Curbside Database Statistical Analysis

Tyler Campbell

2025-07-21

Introduction

Welcome to my next project, a statistical analysis of my database using R! In this analysis, I will show basic visualizations and summary statistics of different entities in my database. I will also run a couple of regressions to test how different entities significantly impact each other. I hope that you enjoy!

Basic Visuals and Summary Stats

```r
library(DBI)
```

```
## Warning: package 'DBI' was built under R version 4.4.3
```

```r
library(RMySQL)
```

```
## Warning: package 'RMySQL' was built under R version 4.4.3
```

```r
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.4.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.3
```

```r
library(broom)
```

```
## Warning: package 'broom' was built under R version 4.4.3
```

```r
con <- dbConnect(
  RMySQL::MySQL(),
  dbname = "heb_curbside",
  host = "localhost",
  port = 3306,
  user = "root",
  password = "Gobaylor#1"
)
```

```r
orders <- dbReadTable(con, "`heb_curbside`.`order`")
head(orders)
```

```
##   order_id order_date cust_id curbie_id order_time
## 1        1 2024-05-03      88        38   20:56:19
## 2        2 2025-05-19      80        28   22:13:08
## 3        3 2023-09-08      86        47   11:50:41
## 4        4 2024-06-06      21         6   12:02:44
## 5        5 2024-02-15       1        31   09:02:05
## 6        6 2023-09-14      10        45   10:35:01
```

```r
order_revenue <- dbReadTable(con, "`heb_curbside`.`order_revenue`")
```

```
## Warning in dbSendQuery(conn, statement, ...): Decimal MySQL column 1 imported
## as numeric
```

```r
labor <- dbReadTable(con, "`heb_curbside`.`labor`")
```

```
## Warning in dbSendQuery(conn, statement, ...): Decimal MySQL column 4 imported
## as numeric
```

```
## Warning in dbSendQuery(conn, statement, ...): Decimal MySQL column 5 imported
## as numeric
```

```r
survey <- dbReadTable(con, "`heb_curbside`.`customer_survey`")
shopper_stats <- dbReadTable(con, "`heb_curbside`.`shopper_stats`")
```

```
## Warning in dbSendQuery(conn, statement, ...): Decimal MySQL column 3 imported
## as numeric
```

```r
curbie_stats <- dbReadTable(con, "`heb_curbside`.`curbie_stats`")
```

```
## Warning in dbSendQuery(conn, statement, ...): Decimal MySQL column 1 imported
## as numeric
```

```r
customer <- dbReadTable(con, "`heb_curbside`.`customer`")
order_personal_shopper <- dbReadTable(con, "`heb_curbside`.`order_personal_shopper`")
personal_shopper <- dbReadTable(con, "`heb_curbside`.`personal_shopper`")
curbie <- dbReadTable(con, "`heb_curbside`.`curbie`")
order_product <- dbReadTable(con, "`heb_curbside`.`order_product`")
```

```r
str(orders)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ order_id  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ order_date: chr  "2024-05-03" "2025-05-19" "2023-09-08" "2024-06-06" ...
##  $ cust_id   : int  88 80 86 21 1 10 27 17 58 85 ...
##  $ curbie_id : int  38 28 47 6 31 45 7 3 27 7 ...
##  $ order_time: chr  "20:56:19" "22:13:08" "11:50:41" "12:02:44" ...
```

```r
str(order_revenue)
```

```
## 'data.frame':    150 obs. of  2 variables:
##  $ order_id   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ order_price: num  193.4 101.5 77 158.4 15.4 ...
```

```r
str(labor)
```

```
## 'data.frame':    150 obs. of  6 variables:
##  $ labor_id   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ week       : chr  "2024-06-17" "2024-07-22" "2024-06-17" "2024-06-17" ...
##  $ shopper_id : int  1 48 19 47 32 25 2 45 49 5 ...
##  $ curbie_id  : int  48 25 8 18 22 16 18 36 25 23 ...
##  $ labor_hours: num  13.5 18.8 14.2 29.2 27.3 ...
##  $ wages      : num  21.5 16.7 14.1 22 14.7 ...
```

```r
str(survey)
```

```
## 'data.frame':    100 obs. of  4 variables:
##  $ survey_id         : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ cust_id           : int  17 2 53 92 89 100 26 50 36 11 ...
##  $ one_to_five_rating: int  4 5 1 1 2 5 3 5 3 2 ...
##  $ comments          : chr  "Will factor thus sure treat guess treat." "Exactly memory level inside
```

```r
str(shopper_stats)
```

```
## 'data.frame':    50 obs. of  5 variables:
##  $ shopper_id     : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ orders_shopped : int  51 201 273 210 57 48 190 224 218 224 ...
##  $ units_shopped  : int  1948 743 2088 2637 1575 1918 2269 2526 2468 1632 ...
##  $ uph            : num  51.4 25.9 61 70.1 31.3 ...
##  $ subs_and_shorts: int  5 7 19 11 6 19 8 8 7 9 ...
```

```r
str(curbie_stats)
```

```
## 'data.frame':    50 obs. of  4 variables:
##  $ curbie_id         : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ retrieval_time_avg: num  10.62 4.85 6.24 6.96 11.53 ...
##  $ orders_retrieved  : int  451 367 131 355 257 371 286 464 450 268 ...
##  $ left_behinds      : int  1 1 3 1 10 5 0 1 1 0 ...
```

```r
orders_full <- orders %>%
  left_join(order_revenue, by = "order_id") %>%
  left_join(customer, by = "cust_id") %>%
  left_join(order_personal_shopper, by = "order_id") %>%
  left_join(personal_shopper, by = "shopper_id") %>%
  left_join(curbie, by = "curbie_id") %>%
  left_join(curbie_stats, by = "curbie_id") %>%
  left_join(shopper_stats, by = "shopper_id") %>%
  left_join(survey, by = "cust_id") %>%
  left_join(labor, by = c("shopper_id", "curbie_id"))
```

```
## Warning in left_join(., survey, by = "cust_id"): Detected an unexpected many-to-many relationship bet
## i Row 15 of `x` matches multiple rows in `y`.
## i Row 85 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```r
# summary statistics for order revenue and wages

library(knitr)
library(dplyr)
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':
##
##     group_rows
```

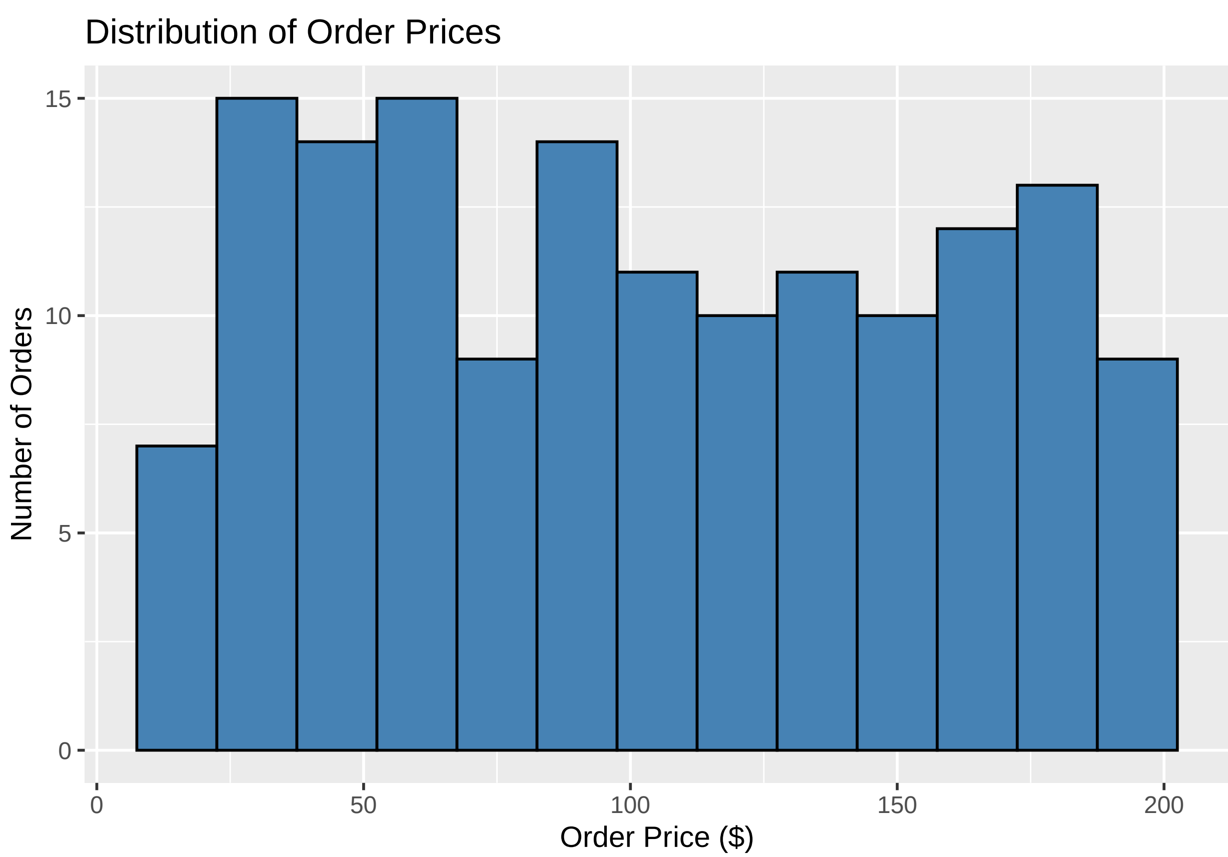
```r
orders_full %>%
  summarise(
    avg_order_price = mean(order_price, na.rm = TRUE),
    min_order_price = min(order_price, na.rm = TRUE),
    max_order_price = max(order_price, na.rm = TRUE),
    avg_wages = mean(wages, na.rm = TRUE),
    total_labor_hours = sum(labor_hours, na.rm = TRUE)
  ) %>%
  kable(digits = 2, caption = "Summary Statistics from Orders Data") %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed"),
                full_width = F,
                position = "center")
```

Table 1: Summary Statistics from Orders Data

avg_order_price	min_order_price	max_order_price	avg_wages	total_labor_hours
101.76	15.4	198.29	17.12	682.74

This is just some basic summary statistics. Everything seems prettu reasonable except for the min and max order prices. Realistically, there will be order prices that are more than 200 dollars and less than 15 dollars, but this is just synthetic data.

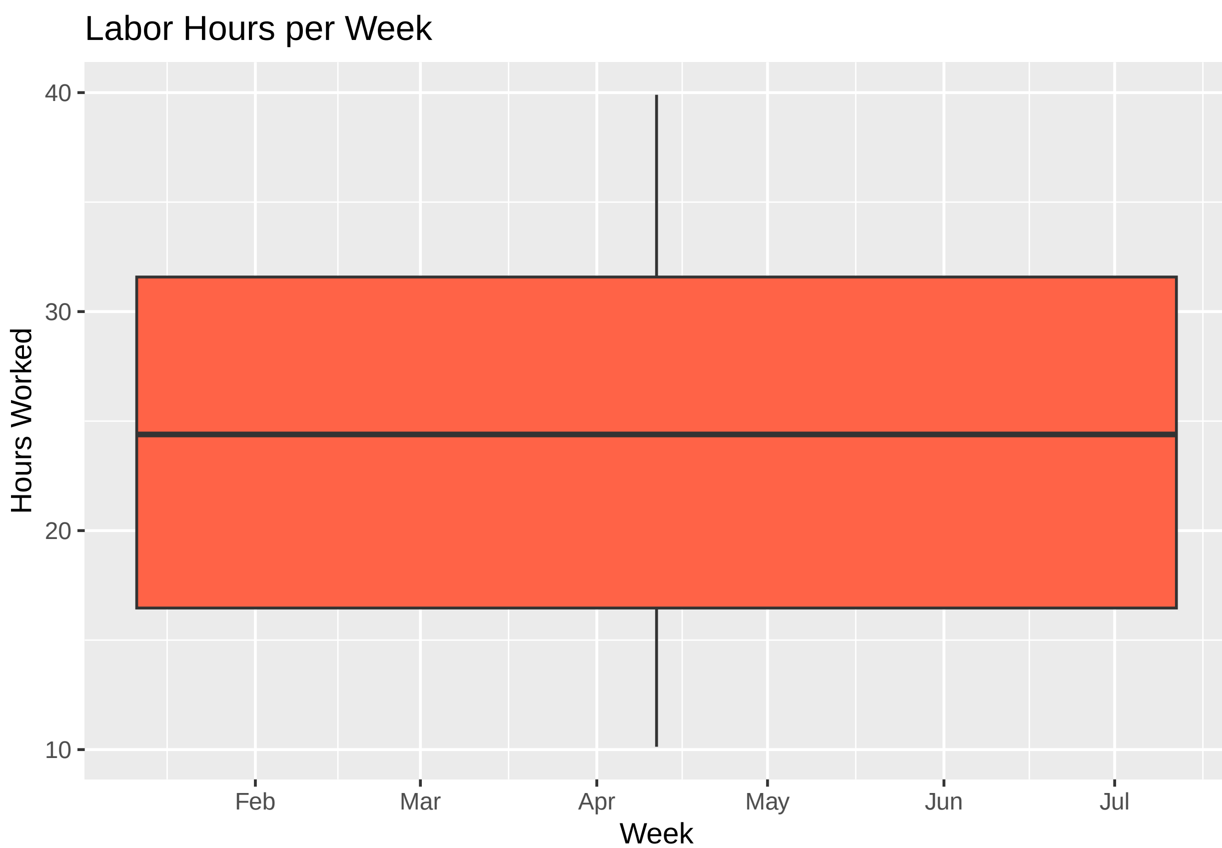
```r
# order price distribution
library(ggplot2)

ggplot(order_revenue, aes(x = order_price)) +
  geom_histogram(binwidth = 15, fill = "steelblue", color = "black") +
  labs(
    title = "Distribution of Order Prices",
    x = "Order Price ($)",
    y = "Number of Orders"
  )
```



This is a bar graph of the price distribution. The majority of the orders are in the 25 and 60 dollar price range. This follows more of a uniform distribution and is not normally distributed. My intuition would think that it would either be more skewed to the right because less people probably purchase highly-priced orders.

```r
#weekly labor
labor$week <- as.Date(labor$week)

ggplot(labor, aes(x = week, y = labor_hours)) +
  geom_boxplot(fill = "tomato") +
  labs(
    title = "Labor Hours per Week",
    x = "Week",
    y = "Hours Worked"
  )
```

```
## Warning: Continuous x aesthetic
## i did you forget `aes(group = ...)`?
```



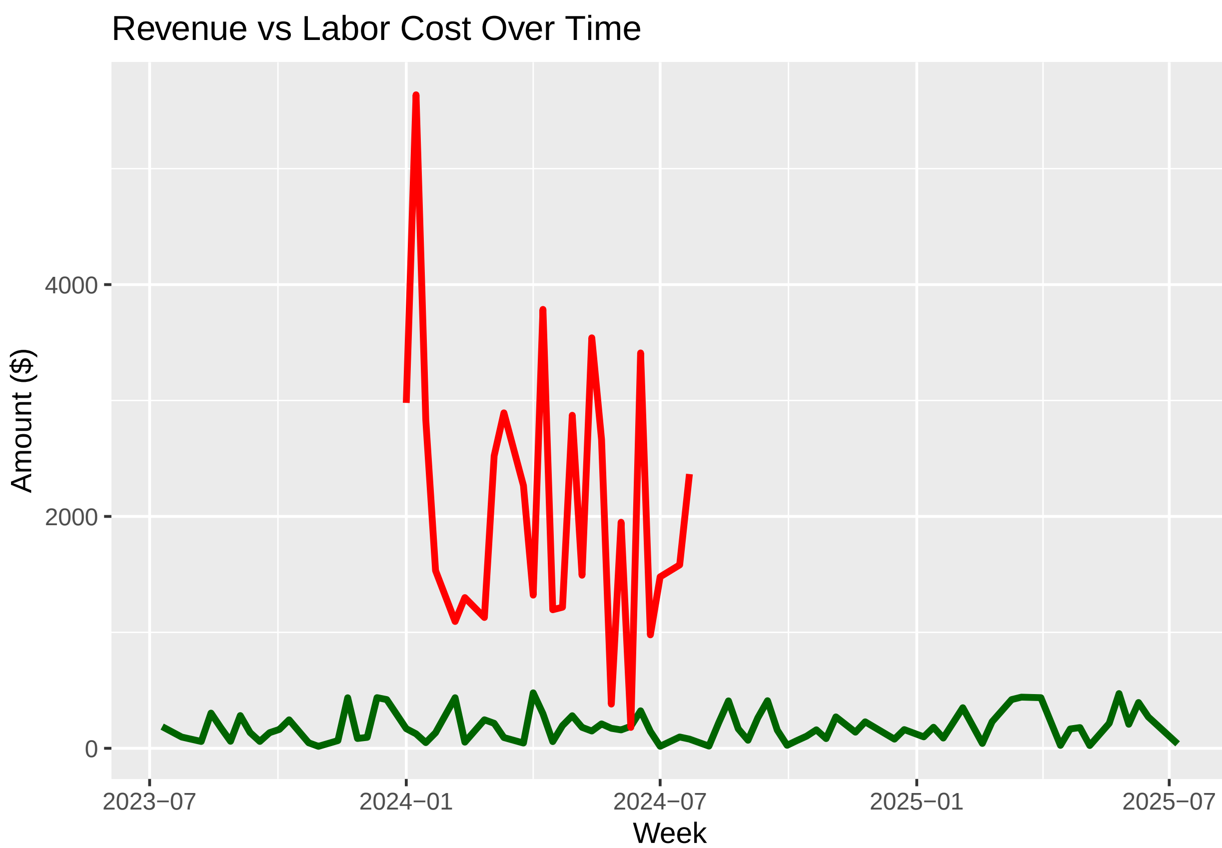
This is a boxplot that shows the number of hours worked by partners during different weeks. It stays the same throughout the year with the majority of hours worked being between 17 and 33 primarily. Is this realistic? Probably not because the number of hours worked might change during different times of the year. For example. At the H-E-B that I work at in Lubbock, a lot of the partners are college students, so they will go home during the summer which causes them to work less.

```r
# Merge labor and order_revenue
orders_with_week <- orders %>%
  mutate(week = as.Date(cut(as.Date(order_date), "week"))) %>%
  left_join(order_revenue, by = "order_id")
```

```r
revenue_by_week <- orders_with_week %>%
  group_by(week) %>%
  summarise(total_revenue = sum(order_price))

labor_by_week <- labor %>%
  group_by(week) %>%
  summarise(total_labor_cost = sum(labor_hours * wages))

profit_data <- left_join(revenue_by_week, labor_by_week, by = "week") %>%
  mutate(profit = total_revenue - total_labor_cost)

ggplot(profit_data, aes(x = week)) +
  geom_line(aes(y = total_revenue), color = "darkgreen", size = 1.2) +
  geom_line(aes(y = total_labor_cost), color = "red", size = 1.2) +
  labs(
    title = "Revenue vs Labor Cost Over Time",
    x = "Week",
    y = "Amount ($)"
  )
```

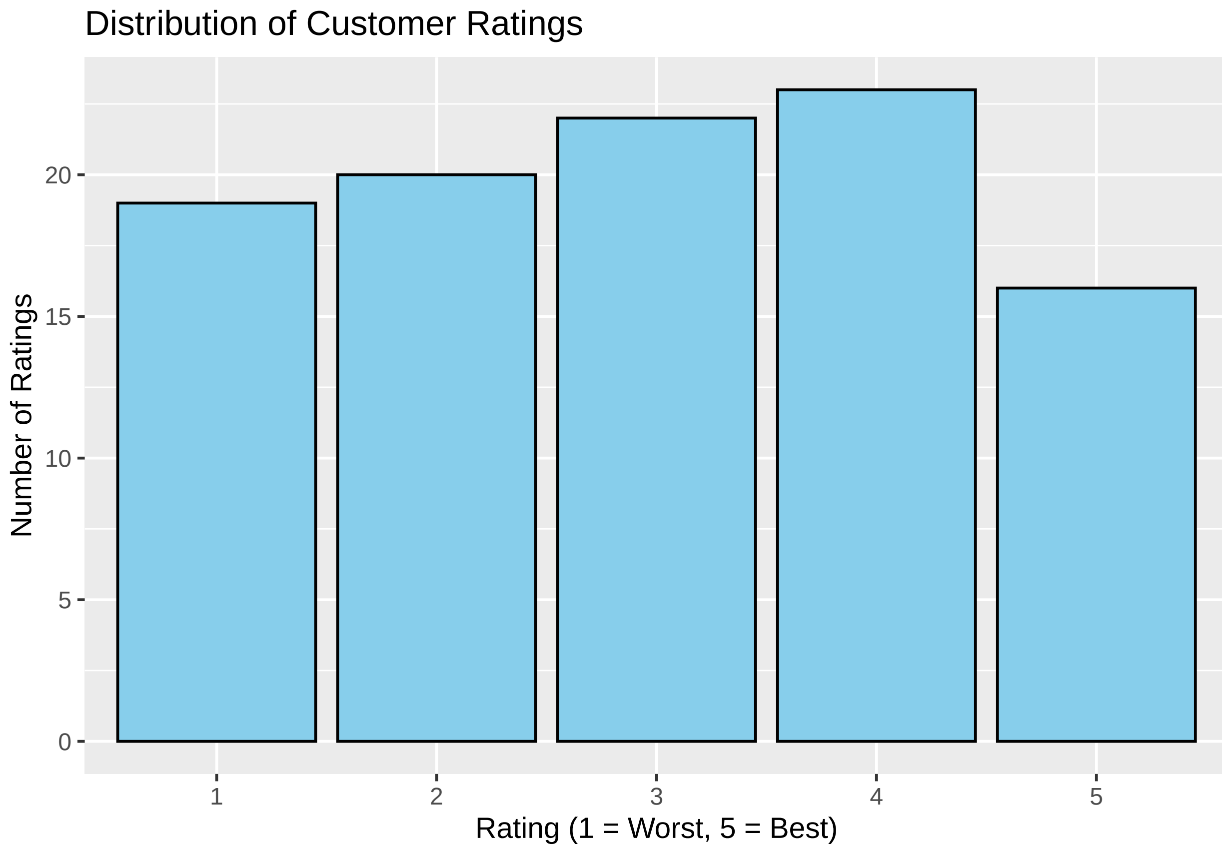
```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: Removed 57 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



This shows the comparison of labor and revenue over time. As you can see, this H-E-B is NOT doing too hot. This is obviously not realistic of what real data would show.

```
ggplot(survey, aes(x = factor(one_to_five_rating))) +
  geom_bar(fill = "skyblue", color = "black") +
  labs(
    title = "Distribution of Customer Ratings",
    x = "Rating (1 = Worst, 5 = Best)",
    y = "Number of Ratings"
  )
```



This shows how many ratings each score got on the 1-5 survey. 4 has the most ratings while 5 has the least. This is probably not very realistic because this average will be close to 3, and it is typically closer to 5.

Linear Regressions

In this section, I will be running regression analysis in order to test the correlation between two different entities and the strength of the models. There will be two tests that I will be running. The first test is whether or not number of items ordered affects the total revenue. The second test is whether or not the order revenue (or order size) generally affects the ratings that customers give on their surveys. Hope you enjoy!

order revenue vs number of items ordered

```r
# Summarize quantity per order
order_items <- order_product %>%
  group_by(order_id) %>%
  summarise(total_items = sum(quantity))

# Join with order revenue
order_qty_rev <- order_items %>%
  left_join(order_revenue, by = "order_id") %>%
  filter(!is.na(order_price))
```

```r
model1 <- lm(order_price ~ total_items, data = order_qty_rev)
summary(model1)
```

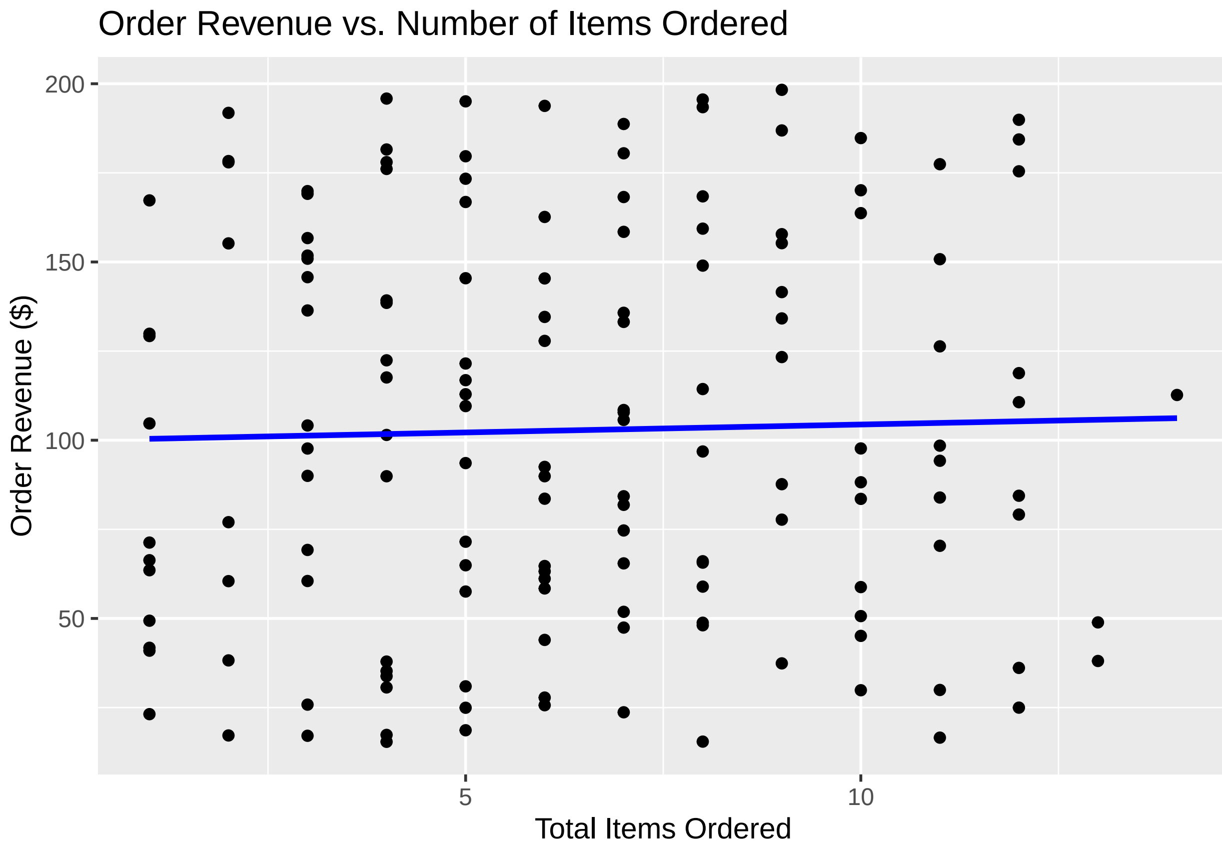
```
##
## Call:
## lm(formula = order_price ~ total_items, data = order_qty_rev)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -88.282 -44.617  -6.528  50.286  94.330
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  99.9490     9.8285  10.169   <2e-16 ***
## total_items   0.4457     1.3709   0.325    0.746
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 55.13 on 148 degrees of freedom
## Multiple R-squared:  0.0007137,  Adjusted R-squared:  -0.006038
## F-statistic: 0.1057 on 1 and 148 DF,  p-value: 0.7455
```

Let's look at the summary statistics for this linear regression. The coefficient for the intercept is 99 which means that when 0 items are ordered, the model predicts a revenue of approximately 99. This provides little statistical significance, and it serves primarily as a baseline. The slope for the line is .45. This means that for every 1 item ordered, the model predicts an increase of about 45 cents in revenue which is a very small amount statistically.

Moving on to the statistical significance section. The p-value of this test is .745. This means that there is very little significance between number of items ordered and order revenue. This means we fail to reject the null. The R-squared is .0007. This means that less than 1 percent of the variation in order revenue is explained by the number of items purchased There is not enough evidence to say that number of items ordered affects the revenue. This, however, seems contrary to my initial intuition. Because this is synthetic data and far from realistic, take this analysis with a grain of salt.

```r
ggplot(order_qty_rev, aes(x = total_items, y = order_price)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(title = "Order Revenue vs. Number of Items Ordered",
       x = "Total Items Ordered",
       y = "Order Revenue ($)")
```

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



Looking at this plot, the line is very flat with a very slight upwards path. This supports what I have previously said with that the revenue does not visibly increase with the number of items ordered. The slight upwards path might indicate there is barely any positive correlation, but it is definitely not enough. The data points are scattered all over the place vertically. This just helps reinforce the lack of correlation.

Customer Rating vs Order Revenue

```r
# Create dataset with customer ratings and order revenue
order_rating_rev <- survey %>%
  left_join(orders, by = "cust_id") %>%
  left_join(order_revenue, by = "order_id") %>%
  select(order_id, cust_id, one_to_five_rating, order_price) %>%
  filter(!is.na(one_to_five_rating), !is.na(order_price))
```

```
## Warning in left_join(., orders, by = "cust_id"): Detected an unexpected many-to-many relationship be
## i Row 1 of `x` matches multiple rows in `y`.
## i Row 96 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```r
# Run the regression
rating_model <- lm(order_price ~ one_to_five_rating, data = order_rating_rev)
summary(rating_model)
```

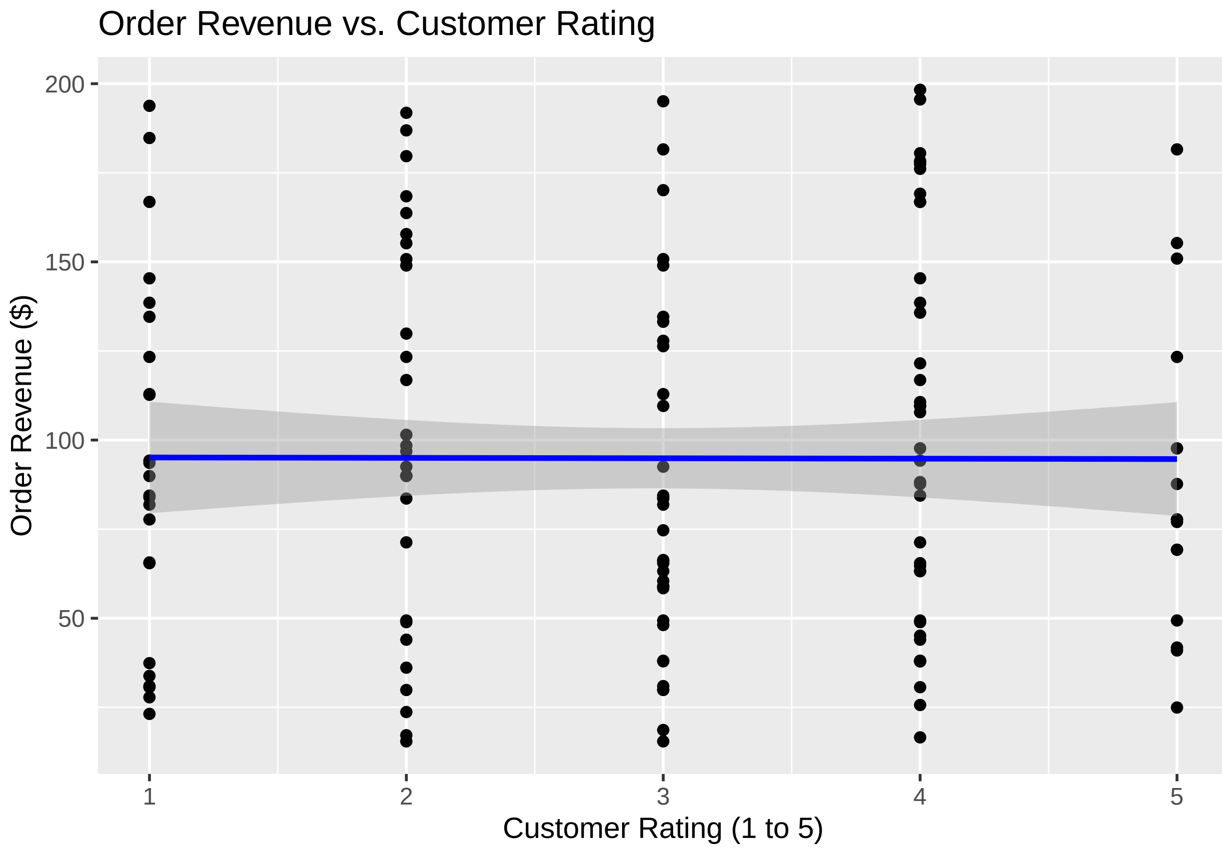
```
##
## Call:
## lm(formula = order_price ~ one_to_five_rating, data = order_rating_rev)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -79.552 -45.435  -6.781  39.532 103.513
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         95.2271    10.9191   8.721 6.64e-15 ***
## one_to_five_rating  -0.1124     3.3809  -0.033    0.974
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 51.26 on 142 degrees of freedom
## Multiple R-squared:  7.789e-06,  Adjusted R-squared:  -0.007034
## F-statistic: 0.001106 on 1 and 142 DF,  p-value: 0.9735
```

Looking at the summary statistics for this next regression, the intercept is 95 which is the prediction of order revenue when the rating is 0. This is not significant since the survey is on a 1-5 scale. The rating slope is -.11 which means that as customer ratings increase, order revenue slightly decreases. However this is not statistically significant either.

The p-value is .97 which is way above .05. This means that we reject the null hypothesis. The rating does not have a significant affect on the revenue. The R-squared is essentially 0. This means that the ratings explain virtually 0% of the variation in order revenue.

```
# Scatterplot with regression line
ggplot(order_rating_rev, aes(x = one_to_five_rating, y = order_price)) +
  geom_point() +
  geom_smooth(method = "lm", color = "blue") +
  labs(
    title = "Order Revenue vs. Customer Rating",
    x = "Customer Rating (1 to 5)",
    y = "Order Revenue ($)"
  )
```

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



The plot is the same as the previous regression that I did. The flat line and scattered plots further prove the lack of relation between survey ratings and order revenue.

Conclusion

In conclusion, in this statistical analysis that I conducted on my H-E-B curbside database, I was able to visualize multiple different entity relationships, such as the distribution of order prices, the comparison of revenue and the number of labor hours among the workers.

Using synthetic data has skewed my intuition solely because of how unrealistic this data is. It is comical to see how poorly the store is doing when looking at revenue and labor costs. However, this has made the project more fun!

Looking at the last portion, the linear regressions, this was the most boring part solely because there was not significant correlation between any of the regressions that I ran.

All in all, I really enjoyed this project using R, and I hope you have gathered some valuable insights of my intuition!