Experiment 3

Question 3

0.1 Outlier Detection & Handling

0.1.1 Preprocessing

[9]:

```
library(tidyverse)
```

[10]:

```
setwd("/home/asus/content/Notes/Semester 4/FDN Lab/Experiments/Experiment␣
 ↪3")
```

[11]:

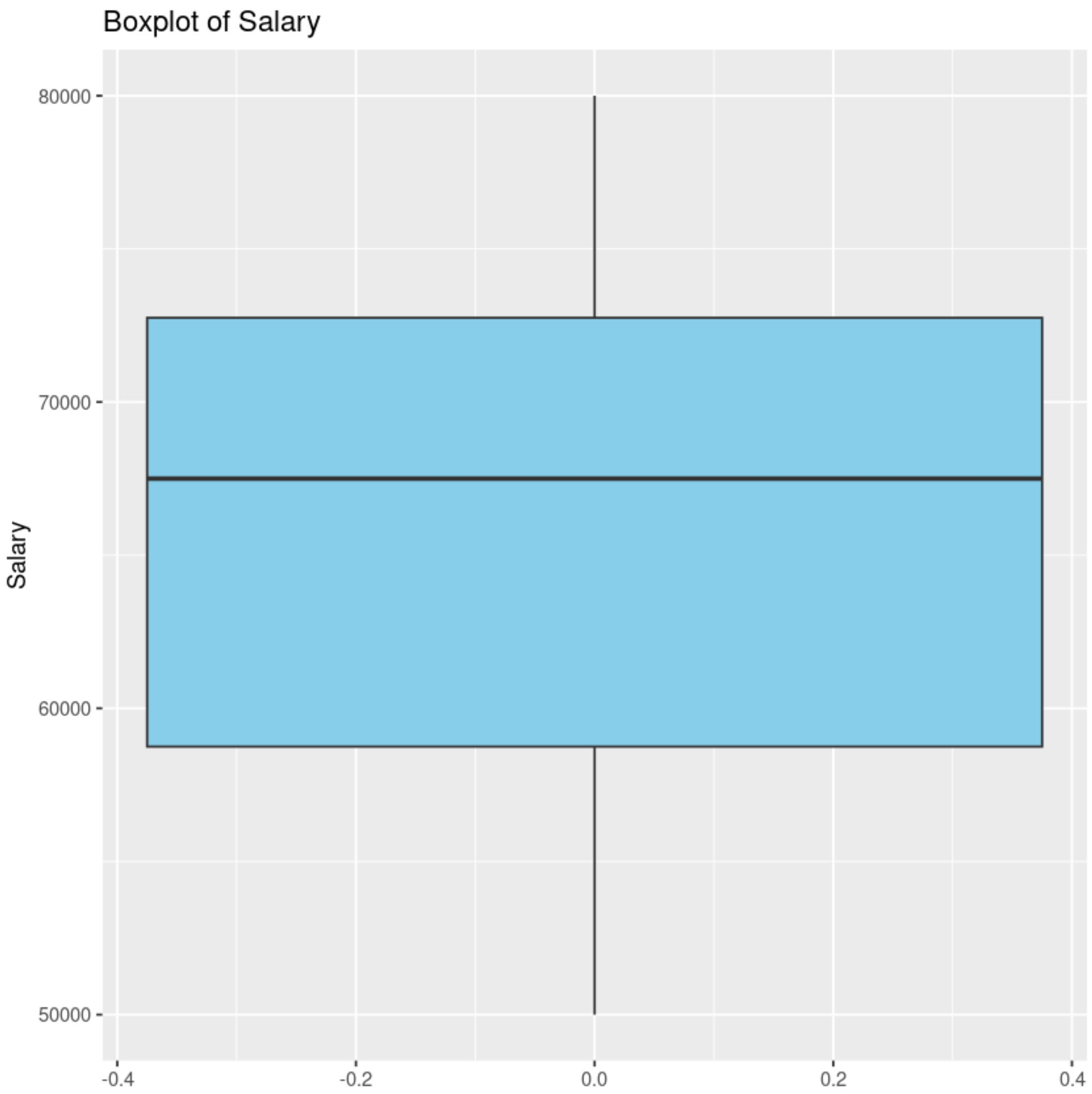
```
df_mean <- data.frame(
  ID = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10),
  Name = c("Alice", "Bob", NA, "David", "Emma", "Frank", NA, "Hannah",␣
 ↪"Ian", "Jack"),
  Age = c(25, NA, 30, 29, NA, 35, 40, NA, 50, 27),
  Salary = c(50000, 60000, 55000, NA, 70000, 75000, 80000, 65000, NA,␣
 ↪72000),
  Score = c(80, 90, NA, 85, 88, 92, NA, 77, 95, Inf)
)
```

Boxplot Visualization to visualize salary data

[12]:

```
# i. Boxplot Visualization to visualize Salary data
ggplot(df_mean, aes(y = Salary)) +
  geom_boxplot(fill = "skyblue", outlier.color = "red", outlier.shape =␣
 ↪16) +
  labs(title = "Boxplot of Salary", y = "Salary")
```

```
Warning message:
“Removed 2 rows containing non-finite outside the scale range
(`stat_boxplot()`).”
```



Z-Score Method (values outside ±3 standard deviations).

```
[13]: # ii. Z-Score Method (Values outside ±3 standard deviations)
df_mean_z <- df_mean %>%
  mutate(Salary_Z = as.numeric(scale(Salary))) %>%  # Convert scale␣
 ↪output to numeric
  filter(abs(Salary_Z) <= 3) %>%  # Remove outliers
  select(-Salary_Z)  # Remove Z-score column
print(df_mean_z)
```

```
  ID  Name Age Salary Score
1  1 Alice  25  50000    80
2  2   Bob  NA  60000    90
3  3  <NA>  30  55000    NA
```

```
4  5   Emma  NA  70000    88
5  6  Frank  35  75000    92
6  7   <NA>  40  80000    NA
7  8 Hannah  NA  65000    77
8 10   Jack  27  72000   Inf
```

iii. IQR Method: Remove values outside Q1 - 1.5$IQR\ and\ Q3 + 1.5$IQR.

[15]:

```
# iii. IQR Method: Remove values outside Q1 - 1.5*IQR and Q3 + 1.5*IQR
Q1 <- quantile(df_mean$Salary, 0.25, na.rm=TRUE)
Q3 <- quantile(df_mean$Salary, 0.75, na.rm=TRUE)
IQR_value <- Q3 - Q1
lower_bound <- Q1 - 1.5 * IQR_value
upper_bound <- Q3 + 1.5 * IQR_value
```

[16]:

```
df_mean_iqr <- df_mean %>%
  filter(Salary >= lower_bound & Salary <= upper_bound)
```

iv. Winsorization: Replace extreme values with percentiles (Winsorize()).

[8]:

```
# iv. Winsorization: Replace extreme values with 5th and 95th percentiles
library(DescTools)
df_mean_winsorized <- df_mean %>%
  mutate(Salary = Winsorize(Salary, probs = c(0.05, 0.95)))
```

```
Error in `mutate()`:
 In argument: `Salary = Winsorize(Salary, probs = c(0.05, 0.95))`.
Caused by error in `Winsorize()`:
! unused argument (probs = c(0.05, 0.95))
Traceback:

1. mutate(., Salary = Winsorize(Salary, probs = c(0.05, 0.95)))
2. mutate.data.frame(., Salary = Winsorize(Salary, probs = c(0.05,
 .     0.95)))
3. mutate_cols(.data, dplyr_quosures(...), by)
4. withCallingHandlers(for (i in seq_along(dots)) {
 .     poke_error_context(dots, i, mask = mask)
 .     context_poke("column", old_current_column)
 .     new_columns <- mutate_col(dots[[i]], data, mask, new_columns)
 . }, error = dplyr_error_handler(dots = dots, mask = mask, bullets =␣
 ↪mutate_bullets,
 .     error_call = error_call, error_class = "dplyr:::mutate_error"),
 .     warning = dplyr_warning_handler(state = warnings_state, mask = mask
 .         error_call = error_call))
5. mutate_col(dots[[i]], data, mask, new_columns)
6. mask$eval_all_mutate(quo)
```

```
7. eval()
8. .handleSimpleError(function (cnd)
 . {
 .     local_error_context(dots, i = frame[[i_sym]], mask = mask)
 .     if (inherits(cnd, "dplyr:::internal_error")) {
 .         parent <- error_cnd(message = bullets(cnd))
 .     }
 .     else {
 .         parent <- cnd
 .     }
 .     message <- c(cnd_bullet_header(action), i = if␣
 ↪(has_active_group_context(mask)) cnd_bullet_cur_group_label())
 .     abort(message, class = error_class, parent = parent, call =␣
 ↪error_call)
 . }, "unused argument (probs = c(0.05, 0.95))", base::
 ↪quote(Winsorize(Salary,
 .     probs = c(0.05, 0.95))))
9. h(simpleError(msg, call))
10. abort(message, class = error_class, parent = parent, call = error_call
11. signal_abort(cnd, .file)
12. signalCondition(cnd)
```

v. Detect & Remove Outliers Using tidyverse (filter())

[17]:
```
# v. Detect & Remove Outliers Using tidyverse (filter method)
df_mean_tidy_outliers <- df_mean %>%
  filter(between(Salary, lower_bound, upper_bound))
```

vi. Detect Outliers in Multiple Columns (apply()).

[19]:
```
# vi. Detect Outliers in Multiple Columns using apply() (Z-score method)
detect_outliers <- function(x) {
  if (is.numeric(x)) {
    z_scores <- scale(x)
    return(abs(z_scores) > 3)
  } else {
    return(rep(FALSE, length(x)))
  }
}

outlier_matrix <- apply(df_mean, 2, detect_outliers)
df_mean_clean <- df_mean[!rowSums(outlier_matrix), ]  # Remove rows with␣
 ↪outliers
```

vii. Create a Clean Dataset After Removing Outliers

[21]:

```
# vii. Create a Clean Dataset After Removing Outliers
df_mean_final <- df_mean_iqr  # Using IQR method for final clean dataset
write.csv(df_mean_final, "Clean_Dataset.csv", row.names = FALSE)
```