WARWICK BUSINESS SCHOOL
THE UNIVERSITY OF WARWICK

Group Name **Students**	Group 32 5663068 5594410 5673899 5611086 5613045 5646841
Module Code	IB98D0
Module Title	Advanced Data Analysis
Submission Deadline	13-Feb-2025 12:00:00 PM
Date Submitted	13-Feb-2025 10:44:08 AM
Word Count	1995
Number of Pages	36
Question Attempted	Your team needs to conduct an analysis of the initial experimental data and write a report discussing your analysis and results, considering the goal of your company. You also need give recommendations to your Analytics Department Manager and the Executive team.
Have you used Artificial Intelligence (AI) in any part of this assignment?	No
If you have ticked "Yes" above, please briefly outline below which AI tool you have used, and what you have used it for. Please note, you must also reference the use of generative AI correctly within your assessment, in line with the guidance provided in your student handbook.	

Table of Contents

1. Executive Summary

This report evaluates the effectiveness of a new loan review model using A/B testing. The model significantly reduced Type I errors, improved recall, and showed potential in enhancing decision-making. While Type II errors remained largely unchanged, further validation through extended testing is recommended to assess its long-term impact.

2. Introduction

The loan approval process faces misclassification errors, particularly Type II errors, where non-viable loans are approved, leading to financial losses. This experiment evaluates whether a new computer model can mitigate such errors and enhance decision-making. An A/B test was conducted, with the existing model serving as the Control Group and the new model as the Treatment Group, to assess its effectiveness in improving loan repayment rates and reducing financial risk.

3. Experimental Design

The experiment aimed to assess whether the new computer model improves loan officers' decision accuracy by comparing outcomes between the two models.

3.1. Hypothesis

Loan reviewers who fully complete all assigned loan reviews will experience an increase in recall improvement before and after viewing AI model predictions when using the new computer model compared to the old computer model.

3.2. Overall Evaluation Criteria (OEC)

The primary OEC is the change in recall scores before and after loan reviewers view the model's predictions, assessing the model’s ability to accurately identify high-risk loans.

Supporting OECs include:

- **Weighted error rate change**: Tracks change in Type I errors (false positives) and Type II errors (false negatives), measuring the model’s impact on misclassification rates.
- **Final recall**: Compares recall performance between groups after incorporating the model’s predictions, offering a clear evaluation of its effectiveness.

This approach ensures an improvement in recall accuracy while minimizing other misclassification errors, providing a comprehensive assessment of the model's overall performance.

4. Data Preparation

Loading the dataset and required libraries into R was the initial stage in the data preparation process, followed by data cleaning, validation and transformation techniques.

4.1. Data Cleaning

The dataset contained 470 entries, all with verified values that met the problem constraints, ensuring no loan officer exceeded the loan or confidence level limit. There were 47 loan officers, 19 in the control and 28 in the treatment group.

Further analysis identified 9 control group officers who did not complete loan reviews on any given day, and this pattern of incomplete reviews was consistent across all days. To maintain data integrity, 90 cases of incomplete loan reviews were removed from the dataset.

4.2. Data Integrity

The variant column was converted to a categorical variable, and a new dataset was created by excluding loan officers who never fully completed any loan reviews, reducing the dataset to 380 entries.

The analysis focused on loan officer performance and the impact of incomplete reviews on key metrics. In 42 cases, officers did not fully review all 10 loans, though the total always summed to 10. This raised concerns about recall, precision, and F1 reliability, as it was unclear which loans were reviewed. To address this, the analysis first checked if the number of loans officers started matched the number they finished. Out of 380 cases, 338 were consistent, 42 had discrepancies.

Due to inconsistencies in loan initiation and completion, recall, precision, and F1 were unreliable for evaluation. Thus, weighted percentage error change was used for this

dataset. For final recall and recall change, only fully completed cases were included, reducing the second dataset to 330 cases.

4.3. Data Transformation

This analysis utilized two datasets: one with 380 cases, to compute the change in percentage error; and one with 330 cases, to evaluate the change in recall and final recall.

The 380-case dataset was aggregated by loan officer and variant, calculating averages for key metrics like Type I error, Type II error and mean predictions by loan type. Type I and Type II error change were determined by comparing initial and final errors. Weighted error change was calculated by assigning Type II errors (false negatives) a weight of 2/3 due to their higher financial risk, while Type I errors (false positives) were weighted at 1/3, reflecting the model's focus on avoiding high-risk mistakes.

A raincloud plot was created to compare Type I, Type II and weighted error change (Appendix A) together, while histograms helped illustrate individual distributions (Appendix C, D, E). Visualisations revealed approximately normal distributions, suitable for t-testing.

In the 330-case dataset, recall change was evaluated based on completed reviews. Loan officer and variant were aggregated, calculating means for applicable metrics. Initial recall was based on loan reviews before seeing model predictions, and final recall was after. Change in recall was calculated by finding the difference between initial and final recall to show the model’s impact on loan officer decision-making.

A scatter plot was created to compare initial and final recall by variant (Appendix B), with histograms for initial, final and change in recall showing approximately normal distributions suitable for t-testing (Appendix F, G, H).

5. Methodology

The study aimed to evaluate the effectiveness of a new AI loan assessment model by conducting various statistical tests. The primary focus was on comparing the performance of a treatment model with a control model for improvement in loan risk classification. T-tests assessed changes in errors and recall rates and mean differences were calculated.

Additionally, effect size analysis and power analysis were conducted to evaluate practical significance and statistical power.

6. Analysis and Results

6.1. Hypothesis Testing(T-tests)

- **Type I Error Change**

  A statistically significant reduction in Type I errors was observed for the treatment group (p = 0.00405), with a mean difference of -6.09% between the groups (Appendix I). This indicates that the AI model reduced rejection of low-risk loans.

- **Type II Error Change**

  No significant difference was found for Type II error change rates between the two groups (p = 0.1691), with a mean difference of -1.13% between the groups (Appendix J). This suggests that the AI model did not reduce acceptance of high-risk loans.

- **Weighted Error Change**

  The treatment group showed a statistically significant reduction in weighted errors (p = 0.00014) as shown in Figure 1, with a mean difference of -2.78% between the groups. The improvement shows overall reduction of error, though this is primarily driven by the reduction in Type I errors.

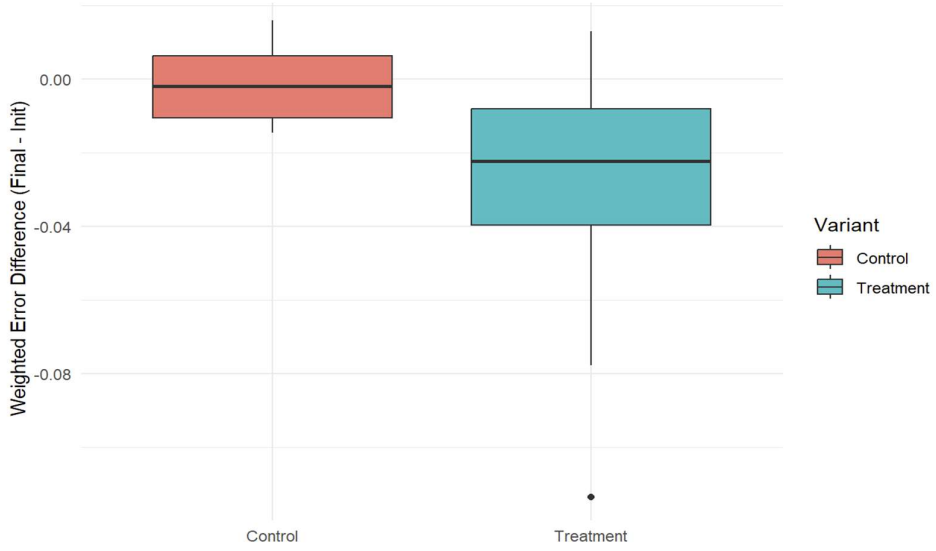


Figure 1: Box Plot of Weighted Error Change by Variant

- **Final Recall**

  A statistically significant increase in final recall was observed for the treatment group (p = 0.02264) as shown in Figure 2, with a mean difference of 11.4% between the groups. This indicates that the AI model increased the proportion of high-risk loans found.

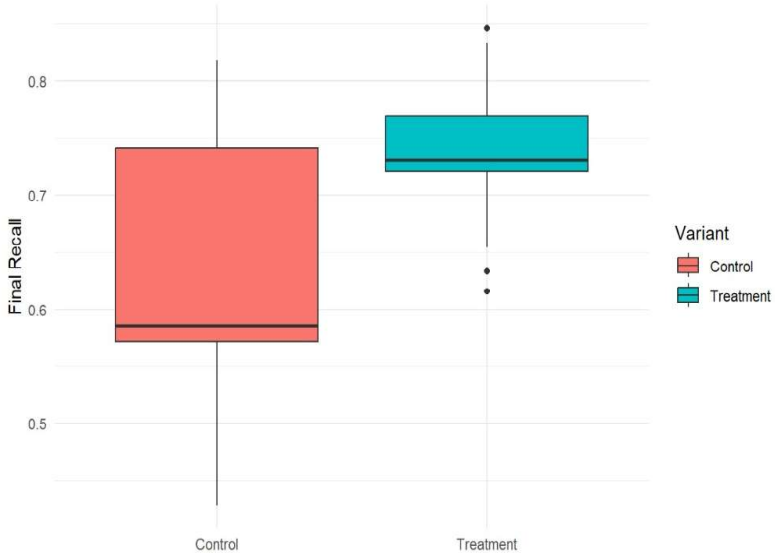


Figure 2: Box Plot of Final Recall by Variant

- **Recall Change**

  A statistically significant increase in recall change was observed for the treatment group (p = 0.01097) as shown in Figure 3, with a mean difference of 7.69% between the groups. This indicates that the AI model increased the proportion of high-risk loans found relative to initial findings.

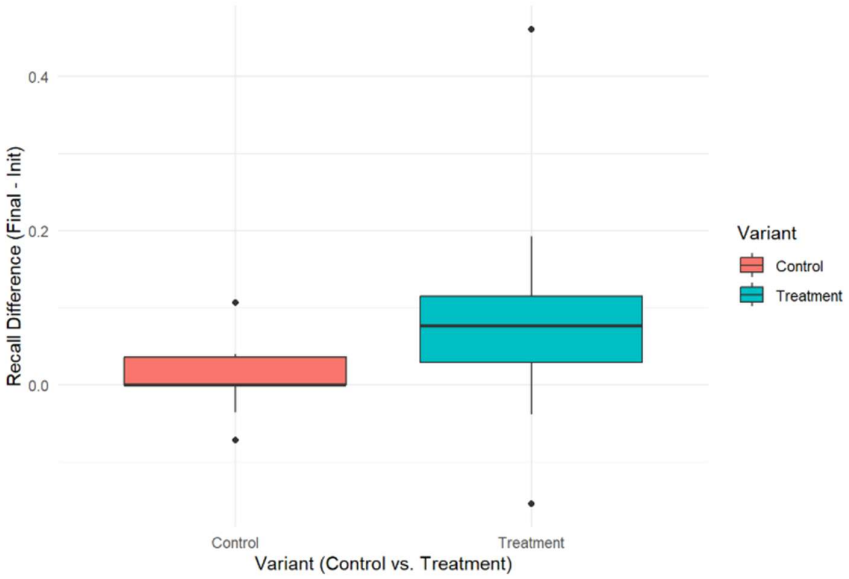


Figure 3: Box Plot of Recall Change by Variant

6.2. Effect Size

Effect size analysis provided insights into the practical significance of the findings:

- **Type I Error**: Cohen's d = -0.72, indicating a moderate effect size, suggesting that the treatment model has a noticeable impact on reducing Type I errors.

- **Type II Error**: Cohen's d = -0.36, indicating a small effect size. Practical impact of the reduction in Type II errors is minimal, suggesting that further improvements in this area may not lead to substantial real-world benefits.

- **Weighted Error**: Cohen's d = -1.08, a large effect size, demonstrating a meaningful reduction in overall errors driven by the decrease in Type I errors.

- **Recall**: Cohen's d for final recall was 1.37, indicating a large effect size, highlighting substantial practical improvement in recall. The effect size for recall change was 0.67, suggesting moderate practical significance and model improvement of recall relative to initial scores.

6.3. Power Analysis

The power analysis was conducted based on 80% power for the Type II error change and indicated for 122 loan officers per group with limited practical significance (d = -0.36). The higher practical significance of Type I error reduction and recall indicate enhancing more suitable domains.

7. Overall Interpretation

The results suggest that the new loan assessment model is more effective in reducing Type I errors than the old model, decreasing the risk of loan officers rejecting low-risk loans. Additionally, the new model was also able to improve recall scores, highlighting its ability to assist loan officers in identifying more high-risk loans overall.

However, Type II error reduction was not significantly different between the old and new loan assessment model, suggesting that the new AI system does not reduce the risk of accepting high-risk loans. The power analysis suggests that the experiment would need to be run for a much longer time to definitively determine the effect of the new model on Type II error change.

However, given the small practical significance of the metric, further development may be better focused on metrics of higher practical significance.

8. Conclusions

This study assessed the performance of a new loan classification model, yielding promising results. The model significantly improved Type I error change rate, thereby preventing the misclassification of low-risk loans as high-risk. Additionally, recall improved, allowing loan officers to better identify high-risk loans.

However, differences in Type II error change rate was not statistically significant, meaning that the new model does not improve on the issue of increased Type II error encountered with the last model. Despite this, the overall improvement in decision-making accuracy, driven by the reduction in Type I errors, led to a substantial improvement in the weighted error metric for the new model. Additionally, the proportion of high-risk loans found also increased as observed in recall change. Thus, while the absolute value of high-risk loans did not decrease between the models, the new model performs better when considering Type II error as a proportion of all high-risk loans reviewed.

It is also important to consider that the lack of significance in Type II errors may be attributed to the small sample size, as the experiment was not run for long and a large proportion of data was not usable. While there is room for refinement, the model's effectiveness in improving loan classification remains evident.

9. Recommendations

Considering the findings, the following recommendations are made:

9.1. Recommendations For the Analytics Manager

1. Stop Experiment

Based on the metrics, the new model is significantly better at loan classification than old model with this difference having practical significance. Thus, the experiment can be stopped. However, if the firm is particularly concerned by Type II error change rate, extending the experiment to increase sample size will ensure more confident conclusions.

2. Modify the Experiment Design

a. Adjust Model Thresholds

Experiment with different thresholds to optimize the balance between Type I and Type II errors. The Type I error rate, which showed a medium effect size, could be adjusted to enhance overall model performance.

b. Segmented Analysis

Assess model effectiveness across loan officer experience levels, loan types, and risk categories. This will ensure consistent performance across subgroups, provide targeted insights, and reduce sampling bias.

c. Improve Data Collection

Significant amounts of data were filtered out when calculating recall and error change metrics as seen in Figure 4. To ensure reliable analysis, better data collection is crucial, loan officers must complete all loan reviews thoroughly to evaluate model performance.

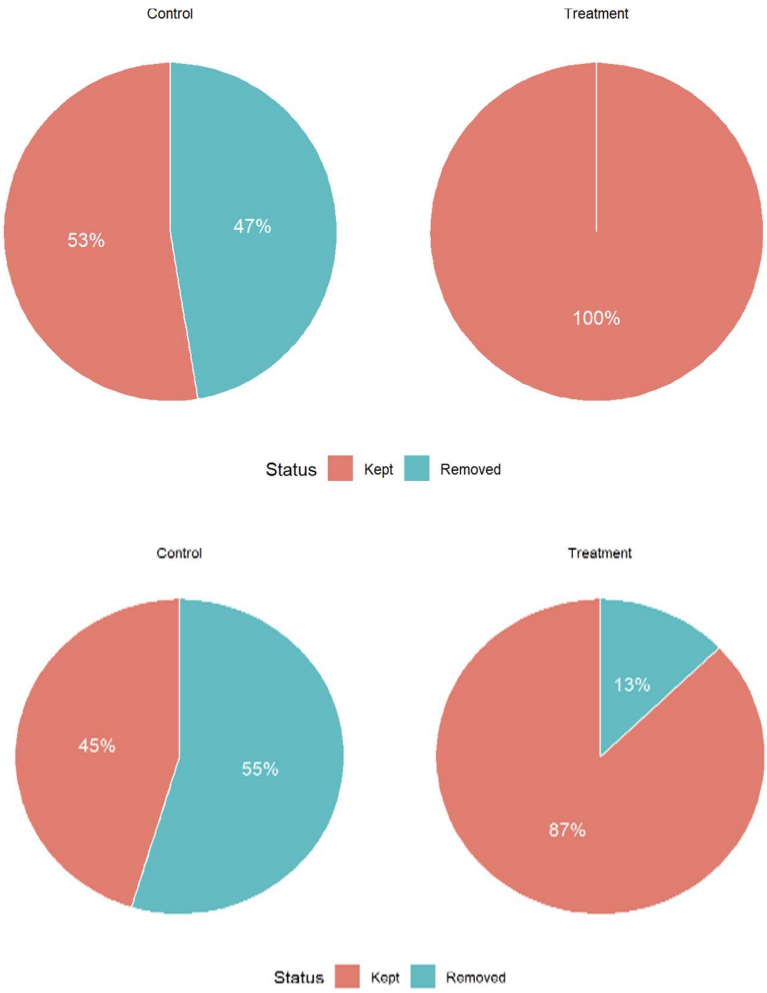


Figure 4: Pie Chart of Data Filtering
for Error Change (Top) and Recall (Bottom) Metrics

9.2. Recommendations for the Executive Team

1. Gradual Model Deployment

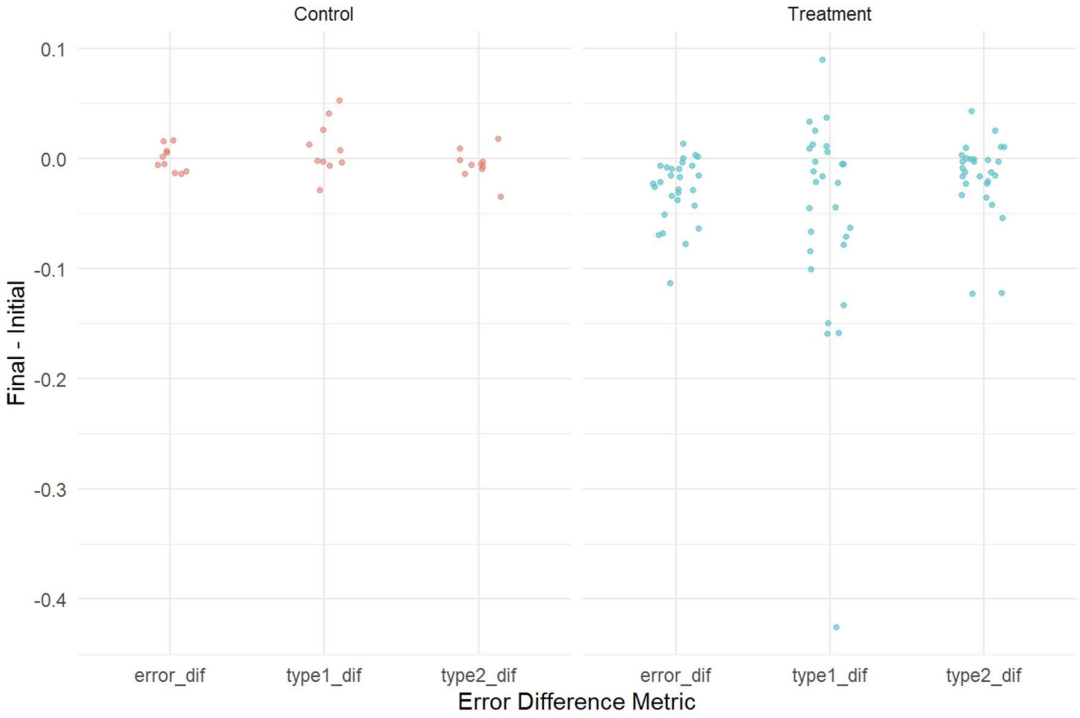
Roll out the model in phases, starting with a small loan officer group. Monitor its effectiveness, fine-tune parameters, and scale up to ensure smooth integration while minimizing risks.

2. Maximizing Cost Savings and ROI Through Data-Driven Insights

Prioritize model adoption to reduce false positives, minimizing unnecessary rejection costs and improving loan approval quality. More accurate risk identification will lower default rates, strengthening financial performance. A cost-benefit analysis will clarify ROI and guide scaling. Continuous refinement will ensure long-term effectiveness and scalability.

10. Appendices

10.1. Appendix A – Raincloud Plot of Type I, Type II, and Weighted Error Differences

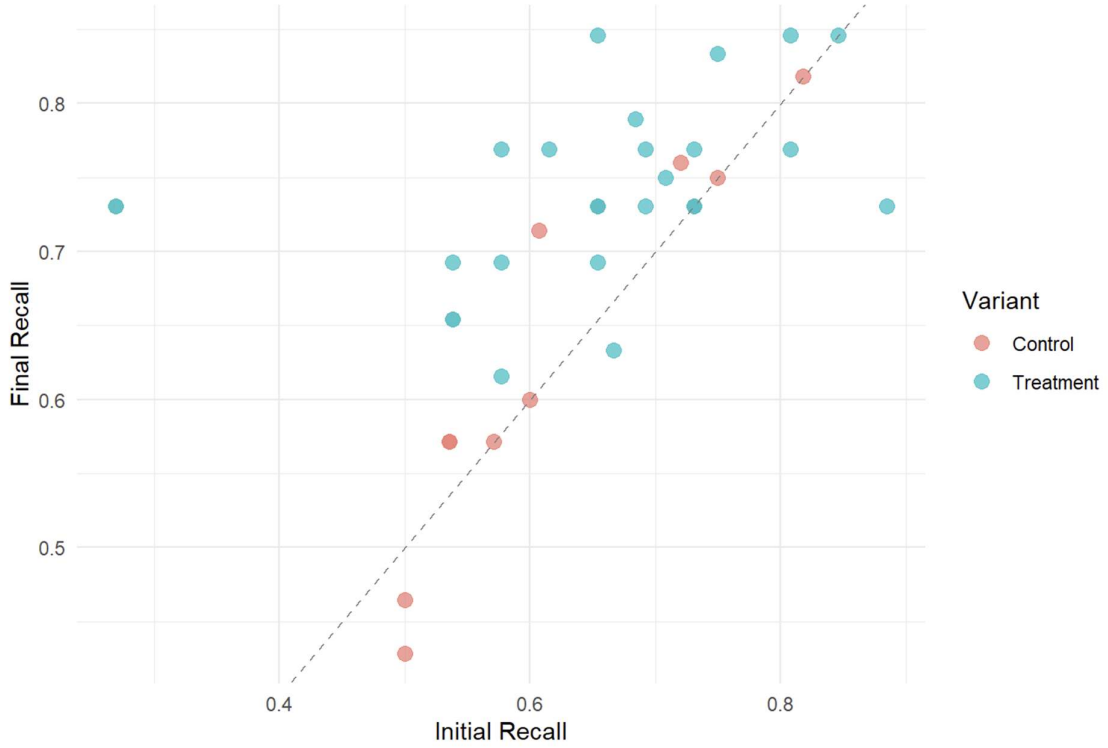


The **Raincloud Plot** displays error differences (Type I, Type II, and weighted) for **Control** vs. **Treatment** groups.

- Evidence of decreased error rate (greater variance and more negative movement), but this occurs for the Treatment group.
- The Control has a tighter cluster around zero, indicating little change.

Statistical significance not confirmed; further testing required.

10.2. Appendix B – Scatter Plot For Initial vs Final Recall By Variant

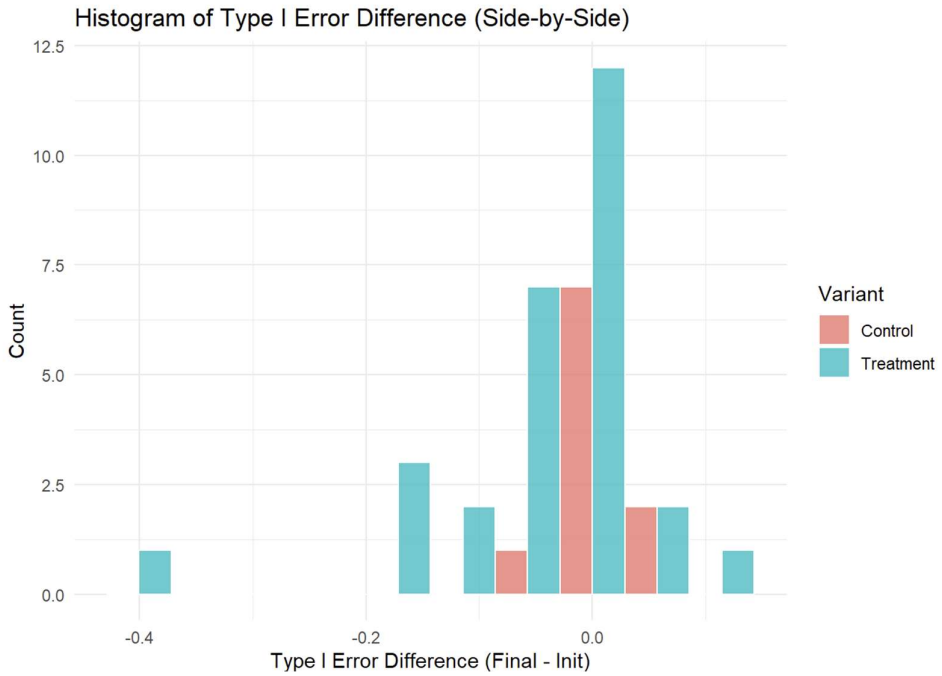


Scatter plot: **initial vs. final recall** of **Control** and **Treatment** groups.

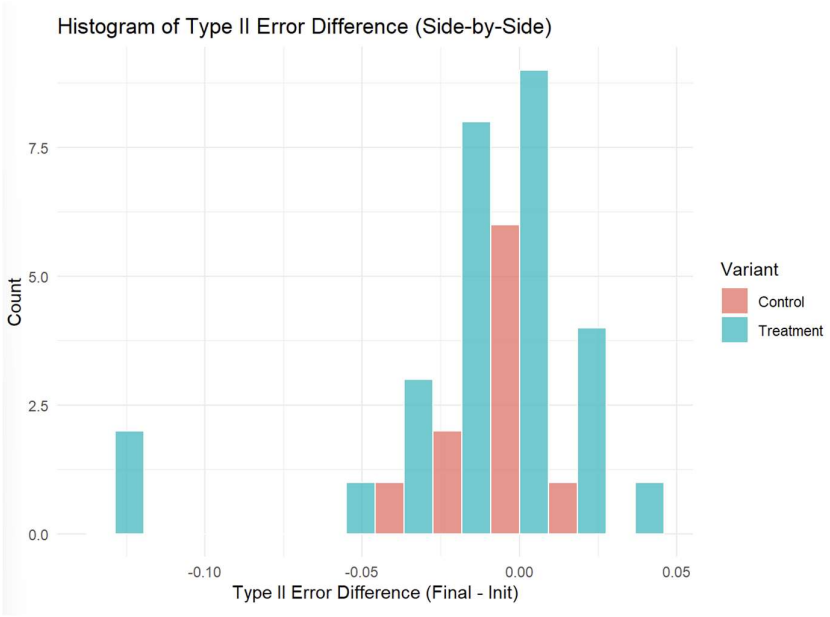
- **If they lie above the diagonal line**,
  - That indicates a recall improvement.
- The **Treatment group** (blue)
  - Has more participants above the line,
    - Indicating better recall improvement than the **Control group** (red).

This emphasizes the efficiency of the Treatment model for improving recall performance.

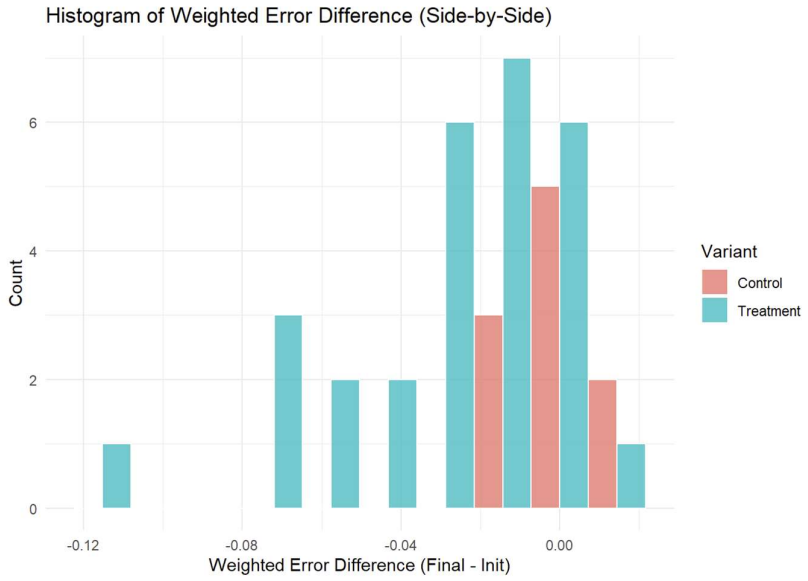
10.3. Appendix C – Histogram of Type I Error Difference By Variant



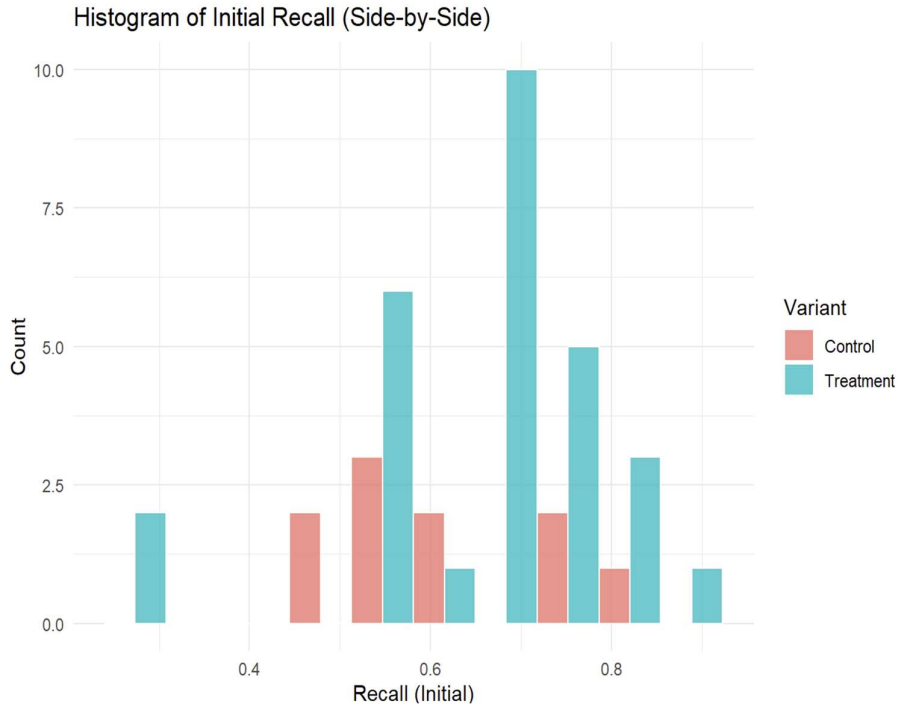
10.4. Appendix D – Histogram of Type II Error Difference By Variant



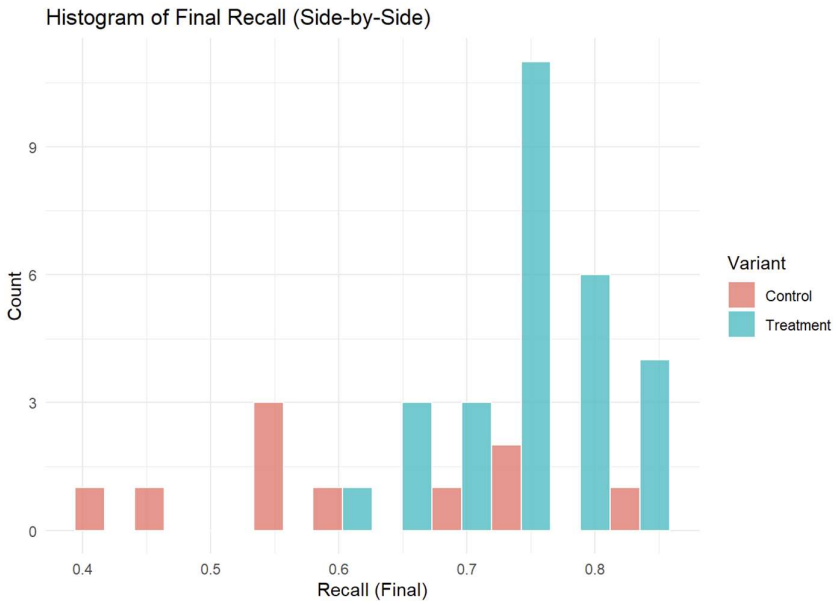
10.5. Appendix E – Histogram of Weighted Error Difference By Variant



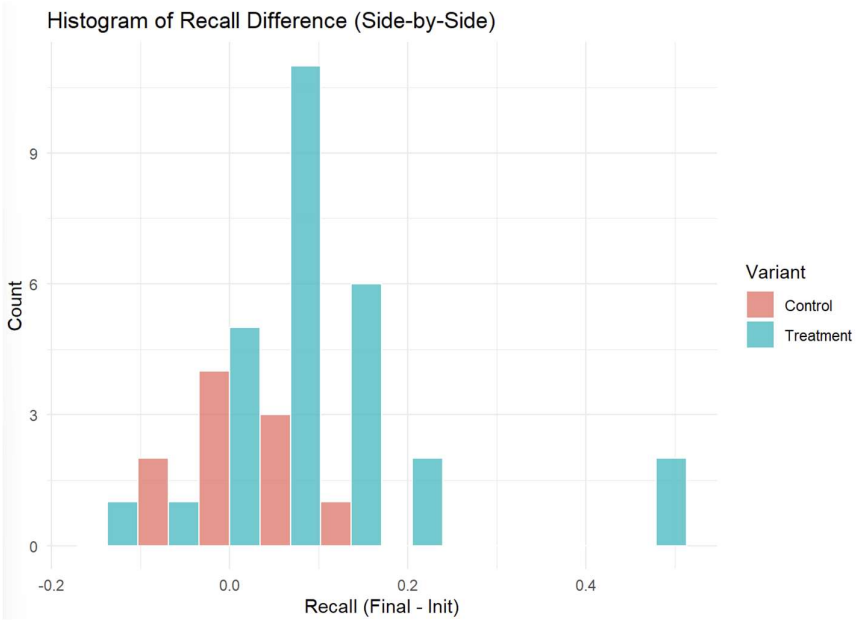
10.6. Appendix F – Histogram of Initial Recall By Variant



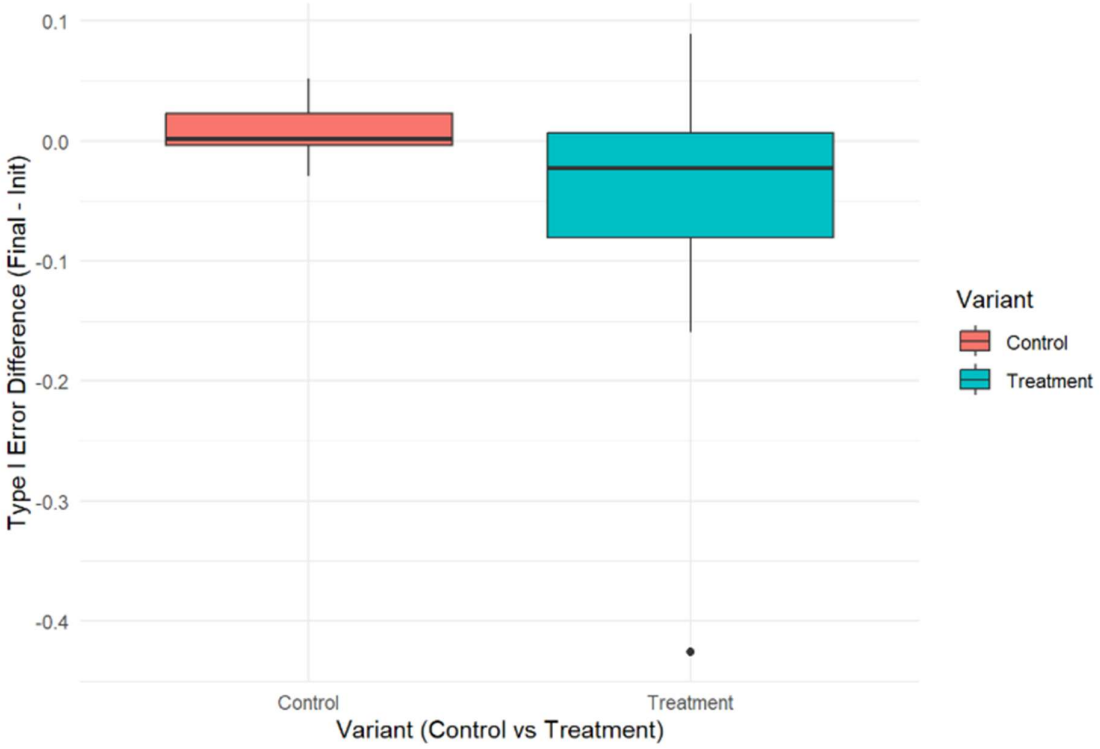
10.7. Appendix G – Histogram Of Finall Recall By Variant



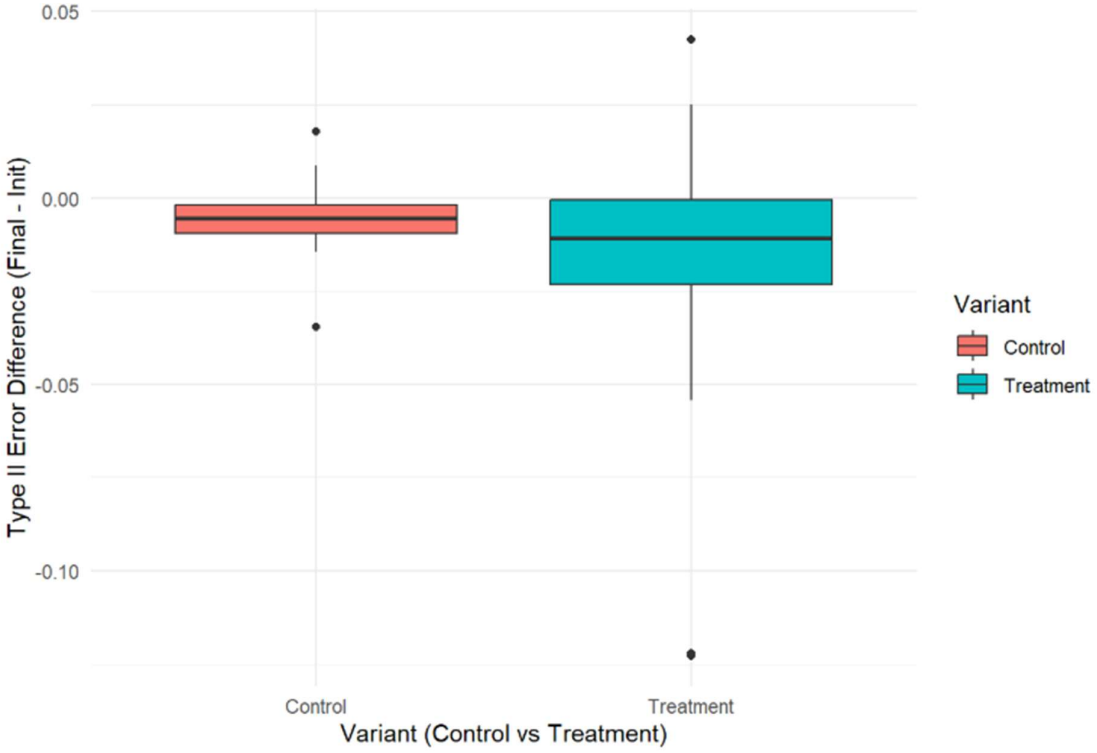
10.8. Appendix H – Histogram of Recall Difference By Variant



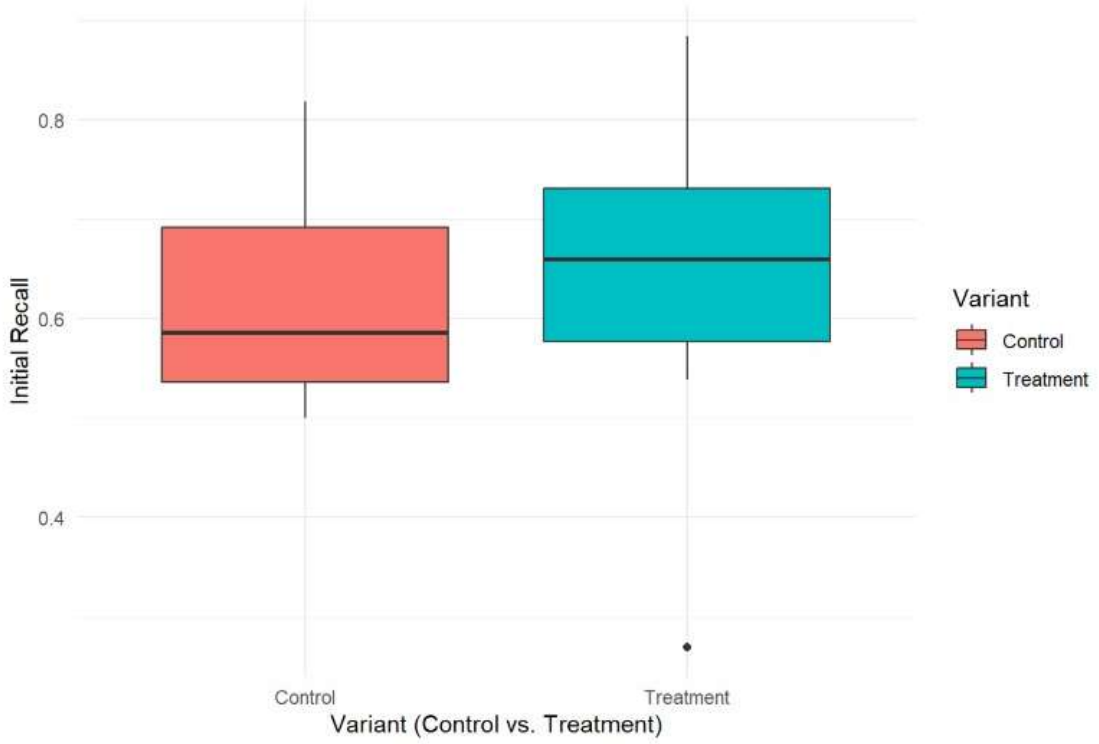
10.9. Appendix I – Boxplot of Type I Error Change Rate by Variant



10.10. Appendix J – Boxplot of Type II Error Change Rate by Variant



10.11. Appendix K – Boxplot of Initial Recall by Variant



10.12. Appendix L – Data Dictionary

Table L: Data Dictionary

The data is provided from the A/B test run within the Loan Review department. The variables are described in the table below:

Variable	Description
Variant	The experimental variant randomly assigned to each loan officer
loanofficer_id	Unique identifier for each loan officer
day	The day of the experiment (e.g. 1 means 1st day, 2 means 2nd day, etc.)
typeI_init	Count of each loan officer's Type I errors (false positives – rejecting good loans) before seeing computer predictions
typeI_fin	Count of each loan officer's Type I errors (false positives – rejecting good loans) after seeing computer predictions
typeII_init	Count of each loan officer's Type II errors (false negatives – approving bad loans) before seeing computer predictions
typeII_fin	Count of each loan officer's Type II errors (false negatives – approving bad loans) after seeing computer predictions
ai_typeI	Count of computer model's Type I errors (false positives – rejecting good loans)
ai_typeII	Count of computer model's Type II errors (false negatives – approving bad loans)
badloans_num	Number of bad loans (loans that defaulted)
goodloans_num	Number of good loans (loans that were paid back on time)
agree_init	Count of each loan officer's agreements with computer predictions before seeing computer predictions
agree_fin	Count of each loan officer's agreements with computer predictions after seeing computer predictions
conflict_init	Count of each loan officer's conflicts with computer predictions before seeing computer predictions
conflict_fin	Count of each loan officer's conflicts with computer predictions after seeing computer predictions
revised_per_ai	Count of each loan officer's decisions that were revised to follow computer predictions
revised_agst_ai	Count of each loan officer's decisions that were revised to go against computer predictions

confidence_init_total	Sum of confidence ratings given by each loan officer to their completed loan review decisions (how sure they were in their decisions) before seeing computer predictions
confidence_fin_total	Sum of confidence ratings given by each loan officer to their completed loan review decisions (how sure they were in their decisions) after seeing computer predictions
complt_init	Count of initial loan review decisions completed by each loan officer before seeing computer predictions
complt_fin	Count of final loan review decisions completed by each loan officer after seeing computer predictions
fully_complt	Count of each loan officer's fully completed loan reviews (in both stages – before and after seeing computer predictions)

10.13. Appendix M – Code

Setup

```
# Load required libraries.
library(dplyr)
library(tidyverse)
options(width = 130)
library(ggplot2)
library(effectsize)
library(pwr)
library(scales)
library(ggdist)
library(ggsignif)
```

Read Data

```
# Load and store the data.
loan_data <- read_csv("ADAproject_2025_data.csv")
```

Step 1. Data Preparation and Observations

Data Quality Check

```
# Check for number of entries.
nrow(loan_data)

# Check for missing values.
colSums(is.na(loan_data))

# Check columns for values which do not match problem requirements. Nothing
violates the 10 loan limit or 1000 score for confidence level.
summary(loan_data)

# Check for number of distinct loan officers.
n_distinct(loan_data$loanofficer_id)

# Check the number of loan officers in the control and treatment groups.
loan_data %>%
  group_by(Variant) %>%
  summarize(unique_officer_count = n_distinct(loanofficer_id))

# Check for the number of entries where no loan reviews are fully completed.
loan_data %>%
  select(Variant, loanofficer_id, fully_complt) %>%
  filter(fully_complt==0)

# It seems that many loan officers who do not fully complete any loan reviews on a
given day consistently do not fully complete loan reviews. Find all loan officers
who no not fully complete any loan reviews.
loan_data %>%
  group_by(loanofficer_id, Variant) %>%
  summarize(not_fully_completed = sum(fully_complt == 0)) %>%
  filter(not_fully_completed > 0)
```

Data Set Creation and Observations

```
# Convert Variant to a factor.
loan_data$Variant <- as.factor(loan_data$Variant)

# Create a new data set without loan officers who do not fully complete any loan
reviews.
loan_data2 <- loan_data %>%
  filter(fully_complt>0)

nrow(loan_data2)
 # Checking if a loan officer is completing a loan if it has been assigned to it.
loan_data2$complt_condition <- ifelse(loan_data2$complt_fin ==
loan_data2$fully_complt, "Yes", "No")
# Check the instances of full completion of loan reviews versus not.
loan_data2 %>%
group_by(complt_condition) %>%
  summarize(n = n())

# Checking if the number of goodloans and badloans add up to 10, i.e. if all the
loans has been marked by the loan officer or not.
loan_data2$loan_number_condition <- ifelse(loan_data2$badloans_num +
loan_data2$goodloans_num == 10, "Yes", "No")

# Check the instances of goodloans and badloans adding up to 10 versus not.
loan_data2 %>%
  group_by(loan_number_condition) %>%
  summarize(n = n())

# Checking if the number of loans the loan officer started working is the same as
the number of loans he gives his decision on.
loan_data2$agree_con_condition <- ifelse(loan_data2$agree_init +
loan_data2$conflict_init == loan_data2$agree_fin + loan_data2$conflict_fin, "Yes",
"No")

# Check the instances in which the number of loans an officer starts working on is
the same as when he finishes.
loan_data2 %>%
  group_by(agree_con_condition) %>%
  summarize(n = n())

# Checking if the loan officer finishes all the loans they start.
loan_data2$finish_condition <- ifelse(loan_data2$complt_fin >=
loan_data2$complt_init, "Yes", "No")

# Check the instances where loan officers finish loans they start versus not.
loan_data2 %>%
  group_by(finish_condition) %>%
  summarize(n = n())

# Check if the number of initial errors is less than or equal to the number of
initial loan reviews an officer makes.
loan_data2$error_init_condition <- ifelse(loan_data2$typeI_init +
loan_data2$typeII_init <= loan_data2$complt_init, "Yes", "No")

# Check the instances where initial errors are less than the number of initial
```

```
loan reviews made versus not.
loan_data2 %>%
  group_by(error_init_condition) %>%
  summarize(n = n())

# Check if the number of final errors is less than or equal to the number of final
loan reviews an officer makes.
loan_data2$error_fin_condition <- ifelse(loan_data2$typeI_fin +
loan_data2$typeII_fin <= loan_data2$complt_fin, "Yes", "No")

# Check the instances where final errors are less than the number of final loan
reviews made versus not.
loan_data2 %>%
  group_by(error_fin_condition) %>%
  summarize(n = n())

# At the end of the observations, we conclude that we cannot use recall with the
current data set, but can calculate weighted percentage error change.
 # To calculate recall, we can consider recall change rate and final recall to
compare the two models. To calculate these metrics, we need initial and final
loans reviewed to both be 10.
loan_data3 <- loan_data2 %>%
  filter(fully_complt == 10)

nrow(loan_data3)
```

Data Transformations

Percentage Error Change Rate

```
# Aggregate the data set without loan officers who do not fully complete any loan
reviews by loan officer and variant in preparation for calculating percentage
error change.
aggregated_data2 <- loan_data2 %>%
  group_by(Variant, loanofficer_id) %>%
  summarise(
    typeI_init_avg = mean(typeI_init),
    typeI_fin_avg = mean(typeI_fin),
    typeII_init_avg = mean(typeII_init),
    typeII_fin_avg = mean(typeII_fin),
    confidence_init_avg = mean(confidence_init_total / complt_init),
    confidence_fin_avg = mean(confidence_fin_total / complt_fin),
    revised_per_ai_avg = mean(revised_per_ai),
    revised_agst_ai_avg = mean(revised_agst_ai),
    agree_diff = mean(agree_fin - agree_init),
    conflict_diff = mean(conflict_fin - conflict_init),
    complt_diff = mean(complt_fin - complt_init),
    badloans_avg = mean(badloans_num),
    goodloans_avg = mean(goodloans_num),
    complt_init_avg = mean(complt_init),
    complt_fin_avg = mean(complt_fin),
    .groups = "drop"
  )

# Calculate initial and final error percentages for the data set without loan
officers who do not fully complete any loan reviews.
aggregated_data2 <- aggregated_data2%>%
```

```
  mutate(type1_init = typeI_init_avg / complt_init_avg,
         type2_init = typeII_init_avg / complt_init_avg,
         type1_fin = typeI_fin_avg / complt_fin_avg,
         type2_fin = typeII_fin_avg / complt_fin_avg
         )

# Calculate change in type I and type II error before and after looking computer
model results. The weight type I and type II error and aggregate to create a
metric that represents both error types. This is for the data set without loan
officers who do not fully complete any loan reviews.
error_change_data <- aggregated_data2%>%
  mutate(type1_dif = type1_fin - type1_init,
         type2_dif = type2_fin - type2_init,
         error_dif = 1/3*type1_dif + 2/3*type2_dif
         )
```

Percentage Error Change Rate Data Visualisations

```
# This code creates two complete pie charts to show how many records are "Removed"
(fully_complt == 0) vs. "Kept" (fully_complt > 0) within each Variant
(Control/Treatment), for the data set excluding loan officers who do not fully
complete any loan reviews. (Data set used for calculating error change rate,
remember we removed 90 rows for this from the original 470 rows)

loan_data %>%
  mutate(
    filtered_out = ifelse(fully_complt == 0, "Removed", "Kept")
  ) %>%
  group_by(Variant, filtered_out) %>%
  summarise(num_records = n(), .groups = "drop") %>%
  group_by(Variant) %>%
  mutate(prop = num_records / sum(num_records)) %>%
  ggplot(aes(x = "", y = prop, fill = filtered_out)) +
    # Stacked bar for creating the polar pie
    geom_bar(stat = "identity", width = 1, color = "white") +
    # Add text labels for the percentage in each slice
    geom_text(
      aes(label = scales::percent(prop, accuracy = 1)),
      position = position_stack(vjust = 0.5),  # Centered in each slice
      color = "white",                         # Text color, adjust if needed
      size = 4                                 # Adjust text size as you prefer
    ) +
    # Turn the stacked bar into a pie chart
    coord_polar("y", start = 0) +
    # One pie chart per variant
    facet_wrap(~ Variant) +
    # (Optional) If you want y-axis to show percentages, though typically hidden
by theme_void()
    scale_y_continuous(labels = percent_format(accuracy = 1)) +
    # Remove background, ticks, and axes for a clean look
    theme_void() +
    theme(
      legend.position = "bottom"
    ) +
    labs(
      title = "Records Kept vs. Removed per Variant",
      fill  = "Status"
    )
```

```
# Histograms for type I error change rate, type II error change rate, and weighted
error change rate.
ggplot(error_change_data, aes(x = type1_dif, fill = Variant)) +
  geom_histogram(
    bins = 10,
    position = "dodge",   # place Control vs. Treatment bars side by side
    alpha = 0.8,
    color = "white"
  ) +
  labs(
    title = "Histogram of Type I Error Difference (Side-by-Side)",
    x = "Type I Error Difference (Final - Init)",
    y = "Count"
  ) +
  theme_minimal()


ggplot(error_change_data, aes(x = type2_dif, fill = Variant)) +
  geom_histogram(
    bins = 10,
    position = "dodge",
    alpha = 0.8,
    color = "white"
  ) +
  labs(
    title = "Histogram of Type II Error Difference (Side-by-Side)",
    x = "Type II Error Difference (Final - Init)",
    y = "Count"
  ) +
  theme_minimal()

ggplot(error_change_data, aes(x = error_dif, fill = Variant)) +
  geom_histogram(
    bins = 10,
    position = "dodge",
    alpha = 0.8,
    color = "white"
  ) +
  labs(
    title = "Histogram of Weighted Error Difference (Side-by-Side)",
    x = "Weighted Error Difference (Final - Init)",
    y = "Count"
  ) +
  theme_minimal()

# Raincloud Plot for error differences.
# Reshape the data similarly.
error_long <- error_change_data %>%
  select(loanofficer_id, Variant, type1_dif, type2_dif, error_dif) %>%
  pivot_longer(
    cols = c(type1_dif, type2_dif, error_dif),
    names_to = "ErrorType",
    values_to = "DiffValue"
  )

# Create a "raincloud" style plot.
ggplot(error_long, aes(x = ErrorType, y = DiffValue, fill = Variant)) +
  stat_halfeye(
```

```
    adjust = 0.5,
    width = 0.6,
    justification = -0.2,
    .width = 0.9,
    point_interval = "mean_sd"
  ) +
  geom_jitter(
    aes(color = Variant),
    width = 0.15,
    alpha = 0.6,
    size = 1
  ) +
  facet_wrap(~ Variant, nrow = 1) +
  labs(
    title = "Raincloud Plot of Error Differences (Type I, Type II, Weighted)",
    x = "Error Difference Metric",
    y = "Final - Initial"
  ) +
  theme_minimal() +
  theme(
    legend.position = "none"
  )
```

Recall

```
# Aggregate the data set excluding loan officers who do not fully complete 10 loan
reviews by loan officer and variant in preparation for calculating recall change
rate and final recall.
aggregated_data3 <- loan_data3 %>%
  group_by(Variant, loanofficer_id) %>%
  summarise(
    typeI_init_avg = mean(typeI_init),
    typeI_fin_avg = mean(typeI_fin),
    typeII_init_avg = mean(typeII_init),
    typeII_fin_avg = mean(typeII_fin),
    confidence_init_avg = mean(confidence_init_total / complt_init),
    confidence_fin_avg = mean(confidence_fin_total / complt_fin),
    revised_per_ai_avg = mean(revised_per_ai),
    revised_agst_ai_avg = mean(revised_agst_ai),
    agree_diff = mean(agree_fin - agree_init),
    conflict_diff = mean(conflict_fin - conflict_init),
    complt_diff = mean(complt_fin - complt_init),
    badloans_avg = mean(badloans_num),
    goodloans_avg = mean(goodloans_num),
    .groups = "drop"
  )

# Calculate initial recall, final recall, and recall change for the data set
excluding loan officers who do not fully complete 10 loan reviews.
recall_data <- aggregated_data3%>%
  mutate(recall_init = (badloans_avg - typeII_init_avg) / badloans_avg,
         recall_fin = (badloans_avg - typeII_fin_avg) / badloans_avg,
         recall_dif = recall_fin - recall_init
         )
```

Recall Data Visualisations

```
# This code creates two complete pie charts to show how many records are "Removed"
(fully_complt != 10) vs. "Kept" (fully_complt == 10) within each Variant
(Control/Treatment), for the data set excluding loan officers who do not fully
complete all loan reviews. (Data set used for calculating recall and recall
change, remember we removed 140 rows for this from the original 470 rows)
loan_data %>%
mutate(
    filtered_out = ifelse(fully_complt != 10, "Removed", "Kept")
  ) %>%
  group_by(Variant, filtered_out) %>%
  summarise(num_records = n(), .groups = "drop") %>%
  group_by(Variant) %>%
  mutate(prop = num_records / sum(num_records)) %>%
  ggplot(aes(x = "", y = prop, fill = filtered_out)) +
    # Stacked bar for creating the polar pie
    geom_bar(stat = "identity", width = 1, color = "white") +
    # Add text labels for the percentage in each slice
    geom_text(
      aes(label = scales::percent(prop, accuracy = 1)),
      position = position_stack(vjust = 0.5),  # Centered in each slice
      color = "white",                         # Text color, adjust if needed
      size = 4                                 # Adjust text size as you prefer
    ) +
    # Turn the stacked bar into a pie chart
    coord_polar("y", start = 0) +
    # One pie chart per variant
    facet_wrap(~ Variant) +
    # (Optional) If you want y-axis to show percentages, though typically hidden
by theme_void()
    scale_y_continuous(labels = percent_format(accuracy = 1)) +
    # Remove background, ticks, and axes for a clean look
    theme_void() +
    theme(
      legend.position = "bottom"
    ) +
    labs(
      title = "Records Kept vs. Removed per Variant",
      fill  = "Status"
    )

# Histograms for initial recall, final recall, and recall change.
ggplot(recall_data, aes(x = recall_init, fill = Variant)) +
  geom_histogram(
    bins = 10,
    position = "dodge",
    alpha = 0.8,
    color = "white"
  ) +
  labs(
    title = "Histogram of Initial Recall (Side-by-Side)",
    x = "Recall (Initial)",
    y = "Count"
  ) +
  theme_minimal()

ggplot(recall_data, aes(x = recall_fin, fill = Variant)) +
  geom_histogram(
```

```
    bins = 10,
    position = "dodge",
    alpha = 0.8,
    color = "white"
  ) +
  labs(
    title = "Histogram of Final Recall (Side-by-Side)",
    x = "Recall (Final)",
    y = "Count"
  ) +
  theme_minimal()

ggplot(recall_data, aes(x = recall_dif, fill = Variant)) +
  geom_histogram(
    bins = 10,
    position = "dodge",
    alpha = 0.8,
    color = "white"
  ) +
  labs(
    title = "Histogram of Recall Difference (Side-by-Side)",
    x = "Recall (Final - Init)",
    y = "Count"
  ) +
  theme_minimal()

# Scatter Plot with diagonal line to compare recall_init vs. recall_fin.
ggplot(recall_data, aes(x = recall_init, y = recall_fin, color = Variant)) +
  geom_point(alpha = 0.7, size = 3) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "gray50") +
  labs(
    title = "Initial vs. Final Recall by Variant",
    x = "Initial Recall",
    y = "Final Recall"
  ) +
  theme_minimal()
```

Step 2. Data Analysis: Hypothesis Testing (T-Tests)

Percentage Error Change Rate

```
# The difference is statistically significant as p<0.05 meaning the null
hypothesis that the difference between Control and Treatment means for type I
error change rate is 0 is rejected.
t.test(
  type1_dif ~ Variant,
 data = error_change_data,
  var.equal = FALSE
)

# The difference isn't statistically significant as p> 0.05 meaning the null
hypothesis that the difference between Control and Treatment means for type II
error change rate is 0 is not rejected.
t.test(
  type2_dif ~ Variant,
  data = error_change_data,
  var.equal = FALSE
```

```
)


# The difference is statistically significant as p< 0.05 meaning the null
hypothesis that the difference between Control and Treatment means for weighted
error change rate is 0 is rejected.
t.test(
  error_dif ~ Variant,
  data = error_change_data,
  var.equal = FALSE
)
```

Recall

```
# The difference is not statistically significant as p> 0.05 meaning the null
hypothesis that the difference between Control and Treatment means for type II
error change rate is 0 is not rejected.
t.test(
  recall_init ~ Variant,
  data = recall_data,
  var.equal = FALSE
)

# The difference is statistically significant as p< 0.05 meaning the null
hypothesis that the difference between Control and Treatment means for final
recall (after looking a model predictions) is 0 is rejected.
t.test(
  recall_fin ~ Variant,
  data = recall_data,
  var.equal = FALSE
)

# The difference is statistically significant as p< 0.05 meaning the null
hypothesis that the difference between Control and Treatment means for recall
change (recall score difference before and after looking at computer model
predictions) is 0 is rejected.
t.test(
  recall_dif ~ Variant,
  data = recall_data,
  var.equal = FALSE
)
```

Step 3. Data Analysis: Compute Difference in Mean OEC between Variants

Percentage Error Change Rate

```
# Calculate mean values of different error-related metrics for each Variant of the
data set without loan officers who do not fully complete any loan reviews.
mean_error_by_variant <- error_change_data %>%
  group_by(Variant) %>%
  summarise(
    mean_type1_dif = mean(type1_dif),
    mean_type2_dif = mean(type2_dif),
    mean_error_dif = mean(error_dif),
    mean_type1_percent = mean(type1_dif) * 100,
```

```
    mean_type2_percent = mean(type2_dif) * 100,
    mean_error_percent = mean(error_dif) * 100,
    .groups = "drop"
  )

mean_comparison_errror <- mean_error_by_variant %>%
  summarise(
    diff_type1_dif = mean_type1_dif[Variant == "Treatment"] -
mean_type1_dif[Variant == "Control"],
    diff_type2_dif = mean_type2_dif[Variant == "Treatment"] -
mean_type2_dif[Variant == "Control"],
    diff_error_dif = mean_error_dif[Variant == "Treatment"] -
mean_error_dif[Variant == "Control"],
    diff_type1_per = mean_type1_percent[Variant == "Treatment"] -
mean_type1_percent[Variant == "Control"],
    diff_type2_per = mean_type2_percent[Variant == "Treatment"] -
mean_type2_percent[Variant == "Control"],
    diff_error_per = mean_error_percent[Variant == "Treatment"] -
mean_error_percent[Variant == "Control"]
  )

print(mean_comparison_errror)
```

Recall

```
# Calculate mean values of different recall metrics for each Variant of the data
set excluding loan officers who do not fully complete 10 loan reviews.
mean_recall_by_variant <- recall_data %>%
  group_by(Variant) %>%
  summarise(
    mean_recall_init = mean(recall_init),
    mean_recall_fin = mean(recall_fin), #Final recall
    mean_recall_dif = mean(recall_dif), #Change in recall
    .groups = "drop"
  )

mean_comparison_recall <- mean_recall_by_variant %>%
  summarise(
    diff_recall_init = mean_recall_init[Variant == "Treatment"] -
mean_recall_init[Variant == "Control"],
    diff_recall_fin = mean_recall_fin[Variant == "Treatment"] -
mean_recall_fin[Variant == "Control"], #Final recall
    diff_recall_dif = mean_recall_dif[Variant == "Treatment"] -
mean_recall_dif[Variant == "Control"] #Recall change
  )

print(mean_comparison_recall)
```

Step 4. Data Analysis: Compute & Interpret Effect Size

Percentage Error Change Rate

```
# Store variants for type I, type II, weighted percentage error change for data
set excluding loan officers who did not fully complete any loan reviews.
control_type1_dif <- error_change_data$type1_dif[error_change_data$Variant ==
```

```
"Control"]
print(control_type1_dif)
treatment_type1_dif <- error_change_data$type1_dif[error_change_data$Variant ==
"Treatment"]
print(treatment_type1_dif)
control_type2_dif <- error_change_data$type2_dif[error_change_data$Variant ==
"Control"]
treatment_type2_dif <- error_change_data$type2_dif[error_change_data$Variant ==
"Treatment"]

control_error_dif <- error_change_data$error_dif[error_change_data$Variant ==
"Control"]
treatment_error_dif <- error_change_data$error_dif[error_change_data$Variant ==
"Treatment"]
 # Calculate effect size for type I error change rate. Medium effect size.
cohens_d(treatment_type1_dif, control_type1_dif)
effectsize::interpret_cohens_d(-0.72)

# Calculate effect size for type II error change rate. Small effect size.
cohens_d(treatment_type2_dif, control_type2_dif)
effectsize::interpret_cohens_d(-0.36)

# Calculate effect size for weighted percentage error change rate. Large effect
size.
cohens_d(treatment_error_dif, control_error_dif)
effectsize::interpret_cohens_d(-1.08)
 # Calculate number of loan officers needed in each variant to reach 80%
statistical power for type II error change.
pwr.t.test(power = .8,
           d = .36,
           sig.level = 0.05,
           type = "two.sample")
```

Recall

```
# Store variants for recall for data set excluding loan officers who did not fully
complete 10 loan reviews.
control_recall_init <- recall_data$recall_init[recall_data$Variant == "Control"]
treatment_recall_init <- recall_data$recall_init[recall_data$Variant ==
"Treatment"]

control_recall_fin <- recall_data$recall_fin[recall_data$Variant == "Control"]
treatment_recall_fin <- recall_data$recall_fin[recall_data$Variant == "Treatment"]

control_recall_dif <- recall_data$recall_dif[recall_data$Variant == "Control"]
treatment_recall_dif <- recall_data$recall_dif[recall_data$Variant == "Treatment"]
 # Calculate effect size for recall
# Calculate effect size for initial recall. Small effect size.
cohens_d(treatment_recall_init, control_recall_init)
effectsize::interpret_cohens_d(0.28)

# Calculate effect size for final recall. Large effect size.
cohens_d(treatment_recall_fin, control_recall_fin)
effectsize::interpret_cohens_d(1.37)

# Calculate effect size for recall change. Medium effect size.
cohens_d(treatment_recall_dif, control_recall_dif)
```

```
effectsize::interpret_cohens_d(0.67)
```

Step 5. Visualization of Key Metrics

Percentage Error Change Rate

```
# Visualize the distribution of Type I / Type II error differences and recall
metrics across Control vs. Treatment groups using boxplots.
ggplot(error_change_data, aes(x = Variant, y = type1_dif, fill = Variant)) +
  geom_boxplot() +
  labs(
    title = "Boxplot of Type I Error Difference by Variant",
    x = "Variant (Control vs Treatment)",
    y = "Type I Error Difference (Final - Init)"
  ) +
  theme_minimal()

ggplot(error_change_data, aes(x = Variant, y = type2_dif, fill = Variant)) +
  geom_boxplot() +
  labs(
    title = "Boxplot of Type II Error Difference by Variant",
    x = "Variant (Control vs Treatment)",
    y = "Type II Error Difference (Final - Init)"
  ) +
  theme_minimal()

ggplot(error_change_data, aes(x = Variant, y = error_dif, fill = Variant)) +
  geom_boxplot() +
  labs(
    title = "Boxplot of Weighted Error Difference by Variant",
    x = "Variant (Control vs Treatment)",
    y = "Weighted Error Difference (Final - Init)"
  ) +
  theme_minimal()
```

Recall

```
# Visualize the Boxplot for Recalls by Variant
# Boxplot for Initial Recall by Variant
ggplot(recall_data, aes(x = Variant, y = recall_init, fill = Variant)) +
  geom_boxplot() +
  labs(
    title = "Boxplot of Initial Recall by Variant",
    x = "Variant (Control vs. Treatment)",
    y = "Initial Recall"
  ) +
  theme_minimal()

# Boxplot for Final Recall by Variant
ggplot(recall_data, aes(x = Variant, y = recall_fin, fill = Variant)) +
  geom_boxplot() +
  labs(
    title = "Boxplot of Final Recall by Variant",
    x = "Variant (Control vs. Treatment)",
    y = "Final Recall"
```

```
  ) +
  theme_minimal()

# Boxplot for Recall Difference (Final - Initial) by Variant
ggplot(recall_data, aes(x = Variant, y = recall_dif, fill = Variant)) +
  geom_boxplot() +
  labs(
    title = "Boxplot of Recall Difference by Variant",
    x = "Variant (Control vs. Treatment)",
    y = "Recall Difference (Final - Init)"
  ) +
  theme_minimal()
```