This is about library analysis done in R.

Data Inputs

- library
- checkouts
- books
- customers

Data Outputs

- data_model

Import all datasets

3

```
library(tidyverse)

files <- c("libraries.csv", "checkouts.csv", "customers.csv", "books.csv")

data_list <- lapply(files, function(file) {
  read_csv(file)
})

names <- c("libraries", "checkouts", "customers", "books")
for (i in 1:length(data_list)) {
  assign(names[i], data_list[[i]])
}
```

Merge all data sets properly and clean data

5: **Step 1**

```
data_model <- checkouts %>%
  inner_join(libraries, by = c("library_id" = "id")) %>%
  mutate(name = tolower(gsub("\\s+", " ", name)),
         city = tools::toTitleCase(tolower(gsub("\\s+", " ", city))),
         region = toupper(gsub("\\s+", " ", region)),
         postal_code = as.numeric(gsub("[^0-9]", "", postal_code)),
         date_checkout = as.Date(date_checkout),
         date_returned = as.Date(date_returned)) %>%
  rename(checkouts_id = id,
         library_name = name,
         library_street_address = street_address,
         library_city = city,
         library_region = region,
         library_postal_code = postal_code)
```

6: **Step 2**

```
data_model <- books %>%
  inner_join(data_model, by = c("id" = "checkouts_id")) %>%
  mutate(title = tolower(gsub("\\s+", " ", title)),
         authors = gsub("\\[|\\]", "", authors),
         categories = gsub("\\[|\\]", "", categories),
         publishedDate = as.Date(publishedDate),
         price = as.numeric(price),
         pages = as.double(pages)) %>%
  rename(book_id = id,
         book_title = title,
         book_authors = authors,
         book_publisher = publisher,
         published_date = publishedDate,
         book_categories = categories,
         book_price = price,
         book_pages = pages)
```

```
Warning in dplyr::inner_join(., data_model, by = c(id = "checkouts_id")) :
  Each row in `x` is expected to match at most 1 row in `y`.
ℹ Row 1 of `x` matches multiple rows.
ℹ If multiple matches are expected, set `multiple = "all"` to silence this
  warning.
Warning: There were 2 warnings in `dplyr::mutate()`.
The first warning was:
ℹ In argument: `price = as.numeric(price)`.
Caused by warning:
! NAs introduced by coercion
ℹ Run `dplyr::last_dplyr_warnings()` to see the 1 remaining warning.
```

7: **Step 3**

```
data_model <- customers %>%
  inner_join(data_model, by = c("id" = "patron_id")) %>%
  mutate(city = tools::toTitleCase(tolower(gsub("\\s+", " ", city))),
         zipcode = as.numeric(gsub("[^0-9]", "", zipcode)),
         zipcode = as.numeric(substr(zipcode, 1, nchar(zipcode) - 1)),
         education = tolower(gsub("\\s+", " ", education)),
         occupation = tolower(gsub("\\s+", " ", occupation)),
         gender = tolower(gender),
         birth_date = as.double(substr(birth_date, 1, 4)),
         date_diff = date_returned - date_checkout,
         late_return = ifelse(date_returned - date_checkout > 28, TRUE, FALSE)) %>%
  rename(customer_id = id,
         customer_name = name,
         customer_street_address = street_address,
         customer_city = city,
         customer_state = state,
         customer_postal_code = zipcode,
         customer_birth_date = birth_date,
         customer_gender = gender,
         customer_education = education,
         customer_occupation = occupation) %>%
  filter(date_diff >= 0,
         date_returned <= Sys.Date(),
         date_checkout <= Sys.Date(),
         lubridate::year(date_checkout) == 2018)
```

8

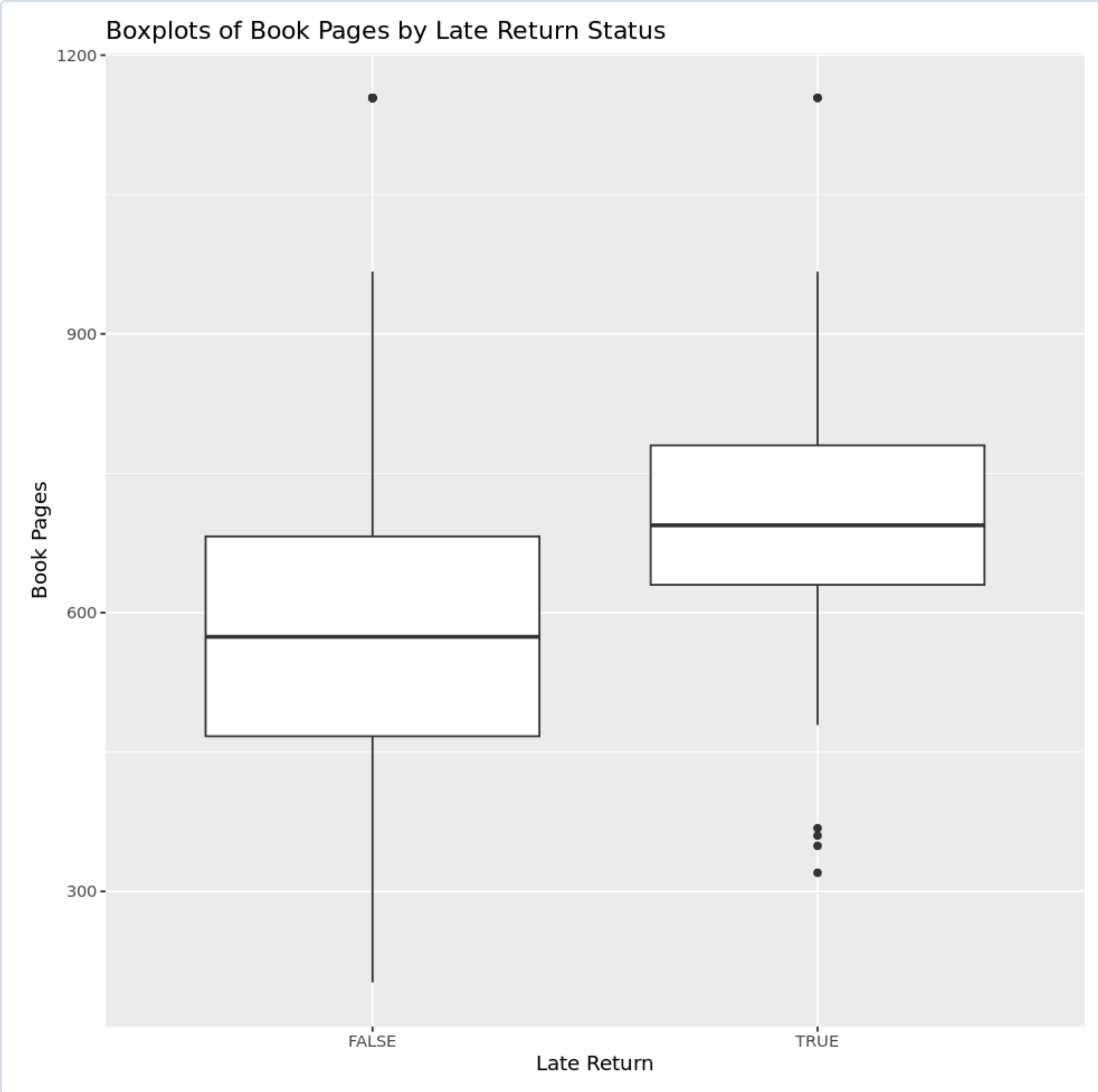
```
summary(data_model$late_return)
```

```
   Mode   FALSE    TRUE 
logical    1256     129 
```

Exploratory Data Analysis

10

```
# Let's see how number of book pages is correlated with late return
data_model %>%
  filter(!is.na(book_pages)) %>%
  ggplot(aes(x = as.factor(late_return), y = book_pages)) +
  geom_boxplot() +
  labs(x = "Late Return", y = "Book Pages") +
  ggtitle("Boxplots of Book Pages by Late Return Status")
```



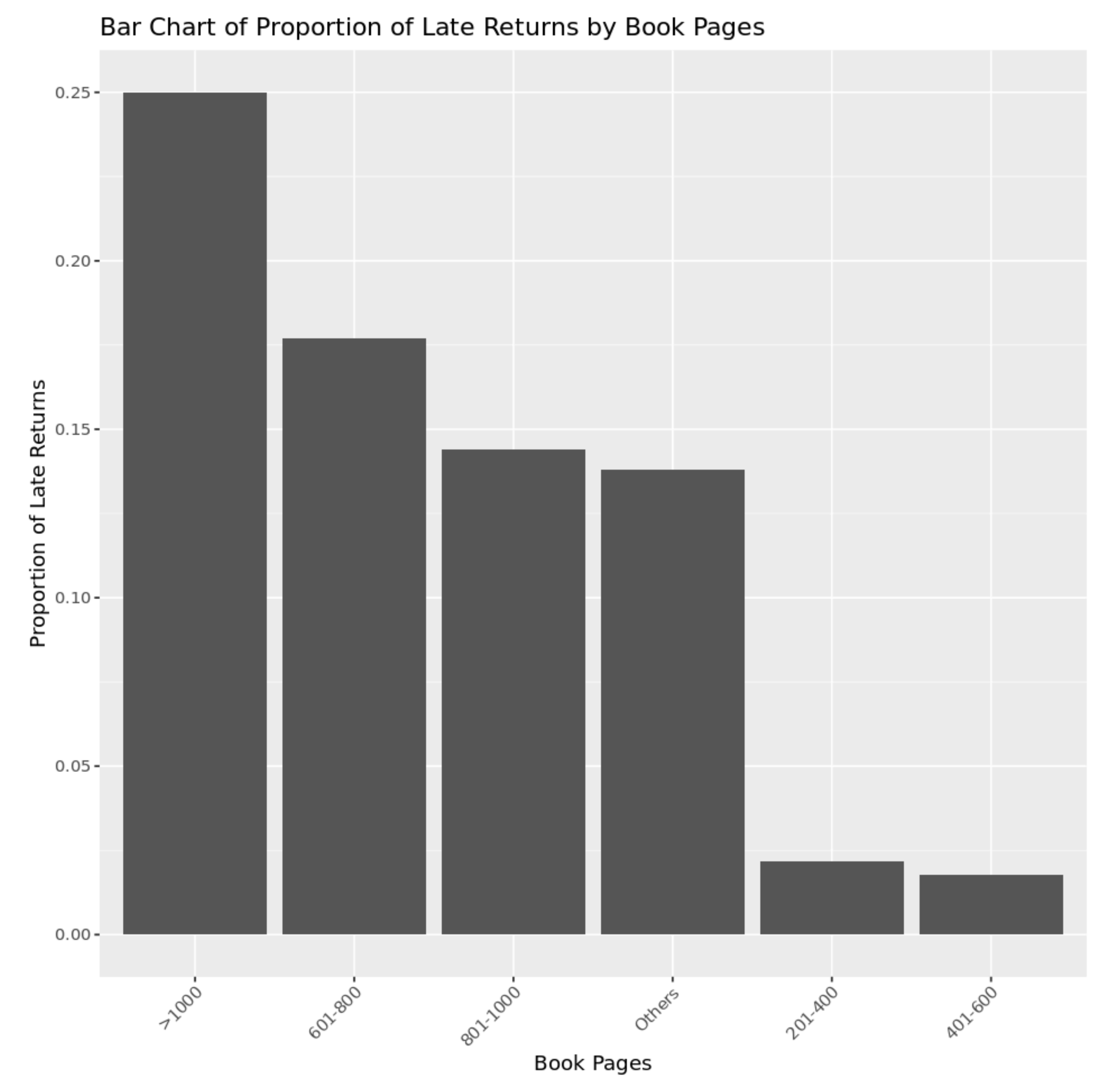

```
# Let's create buckets based on number of book pages and some more structured analysis
data_model <- data_model %>%
  mutate(book_pages_range = case_when(
    book_pages > 200 & book_pages <= 400 ~ "201-400",
    book_pages > 400 & book_pages <= 600 ~ "401-600",
    book_pages > 600 & book_pages <= 800 ~ "601-800",
    book_pages > 800 & book_pages <= 1000 ~ "801-1000",
    book_pages > 1000 ~ ">1000",
    TRUE ~ "Others"))

summary(data_model$book_pages)
```

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
 202.0   481.5   596.0   593.5   698.0  1154.0      58
```

12

```
# Let's figure out the proportion of late returns by book pages
data_model %>%
  group_by(book_pages_range) %>%
  summarise(number_of_checkouts = sum(!is.na(late_return)),
            proportion_late = mean(late_return == TRUE, na.rm = TRUE)) %>%
  arrange(desc(proportion_late)) %>%
  filter(!is.na(book_pages_range)) %>%
  ggplot(aes(x = reorder(book_pages_range, -proportion_late), y = proportion_late)) +
  geom_bar(stat = "identity") +
  labs(x = "Book Pages", y = "Proportion of Late Returns") +
  ggtitle("Bar Chart of Proportion of Late Returns by Book Pages") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



13

```
# Let's figure out how the proportion is related to number of checkots by creating a simple table
data_model %>%
  group_by(book_pages_range) %>%
  summarise(number_of_checkouts = sum(!is.na(late_return)),
            proportion_late = mean(late_return == TRUE, na.rm = TRUE)) %>%
  arrange(desc(proportion_late)) %>%
  filter(!is.na(book_pages_range))
```

```
# A tibble: 6 × 3
  book_pages_range number_of_checkouts proportion_late
  <chr>                          <int>           <dbl>
1 >1000                              8          0.25
2 601-800                          486          0.177
3 801-1000                         139          0.144
4 Others                            58          0.138
5 201-400                          185          0.0216
6 401-600                          509          0.0177
```

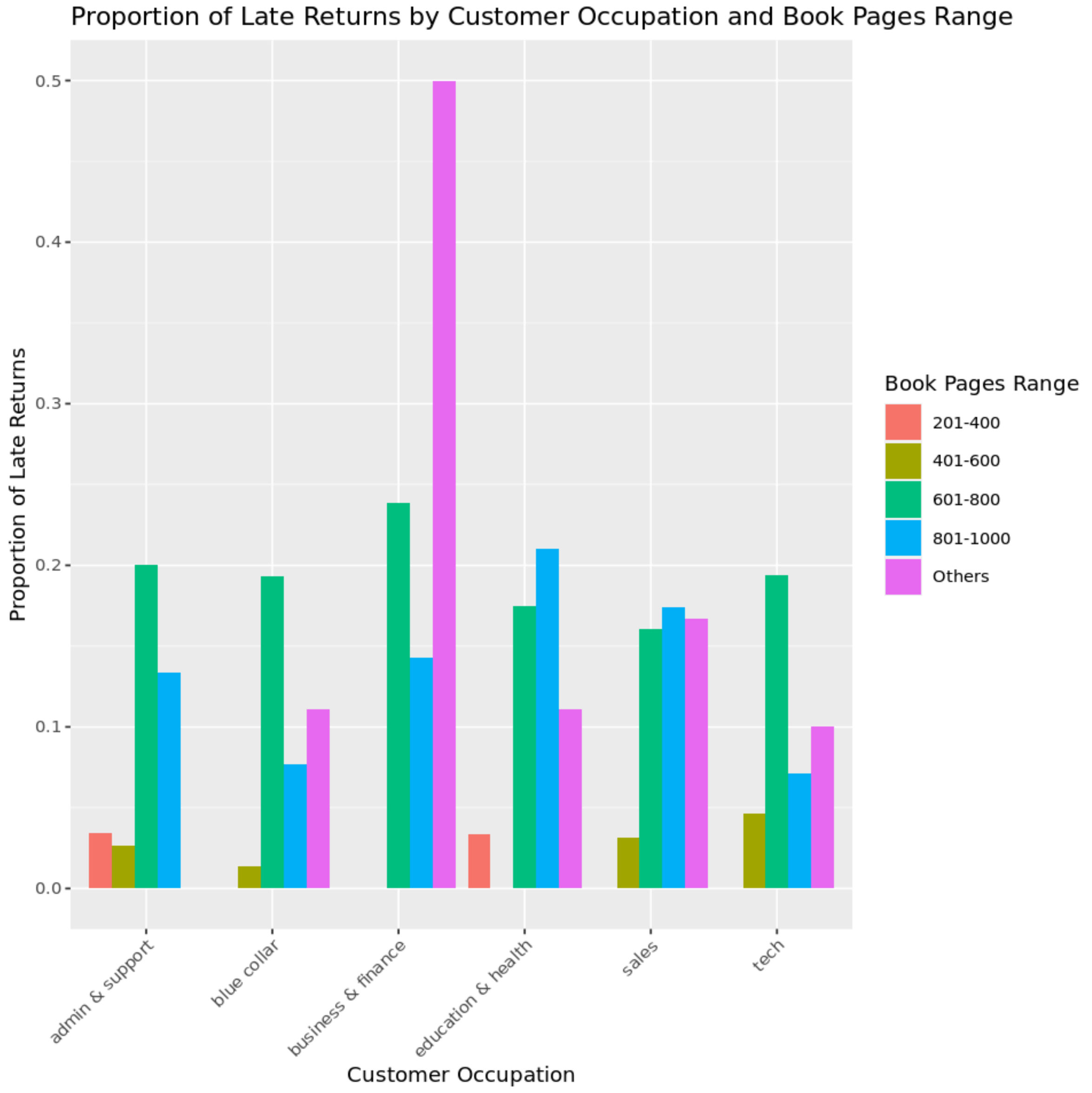
14

```
# Let's figure out how the proportion is related to specific books
data_model %>%
  filter(!is.na(book_title)) %>%
  group_by(book_title) %>%
  summarise(number_of_checkouts = sum(!is.na(late_return)),
            proportion_late = mean(late_return == TRUE, na.rm = TRUE)) %>%
  arrange(desc(proportion_late)) %>%
  head(20)
```

```
 1 resources and opportunities of montana                            6   0.667
 2 the american journal of clinical medicine                         5   0.6
 3 the journal of psychological medicine and mental pathology        5   0.6
 4 the laws of medicine                                              5   0.6
 5 academic search engines                                           2   0.5
 6 american almanac and treasury of facts statistical, financia…     6   0.5
 7 annual report of the general society of mechanics and trades…     6   0.5
 8 financial accounting: a dynamic approach                          2   0.5
 9 financial services, 2e                                            2   0.5
10 immigration services                                              6   0.5
11 international accounting/financial reporting standards guide…     4   0.5
12 invisible engines                                                 2   0.5
13 reports submitted to the council on library resources            4   0.5
14 the commercial and financial chronicle                            4   0.5
15 the resources and attractions of utah                             2   0.5
16 water resources management iv                                     8   0.5
17 planning our resources                                            7   0.429
18 advertising to the american woman, 1900-1999                      8   0.375
19 replies to questionnaires on aircraft engine production cost…     8   0.375
20 advertising the american dream                                    6   0.333
# … with abbreviated variable names ¹number_of_checkouts, ²proportion_late
```

15

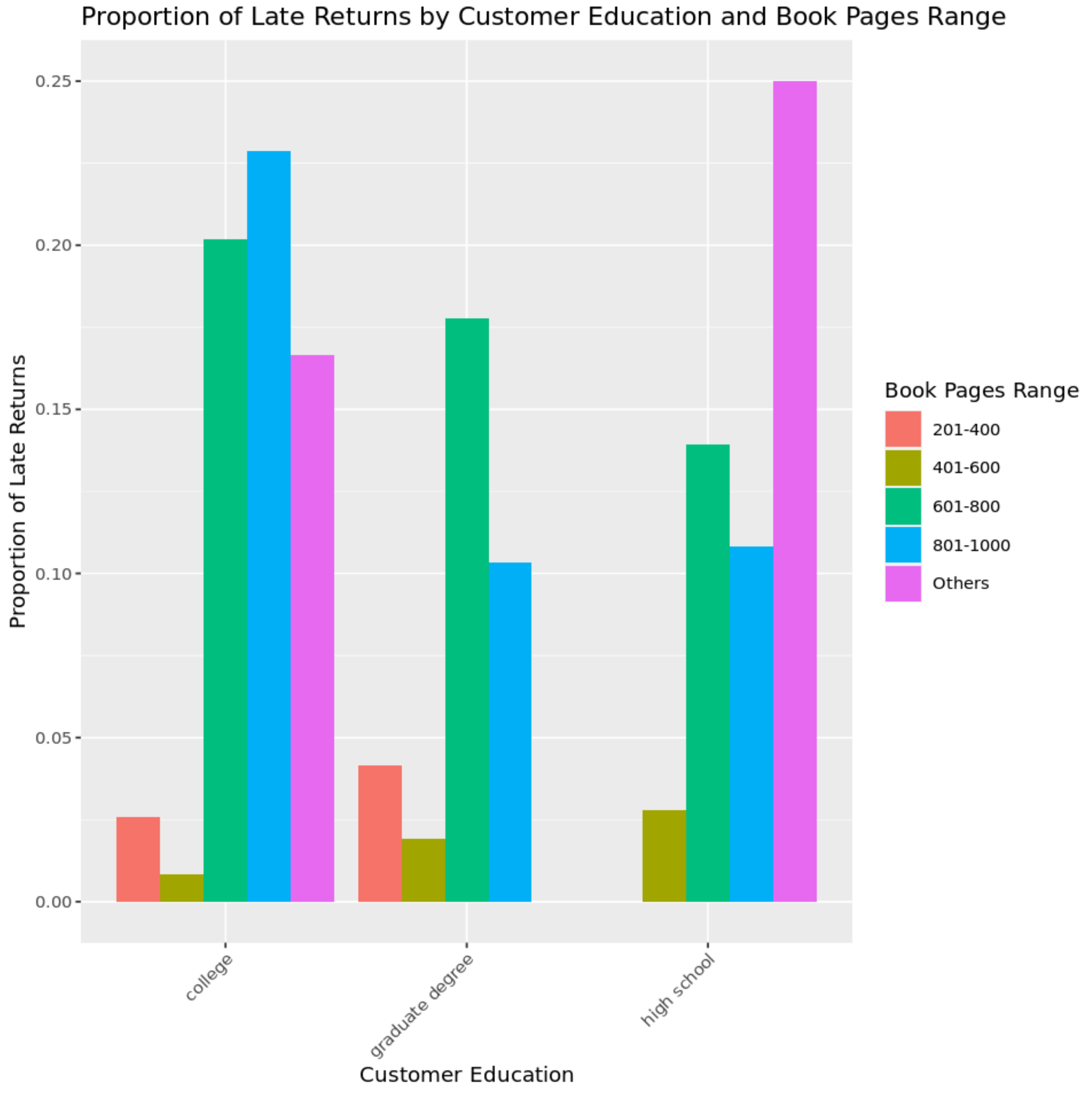
```
# Let's figure out the proportion of late returns by book pages and occupation
data_model %>%
  filter(book_pages_range != ">1000") %>%
  group_by(customer_occupation, book_pages_range) %>%
  summarise(number_of_checkouts = sum(!is.na(late_return)),
            proportion_late = mean(late_return == TRUE, na.rm = TRUE)) %>%
  filter(!is.na(book_pages_range) & customer_occupation != "NA" & customer_occupation != "others") %>%
  ggplot(aes(x = customer_occupation, y = proportion_late, fill = book_pages_range)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Customer Occupation", y = "Proportion of Late Returns", fill = "Book Pages Range") +
  ggtitle("Proportion of Late Returns by Customer Occupation and Book Pages Range") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
`summarise()` has grouped output by 'customer_occupation'. You can override
using the `.groups` argument.
```

16

```
# Let's figure out the proportion of late returns by book pages and education
data_model %>%
filter(book_pages_range != ">1000") %>%
  group_by(customer_education, book_pages_range) %>%
  summarise(number_of_checkouts = sum(!is.na(late_return)),
            proportion_late = mean(late_return == TRUE, na.rm = TRUE)) %>%
  filter(!is.na(book_pages_range) & customer_education != "NA" & customer_education != "others") %>%
  ggplot(aes(x = customer_education, y = proportion_late, fill = book_pages_range)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Customer Education", y = "Proportion of Late Returns", fill = "Book Pages Range") +
  ggtitle("Proportion of Late Returns by Customer Education and Book Pages Range") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
`summarise()` has grouped output by 'customer_education'. You can override
using the `.groups` argument.
```



```r
# create Boolean column called portland_postal_code
portland_postal_codes <- c(97229, 97206, 97080, 97219, 97230, 97202, 97030, 97236,
                           97233, 97203, 97266, 97217, 97211, 97213, 97220, 97214,
                           97212, 97060, 97215, 97239, 97209, 97216, 97201, 97232,
                           97218, 97210, 97272, 97221, 97259, 97255, 97205, 97024,
                           97227, 97271, 97231, 97019, 97014, 97049, 97204, 97208,
                           97258, 97299, 97010, 97282, 97207, 97228, 97238, 97242,
                           97240, 97253, 97251, 97254, 97256, 97280, 97286, 97283,
                           97291, 97290, 97293, 97292, 97296, 97294, 97250, 97252)

data_model <- data_model %>%
  mutate(portland_postal_code = customer_postal_code %in% portland_postal_codes)

summary(data_model$portland_postal_code)
```

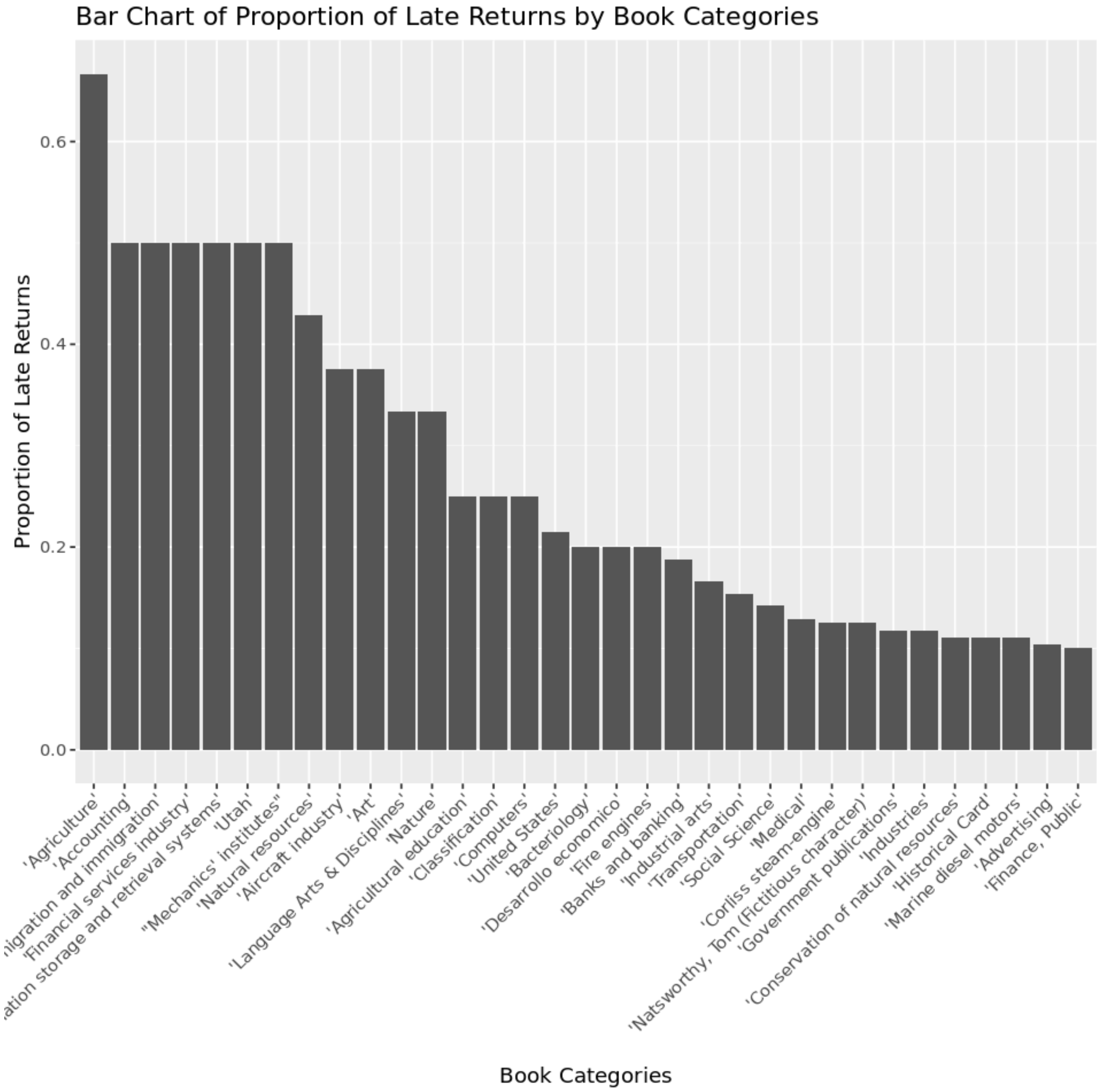
```
   Mode   FALSE    TRUE
logical     216    1169
```

```r
# return the table grouped by education, portland postal code, and book pages to see proportion of late returns
data_model %>%
  filter(!is.na(customer_education) & !is.na(book_pages_range)) %>%
  group_by(customer_education, portland_postal_code, book_pages_range) %>%
  summarise(number_of_books = sum(!is.na(book_id)),
         proportion_late = mean(late_return == TRUE)) %>%
  filter(number_of_books >= 10) %>%
  arrange(desc(proportion_late))
```

```
`summarise()` has grouped output by 'customer_education',
'portland_postal_code'. You can override using the `.groups` argument.
# A tibble: 27 × 5
# Groups:   customer_education, portland_postal_code [8]
   customer_education portland_postal_code book_pages_range number_of_…¹ propo…²
   <chr>              <lgl>                <chr>                   <int>   <dbl>
 1 college            FALSE                601-800                    19  0.579
 2 high school        FALSE                601-800                    24  0.417
 3 others             FALSE                601-800                    18  0.389
 4 graduate degree    FALSE                601-800                    15  0.333
 5 college            TRUE                 801-1000                   28  0.179
 6 graduate degree    TRUE                 601-800                    92  0.152
 7 high school        TRUE                 Others                     14  0.143
 8 others             TRUE                 601-800                   105  0.143
 9 college            TRUE                 601-800                   100  0.13
10 graduate degree    TRUE                 801-1000                   24  0.0833
# … with 17 more rows, and abbreviated variable names ¹number_of_books,
#   ²proportion_late
```

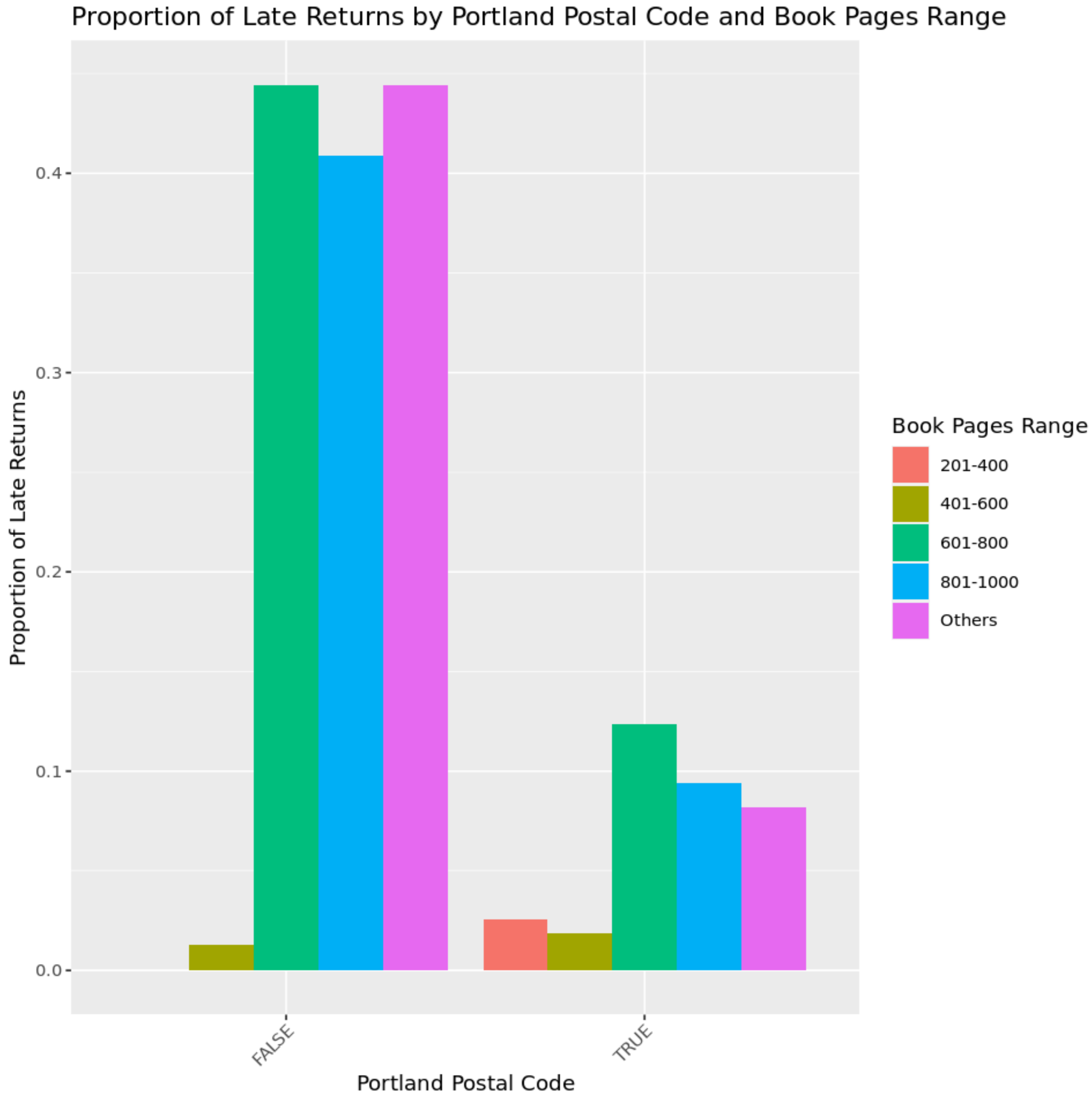
19

```r
# Let's see if book categorization is clean
data_model %>%
  group_by(book_categories) %>%
  summarise(number_of_checkouts = sum(!is.na(late_return)),
            proportion_late = mean(late_return == TRUE, na.rm = TRUE)) %>%
  filter(!is.na(book_categories) & proportion_late >= 0.1) %>%
  arrange(desc(proportion_late)) %>%
  ggplot(aes(x = reorder(book_categories, -proportion_late), y = proportion_late)) +
  geom_bar(stat = "identity") +
  labs(x = "Book Categories", y = "Proportion of Late Returns") +
  ggtitle("Bar Chart of Proportion of Late Returns by Book Categories") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



20

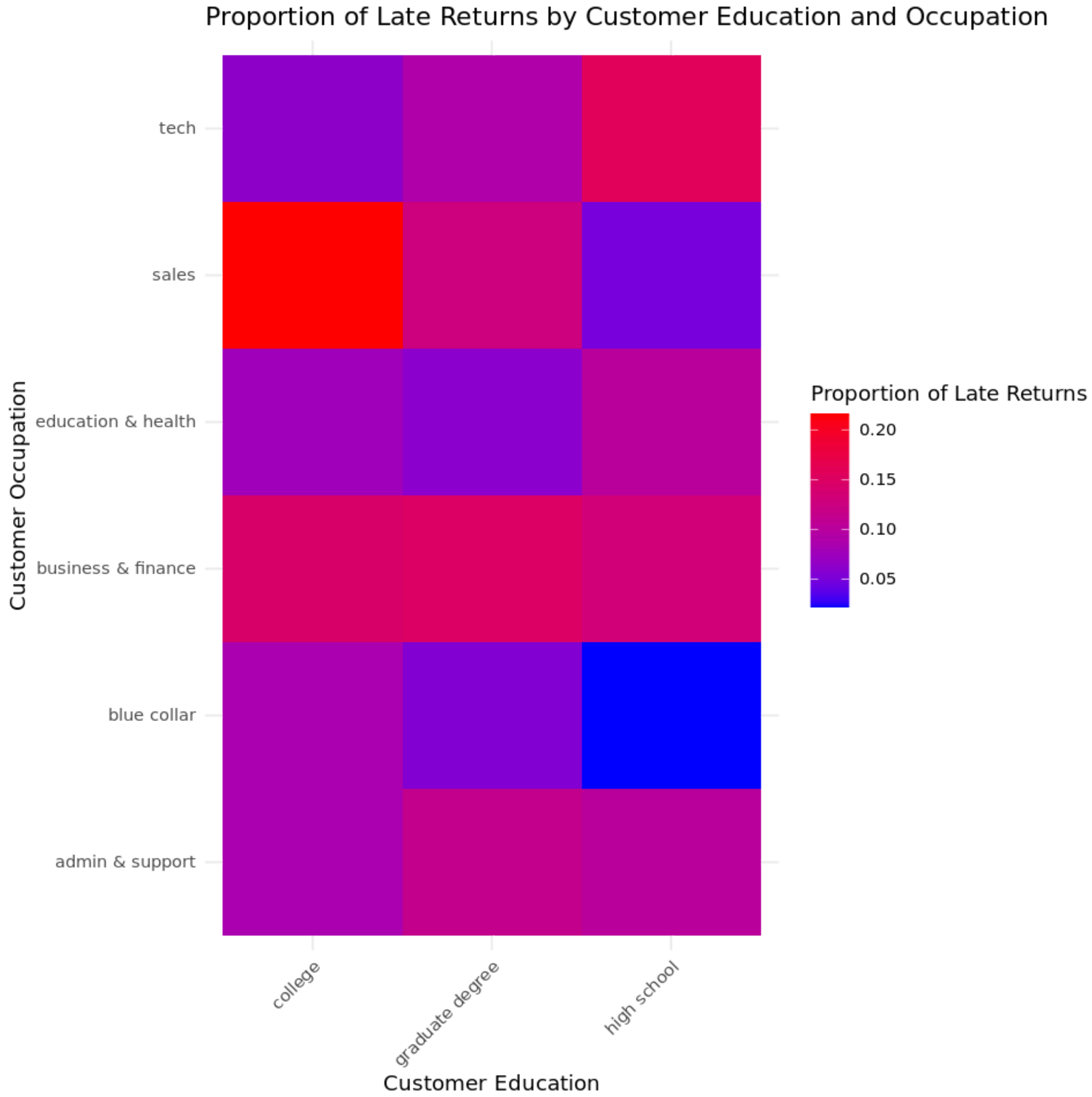
```
# Let's figure out the proportion of late returns by book pages and Portland postal codes
data_model %>%
filter(book_pages_range != ">1000") %>%
  group_by(portland_postal_code, book_pages_range) %>%
  summarise(number_of_checkouts = sum(!is.na(late_return)),
            proportion_late = mean(late_return == TRUE, na.rm = TRUE)) %>%
  filter(!is.na(book_pages_range)) %>%
  ggplot(aes(x = portland_postal_code, y = proportion_late, fill = book_pages_range)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Portland Postal Code", y = "Proportion of Late Returns", fill = "Book Pages Range") +
  ggtitle("Proportion of Late Returns by Portland Postal Code and Book Pages Range") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
`summarise()` has grouped output by 'portland_postal_code'. You can override
using the `.groups` argument.
```



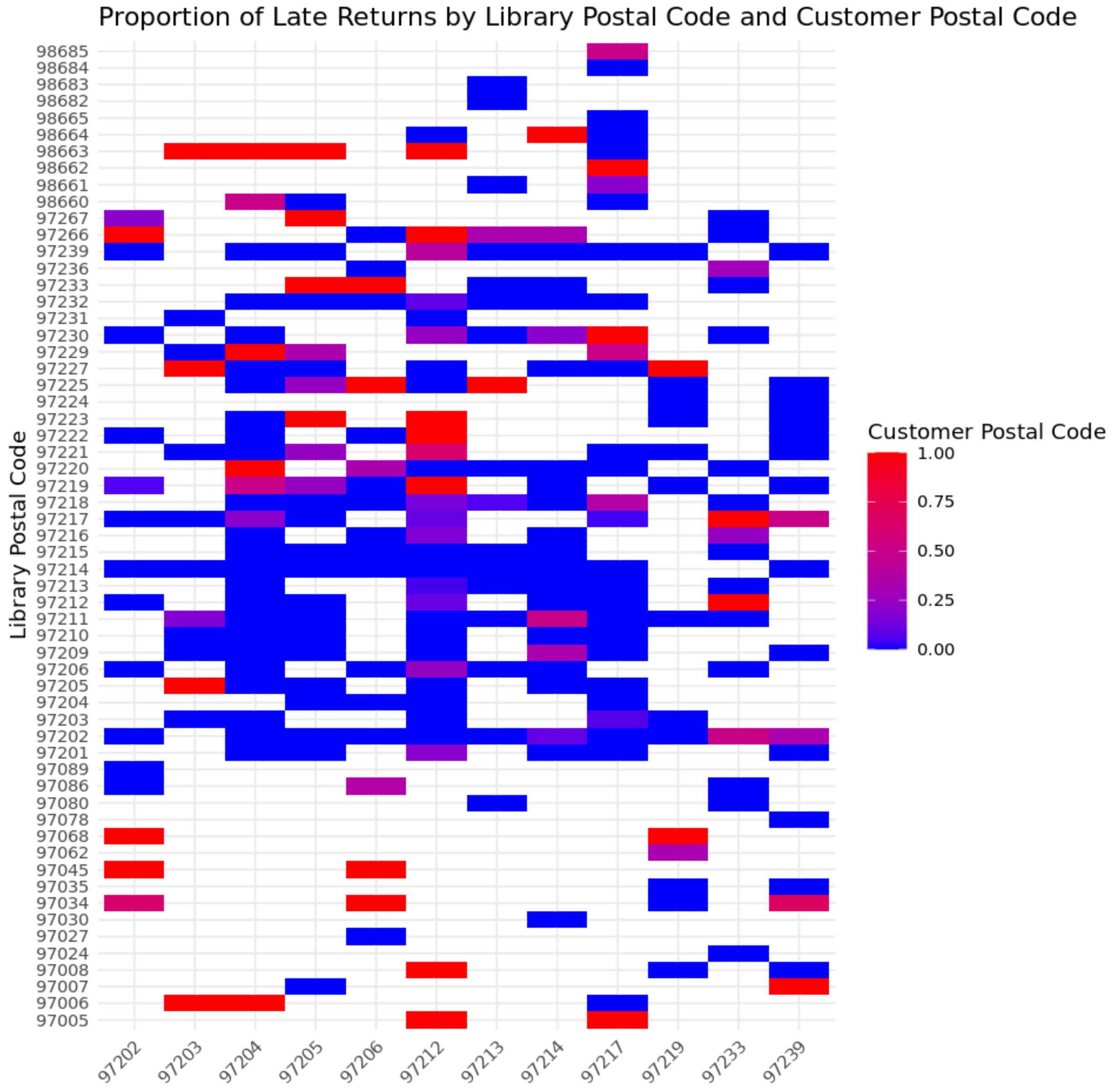
```
# Create a heatmap visualization of the proportion of late returns by occupation and education
data_model %>%
  filter(customer_education != "NA", customer_education != "others",
         customer_occupation != "NA", customer_occupation != "others") %>%
  group_by(customer_education, customer_occupation) %>%
  summarise(proportion_late = mean(late_return == TRUE, na.rm = TRUE), .groups = "drop") %>%
  ggplot(aes(x = customer_education, y = customer_occupation, fill = proportion_late)) +
  geom_tile() +
  scale_fill_gradient(low = "blue", high = "red") +
  labs(x = "Customer Education", y = "Customer Occupation", fill = "Proportion of Late Returns") +
  ggtitle("Proportion of Late Returns by Customer Education and Occupation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Bonus viz for fun!

23

```
# a heatmap visualization of the late return for same postal codes (library and customer postal codes)
data_model %>%
filter(!is.na(library_postal_code)) %>%
  mutate(library_postal_code = as.factor(library_postal_code),
         customer_postal_code = as.factor(customer_postal_code)) %>%
  group_by(library_postal_code, customer_postal_code) %>%
  summarise(proportion_late = mean(late_return == TRUE, na.rm = TRUE), .groups = "drop") %>%
  ggplot(aes(x = library_postal_code, y = customer_postal_code, fill = proportion_late)) +
  geom_tile() +
  scale_fill_gradient(low = "blue", high = "red") +
  labs(x = "", y = "Library Postal Code", fill = "Customer Postal Code") +
  ggtitle("Proportion of Late Returns by Library Postal Code and Customer Postal Code") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Build 4 models and compare their strengths and weakness

25: **Models**

```
# Install the caret package
if (!requireNamespace("caret", quietly = TRUE)) {
  install.packages("caret")
}

# Load the caret package
library(caret)

set.seed(123) # Ensure reproducibility

# Data partition
test_index <- createDataPartition(data_model$late_return, times = 1, p = 0.8, list = FALSE)
test_set <- data_model[test_index, ]
train_set <- data_model[-test_index, ]
```

26

```r
library(pROC)

# Define a list of formulas for the models
formulas <- list(
  late_return ~ book_pages,
  late_return ~ customer_education,
  late_return ~ customer_occupation,
  late_return ~ portland_postal_code
)

# Initialize an empty list to store results
results <- list()

# Loop through each formula
for (i in seq_along(formulas)) {
  # Fit the model
  fit_glm <- glm(formulas[[i]], data=train_set, family = binomial())

  # Predict
  p_hat_glm <- predict(fit_glm, newdata=test_set, type="response")

  # Calculate ROC and best threshold
  roc_obj <- roc(response = test_set$late_return, predictor = p_hat_glm)
  coords_obj <- coords(roc_obj, "best", ret=c("threshold", "accuracy", "sensitivity"),
                       best.method="closest.topleft")

  # Use the best threshold for prediction
  best_threshold <- coords_obj$threshold
  y_hat_glm <- ifelse(p_hat_glm > best_threshold, 1, 0)

  # Calculate metrics
  conf_matrix <- table(Predicted = y_hat_glm, Actual = test_set$late_return)
  accuracy_glm <- sum(diag(conf_matrix)) / sum(conf_matrix)
  sensitivity_glm <- conf_matrix[2, 2] / sum(conf_matrix[2, ])
  specificity_glm <- conf_matrix[1, 1] / sum(conf_matrix[1, ])
  precision_glm <- conf_matrix[2, 2] / sum(conf_matrix[, 2])
  recall_glm <- sensitivity_glm
  f1_glm <- 2 * (precision_glm * recall_glm) / (precision_glm + recall_glm)

  # Store results
  results[[i]] <- list(
    accuracy = accuracy_glm,
    sensitivity = sensitivity_glm,
    specificity = specificity_glm,
    precision = precision_glm,
    recall = recall_glm,
    f1 = f1_glm,
    best_threshold = best_threshold
  )
}

# Convert the list of results to a data frame for easier viewing
results_df <- do.call(rbind, lapply(results, function(x) as.data.frame(t(unlist(x)))))
rownames(results_df) <- c("Model 1", "Model 2", "Model 3", "Model 4")
results_df
```

```
Setting levels: control = FALSE, case = TRUE
Setting direction: controls < cases
Setting levels: control = FALSE, case = TRUE
Setting direction: controls < cases
Setting levels: control = FALSE, case = TRUE
Setting direction: controls < cases
Setting levels: control = FALSE, case = TRUE
Setting direction: controls < cases
         accuracy sensitivity specificity precision     recall        f1
Model 1 0.6154571  0.17536534   0.9776632 0.8659794 0.17536534 0.2916667
Model 2 0.5189394  0.10019646   0.9085923 0.5049505 0.10019646 0.1672131
Model 3 0.4386299  0.09586777   0.9035874 0.5742574 0.09586777 0.1643059
Model 4 0.8205591  0.22857143   0.9314775 0.3846154 0.22857143 0.2867384
        best_threshold
Model 1     0.06982113
Model 2     0.08787879
Model 3     0.09283154
Model 4     0.15386611
```

Build additional for 4 models by going deeper into customer's education with book_pages as a predictor


```
library(pROC)

# Define unique customer education levels excluding NAs
education_levels <- unique(na.omit(train_set$customer_education))

# Initialize an empty list to store results
results <- list()

# Loop through each education level
for (i in seq_along(education_levels)) {
  # Subset the train and test set for the current education level
  train_subset <- subset(train_set, customer_education == education_levels[i])
  test_subset <- subset(test_set, customer_education == education_levels[i])

  # Fit the model
  fit_glm <- glm(late_return ~ book_pages, data=train_subset, family = binomial())

  # Predict
  p_hat_glm <- predict(fit_glm, newdata=test_subset, type="response")

  # Calculate ROC and best threshold
  roc_obj <- roc(response = test_subset$late_return, predictor = p_hat_glm)
  coords_obj <- coords(roc_obj, "best", ret=c("threshold", "accuracy", "sensitivity"),
                       best.method="closest.topleft")

  # Use the best threshold for prediction
  best_threshold <- coords_obj$threshold
  y_hat_glm <- ifelse(p_hat_glm > best_threshold, 1, 0)

  # Calculate metrics
  conf_matrix <- table(Predicted = y_hat_glm, Actual = test_subset$late_return)
  accuracy_glm <- sum(diag(conf_matrix)) / sum(conf_matrix)
  sensitivity_glm <- conf_matrix[2, 2] / sum(conf_matrix[2, ])
  specificity_glm <- conf_matrix[1, 1] / sum(conf_matrix[1, ])
  precision_glm <- conf_matrix[2, 2] / sum(conf_matrix[, 2])
  recall_glm <- sensitivity_glm
  f1_glm <- 2 * (precision_glm * recall_glm) / (precision_glm + recall_glm)

  # Store results
  results[[i]] <- list(
    accuracy = accuracy_glm,
    sensitivity = sensitivity_glm,
    specificity = specificity_glm,
    precision = precision_glm,
    recall = recall_glm,
    f1 = f1_glm,
    best_threshold = best_threshold
  )
}

# Convert the list of results to a data frame for easier viewing
results_df <- do.call(rbind, lapply(results, function(x) as.data.frame(t(unlist(x)))))
rownames(results_df) <- paste("Model", education_levels)
results_df
```

```
Setting levels: control = FALSE, case = TRUE
Setting direction: controls < cases
Setting levels: control = FALSE, case = TRUE
Setting direction: controls < cases
Setting levels: control = FALSE, case = TRUE
Setting direction: controls < cases
Setting levels: control = FALSE, case = TRUE
Setting direction: controls < cases
                       accuracy sensitivity specificity precision    recall
```

```
Model high school     0.6156716   0.1525424   0.9800000 0.8571429 0.1525424
Model graduate degree 0.6695279   0.1647059   0.9594595 0.7000000 0.1647059
Model others          0.6205534   0.1896552   0.9854015 0.9166667 0.1896552
Model college         0.6809339   0.2323232   0.9620253 0.7931034 0.2323232
                             f1 best_threshold
Model high school     0.2589928     0.04943255
Model graduate degree 0.2666667     0.09155432
Model others          0.3142857     0.05877012
Model college         0.3593750     0.08015868
```

Conclusions

- Factors that influence late book returns are:
  - Number of book pages
  - Customer's place of residence
  - Education
  - Occupation (although not as much as education)
  - Specific Book Title

- Recommendations for the library:
  - Start the project of arranging the database by organizing the categorization of books.
  - Create software that will disable logical errors during data entry, such as:
    - Taking a book before the time it actually happened or taking books at a time in the future that has not yet happened.
    - Returning books before the time the book was taken or returning books to a time in the future that has not yet occurred.
  - Be more flexible with regard to books that have a large number of pages and dynamically determine the limit for late book returns.
  - Provide students with benefits, such as more places for reading and more books that are in demand among the student population. Also, provide books in digital formats for this purpose.
  - Enable the return of books by mail or establish cooperation with other libraries located near the place of residence of customers to reduce the probability of late book returns.
  - Find ways to sanction late returns, especially for books where the likelihood of late returns has been extremely high in the past (e.g., books related to sales, business, and finance occupations).