_tots, -n_tots_values)

## two cases, where > 1 dealer is present
tots_inconsistent <- detailed_whiting |>
  group_by(date, dayw) |>
  mutate(
    n_tots = n_distinct(tots)
  ) |>
  filter(n_tots > 1) |>
  arrange(date, dayw, tots, totr)
```

With this step the data cleaning process is finished.

One part of the descriptive analysis of the Whiting data is the analysis aggregated on the daily level. For this purpose a new dataset "detailed_whiting_daily" is constructed.

```
# dataset for the daily-level
detailed_whiting_daily <- detailed_whiting %>%
  group_by(date) %>%
  summarise(
    avg_pric = mean(pric),
    totr = first(totr),
    tots = first(tots),
    dayw = first(dayw),
    n_trsact = n(),
    strate = first(tots)/first(totr)
  )
```

For each date present in the original dataset the average price (*avg_pric*), *totr*, *tots*, number of transactions (*n_trsact*) and sell-through rate (*strate*) are computed. For each computation that involves "totr" or "tots"

the first value of the day can be used. This is possible due to the previous data cleaning step, as described above. The sell-through rate is the share of available stock that was actually sold. It measures how much of the Whiting that was offered for sale at a given day has been sold. More on that later.

Descriptive analysis

The analysis is split in three parts: At first a analysis of the sales on the daily level, second the distribution of price and at last a analysis based on the hours of the day.

In a code chunk that is not printed in this pdf (*'echo=FALSE'*) the different font sizes and some design variables for the plots are defined.

As a starting point, summary statistics for *totr*, *tots* and *n_trsact* are reported.

```
####
# summary of the daily datset
####
detailed_whiting_daily %>%
  select(totr, tots, n_trsact) %>%
  summary()
```

```
##       totr            tots          n_trsact
##  Min.   :  200   Min.   :  490   Min.   : 4.00
##  1st Qu.: 1990   1st Qu.: 3360   1st Qu.:18.00
##  Median : 6080   Median : 5535   Median :25.00
##  Mean   : 5881   Mean   : 5760   Mean   :25.05
##  3rd Qu.: 7927   3rd Qu.: 7495   3rd Qu.:32.50
##  Max.   :15940   Max.   :15455   Max.   :57.00
```

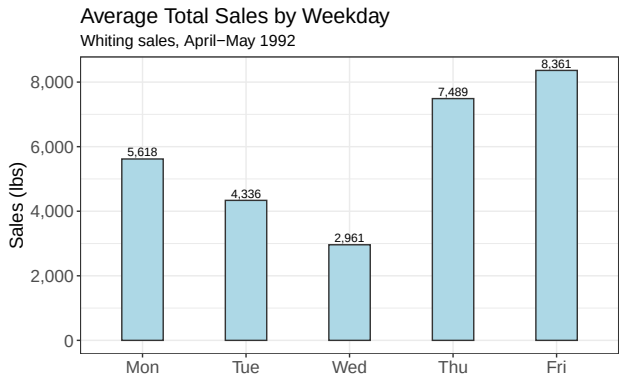
```
detailed_whiting_daily %>%
  select(totr, tots, n_trsact) %>%
  summarise(across(everything(), sd, na.rm = TRUE))
```

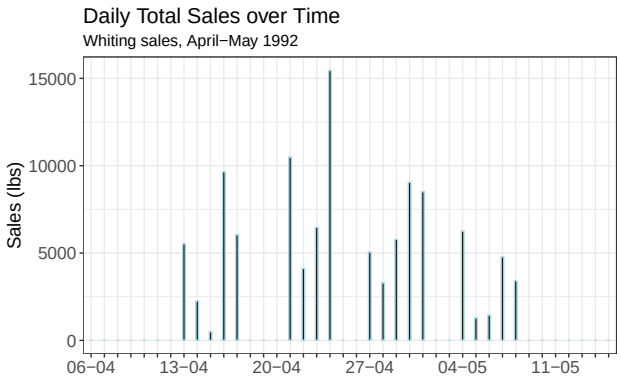
```
## # A tibble: 1 x 3
##    totr  tots n_trsact
##   <dbl> <dbl>    <dbl>
## 1 4381. 3691.     12.3
```

As the summary indicates, total sales and therefore the total received amount of Whiting in lbs inherit a decent amount of variation. The amount of transaction per day also shows a broad range of values, with the minimum of four and a maximum of 57.

To gain a first insight in the properties of the total sales of Whiting in the period of April to May 1992 a bar-chart and a time-series plot are used.

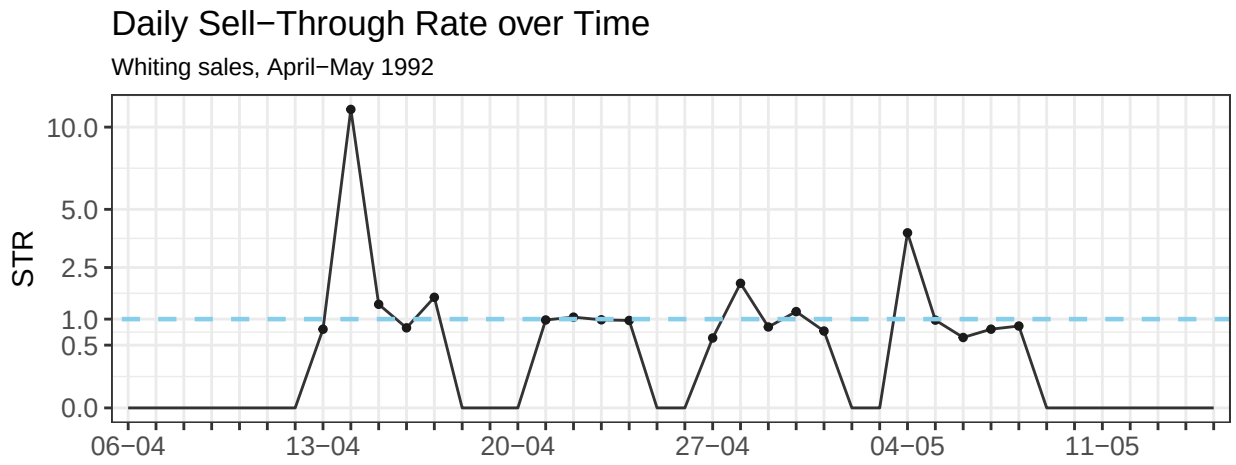
```
plot_average_sales_by_weekday
plot_daily_sales_over_time
```





Plot Beschreibung.

```
plot_daily_str_over_time
```



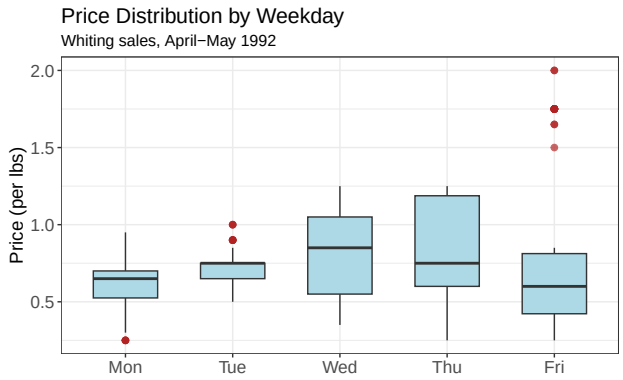
Plot Beschreibung.

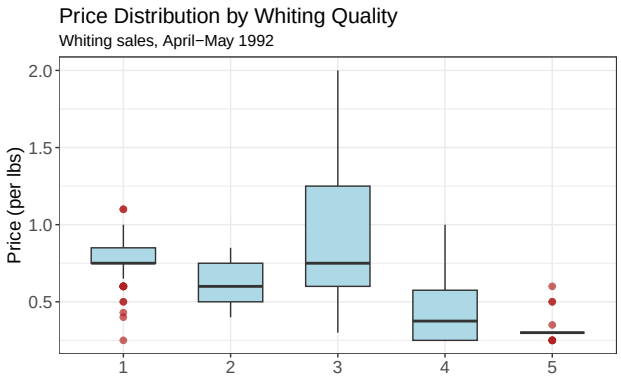
```
####
# correlation between tots (total dailiy sales) and avg_pric (average price)
####
cor(detailed_whiting_daily$avg_pric,detailed_whiting_daily$tots)
```

```
## [1] -0.4236805
```

Korrelation Beschreibung.

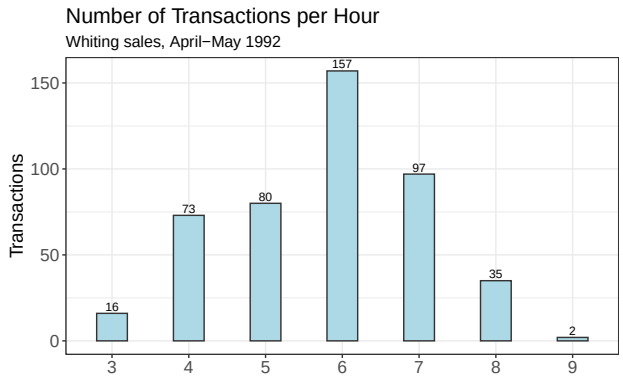
```
plot_price_distr_by_weekday
plot_price_distr_by_quality
```

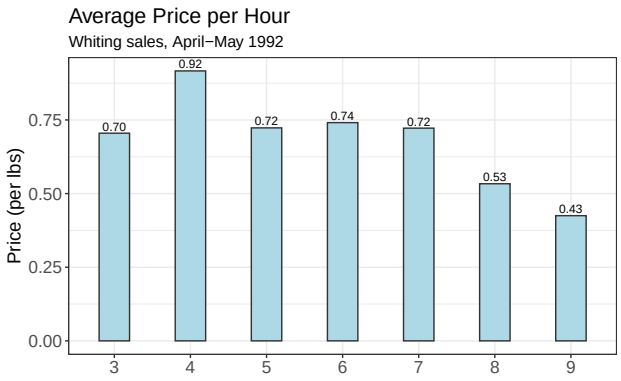




Plot Beschreibung.

```
plot_transactions_per_hour
plot_average_price_per_hour
```





Plot Beschreibung.

Task 3: Moderated Regression

3.1 Hypotheses

In this task, we investigate whether the price sensitivity of individual customers depends on specific contextual characteristics of the transaction. Based on plausibility considerations, the following three hypotheses were developed:

Hypothesis 1 (H1): Moderation by Product Quality (`qual`)

- **Hypothesis:** The quality of the fish has an influence on the price sensitivity of customers.
- **Rationale:** We expect that higher quality (`qual`) leads to *lower* price sensitivity. A high-quality product, which might be sold to expensive restaurants with higher margins, justifies a higher price and makes customers less susceptible to price fluctuations.

Hypothesis 2 (H2): Moderation by Payment Method (`cash`)

- **Hypothesis:** The use of cash versus charge (invoice) influences price sensitivity.
- **Rationale:** Paying with cash may have a higher "emotional value" (or "pain of paying") as the amount paid is immediately visible, rather than just appearing on a bill later. We, therefore, expect that cash transactions lead to *higher* price sensitivity.

Hypothesis 3 (H3): Moderation by Establishment Type (`estb`)

- **Hypothesis:** Price sensitivity depends on the customer's type of establishment.
- **Rationale:** "Fry shops" (`f`) likely operate on lower margins for their final products and thus have a stronger incentive to watch purchase prices than "Stores" (`s`). We, therefore, expect that "fry shops" will exhibit *higher* price sensitivity.

```
#1 data preperation
detailed_data_prep <- detailed_data %>%
  filter(!is.na(pric),
         !is.na(quan),
         type == "w") %>%
  arrange(date) %>%

  # 1.1 standardization
  group_by(cusn) %>%
  mutate(Qty_Dev = quan - mean(quan, na.rm = TRUE)) %>% #target variable
  ungroup() %>%

  # 1.2 mean centering and dummy creation
  mutate(
    price_c = as.numeric(scale(pric, center = TRUE, scale = FALSE)), #main regressor
    quality_c = as.numeric(scale(qual, center = TRUE, scale = FALSE)), # Moderator 1
    cash_dummy = if_else(cash == 'c', 1, 0), # Moderator 2
    estb = as.factor(estb), # Moderator 3
  )
```

- `filter(...)`: This is the first cleaning step.
  - `!is.na(pric), !is.na(quan)`: We remove any rows where either price (`pric`) or quantity (`quan`) are missing. We cannot model a price-demand relationship without a price or a quantity, so these rows are unusable for our model.
  - `type == "w"`: We filter the dataset to only include "Whiting". This ensures our analysis is focused on a single product, as combining different fish types would introduce confounding variables.
- `arrange(date)`: This step sorts the resulting data by date. While not strictly required for this regression, it's good practice to organize time-series data chronologically, which can help in identifying patterns or debugging later.
- `group_by(cusn)`: This crucial step groups the data by customer number (`cusn`). It doesn't change the data itself but tells the following `mutate` function to perform its calculations *within* each customer's group.
- `mutate(Qty_Dev = ...)`: This creates our new target (dependent) variable, `Qty_Dev`.
  - **Justification**: The problem set notes that customers are different sizes. Simply modeling `quan` (quantity) would be misleading, as a large restaurant will always buy more than a small shop, regardless of price.

  - **Action**: By calculating `quan - mean(quan, na.rm = TRUE)`, we create a **within-customer standardized variable**. `Qty_Dev` now represents how much *more* or *less* a customer bought on a specific day compared to their *own* average. This isolates their behavioral deviation and is a much more accurate variable for modeling price sensitivity.

- **`ungroup()`**: This step removes the grouping. It's essential "housekeeping" to ensure that the next `mutate` call performs its calculations (like mean-centering) on the *entire* dataset, not on a per-customer basis.

- **`mutate(...)`**: This final step creates all the predictor variables (regressors) we need for our models.
  - **`price_c = ...`**: This **mean-centers** the `pric` variable, as recommended in the lecture. Mean-centering (`center = TRUE`, `scale = FALSE`) subtracts the overall average price from each transaction's price. This makes the coefficients in our moderated regression models much easier to interpret. Specifically, the main effect of price will now represent the price sensitivity at the *average* level of the moderator.
  - **`quality_c = ...`**: This likewise mean-centers our first moderator, `qual` (quality).
  - **`cash_dummy = ...`**: This converts the categorical `cash` variable into a numeric **dummy variable** for our second hypothesis. The model can interpret "1" (for cash) and "0" (for non-cash), but it cannot interpret the original 'c' and 'h' letters.
  - **`estb = as.factor(estb)`**: This converts the establishment type (`estb`) variable into a **factor**. This tells R that `estb` is a categorical variable. When we include it in the `lm()` function, R will automatically create the necessary dummy variables for each establishment type, allowing us to test our third hypothesis.

```
# additional steps for establishment
detailed_data_prep %>% group_by(estb) %>% count()
```

```
## # A tibble: 9 x 2
## # Groups:   estb [9]
##   estb      n
##   <fct> <int>
## 1 d         2
## 2 f        81
## 3 fd        1
## 4 s       307
## 5 sd        2
## 6 sf       40
## 7 sh        1
## 8 sr        3
## 9 <NA>     41
```

```
detailed_data_perep_estb = detailed_data_prep %>%
  filter(estb %in% c("s", "f", "sf"))
```

This code block performs a crucial diagnostic check and a subsequent filtering action specifically to prepare for testing Hypothesis 3 (moderation by establishment type).

- **`detailed_data_prep %>% group_by(estb) %>% count()`**
  - **What it does**: This line is a **diagnostic check**. It groups the prepared data by the establishment type (`estb`) and counts the number of observations (transactions) for each type.

  - **Why it's done**: Before using a categorical variable as a moderator, we must check its distribution. The output of this count (seen in the previous step) reveals that while some categories like 's' (store) and 'f' (fry shop) have many observations, other categories have very few (e.g., only 1, 2, or 3).

- `detailed_data_perep_estb = ... filter(estb %in% c("s", "f", "sf"))`
  - **What it does**: This line **creates a new, filtered dataset** named `detailed_data_perep_estb`. It includes only the rows where the establishment type is one of the three most common: "s", "f", or "sf".
  - **Why it's done**: This is a **critical step for statistical stability**. Attempting to run a regression or moderation analysis on a categorical level with only 1 or 2 observations is statistically unreliable; the model cannot produce a stable estimate for such a small group. It can lead to model errors or highly misleading results. By filtering down to the well-represented groups, we ensure that our analysis for Hypothesis 3 is robust and that the results are meaningful. This new dataset will be used *only* for the H3 analysis.

```
#2.1 Moderated Model 1
quality_model = lm(Qty_Dev ~ price_c + quality_c, data = detailed_data_prep)
quality_model_moderation = lm(Qty_Dev ~ price_c*quality_c, data = detailed_data_prep)
summary(quality_model)
```

```
##
## Call:
## lm(formula = Qty_Dev ~ price_c + quality_c, data = detailed_data_prep)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -458.26  -23.08   -1.96   16.28  963.39
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.6419     5.0844   0.126    0.900
## price_c     -19.7349    15.4777  -1.275    0.203
## quality_c    -0.8560     4.7298  -0.181    0.856
##
## Residual standard error: 108.7 on 454 degrees of freedom
##   (21 Beobachtungen als fehlend gelöscht)
## Multiple R-squared:  0.003572,   Adjusted R-squared:  -0.0008177
## F-statistic: 0.8137 on 2 and 454 DF,  p-value: 0.4439
```

```
summary(quality_model_moderation)
```

```
##
## Call:
## lm(formula = Qty_Dev ~ price_c * quality_c, data = detailed_data_prep)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -456.67  -23.21   -2.21   16.08  963.27
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)         0.0707     5.2992   0.013    0.989
## price_c           -15.1822    19.4567  -0.780    0.436
## quality_c          -1.9473     5.5113  -0.353    0.724
## price_c:quality_c  -8.9002    23.0116  -0.387    0.699
##
## Residual standard error: 108.8 on 453 degrees of freedom
##   (21 Beobachtungen als fehlend gelöscht)
## Multiple R-squared:  0.003901,   Adjusted R-squared:  -0.002696
## F-statistic: 0.5913 on 3 and 453 DF,  p-value: 0.6209
```

```
anova(quality_model, quality_model_moderation)
```

```
## Analysis of Variance Table
##
## Model 1: Qty_Dev ~ price_c + quality_c
## Model 2: Qty_Dev ~ price_c * quality_c
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1    454 5363293
## 2    453 5361522  1    1770.5 0.1496 0.6991
```

```
lrtest(quality_model, quality_model_moderation)
```

```
## Likelihood ratio test
##
## Model 1: Qty_Dev ~ price_c + quality_c
## Model 2: Qty_Dev ~ price_c * quality_c
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   4 -2789.6
## 2   5 -2789.5  1 0.1509     0.6977
```

```
#2.2 Moderated Model 2
cash_model = lm(Qty_Dev ~ price_c+cash_dummy, data=detailed_data_prep)
cash_model_moderation = lm(Qty_Dev ~ price_c*cash_dummy, data=detailed_data_prep)
summary(cash_model)
```

```
##
## Call:
## lm(formula = Qty_Dev ~ price_c + cash_dummy, data = detailed_data_prep)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -455.49  -23.55   -0.94   14.87  965.08
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.9983     9.1851   0.109    0.913
## price_c     -18.2086    15.5898  -1.168    0.243
## cash_dummy   -1.5193    10.9672  -0.139    0.890
##
## Residual standard error: 108.2 on 472 degrees of freedom
##   (3 Beobachtungen als fehlend gelöscht)
## Multiple R-squared:  0.002882,   Adjusted R-squared:  -0.001343
## F-statistic: 0.6821 on 2 and 472 DF,  p-value: 0.5061
```

```r
summary(cash_model_moderation)
```

```
##
## Call:
## lm(formula = Qty_Dev ~ price_c * cash_dummy, data = detailed_data_prep)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -456.53  -23.23   -0.50   16.07  965.35
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -0.3933     9.3002  -0.042    0.966
## price_c              7.1742    30.7550   0.233    0.816
## cash_dummy          -0.3788    11.0327  -0.034    0.973
## price_c:cash_dummy -34.1625    35.6796  -0.957    0.339
##
## Residual standard error: 108.2 on 471 degrees of freedom
##   (3 Beobachtungen als fehlend gelöscht)
## Multiple R-squared:  0.004819,   Adjusted R-squared:  -0.00152
## F-statistic: 0.7602 on 3 and 471 DF,  p-value: 0.5168
```

```r
anova(cash_model, cash_model_moderation)
```

```
## Analysis of Variance Table
##
## Model 1: Qty_Dev ~ price_c + cash_dummy
## Model 2: Qty_Dev ~ price_c * cash_dummy
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1    472 5526664
## 2    471 5515928  1     10736 0.9168 0.3388
```

```r
lrtest(cash_model, cash_model_moderation)
```

```
## Likelihood ratio test
##
## Model 1: Qty_Dev ~ price_c + cash_dummy
## Model 2: Qty_Dev ~ price_c * cash_dummy
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   4 -2897.4
## 2   5 -2897.0  1 0.9237     0.3365
```

```r
#2.3 Moderated Model 3
estb_model = lm(Qty_Dev ~ price_c + estb, data = detailed_data_perep_estb)
estb_model_moderation = lm(Qty_Dev ~ price_c * estb, data = detailed_data_perep_estb)
summary(estb_model)
```

```
##
## Call:
## lm(formula = Qty_Dev ~ price_c + estb, data = detailed_data_perep_estb)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -457.01  -24.83   -0.38   17.31  966.03
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.465     12.577  -0.116    0.907
## price_c      -21.989     16.571  -1.327    0.185
## estbs          1.905     14.158   0.135    0.893
## estbsf         3.234     21.925   0.148    0.883
##
## Residual standard error: 112.7 on 424 degrees of freedom
## Multiple R-squared:  0.004136,   Adjusted R-squared:  -0.00291
## F-statistic: 0.5869 on 3 and 424 DF,  p-value: 0.6238
```

```
summary(estb_model_moderation)
```

```
##
## Call:
## lm(formula = Qty_Dev ~ price_c * estb, data = detailed_data_perep_estb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -457.49  -24.14   -1.02   15.34  963.48
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.9751    12.8901   0.076    0.940
## price_c         13.8828    43.7572   0.317    0.751
## estbs           -0.4756    14.4158  -0.033    0.974
## estbsf           2.5946    22.3083   0.116    0.907
## price_c:estbs  -38.7734    47.9334  -0.809    0.419
## price_c:estbsf -58.2433    62.6070  -0.930    0.353
##
## Residual standard error: 112.9 on 422 degrees of freedom
## Multiple R-squared:  0.006358,   Adjusted R-squared:  -0.005415
## F-statistic:  0.54 on 5 and 422 DF,  p-value: 0.746
```

```
anova(estb_model, estb_model_moderation)
```

```
## Analysis of Variance Table
##
## Model 1: Qty_Dev ~ price_c + estb
## Model 2: Qty_Dev ~ price_c * estb
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1    424 5388709
## 2    422 5376685  2     12023 0.4718 0.6242
```

```
lrtest(estb_model, estb_model_moderation)
```

```
## Likelihood ratio test
##
```

```
## Model 1: Qty_Dev ~ price_c + estb
## Model 2: Qty_Dev ~ price_c * estb
##   #Df  LogLik Df Chisq Pr(>Chisq)
## 1   5 -2627.6
## 2   7 -2627.1  2 0.956       0.62
```