# Exploratory Data Analysis

Andy Wang 305609128

2024-07-13

## Overview

### Files

1. `customer_info_train.csv`, `customer_info_test.csv` - contains customer demographic info
2. `amazon_order_details_train.csv`, `amazon_order_details_test.csv` - contains each order's specific info
3. `train.csv`, `test.csv` - primary data file used for modelling

### Imports

```
library(dplyr)
```

```
##
## 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
```

```
library(ggplot2)
library(tidymodels)
```

```
## -- Attaching packages ------------------------------------- tidymodels 1.2.0 --
```

```
## v broom        1.0.6      v rsample      1.2.1
## v dials        1.2.1      v tibble       3.2.1
## v infer        1.0.7      v tidyr        1.3.1
## v modeldata    1.4.0      v tune         1.2.1
## v parsnip      1.2.1      v workflows    1.1.4
## v purrr        1.0.2      v workflowsets 1.1.0
## v recipes      1.0.10     v yardstick    1.3.1
```

```
## -- Conflicts ---------------------------------------- tidymodels_conflicts() --
## x purrr::discard() masks scales::discard()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x recipes::step()  masks stats::step()
## * Search for functions across packages at https://www.tidymodels.org/find/
```

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

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

## File 1: `train.csv`

```
train_path <- "../ucla-stats-101-c-2024-su-regression/train.csv"
train <- read.csv(train_path)
#train <- train %>% select(-order_totals)
head(train)
```

```
##   q_demos_state year month order_totals log_total count count_female count_male
## 1       Alabama 2018     1      1774.01  3.248956    53           49          4
## 2       Alabama 2018     2      2015.14  3.304305    49           47          2
## 3       Alabama 2018     3      1689.01  3.227632    51           48          3
## 4       Alabama 2018     4      3303.88  3.519024    47           42          5
## 5       Alabama 2018     5      1922.96  3.283970    43           41          2
## 6       Alabama 2018     6      2497.19  3.397452    62           57          5
##   count_less5 count_5to10 count_over10 count_hh1 count_hh2 count_hh3 count_hh4
## 1          17          33            3        15         9        17        12
## 2          19          26            4        15         4        18        12
## 3          18          30            3        15        10        15        11
## 4          19          27            1         9         6        17        15
## 5          11          30            2        10         7        16        10
## 6          22          39            1        15         8        23        16
##   count_howmany1 count_howmany2 count_howmany3 count_howmany4 count_1824
## 1             41              7              1              4          1
## 2             36              5              1              7          0
```

```
## 3             40             6             0             5          0
## 4             39             3             0             5          0
## 5             31             5             0             7          0
## 6             44             8             5             5          1
##   count_2534 count_3544 count_4554 count_5564 count_65up count_und25k
## 1         10         10         29          3          0            1
## 2          6          7         29          7          0            4
## 3          8         10         26          7          0            4
## 4         11          9         17         10          0            5
## 5          6          9         23          5          0            1
## 6          7         13         34          7          0            2
##   count_2549k count_5074k count_7599k count_100149k count_150kup count_lessHS
## 1          23           8           0            19            2            0
## 2          18           3           2            17            5            0
## 3          17           7           0            19            4            0
## 4          14          10           0            12            6            0
## 5          11           7           1            17            6            0
## 6          17          10           2            26            5            0
##   count_HS count_B count_G
## 1        8      29      16
## 2       13      24      12
## 3        8      24      19
## 4       18      15      14
## 5        5      22      16
## 6       14      28      20
```

## Variables

index variables - `q_demos_state` - US state where orders were placed - `year` - year where orders were placed - `month` - month where orders were placed

user info - `count` - total number of orders placed - `count_female` - number of orders placed by those who responded female to survey question - `count_male` - number of orders placed by those who responded male to survey question - `count_less5` - number of orders placed by those who responded "place less than 5 orders per month on Amazon" - `count_5to10` - "5 to 10 orders per month on Amazon" - `count_over10` - "over 10 orders per month on Amazon"

household size variables - `count_hh1` - household size = 1 - `count_hh2` - household size = 2 - `count_hh3` - household size = 3 - `count_hh4` - household size = 4+

account sharing variables - `count_howmany1` - how many people in the household use/share the amazon account = 1 - `count_howmany2` - how many people in the household use/share the amazon account = 2 - `count_howmany3` - how many people in the household use/share the amazon account = 3 - `count_howmany4` - how many people in the household use/share the amazon account = 4

customer age variables - `count_1824` - age 18-24 - `count_2534`
- `count_3544`
- `count_4554`
- `count_5564`
- `count_65up`

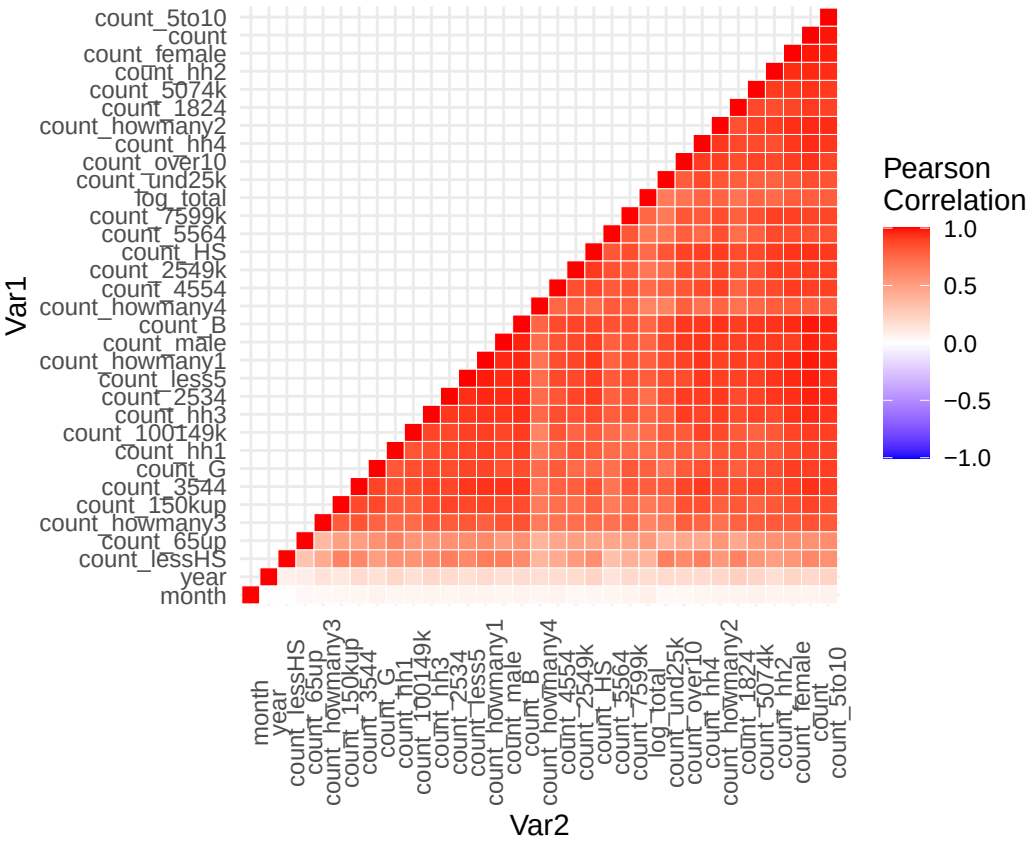
customer income variables - `count_und25k` - under 25k - `count_2549k` - 25k to 49k - `count_5074k`
- `count_7599k`
- `count_100149k` - 100k to 149k - `count_150kup` - over 150k

customer education variables - `count_lessHS` - less than HS - `count_HS` - HS diploma - `count_B` - Bachelor's degree - `count_G` - graduate / professional degree

target variable - `order_totals` - total order price - `log_total` - log of total order price

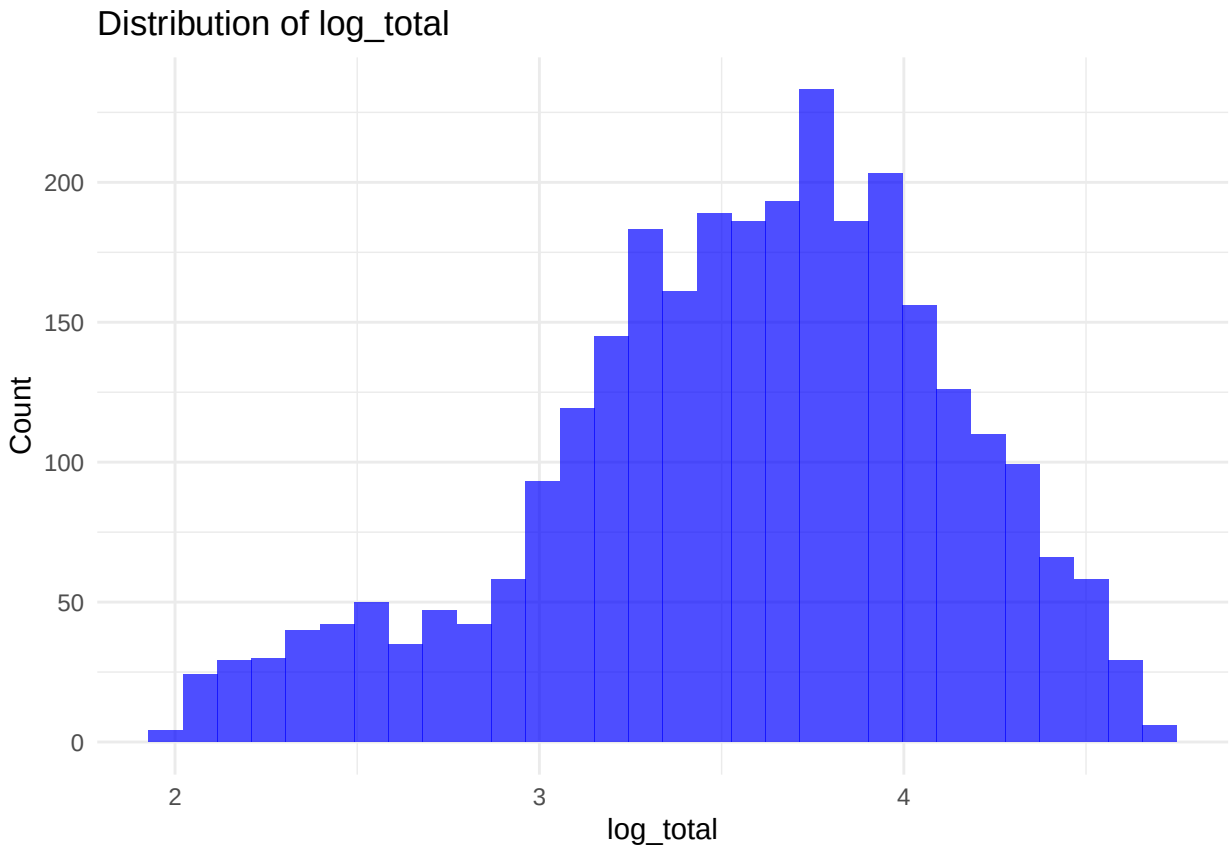
## Correlation Matrix

```
correlation_mat <- train %>%
  select(-order_totals) %>%
  select_if(is.numeric) %>%
  cor()

melted_cormat <- melt(correlation_mat)

# Get upper triangle of the correlation matrix
get_upper_tri <- function(cormat){
  cormat[lower.tri(cormat)]<- NA
  return(cormat)
}

# Use correlation between variables as distance
reorder_cormat <- function(cormat){
  dd <- as.dist((1-cormat)/2)
  hc <- hclust(dd)
  cormat <-cormat[hc$order, hc$order]
}

# Reorder the correlation matrix
cormat <- reorder_cormat(correlation_mat)
upper_tri <- get_upper_tri(cormat)

# Melt the correlation matrix
melted_cormat <- melt(upper_tri, na.rm = TRUE)

# Create a heatmap
ggplot(melted_cormat, aes(Var2, Var1, fill = value)) +
  geom_tile(color = "white") +
  scale_fill_gradient2(
    low = "blue",
    high = "red",
    mid = "white",
    midpoint = 0,
    limit = c(-1, 1),
    space = "Lab",
    name = "Pearson\nCorrelation"
  ) +
  theme_minimal() + # minimal theme
  theme(
    axis.text.x = element_text(angle = 90, size = 9)
  ) +
  coord_fixed()
```




## Distribution Plots

```
# Distribution Plots
# Histogram of log_total
ggplot(train, aes(x = log_total)) +
  geom_histogram(bins = 30, fill = "blue", alpha = 0.7) +
  theme_minimal() +
  labs(title = "Distribution of log_total", x = "log_total", y = "Count")
```

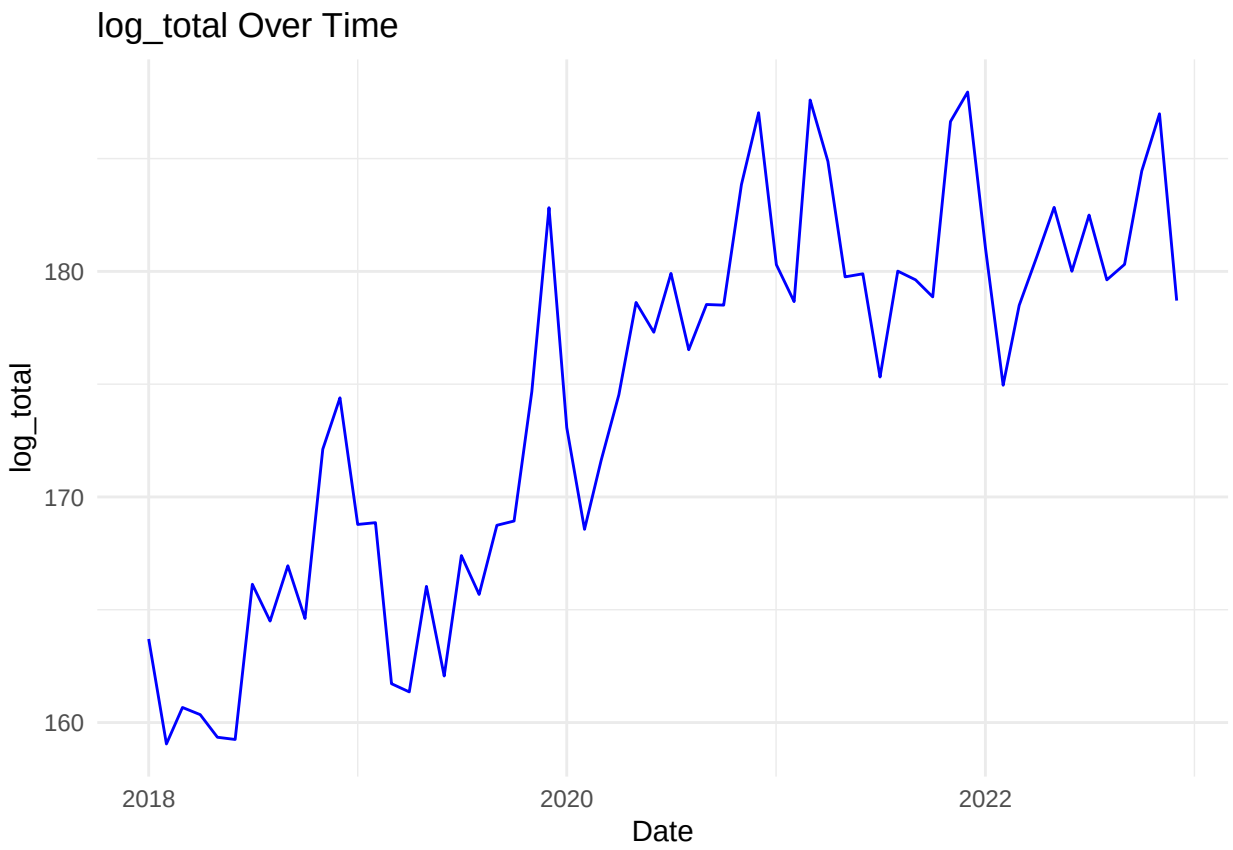



### Time Series Plots

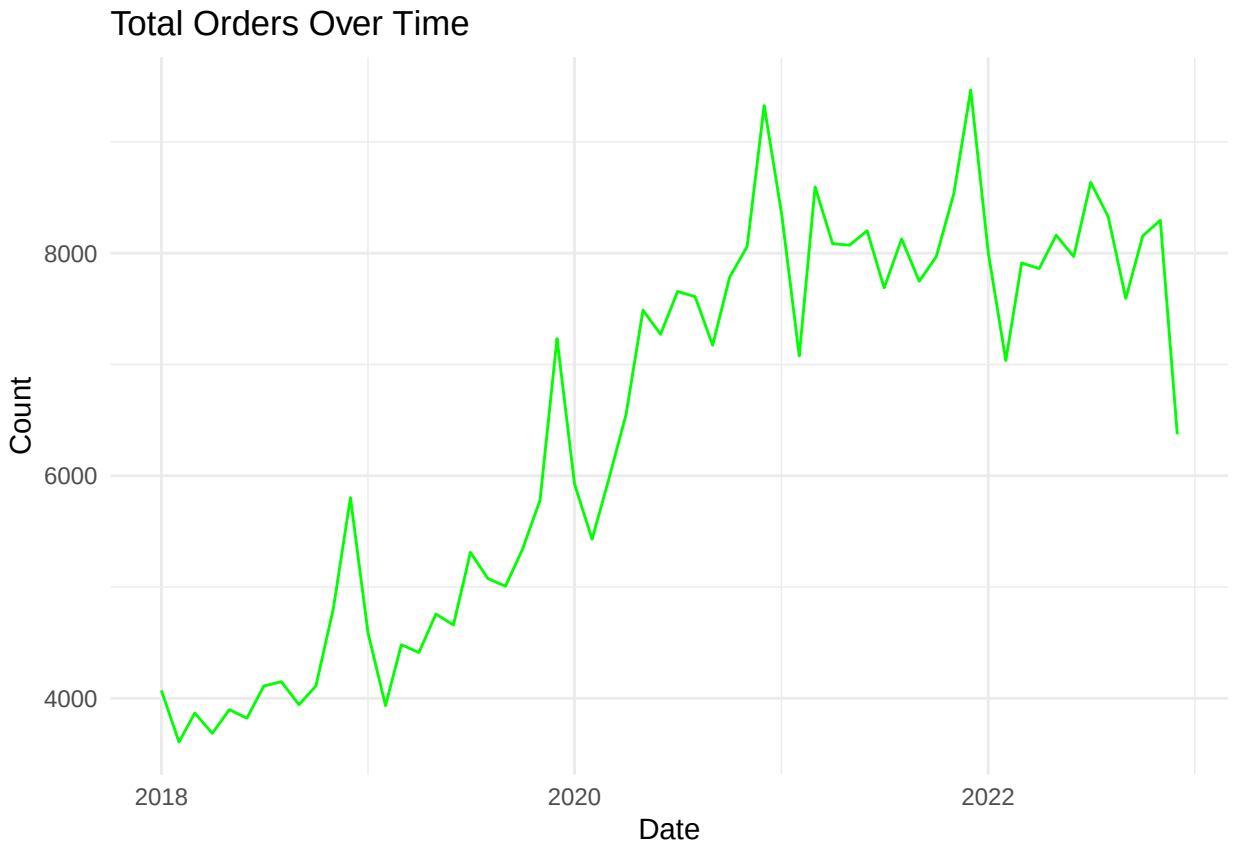
```
# Time Series Plots
# Line Plot of log_total over time
timeseries <- train %>%
  group_by(year, month) %>%
  summarise(log_total = sum(log_total), count = sum(count)) %>%
  mutate(date = ymd(paste(year, month, "01", sep = "-"))) %>%
  select(date, log_total, count) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
## Adding missing grouping variables: `year`
```

```
ggplot(timeseries, aes(x = date, y = log_total)) +
  geom_line(color = "blue") +
  theme_minimal() +
  labs(title = "log_total Over Time", x = "Date", y = "log_total")
```

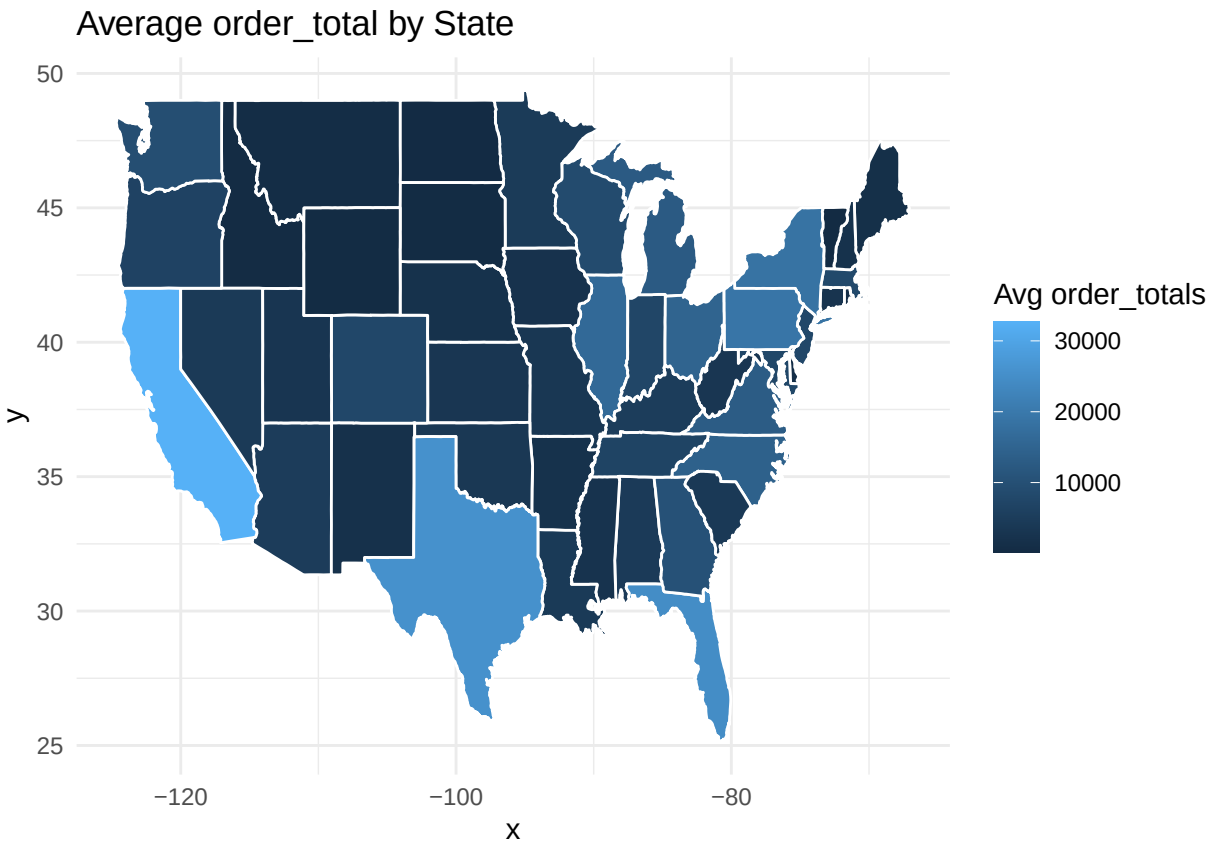



```
# Line Plot of count over time
ggplot(timeseries, aes(x = date, y = count)) +
  geom_line(color = "green") +
  theme_minimal() +
  labs(title = "Total Orders Over Time", x = "Date", y = "Count")
```



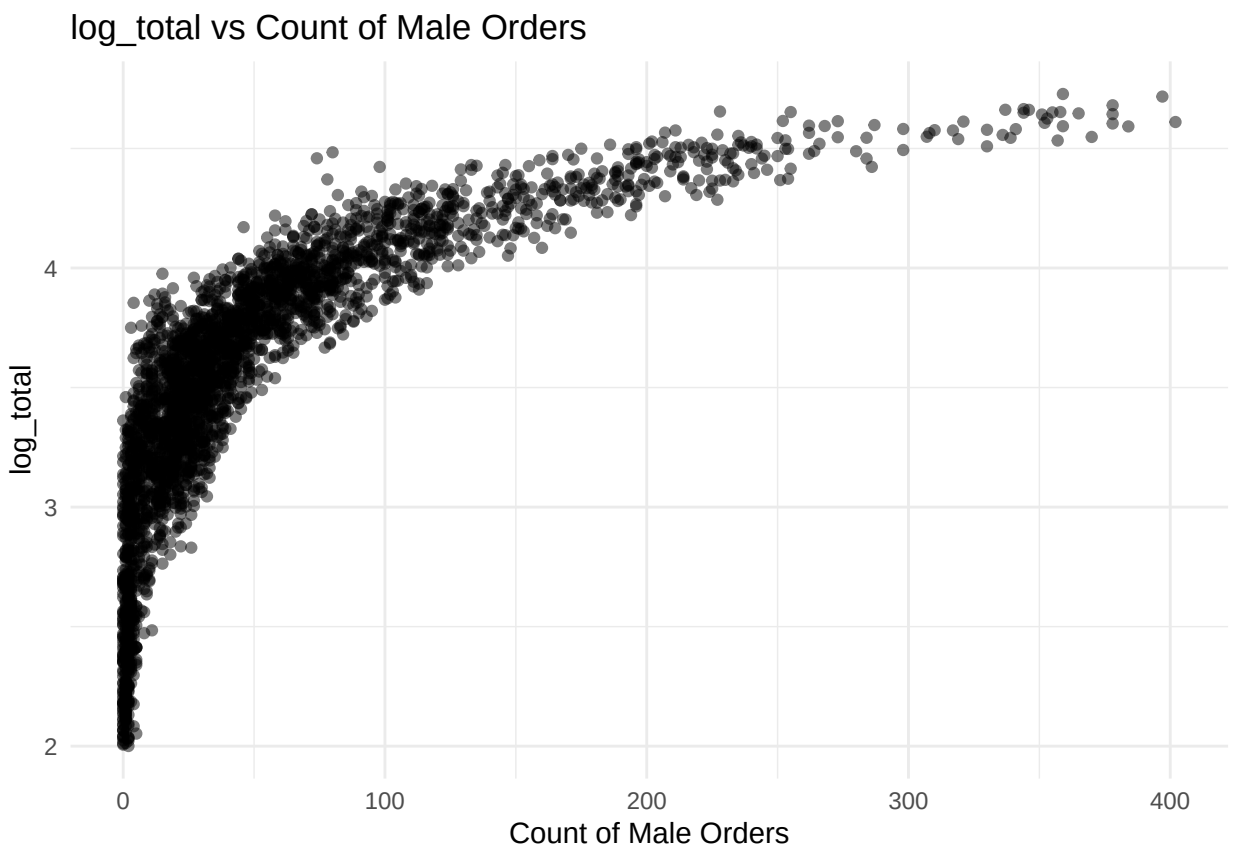

## Geographical Plots

```r
# Geographical Plots
# Assuming you have a state map dataset named `us_states` with state abbreviations
us_states <- map_data("state")

states <- train %>%
  mutate(q_demos_state = tolower(q_demos_state)) %>%
  group_by(q_demos_state) %>%
  summarize(avg_total = mean(order_totals)) %>%
  select(q_demos_state, avg_total) %>%
  ungroup()

# Merge the state data
states <- us_states %>%
  left_join(states, by = c("region" = "q_demos_state"))

ggplot(states, aes(map_id = region, fill = avg_total)) +
  geom_map(map = us_states, color = "white") +
  expand_limits(x = us_states$long, y = us_states$lat) +
  theme_minimal() +
  labs(title = "Average order_total by State", fill = "Avg order_totals")
```

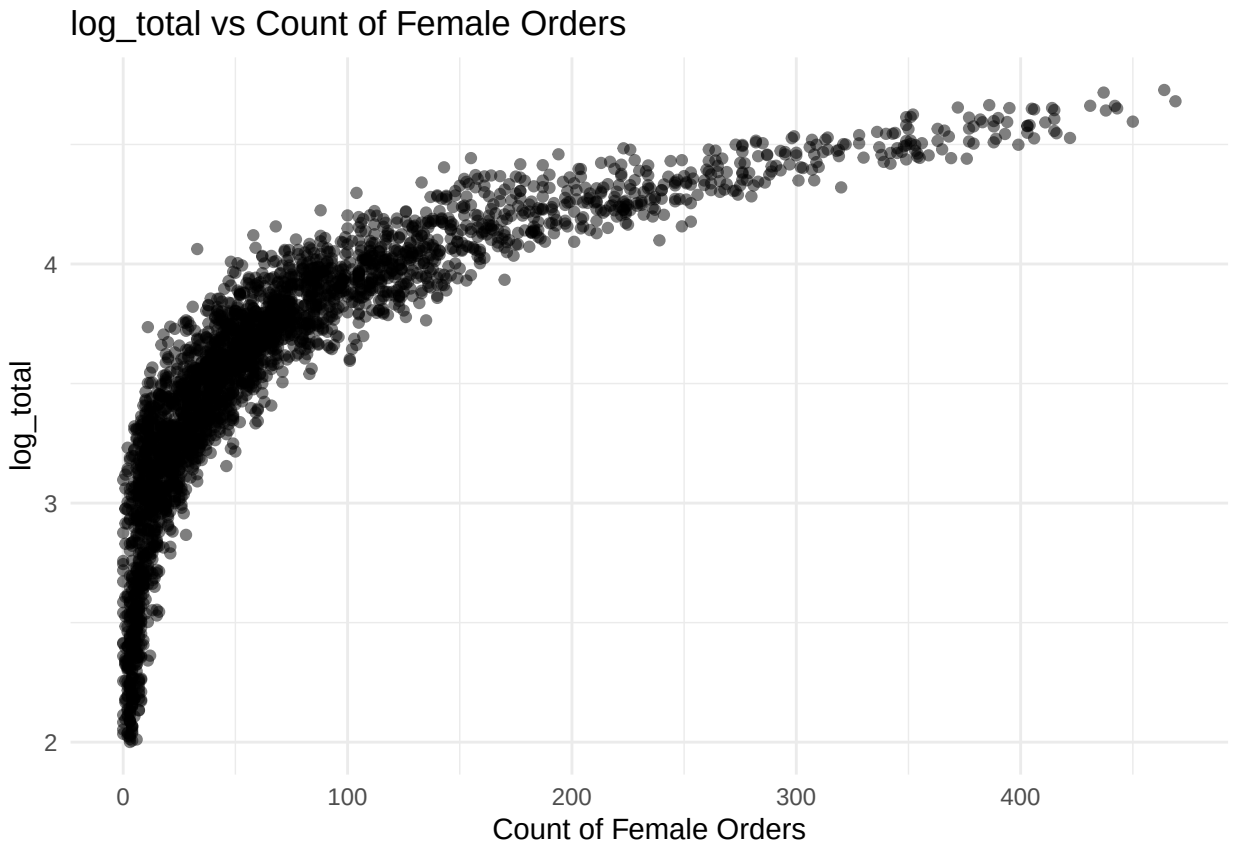



### Scatter Plots

```
# Scatter Plots
# Scatter plot of log_total vs count_male
ggplot(train, aes(x = count_male, y = log_total)) +
  geom_point(alpha = 0.5) +
  theme_minimal() +
  labs(title = "log_total vs Count of Male Orders", x = "Count of Male Orders", y = "log_total")
```



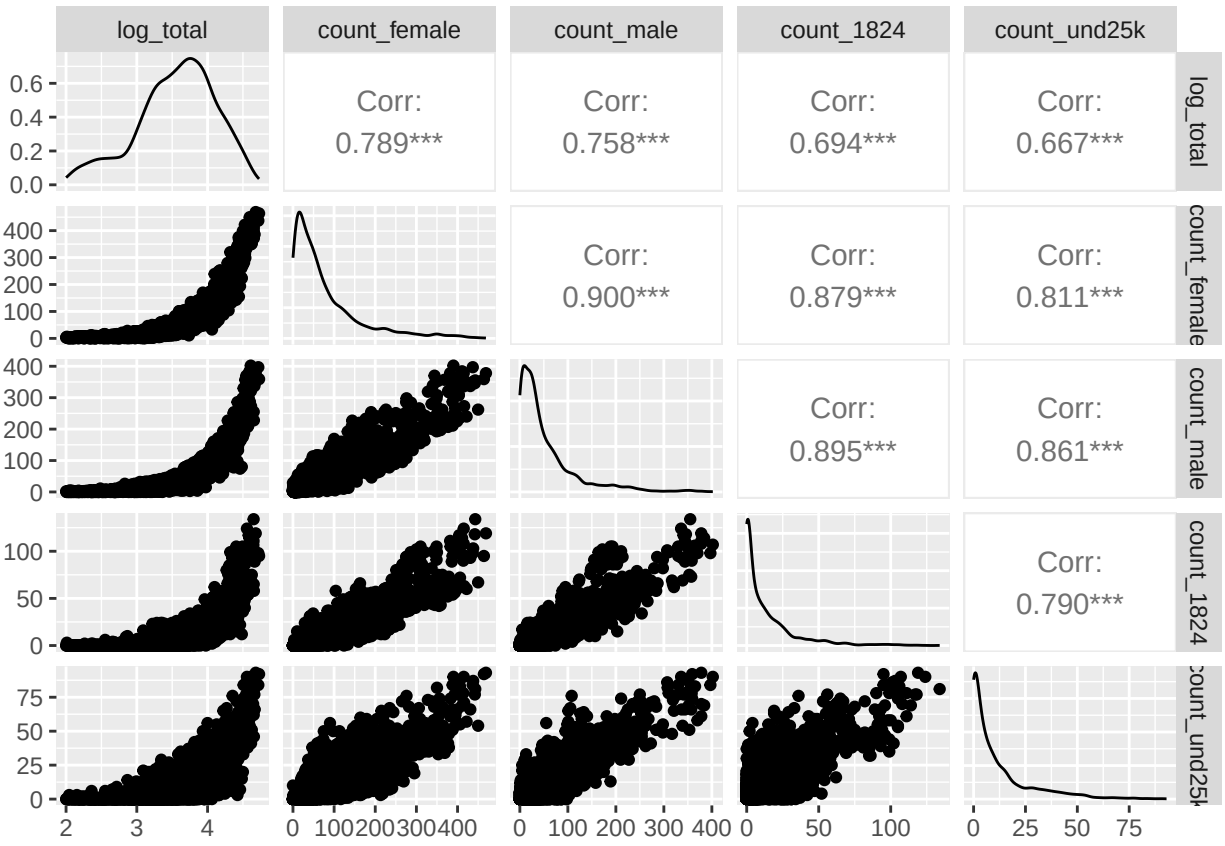

```r
# Scatter plot of log_total vs count_female
ggplot(train, aes(x = count_female, y = log_total)) +
  geom_point(alpha = 0.5) +
  theme_minimal() +
  labs(title = "log_total vs Count of Female Orders", x = "Count of Female Orders", y = "log_total")
```




```
# Pair Plot
# Using GGally package for a pair plot
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
ggpairs(train %>% select(log_total, count_female, count_male, count_1824, count_und25k))
```



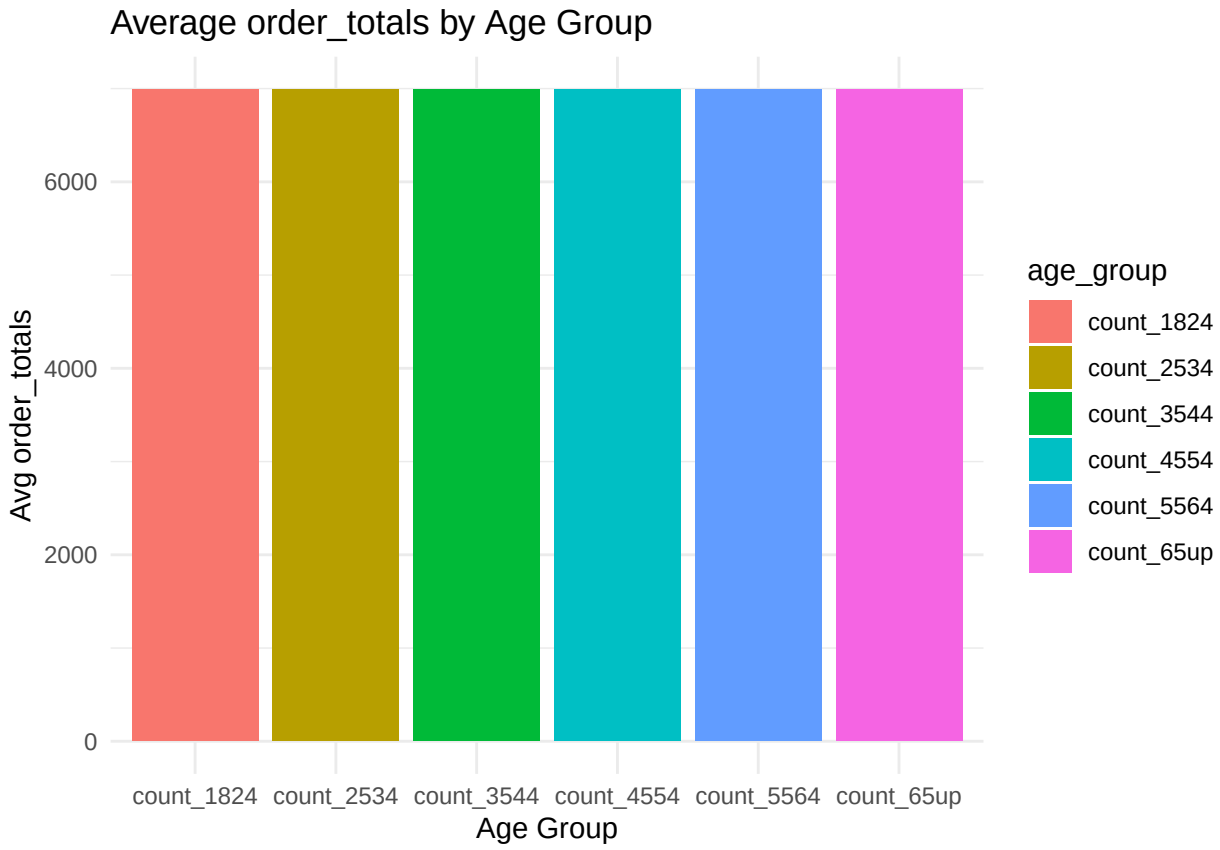

## Bar Plots TODO: FIX THIS

```
# Bar Plots
# Bar Plot of average order_totals by age group
age_vars <- c("count_1824", "count_2534", "count_3544", "count_4554", "count_5564", "count_65up")
ages <- train %>%
  select(q_demos_state, year, month, order_totals, all_of(age_vars)) %>%
  pivot_longer(cols = age_vars, names_to = "age_group", values_to = "count")
```

```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use ‘all_of()‘ or ‘any_of()‘ instead.
##   # Was:
##   data %>% select(age_vars)
##
##   # Now:
##   data %>% select(all_of(age_vars))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
ages <- ages %>%
  group_by(age_group) %>%
```

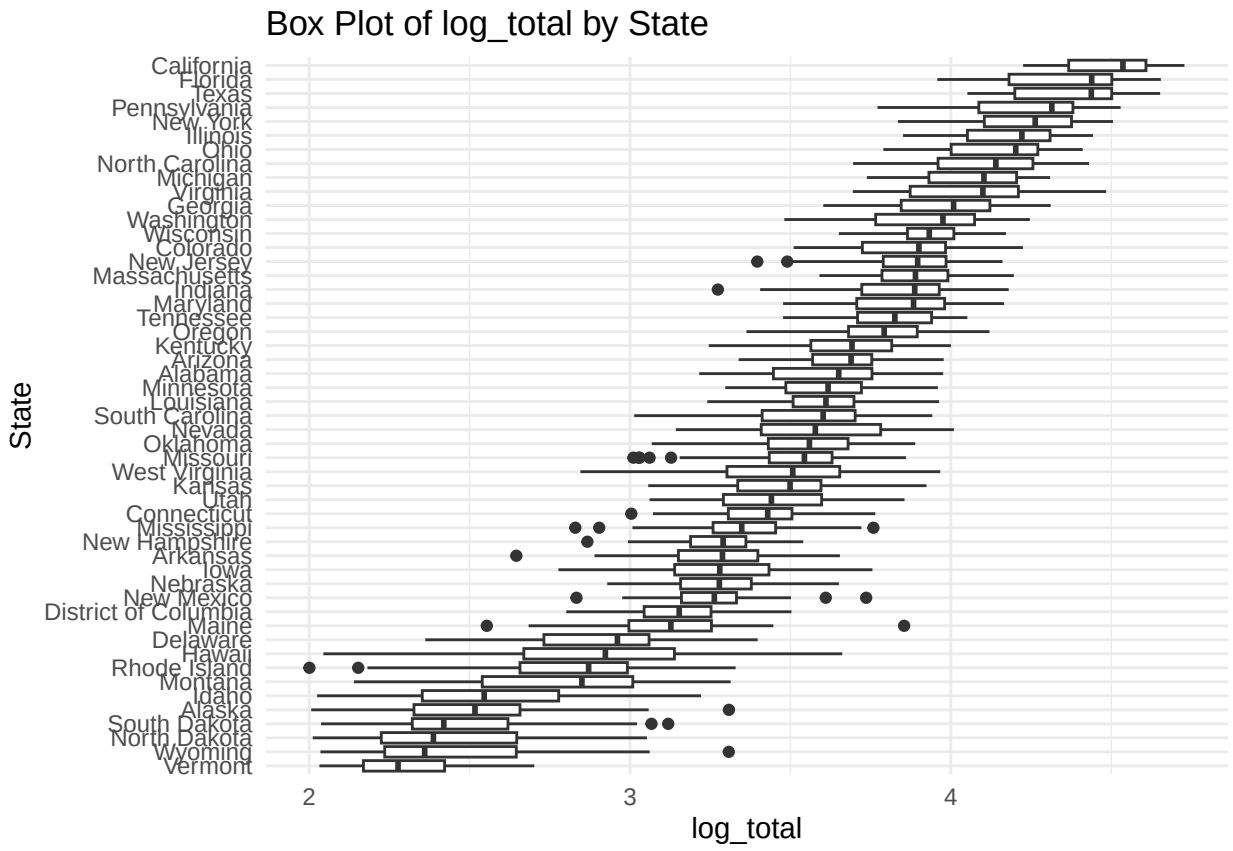
```
  summarize(avg_total = mean(order_totals))

ggplot(ages, aes(x = age_group, y = avg_total, fill = age_group)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Average order_totals by Age Group", x = "Age Group", y = "Avg order_totals")
```



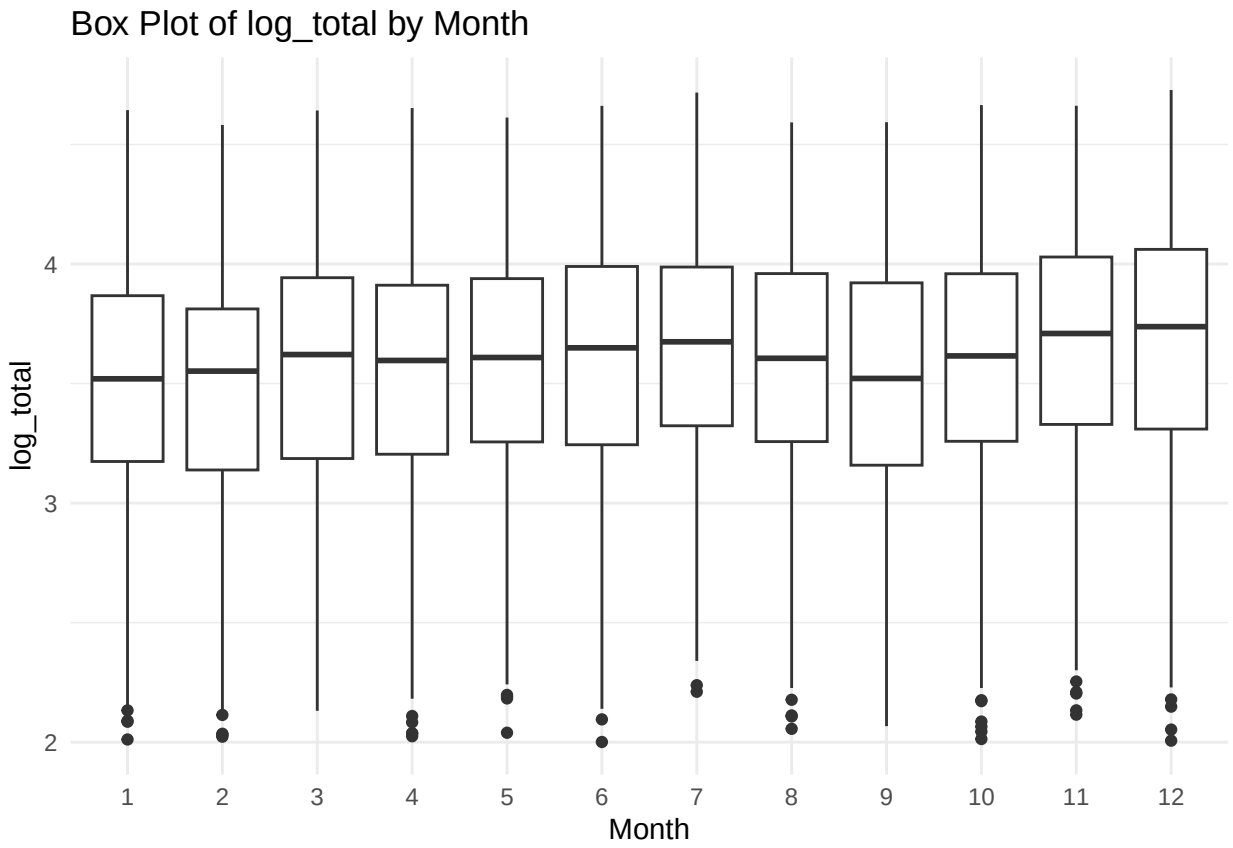

## Box Plots

```
# Box Plots
# Box Plot of log_total by state
# Calculate the median log_total for each state
state_medians <- train %>%
  group_by(q_demos_state) %>%
  summarize(median_log_total = median(log_total)) %>%
  arrange(median_log_total)

# Reorder the levels of q_demos_state based on the median log_total
state_medians <- train %>%
  mutate(q_demos_state = factor(q_demos_state, levels = state_medians$q_demos_state))

# Box Plot of log_total by state with sorted states
ggplot(state_medians, aes(x = log_total, y = q_demos_state)) +
  geom_boxplot() +
```

```
  theme_minimal() +
  labs(title = "Box Plot of log_total by State", y = "State", x = "log_total")
```




```
# Box Plot of log_total by month
ggplot(train, aes(x = factor(month), y = log_total)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Box Plot of log_total by Month", x = "Month", y = "log_total")
```


Box Plot of log_total by Month
log_total
4
3
2
1
2
3
4
5
6
7
8
9
10
11
12
Month