R Notebook

```
library(english)
source("C:\\xampp\\htdocs\\TacticalDataDrivenDecisionSystem\\r_library\\ch2.r")
source("C:\\xampp\\htdocs\\TacticalDataDrivenDecisionSystem\\r_library\\ch3.r")
source("C:\\xampp\\htdocs\\TacticalDataDrivenDecisionSystem\\r_library\\ch4.r")




# Load required packages
library(RMySQL)
```

```
## Loading required package: DBI
```

```
library(ggplot2)
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(tidyr)
library(lubridate)
```

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

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

```r
library(ggpubr)


# Database connection
con <- dbConnect(MySQL(),
                 user = "root",         # Replace with your MySQL username
                 password = "",         # Replace with your MySQL password
                 dbname = "pcc_forecast_db",
                 host = "127.0.0.1")

# Fetch data from all tables
accom_data <- dbGetQuery(con, "SELECT * FROM htaaccommodationchoices")
air_data <- dbGetQuery(con, "SELECT * FROM htaallvisitorsbyair")
travel_data <- dbGetQuery(con, "SELECT * FROM htamethoodoftrave")
purpose_data <- dbGetQuery(con, "SELECT * FROM htapurposeoftrip")
ticket_data <- dbGetQuery(con, "SELECT * FROM htaticketing")
tenant_data <- dbGetQuery(con, "SELECT * FROM tenantsales")
```

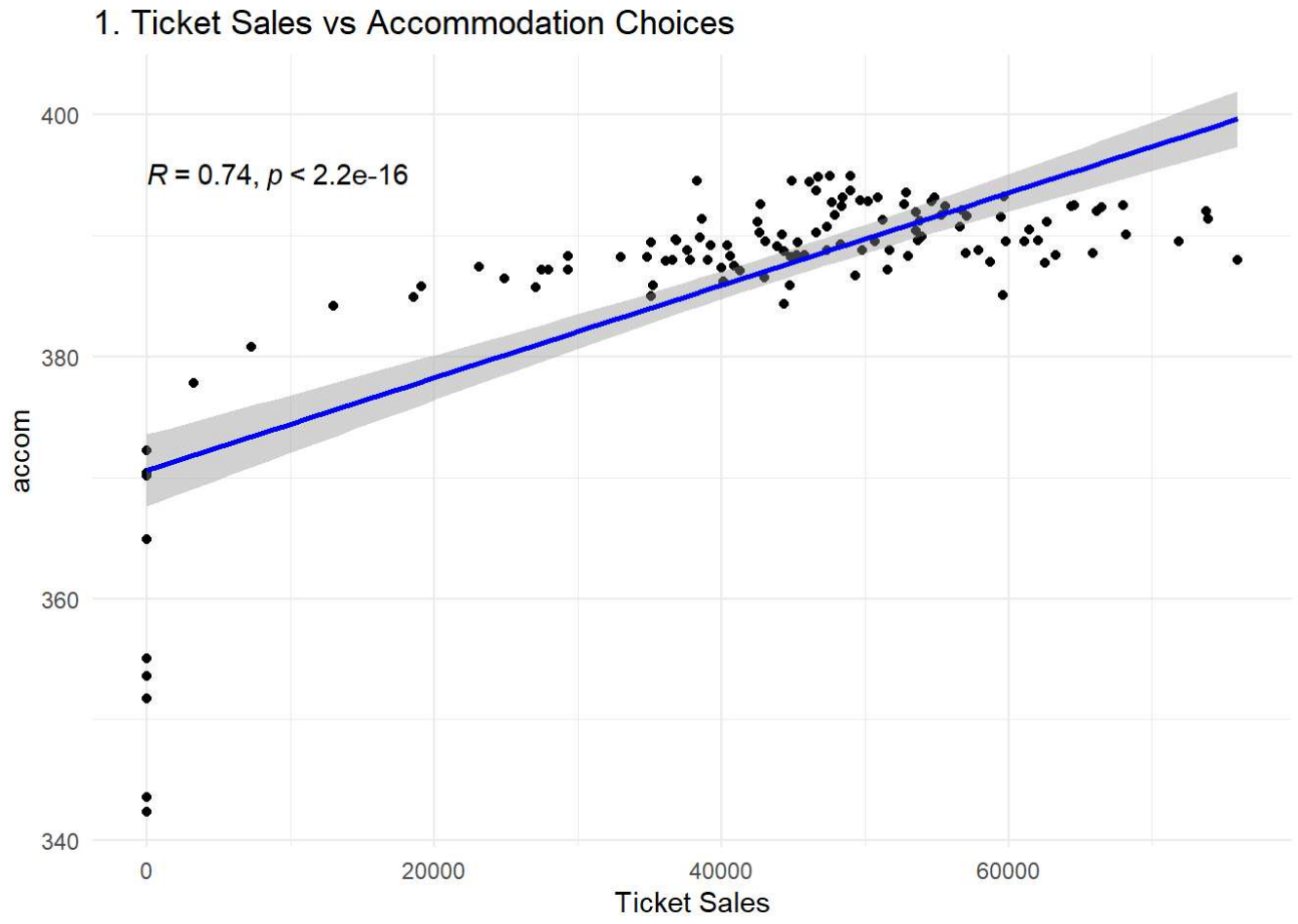
```
# Prepare ticket_data with a month-year column
if(nrow(ticket_data) > 0) {
  ticket_data$date <- as.Date(paste(ticket_data$Year, ticket_data$Month, "01", sep = "-"), forma
t = "%Y-%B-%d")
  ticket_data$month_year <- format(ticket_data$date, "%Y-%m")
  ticket_sum <- ticket_data %>%
    group_by(month_year) %>%
    summarise(total_tickets = sum(TicketSell, na.rm = TRUE))
}

# Function to calculate and plot correlation
plot_correlation <- function(data1, data2, x_var, y_var, title, group_var = NULL) {
  if(nrow(data1) > 0 && nrow(data2) > 0) {
    # Merge data by month_year
    merged_data <- merge(data1, data2, by = "month_year", all = FALSE)

    # Calculate correlation
    cor_value <- cor(merged_data[[x_var]], merged_data[[y_var]], use = "complete.obs", method =
"pearson")
    cat(sprintf("%s - Pearson Correlation: %.3f\n", title, cor_value))

    # Create scatter plot
    if(is.null(group_var)) {
      p <- ggplot(merged_data, aes_string(x = x_var, y = y_var)) +
        geom_point() +
        geom_smooth(method = "lm", color = "blue") +
        labs(title = title, x = "Ticket Sales", y = gsub("total_", "", y_var)) +
        stat_cor(method = "pearson", label.x = min(merged_data[[x_var]], na.rm = TRUE),
                 label.y = max(merged_data[[y_var]], na.rm = TRUE)) +
        theme_minimal()
    } else {
      p <- ggplot(merged_data, aes_string(x = x_var, y = y_var, color = group_var)) +
        geom_point() +
        geom_smooth(method = "lm") +
        labs(title = title, x = "Ticket Sales", y = gsub("total_", "", y_var)) +
        stat_cor(method = "pearson", aes_string(color = group_var), label.x = min(merged_data[[x
_var]], na.rm = TRUE)) +
        theme_minimal()
    }
    print(p)
    Sys.sleep(1)
    return(cor_value)
  } else {
    cat(sprintf("%s - Insufficient data for correlation\n", title))
    return(NA)
  }
}

# 1. Ticket Sales vs Accommodation Choices (by Group)
if(nrow(accom_data) > 0 && nrow(ticket_data) > 0) {
  accom_sum <- accom_data %>%
    group_by(`YYYY-MM`) %>%
```

```
    summarise(total_accom = sum(value, na.rm = TRUE)) %>%
    rename(month_year = `YYYY-MM`)
  cor_accom <- plot_correlation(ticket_sum, accom_sum, "total_tickets", "total_accom",
                                "1. Ticket Sales vs Accommodation Choices")
}
```

```
## 1. Ticket Sales vs Accommodation Choices - Pearson Correlation: 0.742
```

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## ℹ Please use tidy evaluation idioms with `aes()`.
## ℹ See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

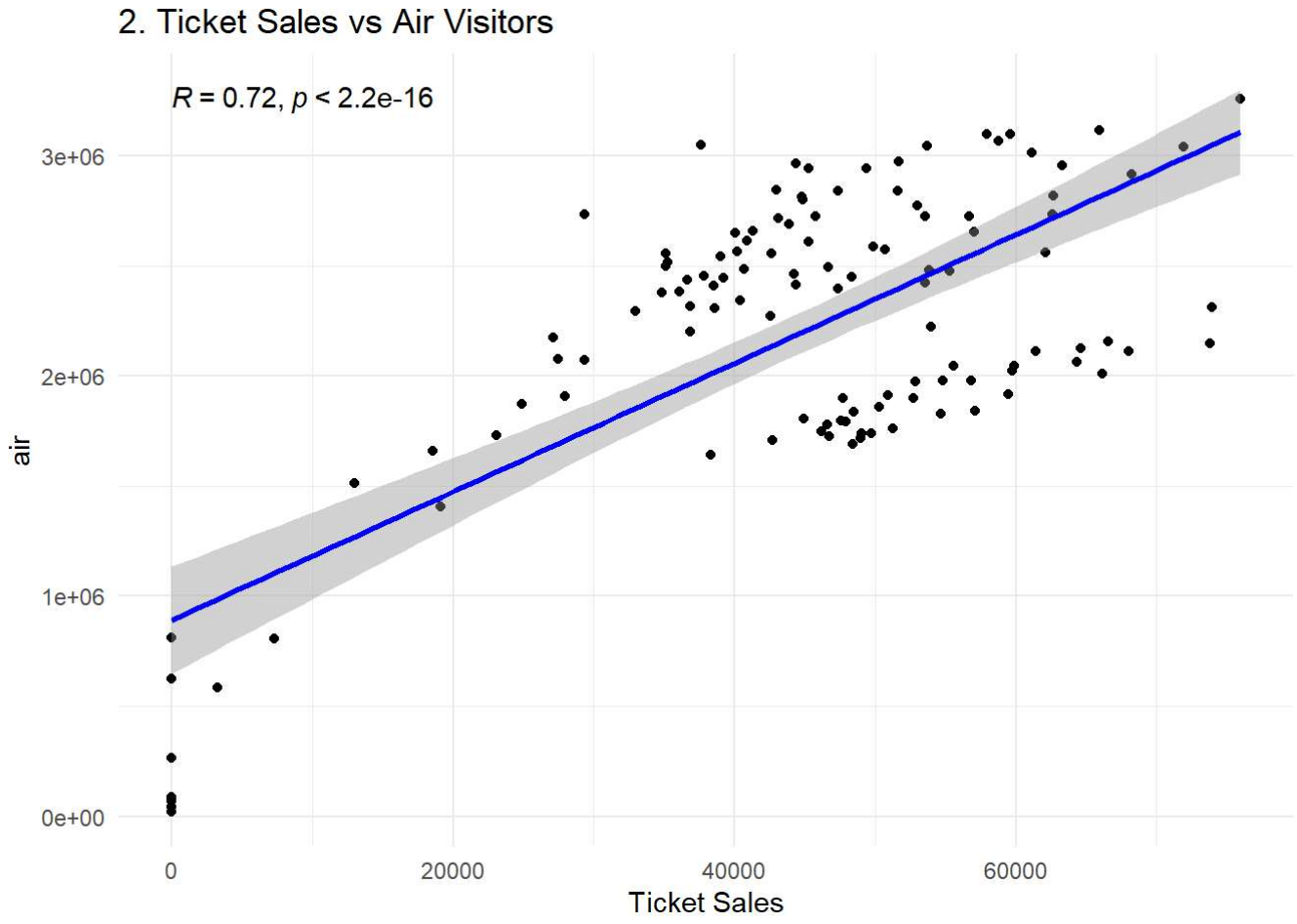
```
## `geom_smooth()` using formula = 'y ~ x'
```



```
# 2. Ticket Sales vs Air Visitors (by Group)
if(nrow(air_data) > 0 && nrow(ticket_data) > 0) {
  air_sum <- air_data %>%
    group_by(`YYYY-MM`) %>%
    summarise(total_air = sum(value, na.rm = TRUE)) %>%
    rename(month_year = `YYYY-MM`)
  cor_air <- plot_correlation(ticket_sum, air_sum, "total_tickets", "total_air",
                              "2. Ticket Sales vs Air Visitors")
}
```

```
## 2. Ticket Sales vs Air Visitors - Pearson Correlation: 0.717
```

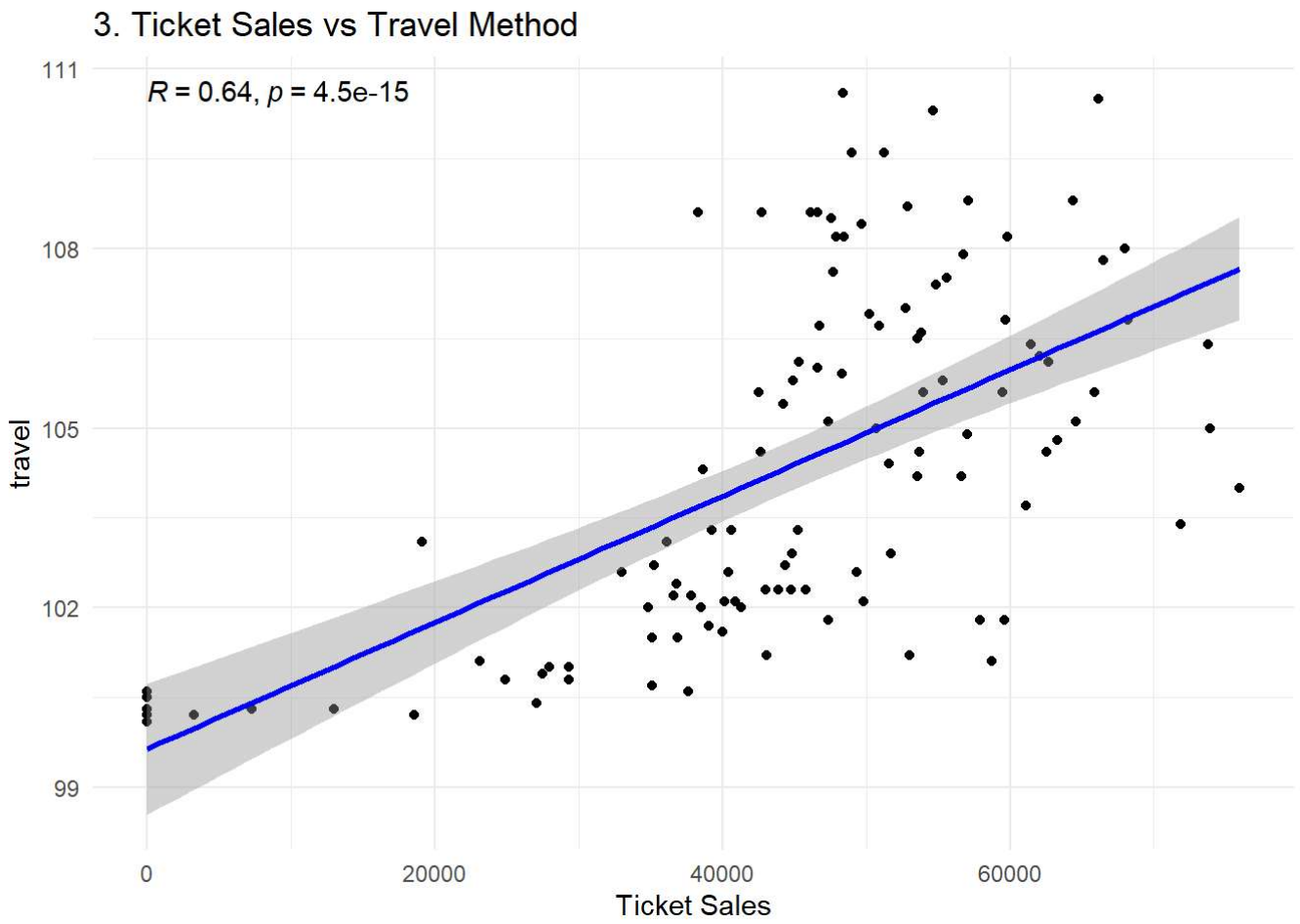
```
## `geom_smooth()` using formula = 'y ~ x'
```




```
# 3. Ticket Sales vs Travel Method (by Group)
if(nrow(travel_data) > 0 && nrow(ticket_data) > 0) {
  travel_sum <- travel_data %>%
    group_by(`YYYY-MM`) %>%
    summarise(total_travel = sum(value, na.rm = TRUE)) %>%
    rename(month_year = `YYYY-MM`)
  cor_travel <- plot_correlation(ticket_sum, travel_sum, "total_tickets", "total_travel",
                                 "3. Ticket Sales vs Travel Method")
}
```

```
## 3. Ticket Sales vs Travel Method - Pearson Correlation: 0.638
```

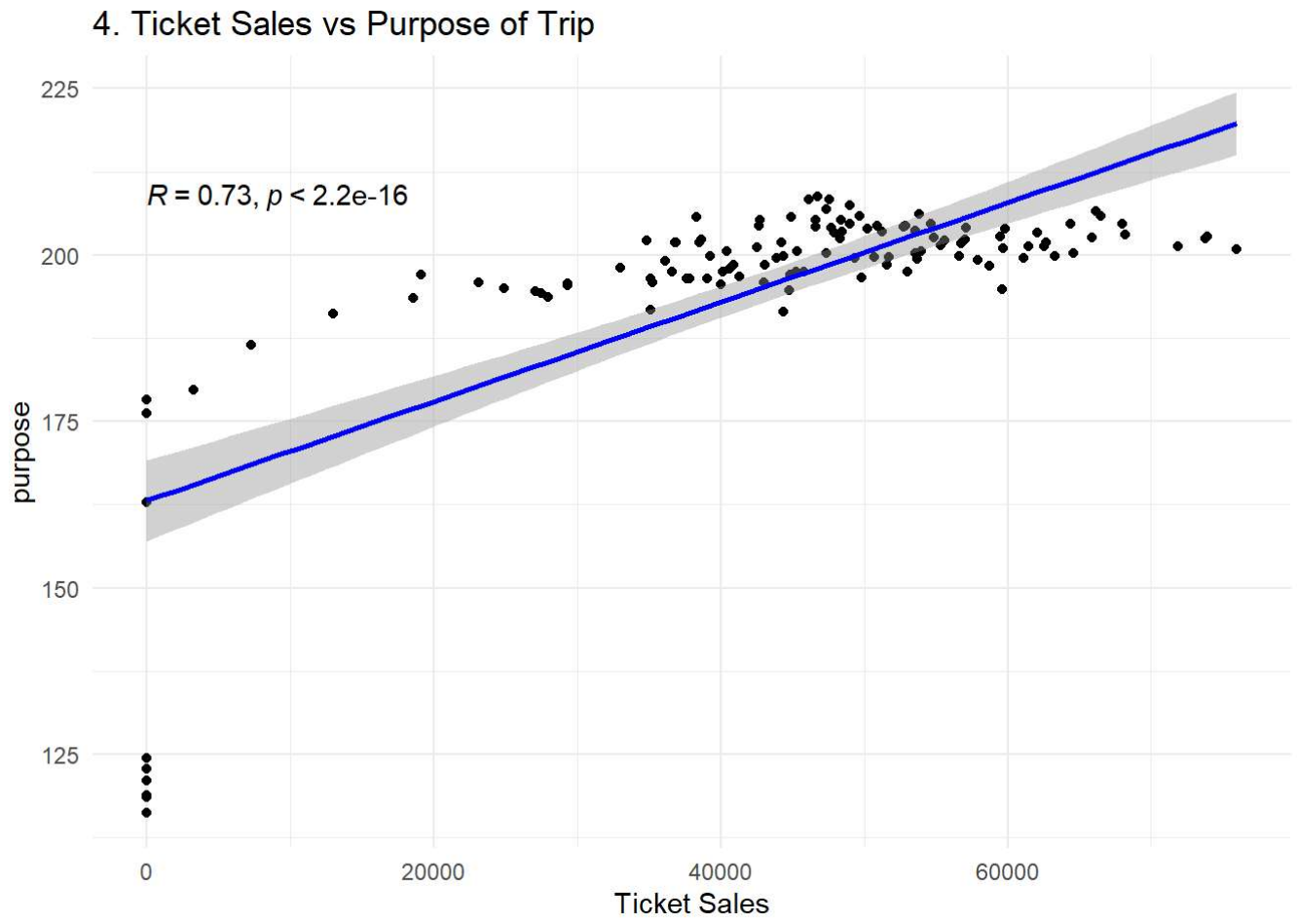
```
## `geom_smooth()` using formula = 'y ~ x'
```




```
# 4. Ticket Sales vs Purpose of Trip (by Group)
if(nrow(purpose_data) > 0 && nrow(ticket_data) > 0) {
  purpose_sum <- purpose_data %>%
    group_by(`YYYY-MM`) %>%
    summarise(total_purpose = sum(value, na.rm = TRUE)) %>%
    rename(month_year = `YYYY-MM`)
  cor_purpose <- plot_correlation(ticket_sum, purpose_sum, "total_tickets", "total_purpose",
                                  "4. Ticket Sales vs Purpose of Trip")
}
```

```
## 4. Ticket Sales vs Purpose of Trip - Pearson Correlation: 0.727
```

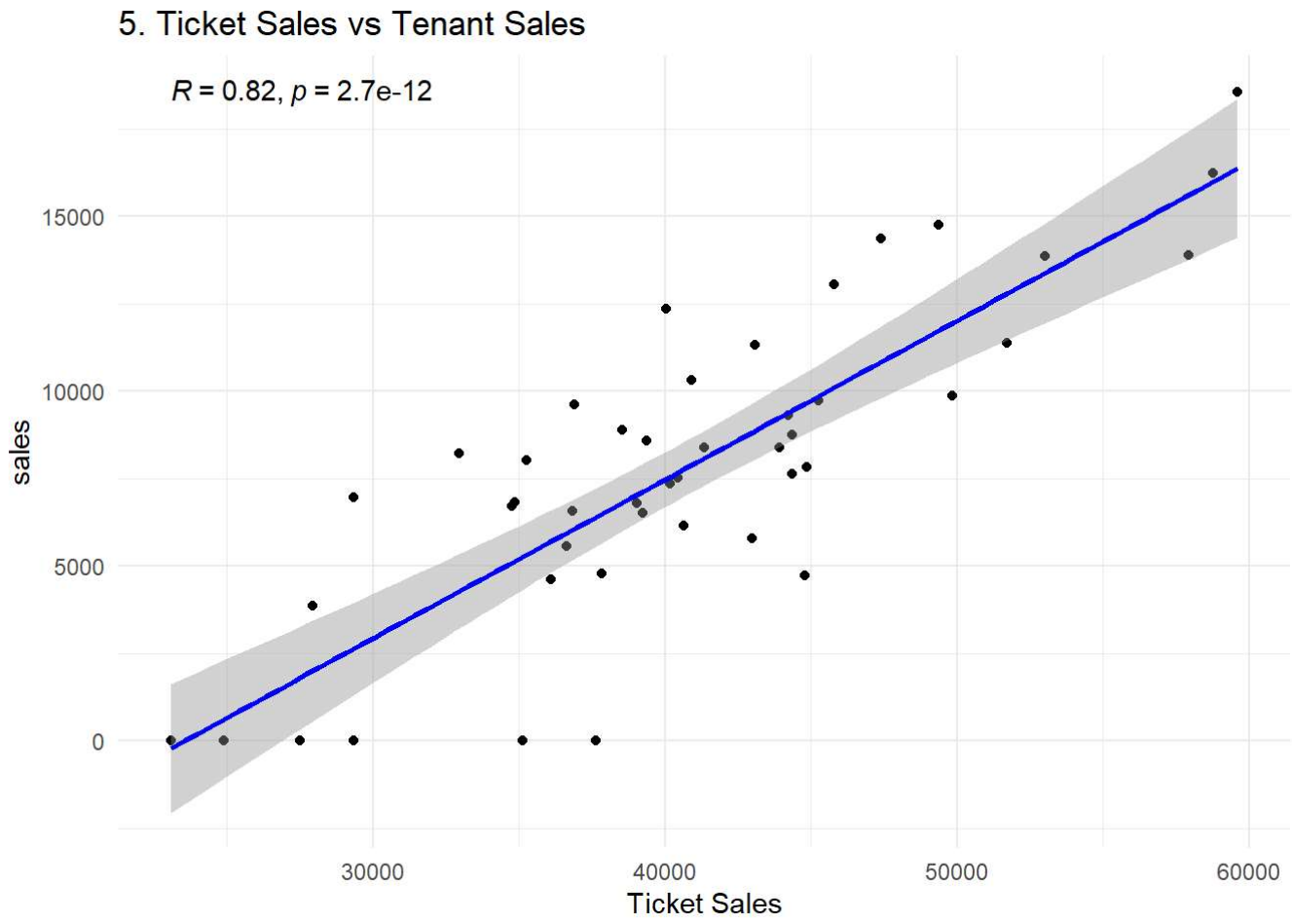
```
## `geom_smooth()` using formula = 'y ~ x'
```




```
# 5. Ticket Sales vs Tenant Sales
if(nrow(tenant_data) > 0 && nrow(ticket_data) > 0) {
  tenant_sum <- tenant_data %>%
    mutate(month_year = format(as.Date(Date), "%Y-%m")) %>%
    group_by(month_year) %>%
    summarise(total_sales = sum(`Net Sales`, na.rm = TRUE))
  cor_tenant <- plot_correlation(ticket_sum, tenant_sum, "total_tickets", "total_sales",
                                 "5. Ticket Sales vs Tenant Sales")
}
```

```
## 5. Ticket Sales vs Tenant Sales - Pearson Correlation: 0.821
```

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

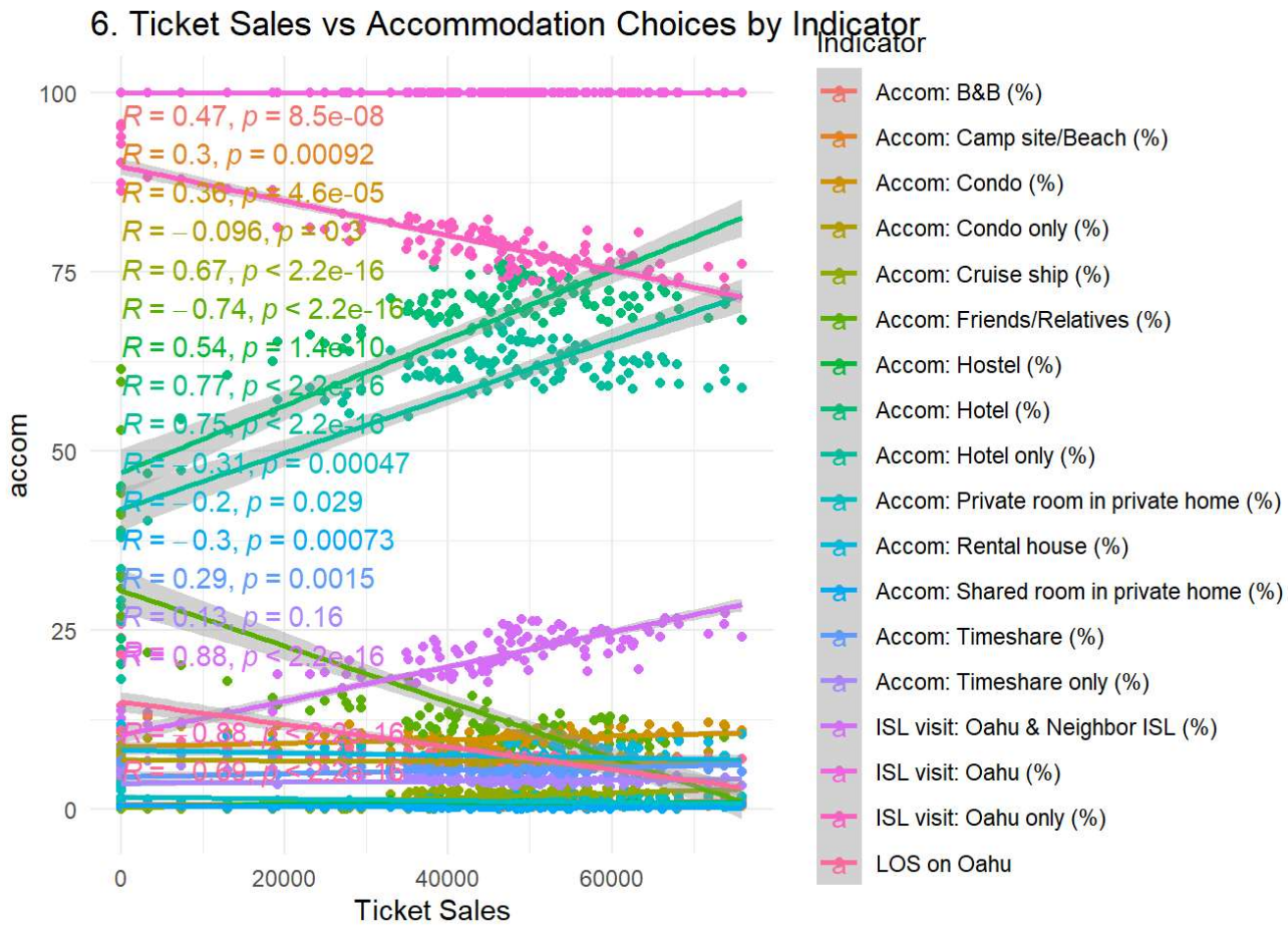



```
# 6. Ticket Sales vs Accommodation Choices by Indicator
if(nrow(accom_data) > 0 && nrow(ticket_data) > 0) {
  accom_by_ind <- accom_data %>%
    group_by(`YYYY-MM`, Indicator) %>%
    summarise(total_accom = sum(value, na.rm = TRUE)) %>%
    rename(month_year = `YYYY-MM`)
  cor_accom_ind <- plot_correlation(ticket_sum, accom_by_ind, "total_tickets", "total_accom",
                                    "6. Ticket Sales vs Accommodation Choices by Indicator", "Ind
icator")
}
```

```
## `summarise()` has grouped output by 'YYYY-MM'. You can override using the
## `.groups` argument.
```

```
## 6. Ticket Sales vs Accommodation Choices by Indicator - Pearson Correlation: 0.012
```

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

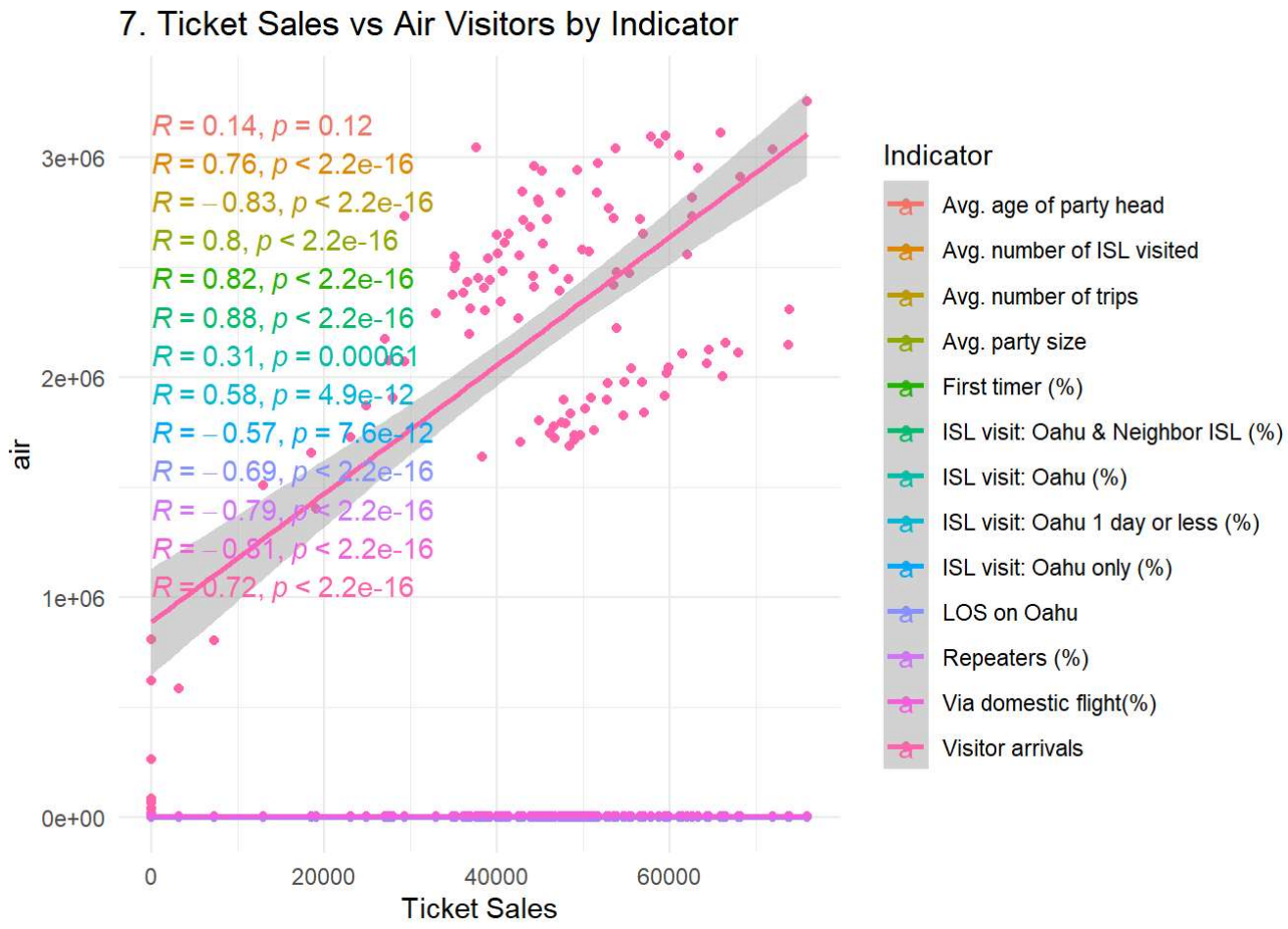



```
# 7. Ticket Sales vs Air Visitors by Indicator
if(nrow(air_data) > 0 && nrow(ticket_data) > 0) {
  air_by_ind <- air_data %>%
    group_by(`YYYY-MM`, Indicator) %>%
    summarise(total_air = sum(value, na.rm = TRUE)) %>%
    rename(month_year = `YYYY-MM`)
  cor_air_ind <- plot_correlation(ticket_sum, air_by_ind, "total_tickets", "total_air",
                                  "7. Ticket Sales vs Air Visitors by Indicator", "Indicator")
}
```

```
## `summarise()` has grouped output by 'YYYY-MM'. You can override using the
## `.groups` argument.
```

```
## 7. Ticket Sales vs Air Visitors by Indicator - Pearson Correlation: 0.066
```

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

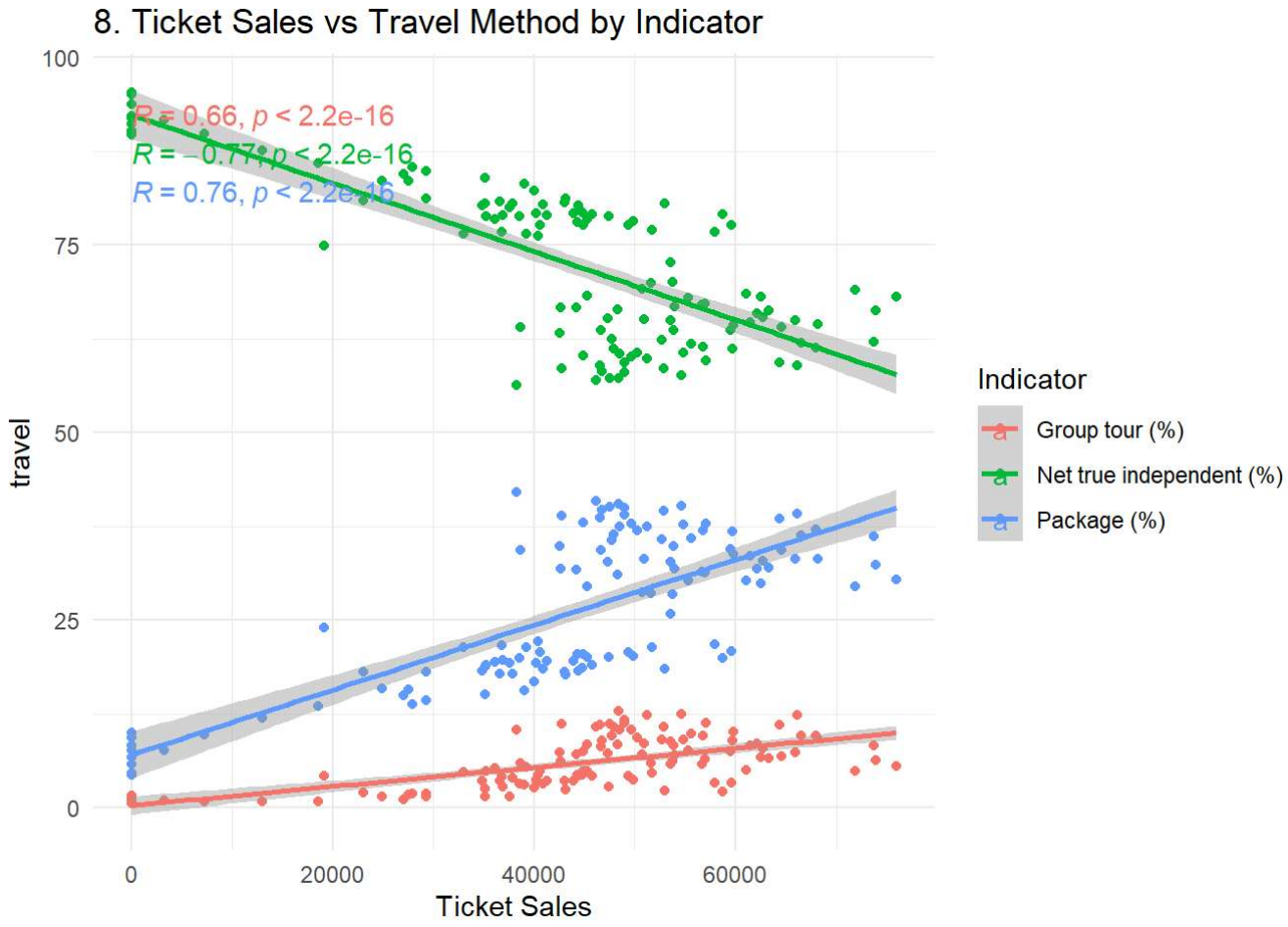



```
# 8. Ticket Sales vs Travel Method by Indicator
if(nrow(travel_data) > 0 && nrow(ticket_data) > 0) {
  travel_by_ind <- travel_data %>%
    group_by(`YYYY-MM`, Indicator) %>%
    summarise(total_travel = sum(value, na.rm = TRUE)) %>%
    rename(month_year = `YYYY-MM`)
  cor_travel_ind <- plot_correlation(ticket_sum, travel_by_ind, "total_tickets", "total_travel",
                                     "8. Ticket Sales vs Travel Method by Indicator", "Indicato
r")
}
```

```
## `summarise()` has grouped output by 'YYYY-MM'. You can override using the
## `.groups` argument.
```

```
## 8. Ticket Sales vs Travel Method by Indicator - Pearson Correlation: 0.022
```

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

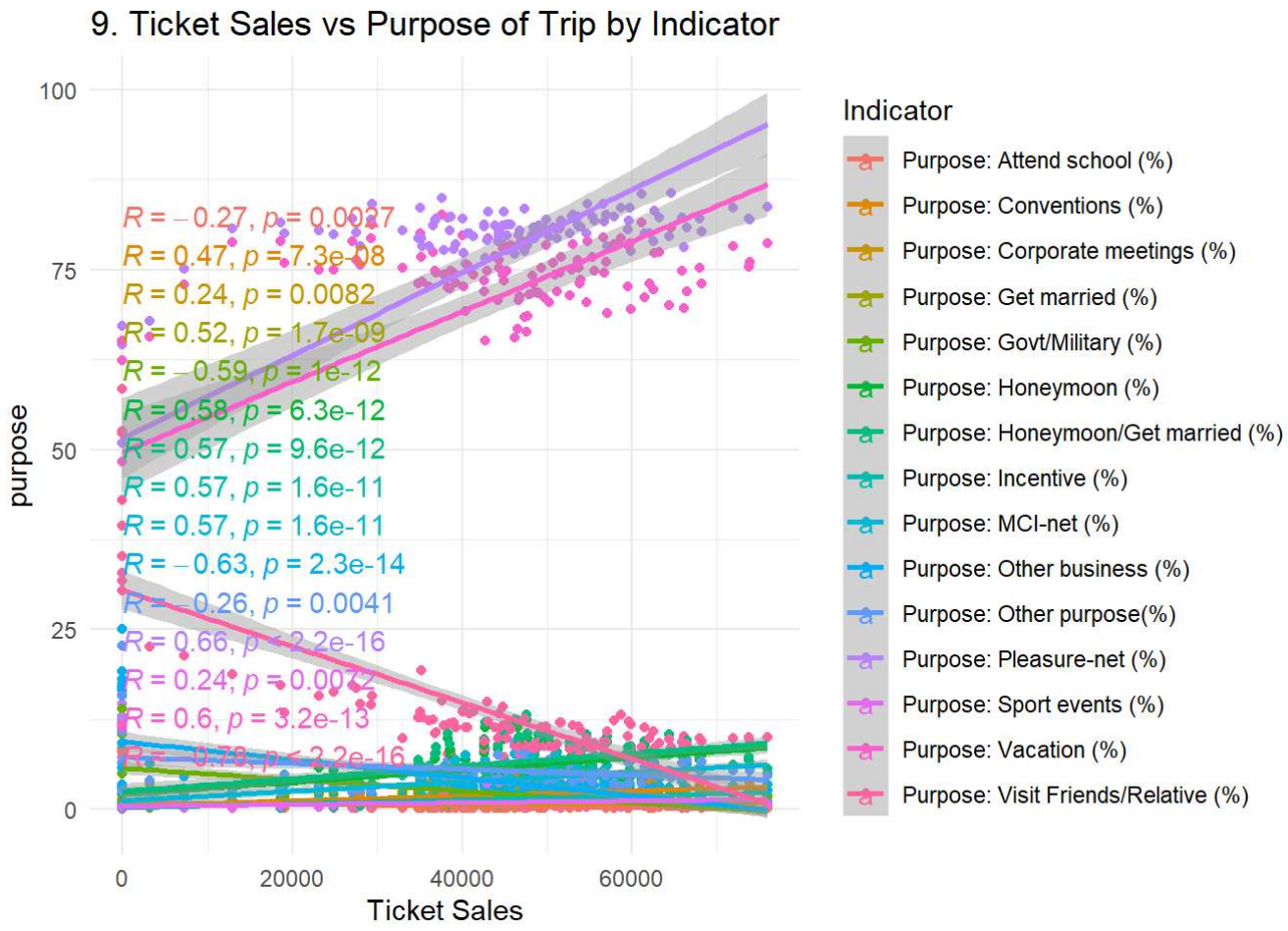



```
# 9. Ticket Sales vs Purpose of Trip by Indicator
if(nrow(purpose_data) > 0 && nrow(ticket_data) > 0) {
  purpose_by_ind <- purpose_data %>%
    group_by(`YYYY-MM`, Indicator) %>%
    summarise(total_purpose = sum(value, na.rm = TRUE)) %>%
    rename(month_year = `YYYY-MM`)
  cor_purpose_ind <- plot_correlation(ticket_sum, purpose_by_ind, "total_tickets", "total_purpos
e",
                                      "9. Ticket Sales vs Purpose of Trip by Indicator", "Indicat
or")
}
```

```
## `summarise()` has grouped output by 'YYYY-MM'. You can override using the
## `.groups` argument.
```

```
## 9. Ticket Sales vs Purpose of Trip by Indicator - Pearson Correlation: 0.036
```

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




```
# Summary of correlation values
cat("\nSummary of Pearson Correlation Coefficients:\n")
```

```
##
## Summary of Pearson Correlation Coefficients:
```

```
cat(sprintf("1. Ticket Sales vs Accommodation Choices: %.3f\n", cor_accom))
```

```
## 1. Ticket Sales vs Accommodation Choices: 0.742
```

```
cat(sprintf("2. Ticket Sales vs Air Visitors: %.3f\n", cor_air))
```

```
## 2. Ticket Sales vs Air Visitors: 0.717
```

```
cat(sprintf("3. Ticket Sales vs Travel Method: %.3f\n", cor_travel))
```

```
## 3. Ticket Sales vs Travel Method: 0.638
```

```
cat(sprintf("4. Ticket Sales vs Purpose of Trip: %.3f\n", cor_purpose))
```

```
## 4. Ticket Sales vs Purpose of Trip: 0.727
```

```
cat(sprintf("5. Ticket Sales vs Tenant Sales: %.3f\n", cor_tenant))
```

```
## 5. Ticket Sales vs Tenant Sales: 0.821
```

```
cat(sprintf("6. Ticket Sales vs Accommodation Choices by Indicator: %.3f (avg across Indicators)
\n", cor_accom_ind))
```

```
## 6. Ticket Sales vs Accommodation Choices by Indicator: 0.012 (avg across Indicators)
```

```
cat(sprintf("7. Ticket Sales vs Air Visitors by Indicator: %.3f (avg across Indicators)\n", cor_
air_ind))
```

```
## 7. Ticket Sales vs Air Visitors by Indicator: 0.066 (avg across Indicators)
```

```
cat(sprintf("8. Ticket Sales vs Travel Method by Indicator: %.3f (avg across Indicators)\n", cor
_travel_ind))
```

```
## 8. Ticket Sales vs Travel Method by Indicator: 0.022 (avg across Indicators)
```

```
cat(sprintf("9. Ticket Sales vs Purpose of Trip by Indicator: %.3f (avg across Indicators)\n", c
or_purpose_ind))
```

```
## 9. Ticket Sales vs Purpose of Trip by Indicator: 0.036 (avg across Indicators)
```

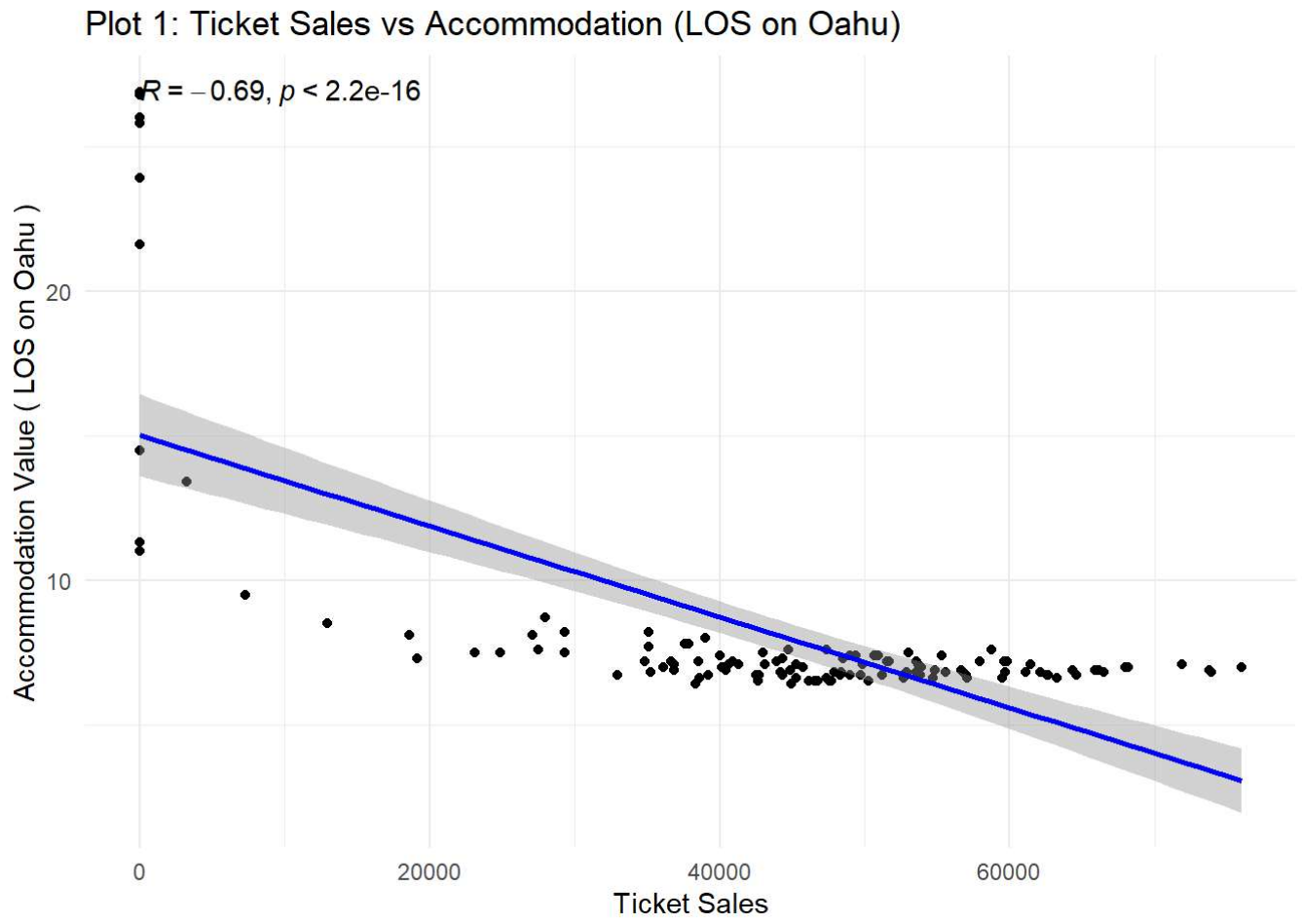
```
# Prepare ticket_data with a month-year column
if(nrow(ticket_data) > 0) {
  ticket_data$date <- as.Date(paste(ticket_data$Year, ticket_data$Month, "01", sep = "-"), forma
t = "%Y-%B-%d")
  ticket_data$month_year <- format(ticket_data$date, "%Y-%m")
  ticket_sum <- ticket_data %>%
    group_by(month_year) %>%
    summarise(total_tickets = sum(TicketSell, na.rm = TRUE))
}

# Prepare accommodation data and get unique Indicators
if(nrow(accom_data) > 0) {
  accom_data$month_year <- accom_data$`YYYY-MM`
  unique_indicators <- unique(accom_data$Indicator)
  num_indicators <- length(unique_indicators)
  cat(sprintf("Number of unique Indicators found: %d\n", num_indicators))

  # Limit to 18 or fewer indicators
  if(num_indicators > 18) {
    unique_indicators <- unique_indicators[1:18]
    cat("Limiting to first 18 Indicators.\n")
  }
} else {
  cat("No data in htaaccommodationchoices table.\n")
  unique_indicators <- character(0)
}
```

```
## Number of unique Indicators found: 18
```

```
# Function to plot correlation for a specific Indicator
plot_correlation_by_indicator <- function(ticket_data, accom_data, indicator, plot_num) {
  # Filter accommodation data for the specific Indicator
  accom_subset <- accom_data %>%
    filter(Indicator == indicator) %>%
    group_by(month_year) %>%
    summarise(total_accom = sum(value, na.rm = TRUE))

  # Merge with ticket data
  merged_data <- merge(ticket_data, accom_subset, by = "month_year", all = FALSE)

  if(nrow(merged_data) > 1) {  # Need at least 2 points for correlation
    # Calculate correlation
    cor_value <- cor(merged_data$total_tickets, merged_data$total_accom, use = "complete.obs", m
ethod = "pearson")
    cat(sprintf("Plot %d - Ticket Sales vs %s - Pearson Correlation: %.3f\n", plot_num, indicato
r, cor_value))

    # Create scatter plot
    p <- ggplot(merged_data, aes(x = total_tickets, y = total_accom)) +
      geom_point() +
      geom_smooth(method = "lm", color = "blue") +
      labs(title = sprintf("Plot %d: Ticket Sales vs Accommodation (%s)", plot_num, indicator),
           x = "Ticket Sales", y = paste("Accommodation Value (", indicator, ")")) +
      stat_cor(method = "pearson", label.x = min(merged_data$total_tickets, na.rm = TRUE),
               label.y = max(merged_data$total_accom, na.rm = TRUE)) +
      theme_minimal()
    print(p)
    Sys.sleep(1)
    return(cor_value)
  } else {
    cat(sprintf("Plot %d - Ticket Sales vs %s - Insufficient data for correlation\n", plot_num,
indicator))
    return(NA)
  }
}

# Generate up to 18 correlation plots
correlation_values <- vector("numeric", min(18, length(unique_indicators)))
if(nrow(ticket_data) > 0 && nrow(accom_data) > 0) {
  for(i in seq_along(unique_indicators)) {
    indicator <- unique_indicators[i]
    correlation_values[i] <- plot_correlation_by_indicator(ticket_sum, accom_data, indicator, i)
  }
}
```

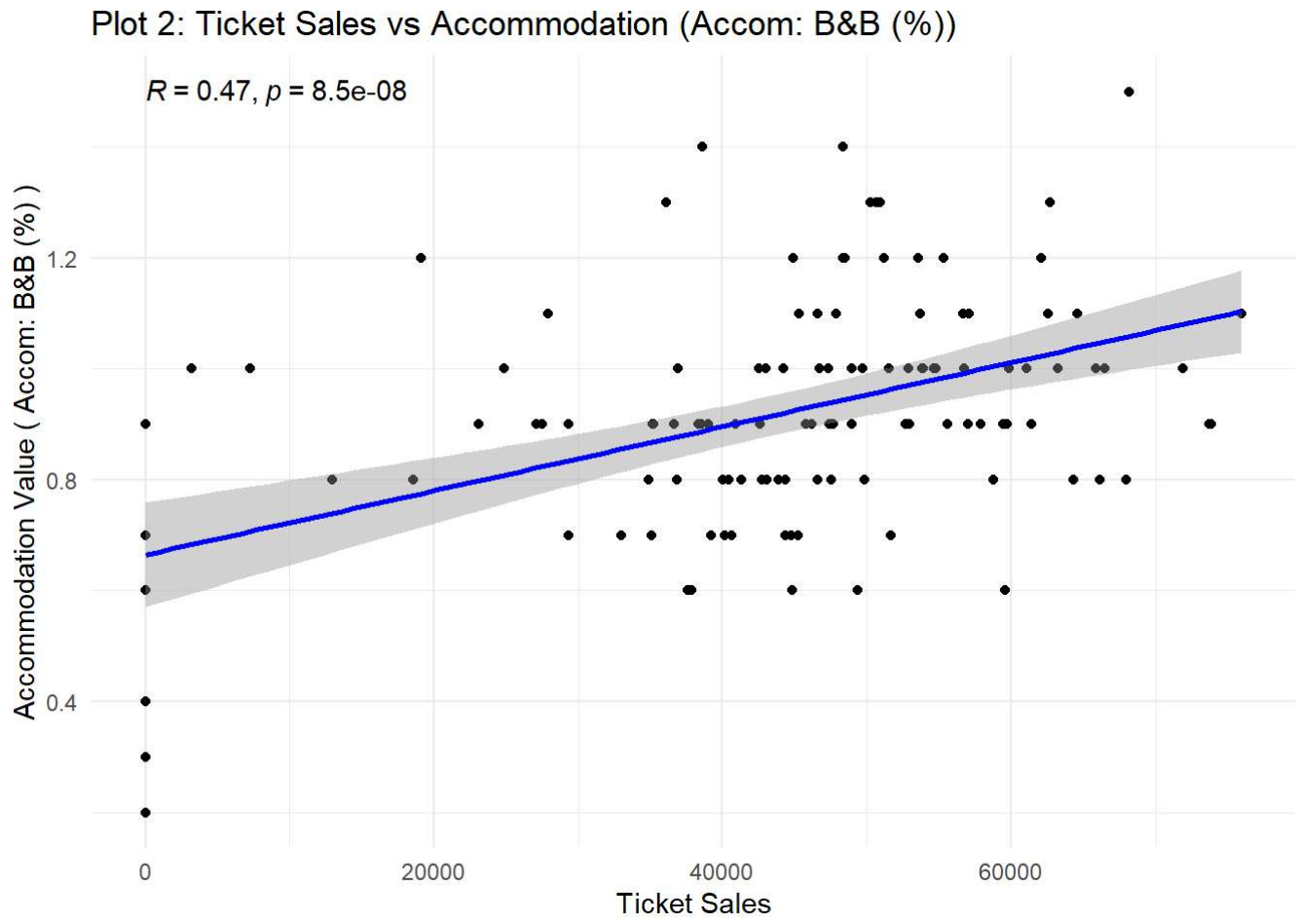
```
## Plot 1 - Ticket Sales vs LOS on Oahu - Pearson Correlation: -0.690
```

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



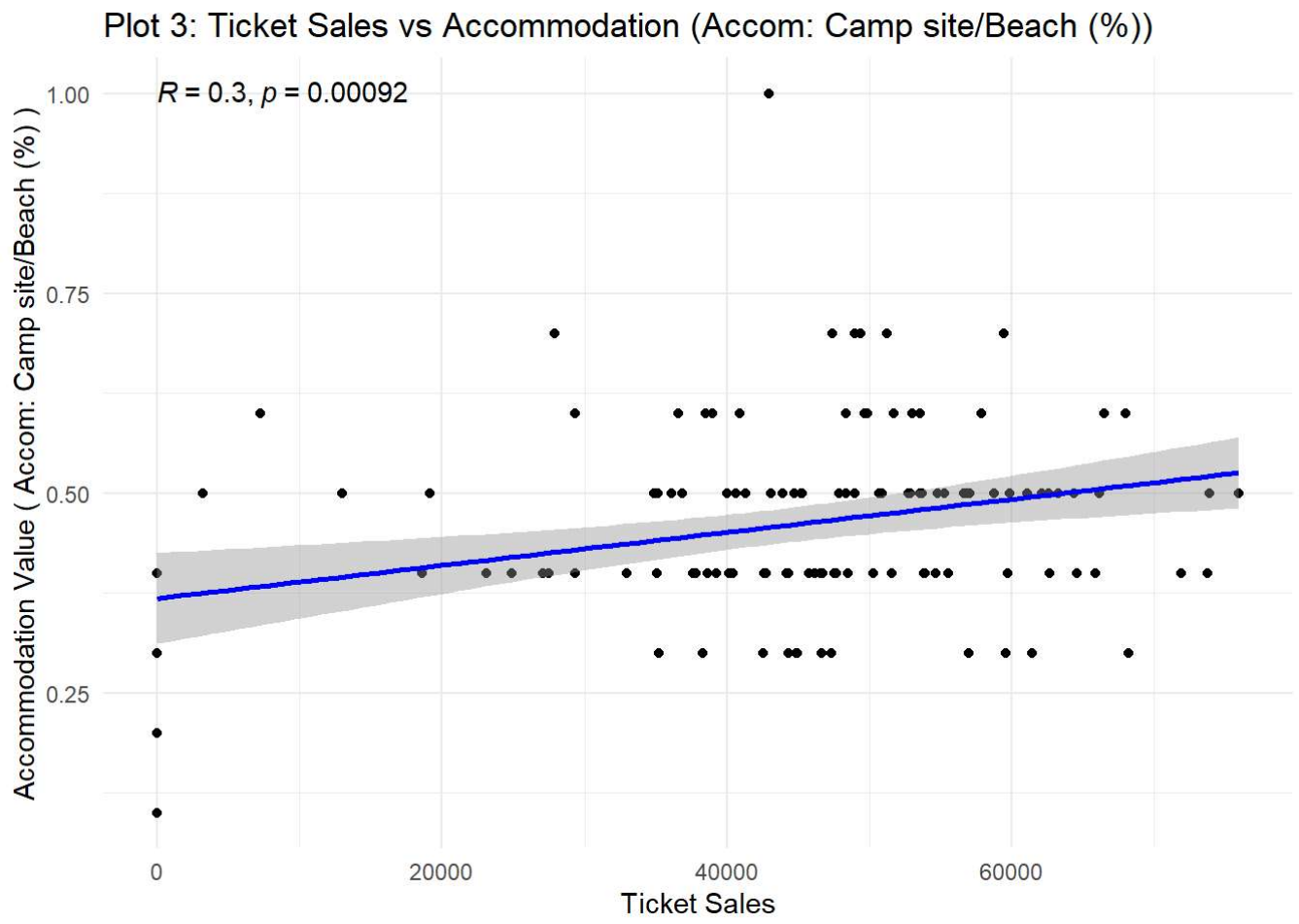

```
## Plot 2 - Ticket Sales vs Accom: B&B (%) - Pearson Correlation: 0.465
```

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



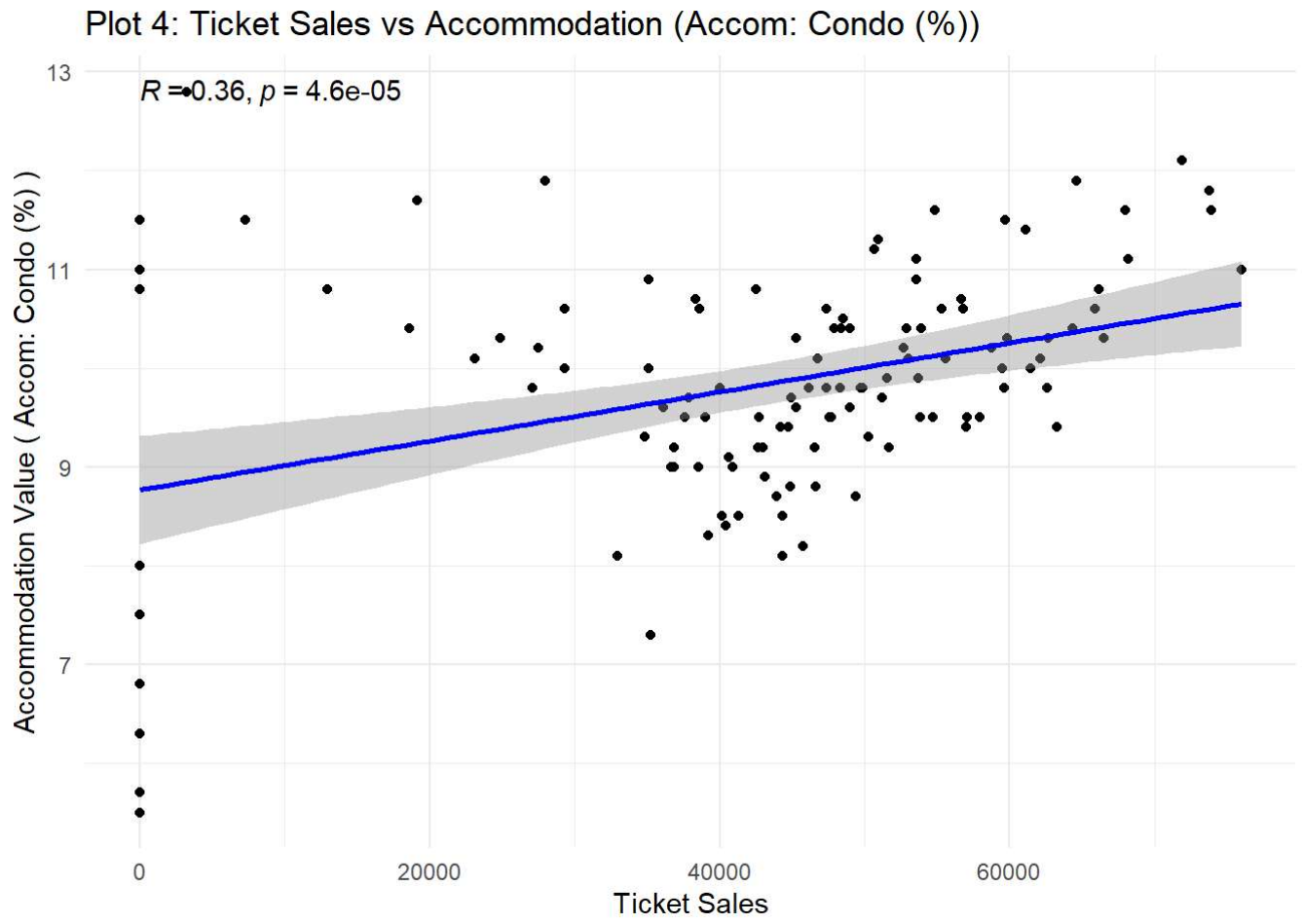

```
## Plot 3 - Ticket Sales vs Accom: Camp site/Beach (%) - Pearson Correlation: 0.299
```

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



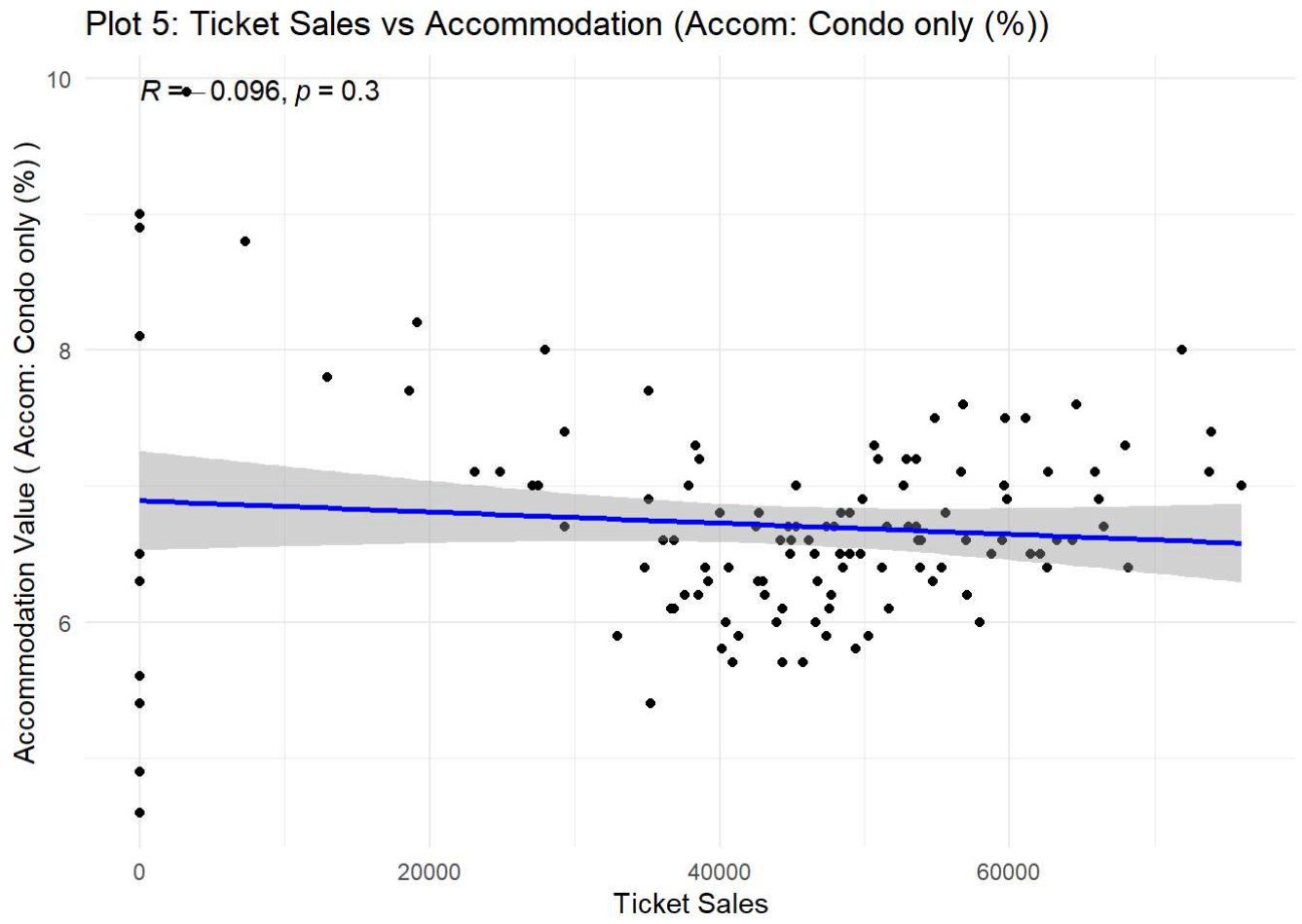

```
## Plot 4 - Ticket Sales vs Accom: Condo (%) - Pearson Correlation: 0.363
```

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



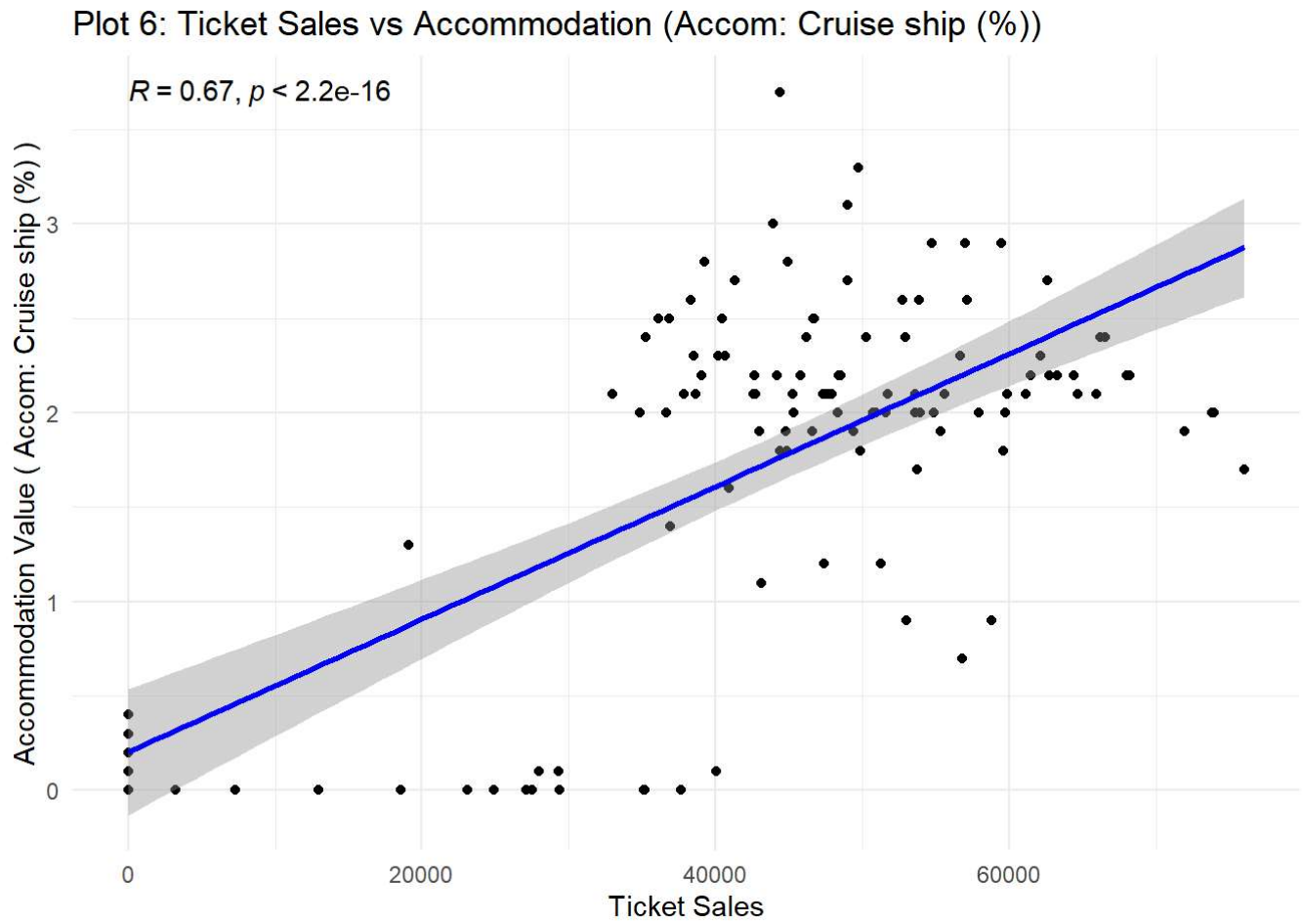

```
## Plot 5 - Ticket Sales vs Accom: Condo only (%) - Pearson Correlation: -0.096
```

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



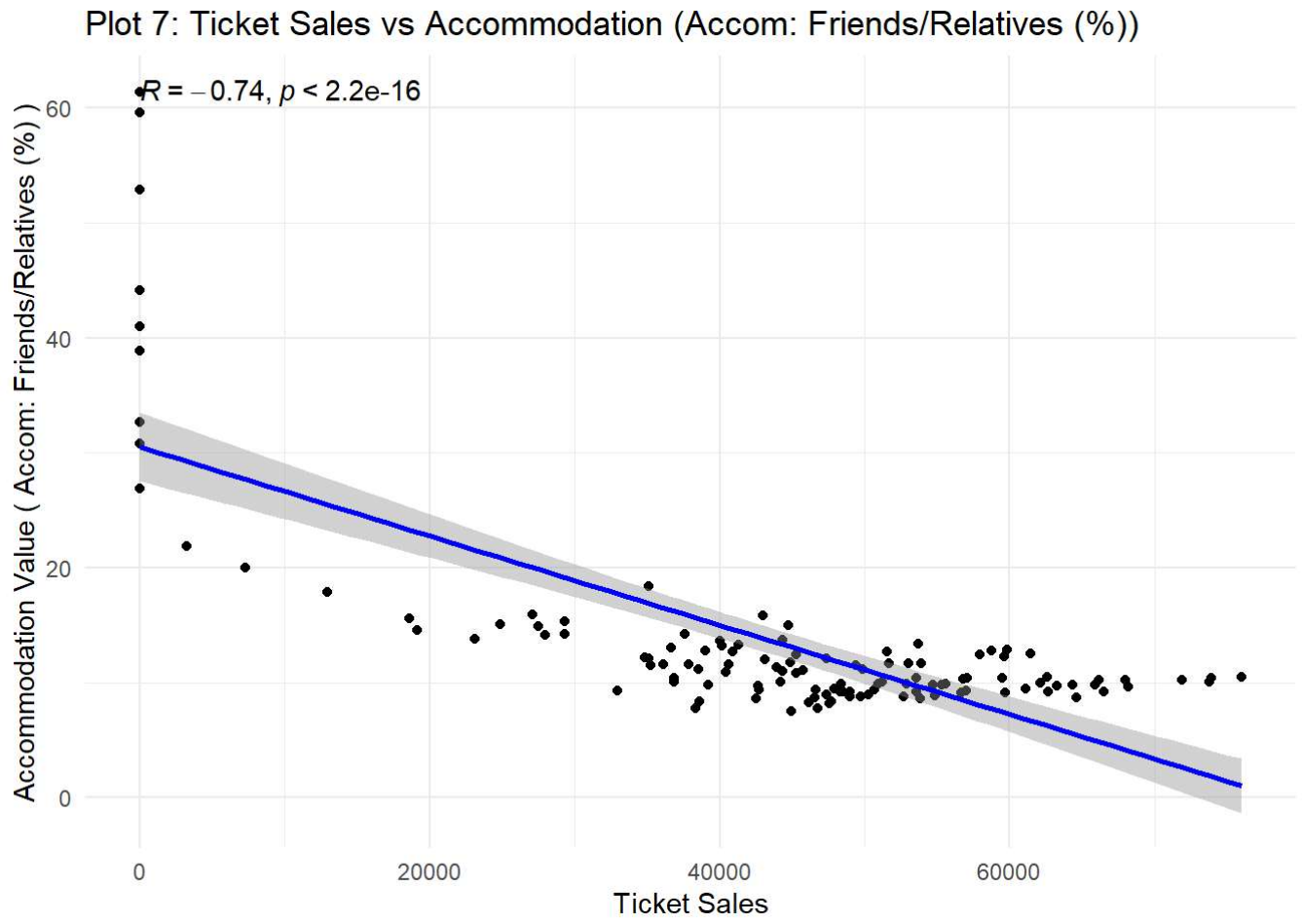

```
## Plot 6 - Ticket Sales vs Accom: Cruise ship (%) - Pearson Correlation: 0.673
```

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



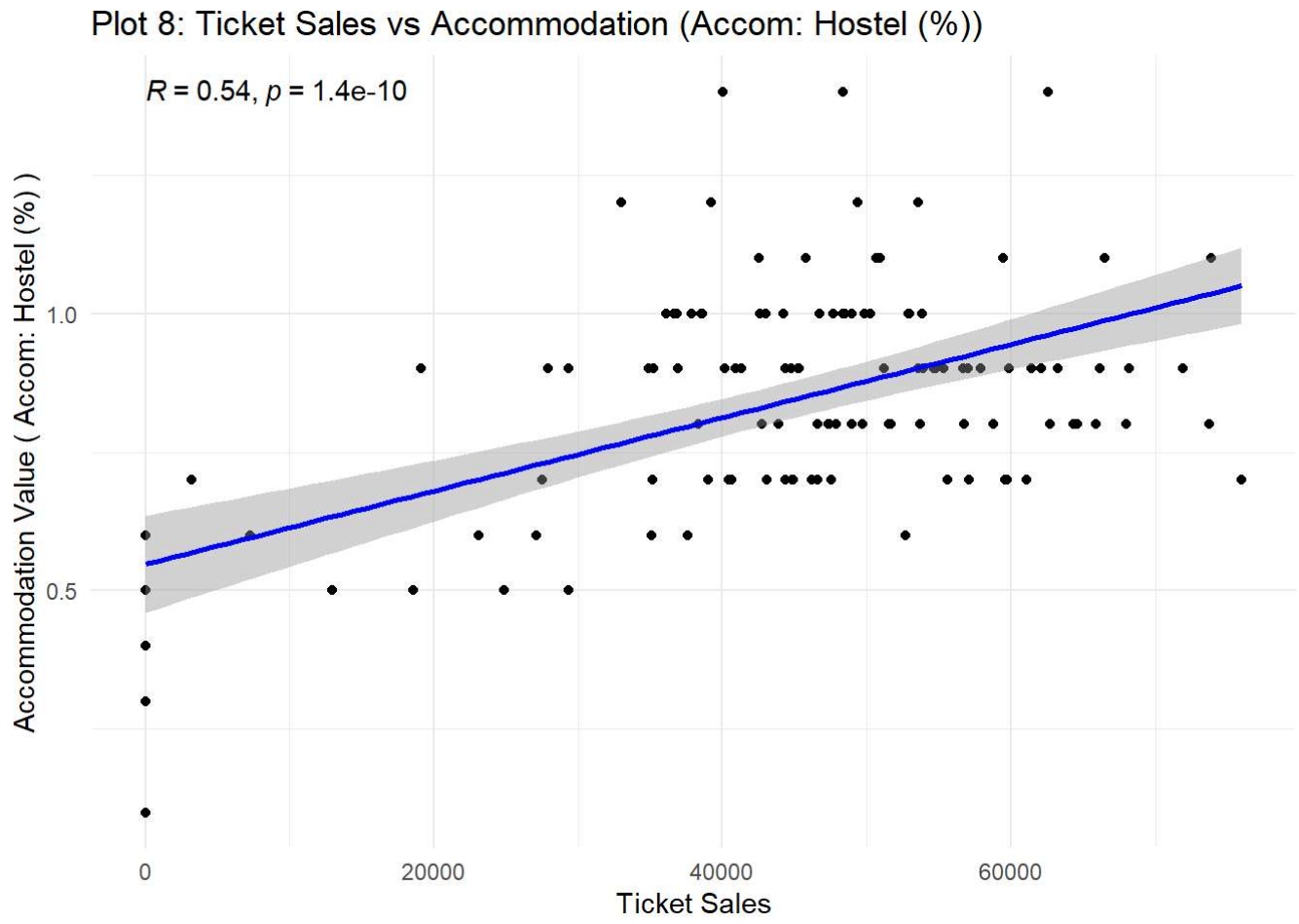

```
## Plot 7 - Ticket Sales vs Accom: Friends/Relatives (%) - Pearson Correlation: -0.744
```

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



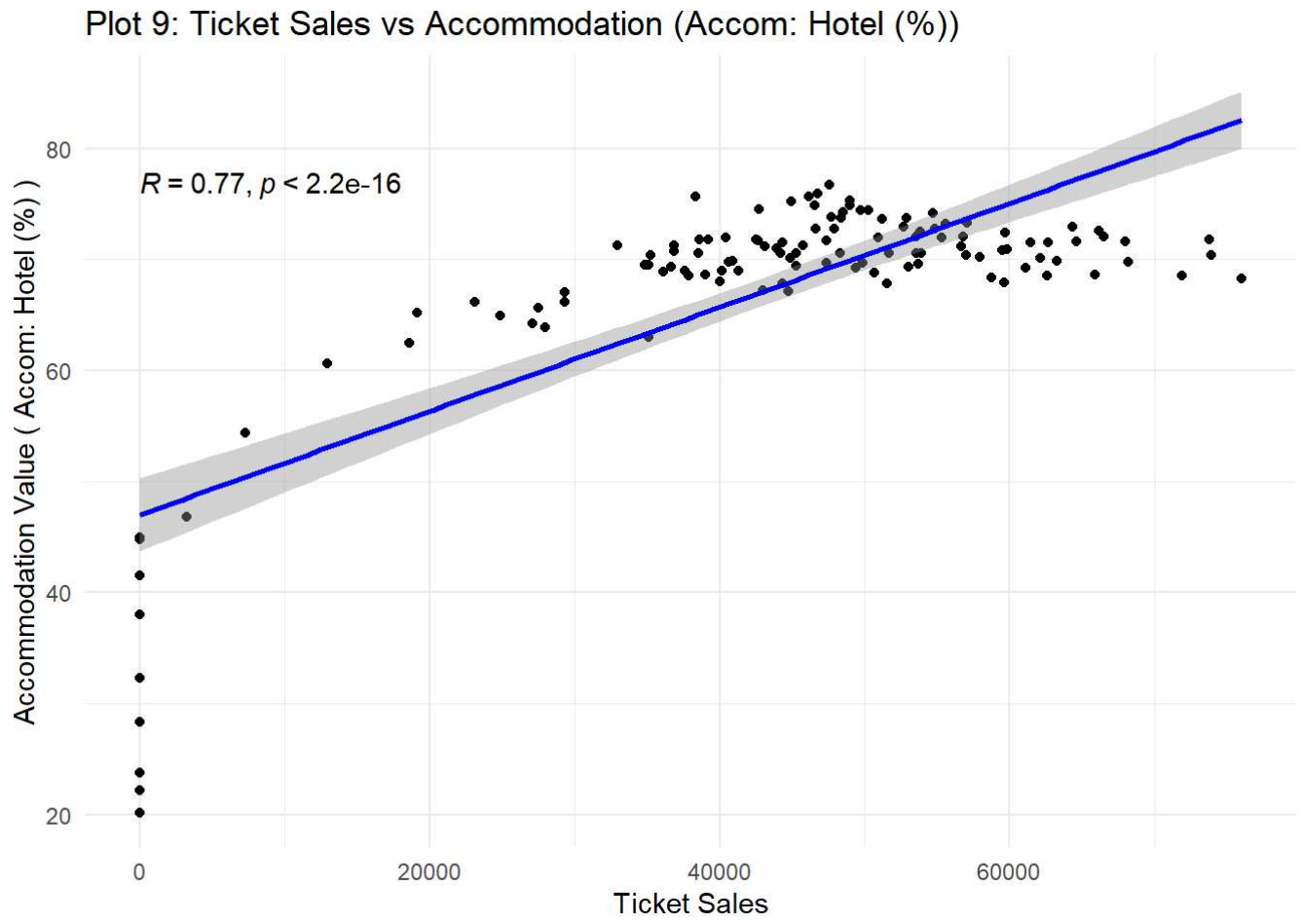

```
## Plot 8 - Ticket Sales vs Accom: Hostel (%) - Pearson Correlation: 0.544
```

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



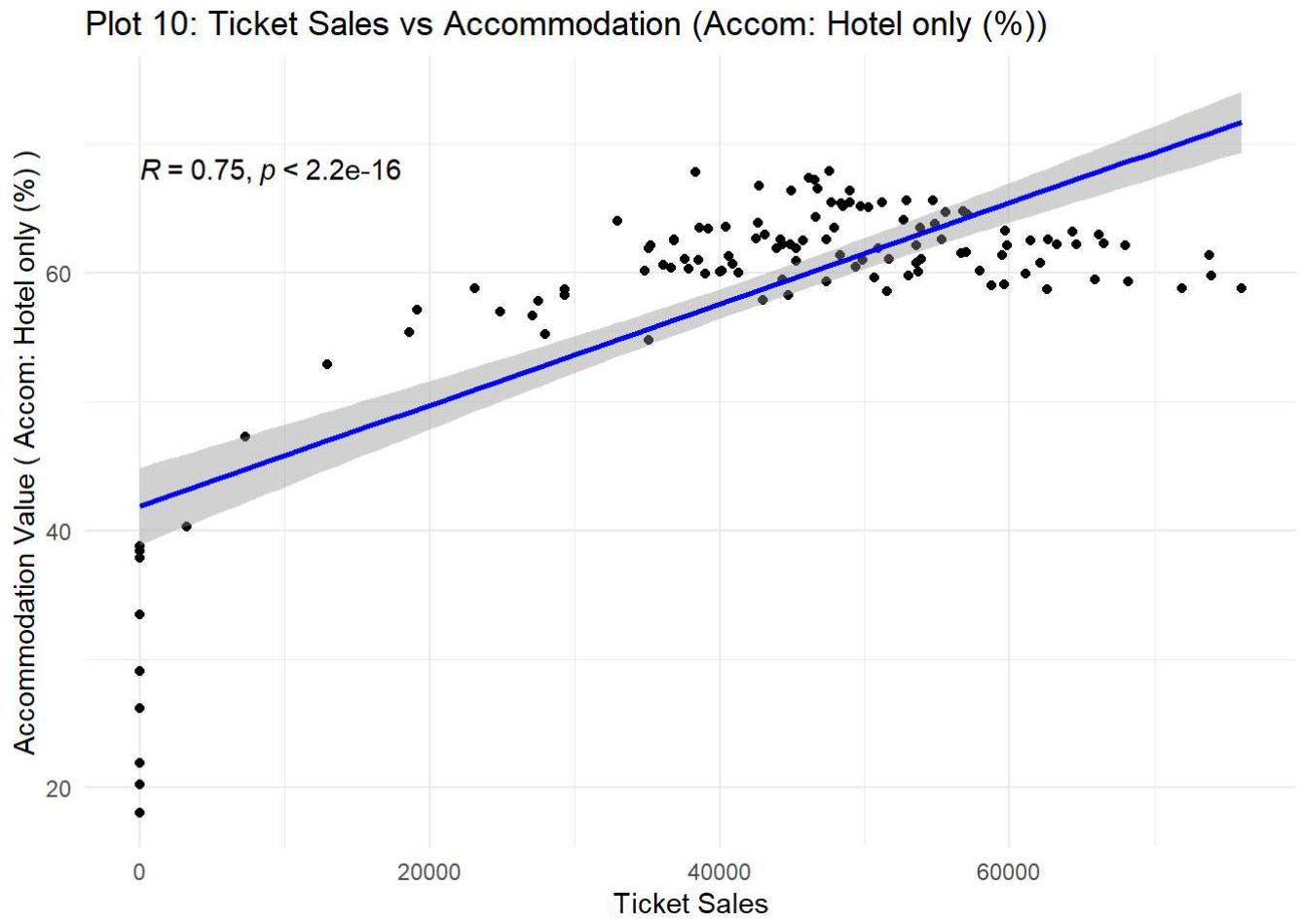

```
## Plot 9 - Ticket Sales vs Accom: Hotel (%) - Pearson Correlation: 0.774
```

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



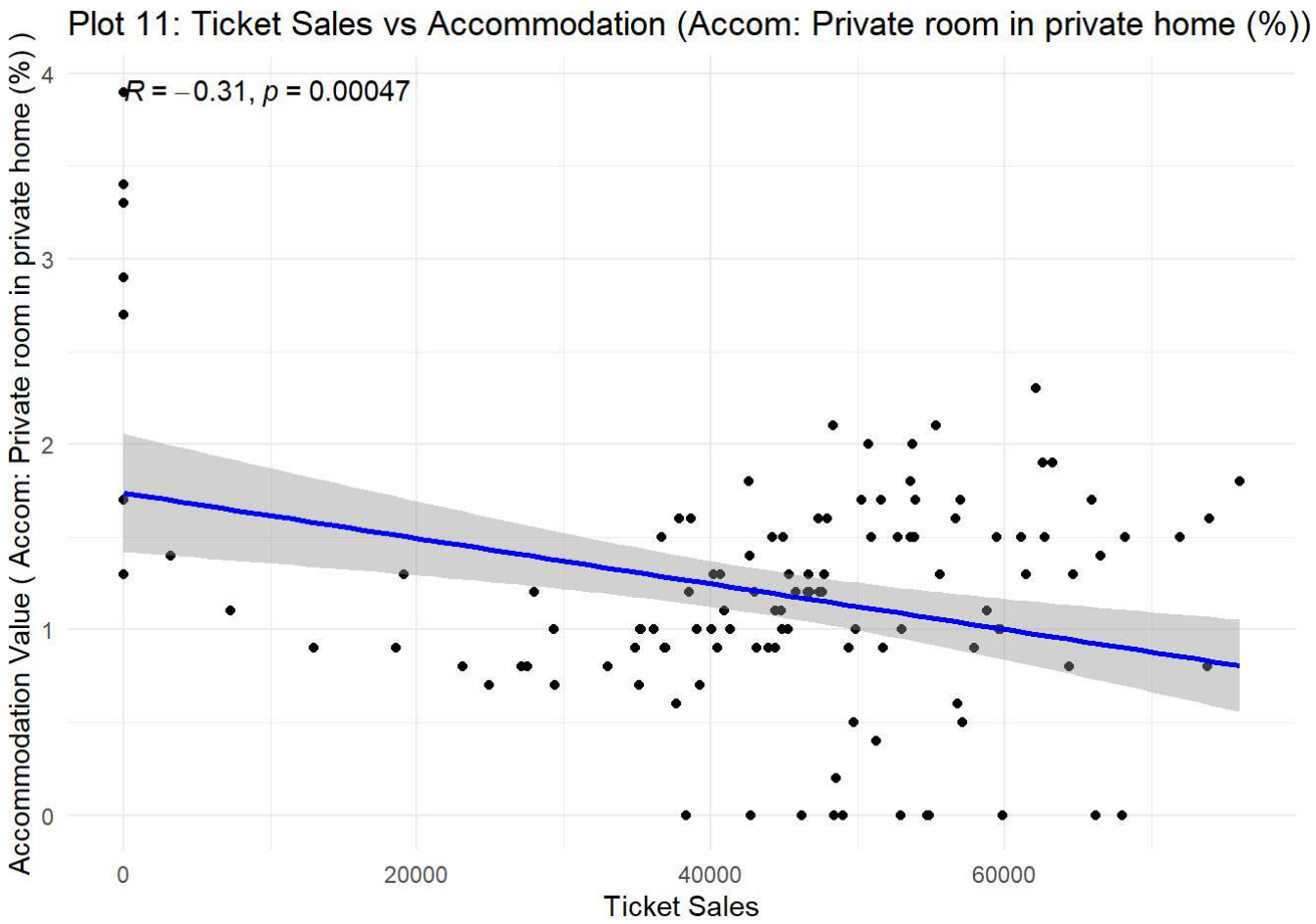

```
## Plot 10 - Ticket Sales vs Accom: Hotel only (%) - Pearson Correlation: 0.748
```

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



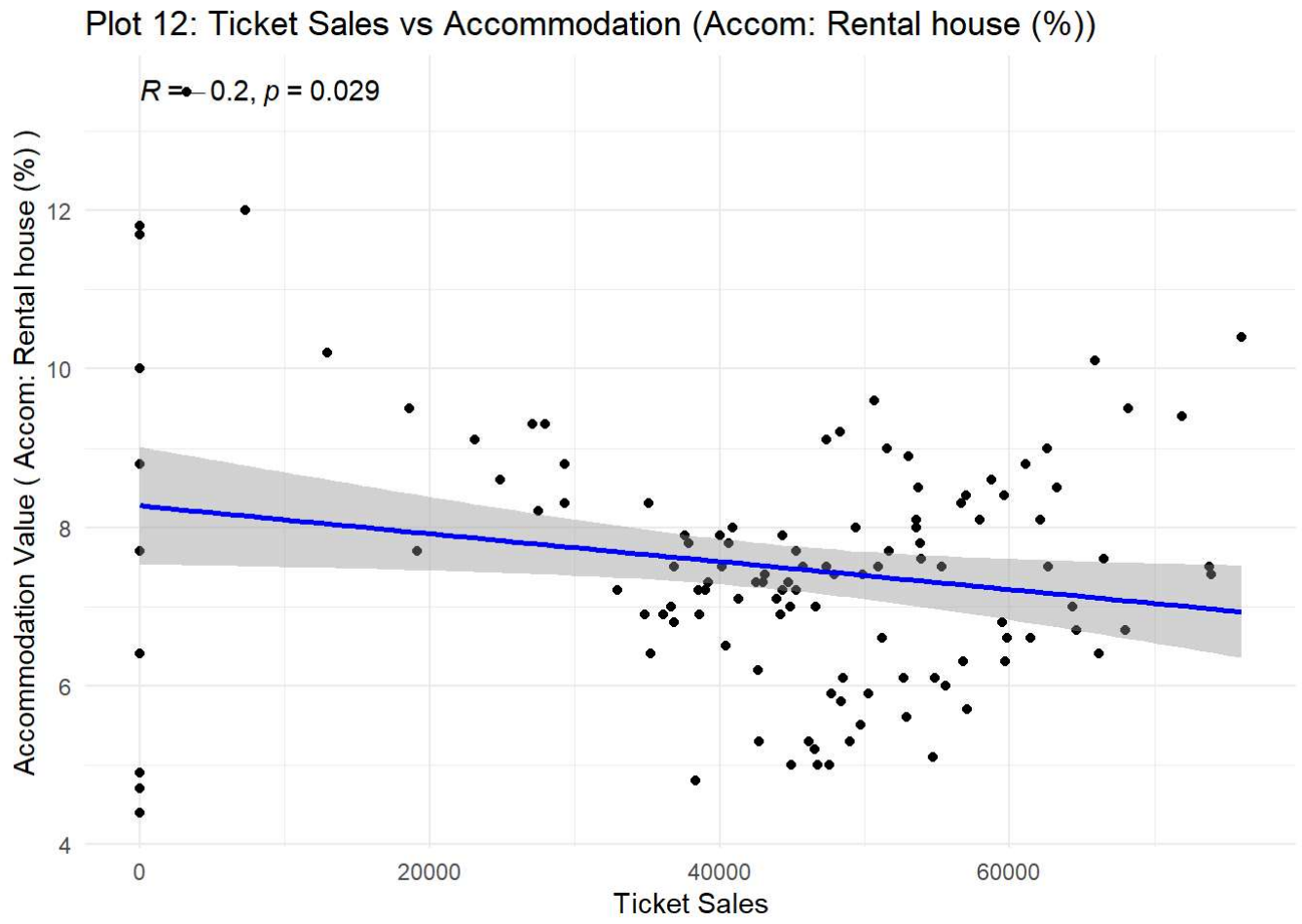

```
## Plot 11 - Ticket Sales vs Accom: Private room in private home (%) - Pearson Correlation: -0.3
14
```

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



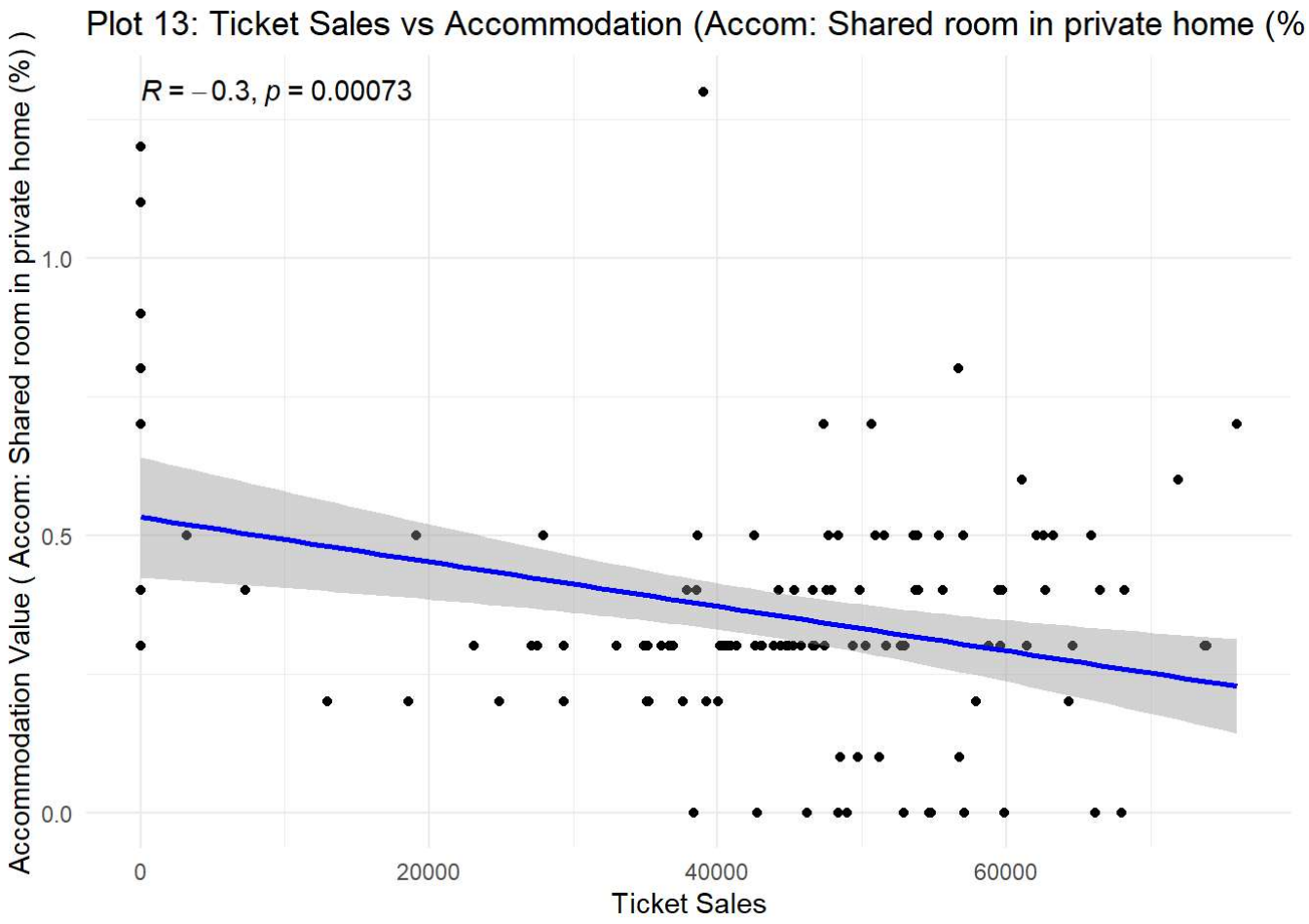

```
## Plot 12 - Ticket Sales vs Accom: Rental house (%) - Pearson Correlation: -0.200
```

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



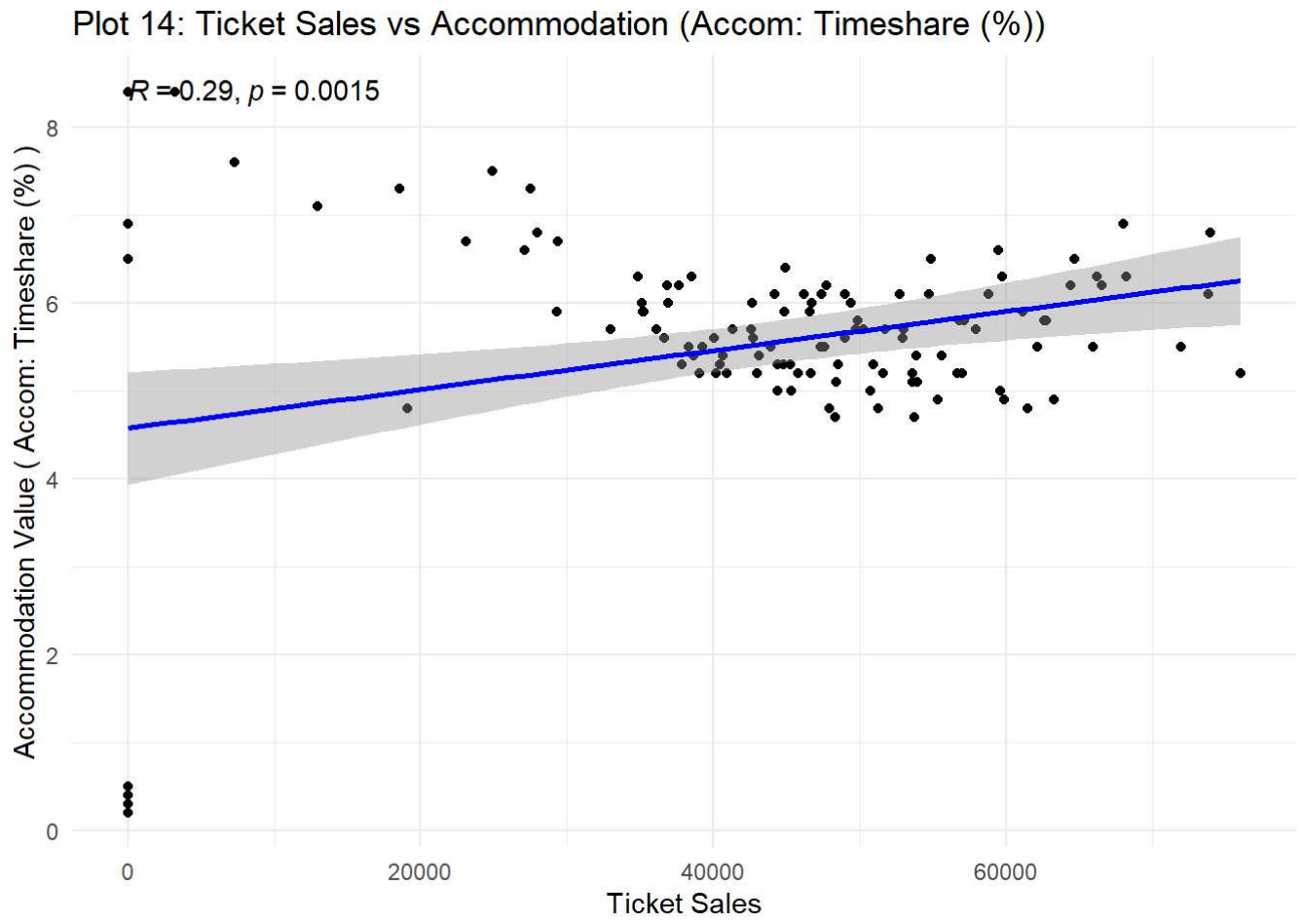

```
## Plot 13 - Ticket Sales vs Accom: Shared room in private home (%) - Pearson Correlation: -0.30
4
```

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



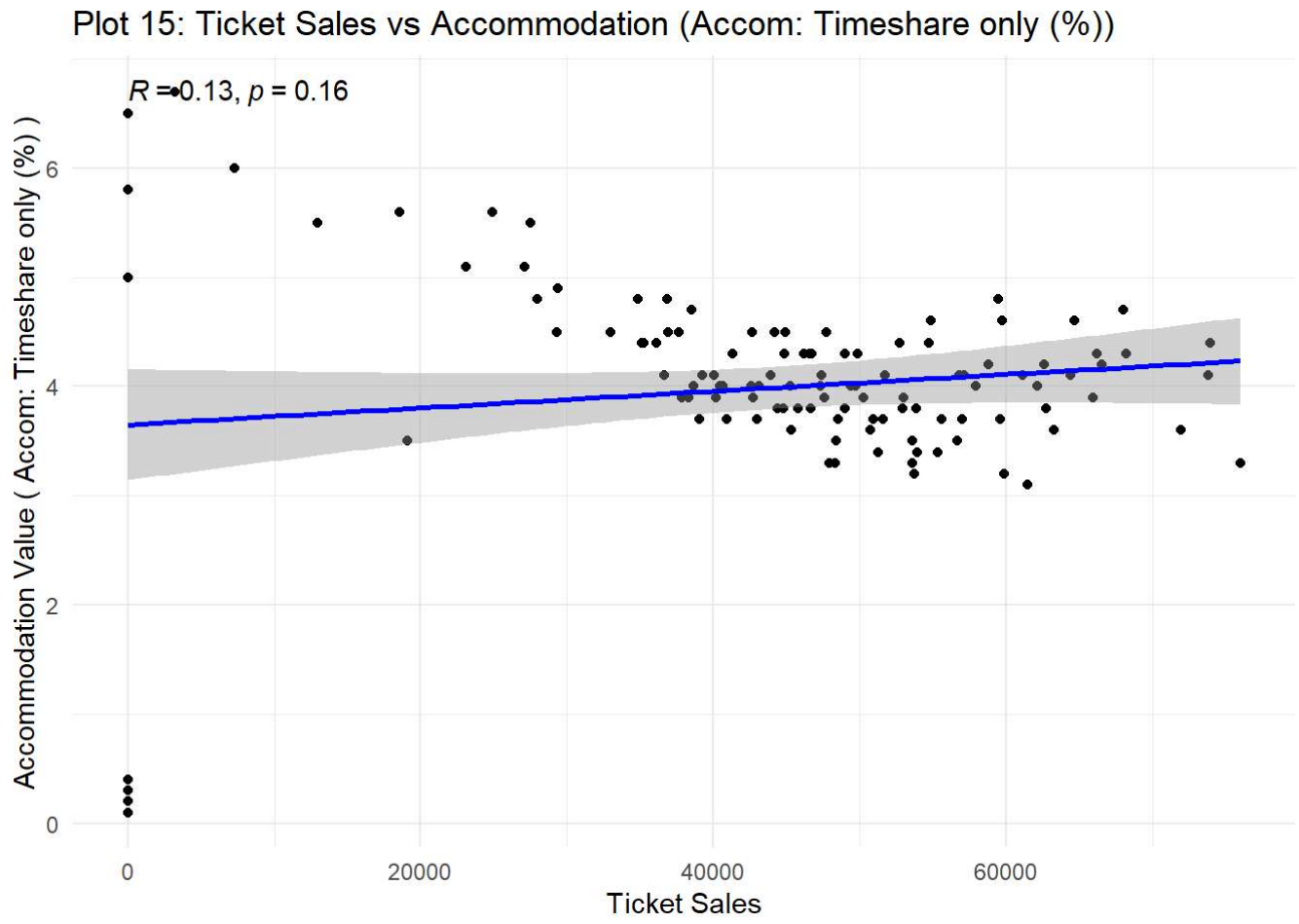

```
## Plot 14 - Ticket Sales vs Accom: Timeshare (%) - Pearson Correlation: 0.286
```

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




```
## Plot 15 - Ticket Sales vs Accom: Timeshare only (%) - Pearson Correlation: 0.129
```

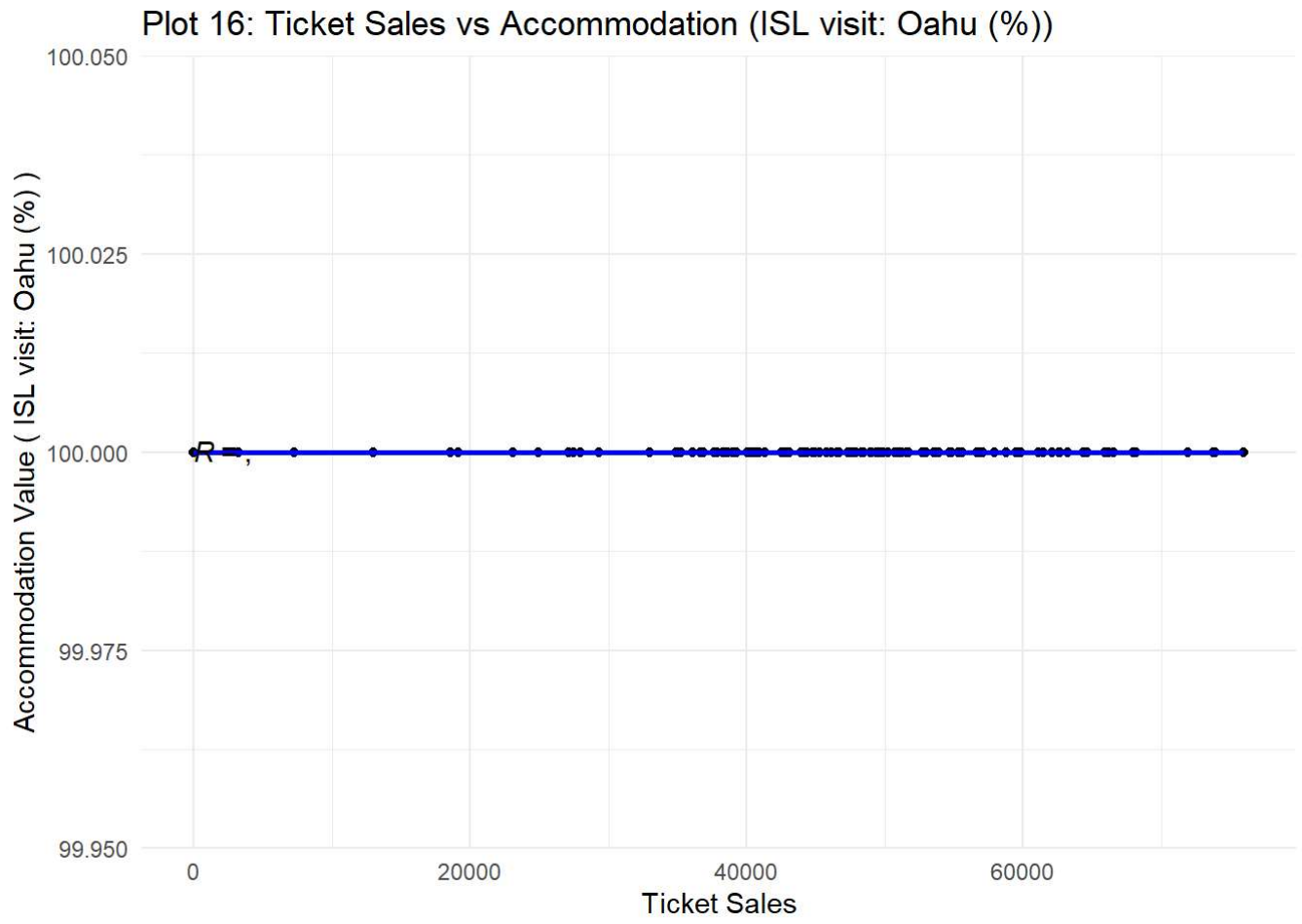
```
## `geom_smooth()` using formula = 'y ~ x'
```




```
## Warning in cor(merged_data$total_tickets, merged_data$total_accom, use =
## "complete.obs", : the standard deviation is zero
```

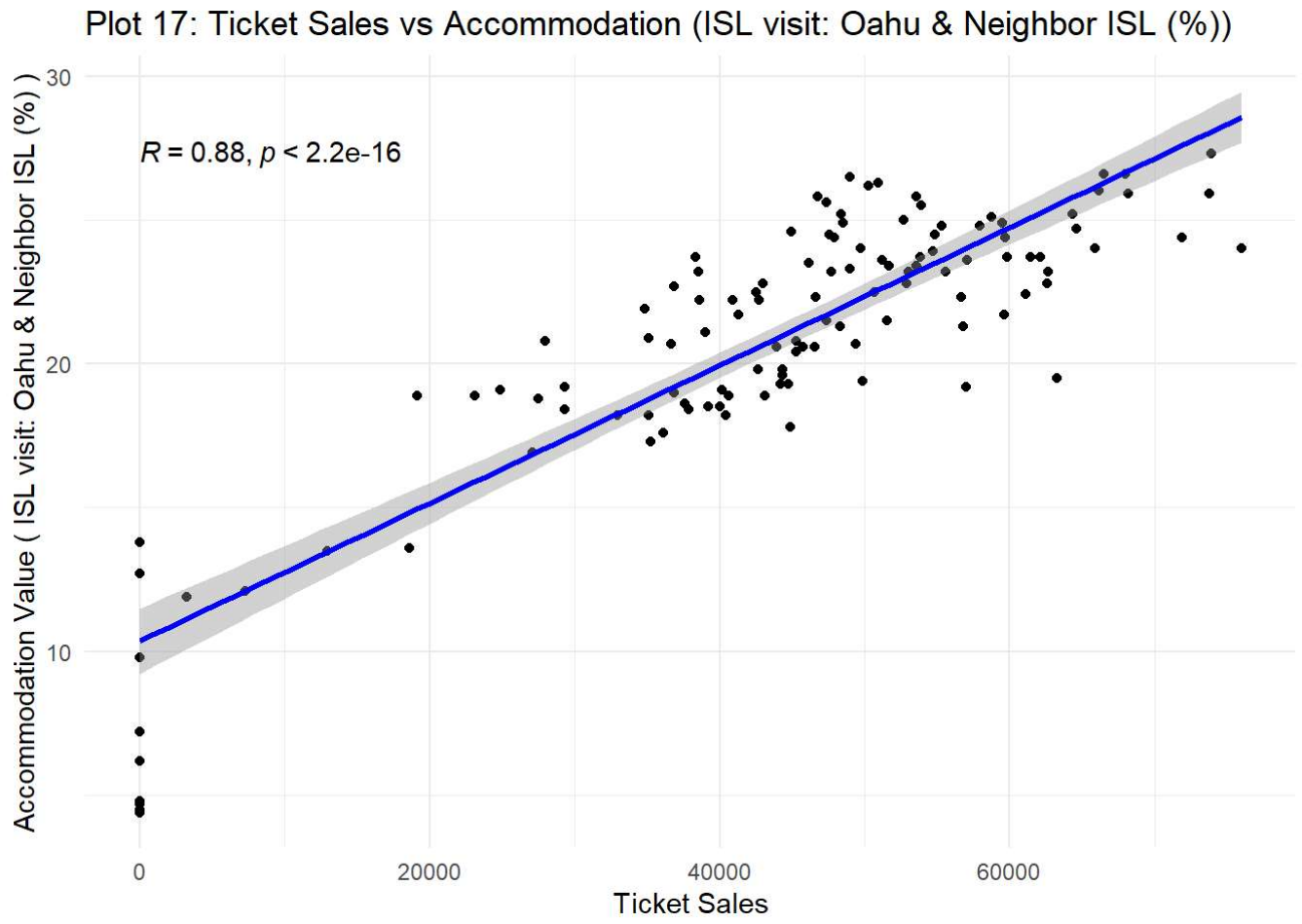
```
## Plot 16 - Ticket Sales vs ISL visit: Oahu (%) - Pearson Correlation: NA
```

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



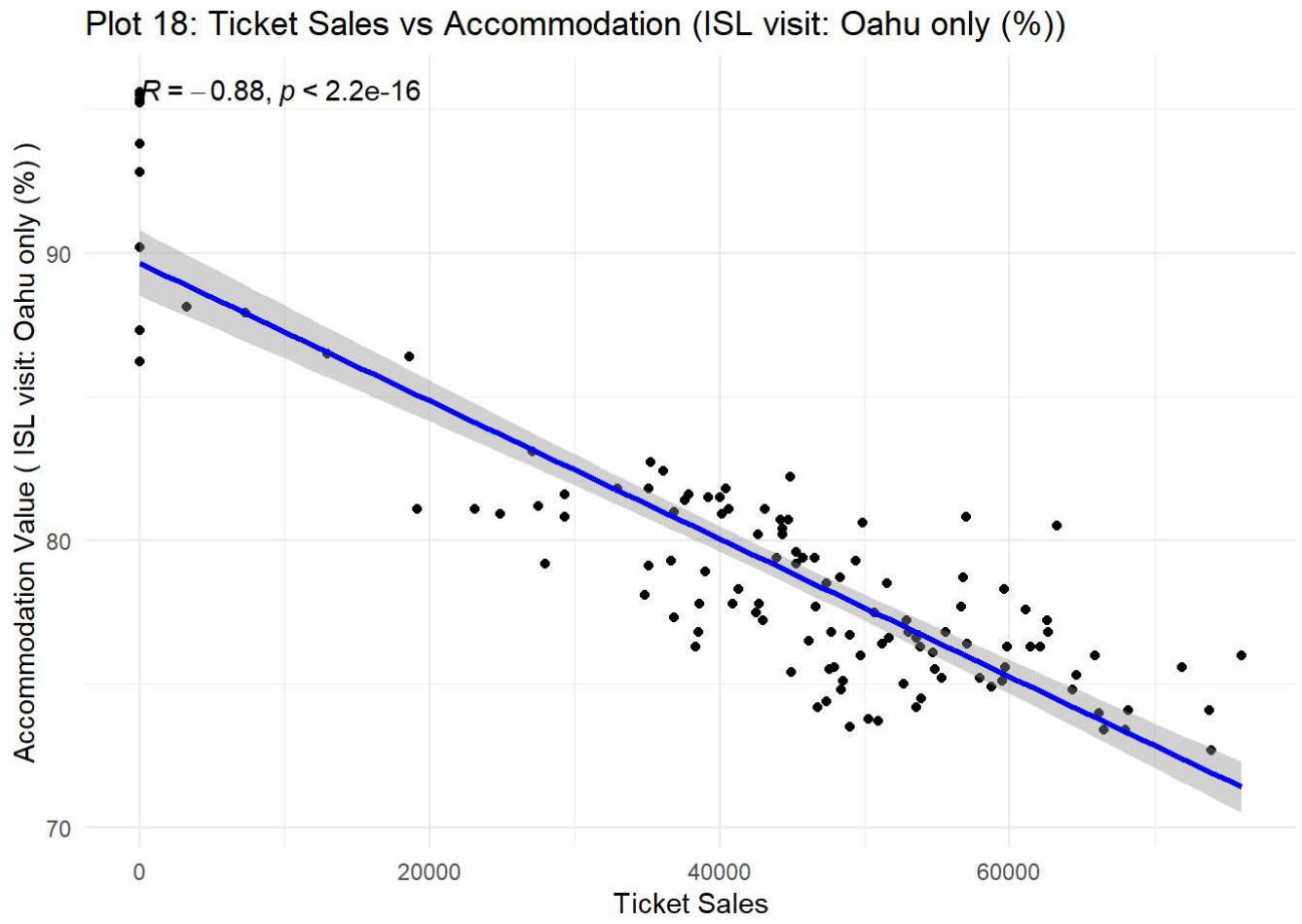

```
## Plot 17 - Ticket Sales vs ISL visit: Oahu & Neighbor ISL (%) - Pearson Correlation: 0.875
```

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




```
## Plot 18 - Ticket Sales vs ISL visit: Oahu only (%) - Pearson Correlation: -0.875
```

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




```
# Pad with NA if fewer than 18 Indicators
if(length(unique_indicators) < 18) {
  for(i in (length(unique_indicators) + 1):18) {
    cat(sprintf("Plot %d - No additional Indicator available\n", i))
    correlation_values[i] <- NA
  }
}

# Summary of correlation values
cat("\nSummary of Pearson Correlation Coefficients:\n")
```

```
##
## Summary of Pearson Correlation Coefficients:
```

```
for(i in 1:18) {
  if(i <= length(unique_indicators)) {
    indicator <- unique_indicators[i]
    cor_val <- correlation_values[i]
    if(!is.na(cor_val)) {
      cat(sprintf("Plot %d - Ticket Sales vs %s: %.3f\n", i, indicator, cor_val))
    } else {
      cat(sprintf("Plot %d - Ticket Sales vs %s: Insufficient data\n", i, indicator))
    }
  } else {
    cat(sprintf("Plot %d - No Indicator available: NA\n", i))
  }
}
```

```
## Plot 1 - Ticket Sales vs LOS on Oahu: -0.690
## Plot 2 - Ticket Sales vs Accom: B&B (%): 0.465
## Plot 3 - Ticket Sales vs Accom: Camp site/Beach (%): 0.299
## Plot 4 - Ticket Sales vs Accom: Condo (%): 0.363
## Plot 5 - Ticket Sales vs Accom: Condo only (%): -0.096
## Plot 6 - Ticket Sales vs Accom: Cruise ship (%): 0.673
## Plot 7 - Ticket Sales vs Accom: Friends/Relatives (%): -0.744
## Plot 8 - Ticket Sales vs Accom: Hostel (%): 0.544
## Plot 9 - Ticket Sales vs Accom: Hotel (%): 0.774
## Plot 10 - Ticket Sales vs Accom: Hotel only (%): 0.748
## Plot 11 - Ticket Sales vs Accom: Private room in private home (%): -0.314
## Plot 12 - Ticket Sales vs Accom: Rental house (%): -0.200
## Plot 13 - Ticket Sales vs Accom: Shared room in private home (%): -0.304
## Plot 14 - Ticket Sales vs Accom: Timeshare (%): 0.286
## Plot 15 - Ticket Sales vs Accom: Timeshare only (%): 0.129
## Plot 16 - Ticket Sales vs ISL visit: Oahu (%): Insufficient data
## Plot 17 - Ticket Sales vs ISL visit: Oahu & Neighbor ISL (%): 0.875
## Plot 18 - Ticket Sales vs ISL visit: Oahu only (%): -0.875
```

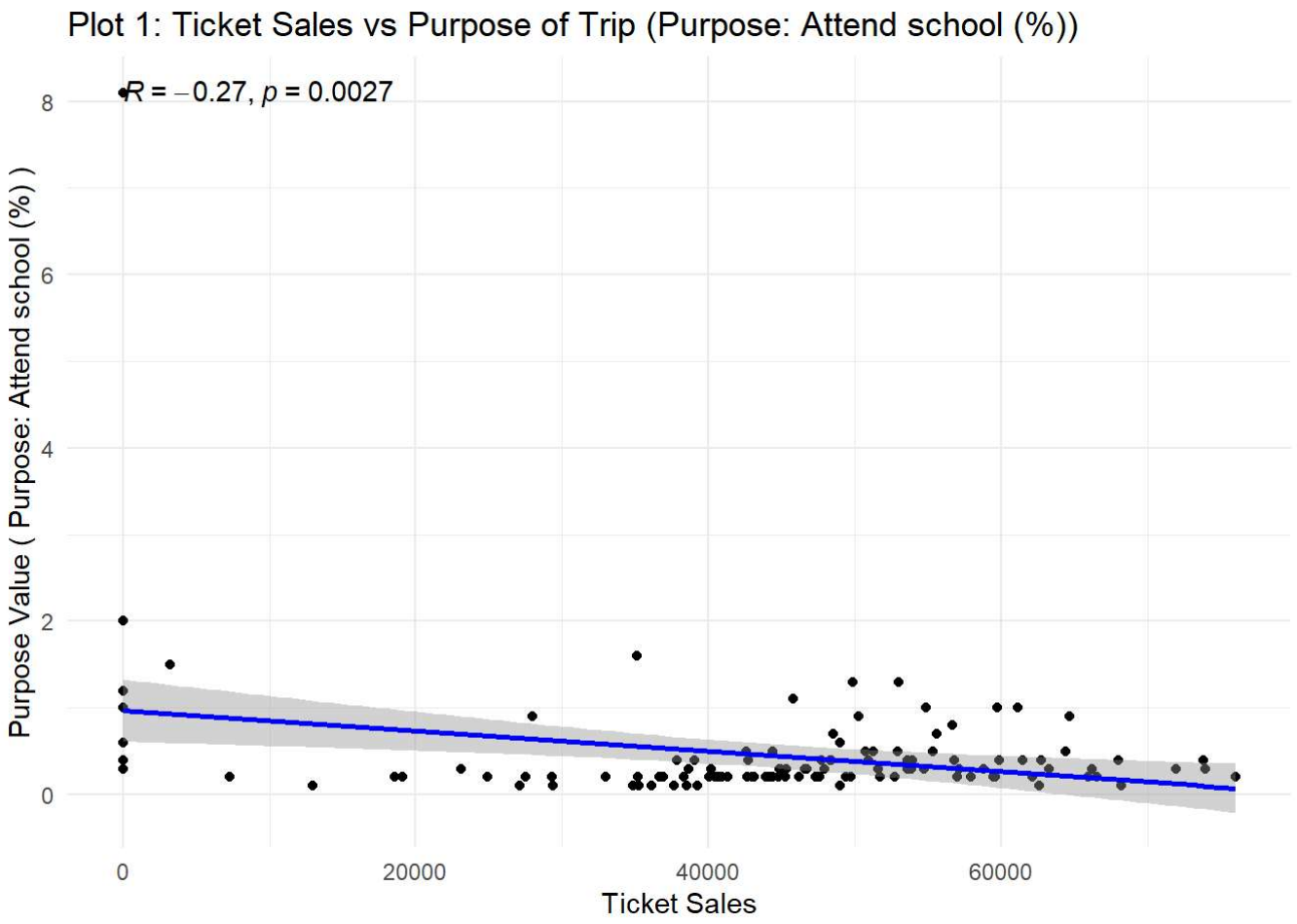
```
# Prepare ticket_data with a month-year column
if(nrow(ticket_data) > 0) {
  ticket_data$date <- as.Date(paste(ticket_data$Year, ticket_data$Month, "01", sep = "-"), forma
t = "%Y-%B-%d")
  ticket_data$month_year <- format(ticket_data$date, "%Y-%m")
  ticket_sum <- ticket_data %>%
    group_by(month_year) %>%
    summarise(total_tickets = sum(TicketSell, na.rm = TRUE))
}

# Prepare purpose of trip data and get unique Indicators
if(nrow(purpose_data) > 0) {
  purpose_data$month_year <- purpose_data$`YYYY-MM`
  unique_indicators <- unique(purpose_data$Indicator)
  num_indicators <- length(unique_indicators)
  cat(sprintf("Number of unique Indicators found: %d\n", num_indicators))

  # Limit to 15 or fewer indicators
  if(num_indicators > 15) {
    unique_indicators <- unique_indicators[1:15]
    cat("Limiting to first 15 Indicators.\n")
  }
} else {
  cat("No data in htapurposeoftrip table.\n")
  unique_indicators <- character(0)
}
```

```
## Number of unique Indicators found: 15
```

```
# Function to plot correlation for a specific Indicator
plot_correlation_by_indicator <- function(ticket_data, purpose_data, indicator, plot_num) {
  # Filter purpose data for the specific Indicator
  purpose_subset <- purpose_data %>%
    filter(Indicator == indicator) %>%
    group_by(month_year) %>%
    summarise(total_purpose = sum(value, na.rm = TRUE))

  # Merge with ticket data
  merged_data <- merge(ticket_data, purpose_subset, by = "month_year", all = FALSE)

  if(nrow(merged_data) > 1) {  # Need at least 2 points for correlation
    # Calculate correlation
    cor_value <- cor(merged_data$total_tickets, merged_data$total_purpose, use = "complete.obs",
method = "pearson")
    cat(sprintf("Plot %d - Ticket Sales vs %s - Pearson Correlation: %.3f\n", plot_num, indicato
r, cor_value))

    # Create scatter plot
    p <- ggplot(merged_data, aes(x = total_tickets, y = total_purpose)) +
      geom_point() +
      geom_smooth(method = "lm", color = "blue") +
      labs(title = sprintf("Plot %d: Ticket Sales vs Purpose of Trip (%s)", plot_num, indicato
r),
           x = "Ticket Sales", y = paste("Purpose Value (", indicator, ")")) +
      stat_cor(method = "pearson", label.x = min(merged_data$total_tickets, na.rm = TRUE),
               label.y = max(merged_data$total_purpose, na.rm = TRUE)) +
      theme_minimal()
    print(p)
    Sys.sleep(1)
    return(cor_value)
  } else {
    cat(sprintf("Plot %d - Ticket Sales vs %s - Insufficient data for correlation\n", plot_num,
indicator))
    return(NA)
  }
}

# Generate up to 15 correlation plots
correlation_values <- vector("numeric", min(15, length(unique_indicators)))
if(nrow(ticket_data) > 0 && nrow(purpose_data) > 0) {
  for(i in seq_along(unique_indicators)) {
    indicator <- unique_indicators[i]
    correlation_values[i] <- plot_correlation_by_indicator(ticket_sum, purpose_data, indicator,
i)
  }
}
```

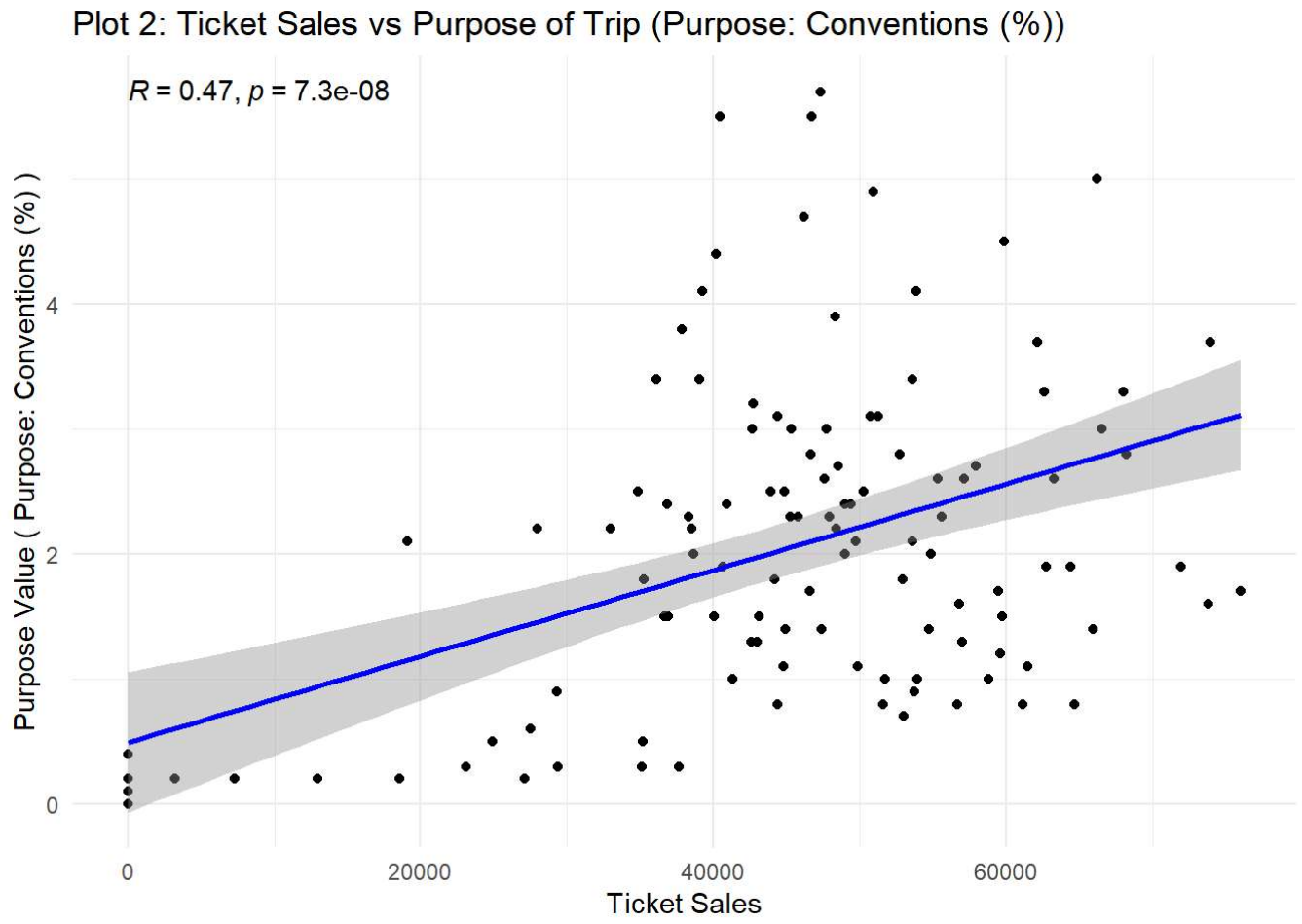
```
## Plot 1 - Ticket Sales vs Purpose: Attend school (%) - Pearson Correlation: -0.271
```

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



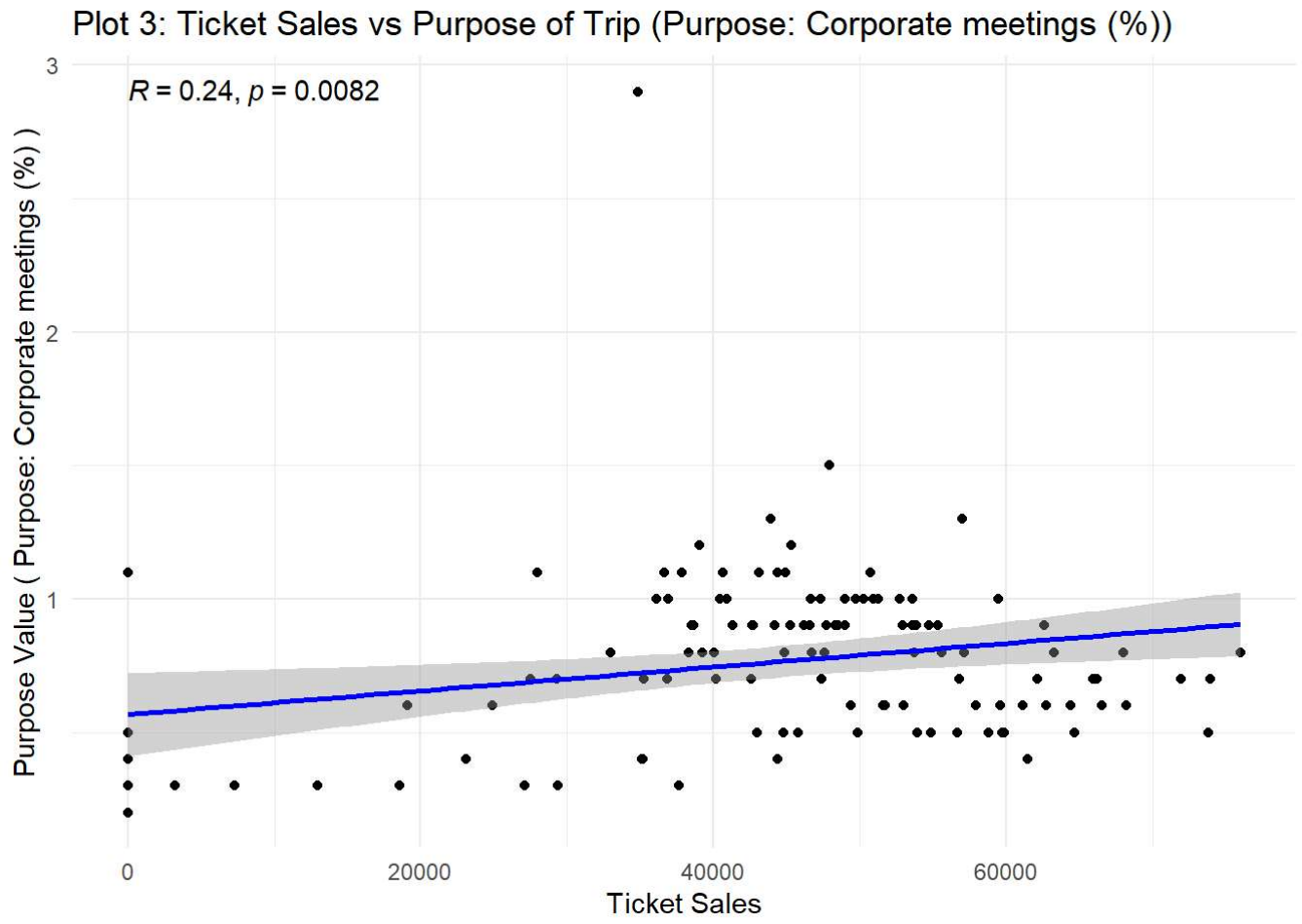

```
## Plot 2 - Ticket Sales vs Purpose: Conventions (%) - Pearson Correlation: 0.468
```

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



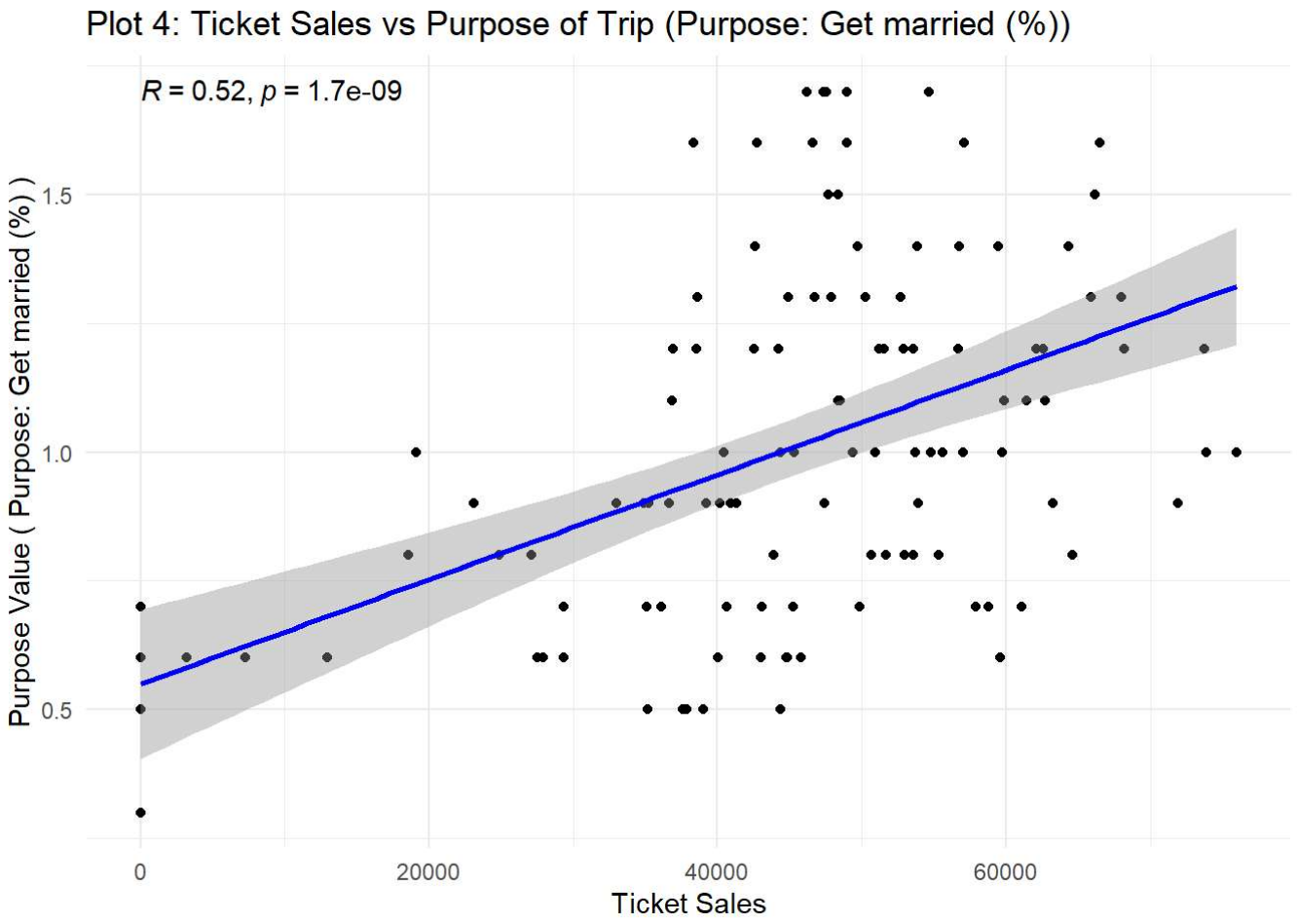

```
## Plot 3 - Ticket Sales vs Purpose: Corporate meetings (%) - Pearson Correlation: 0.240
```

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



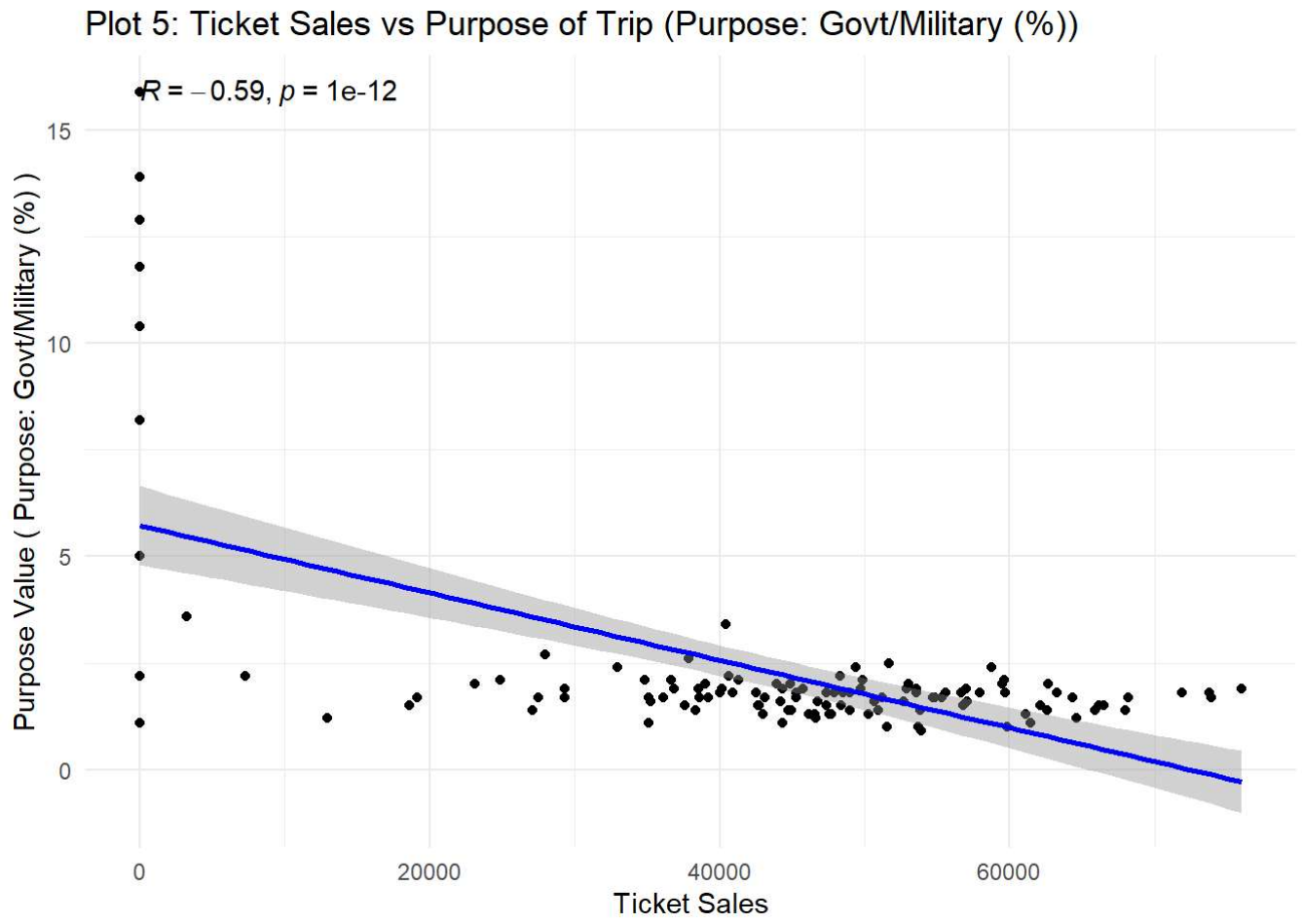

```
## Plot 4 - Ticket Sales vs Purpose: Get married (%) - Pearson Correlation: 0.515
```

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



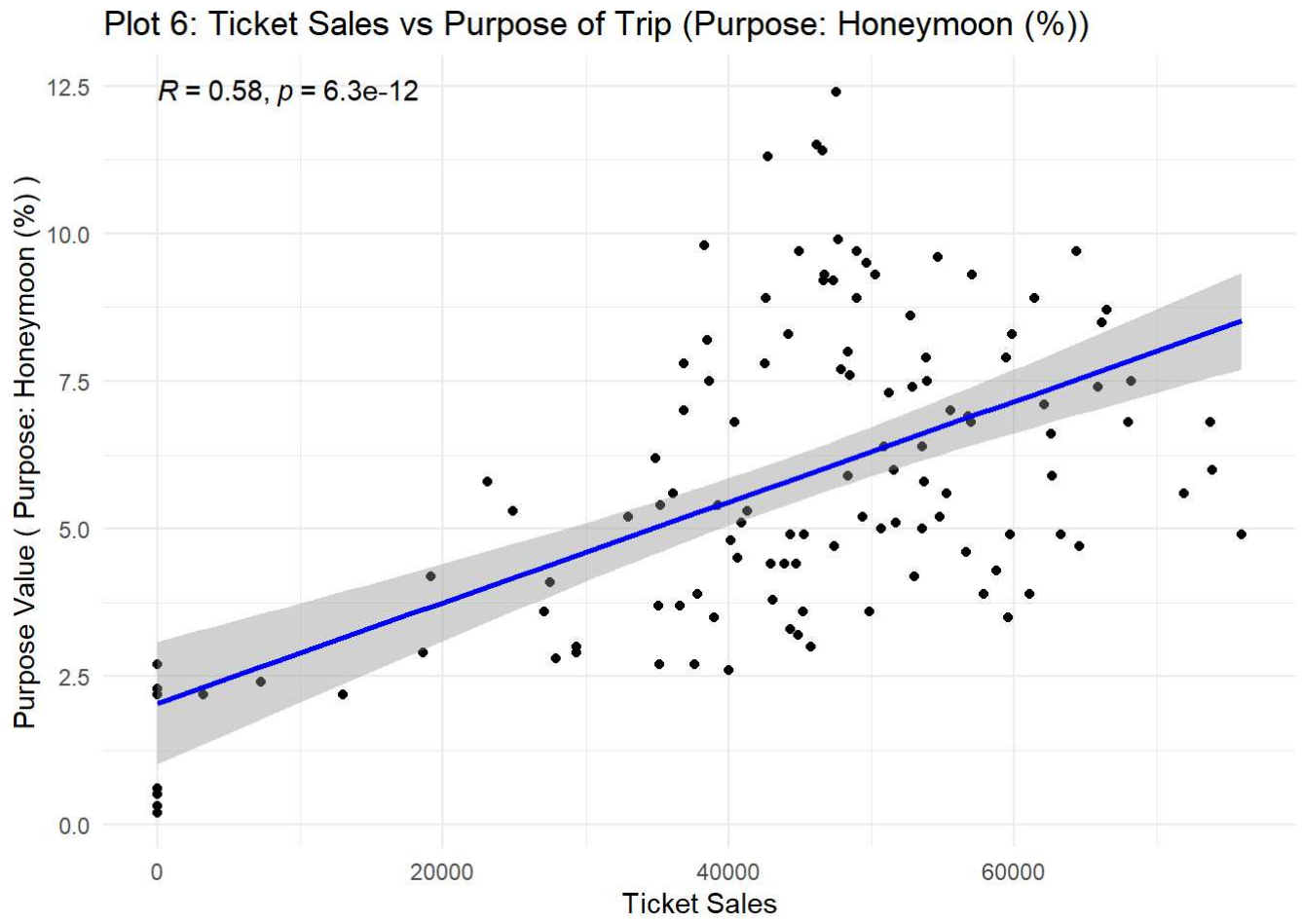

```
## Plot 5 - Ticket Sales vs Purpose: Govt/Military (%) - Pearson Correlation: -0.592
```

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



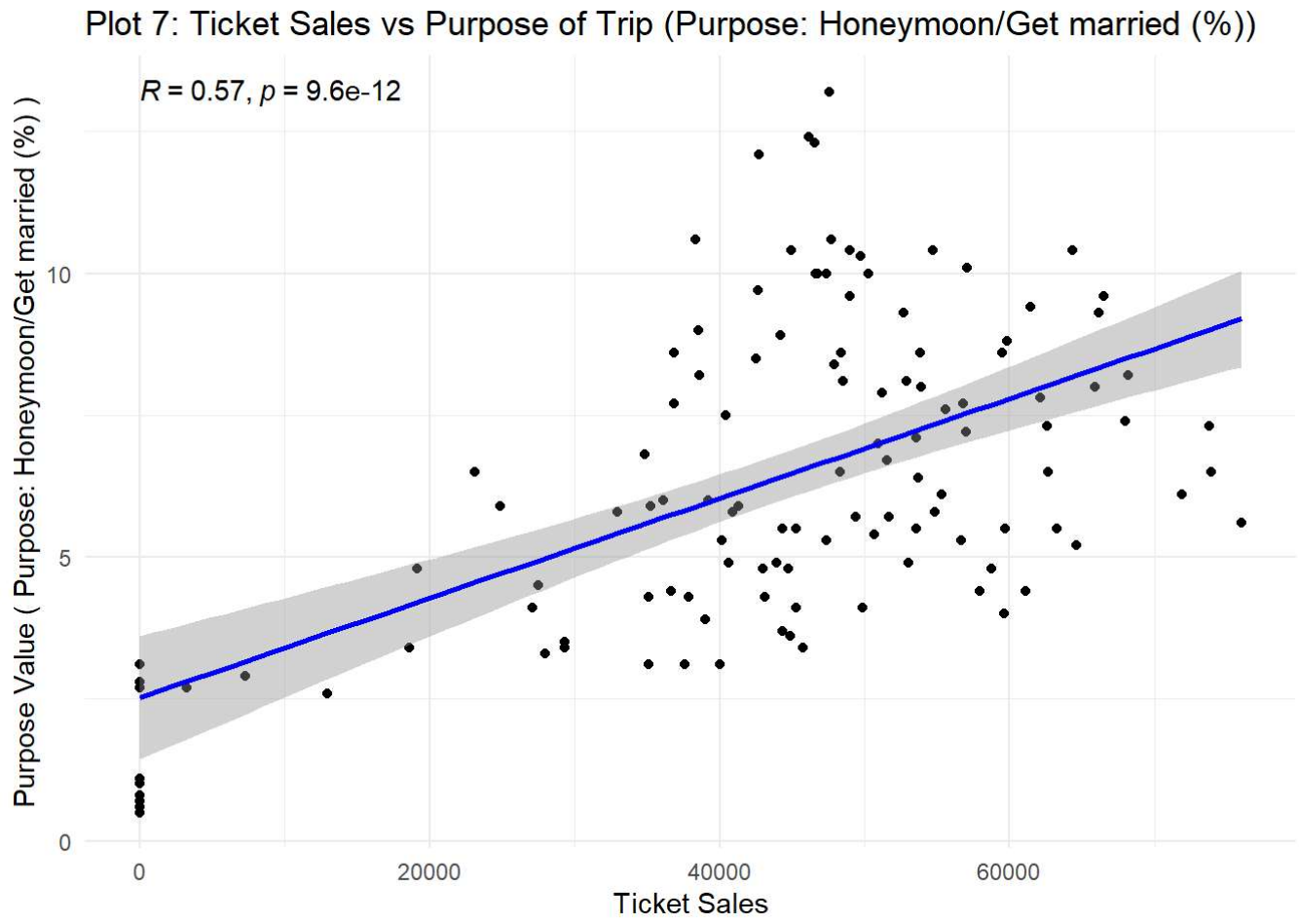

```
## Plot 6 - Ticket Sales vs Purpose: Honeymoon (%) - Pearson Correlation: 0.575
```

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



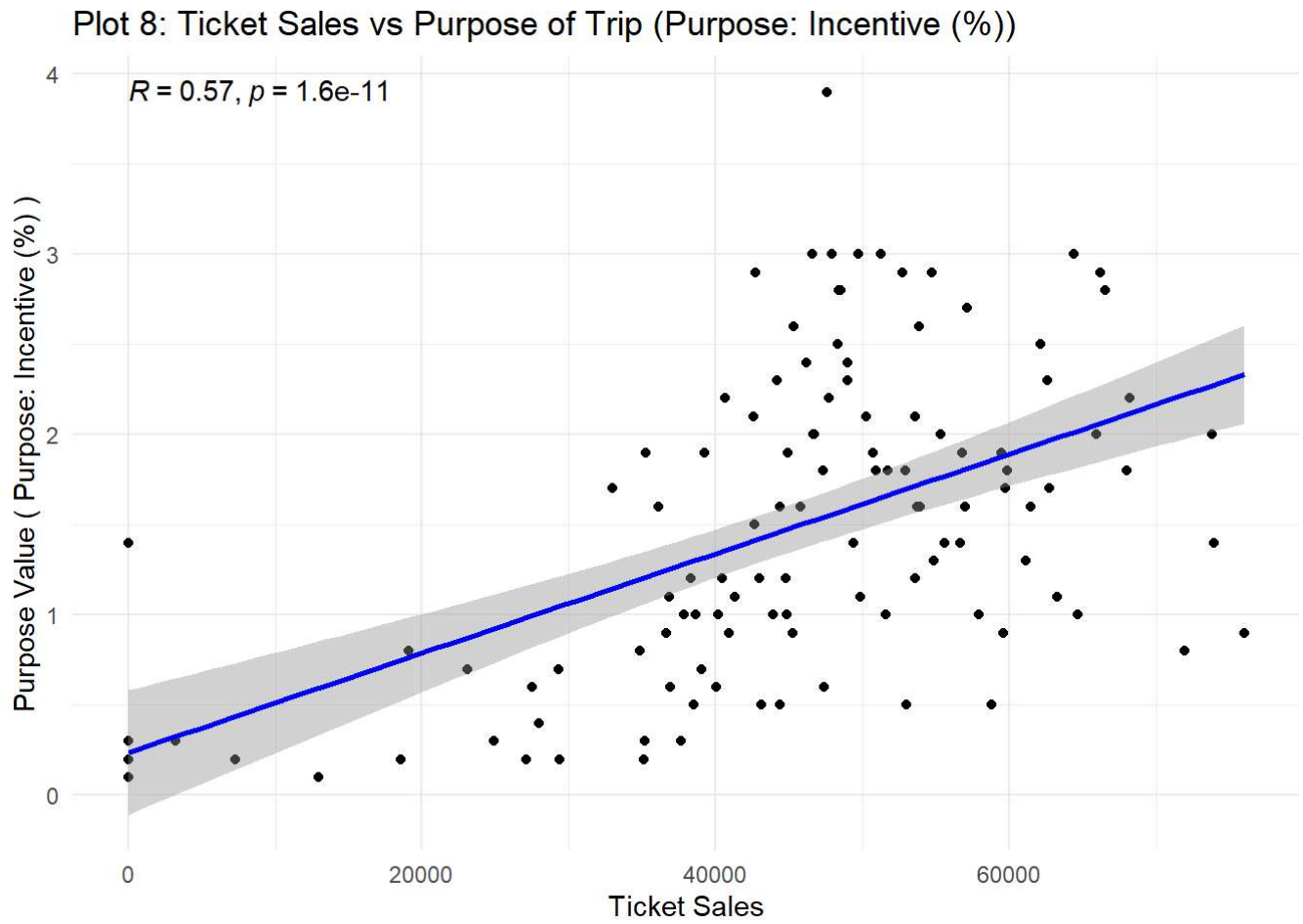

```
## Plot 7 - Ticket Sales vs Purpose: Honeymoon/Get married (%) - Pearson Correlation: 0.571
```

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



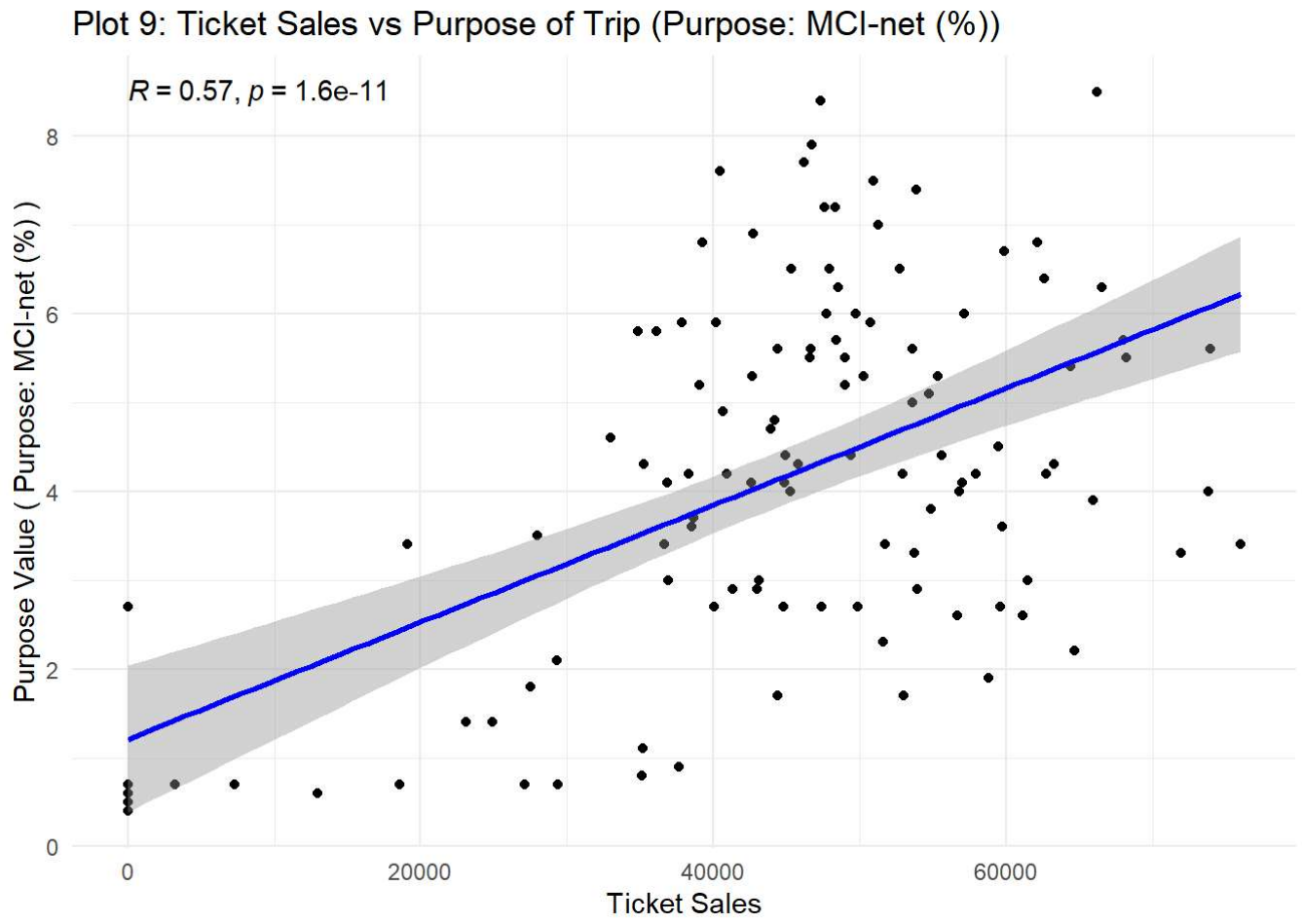

```
## Plot 8 - Ticket Sales vs Purpose: Incentive (%) - Pearson Correlation: 0.566
```

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



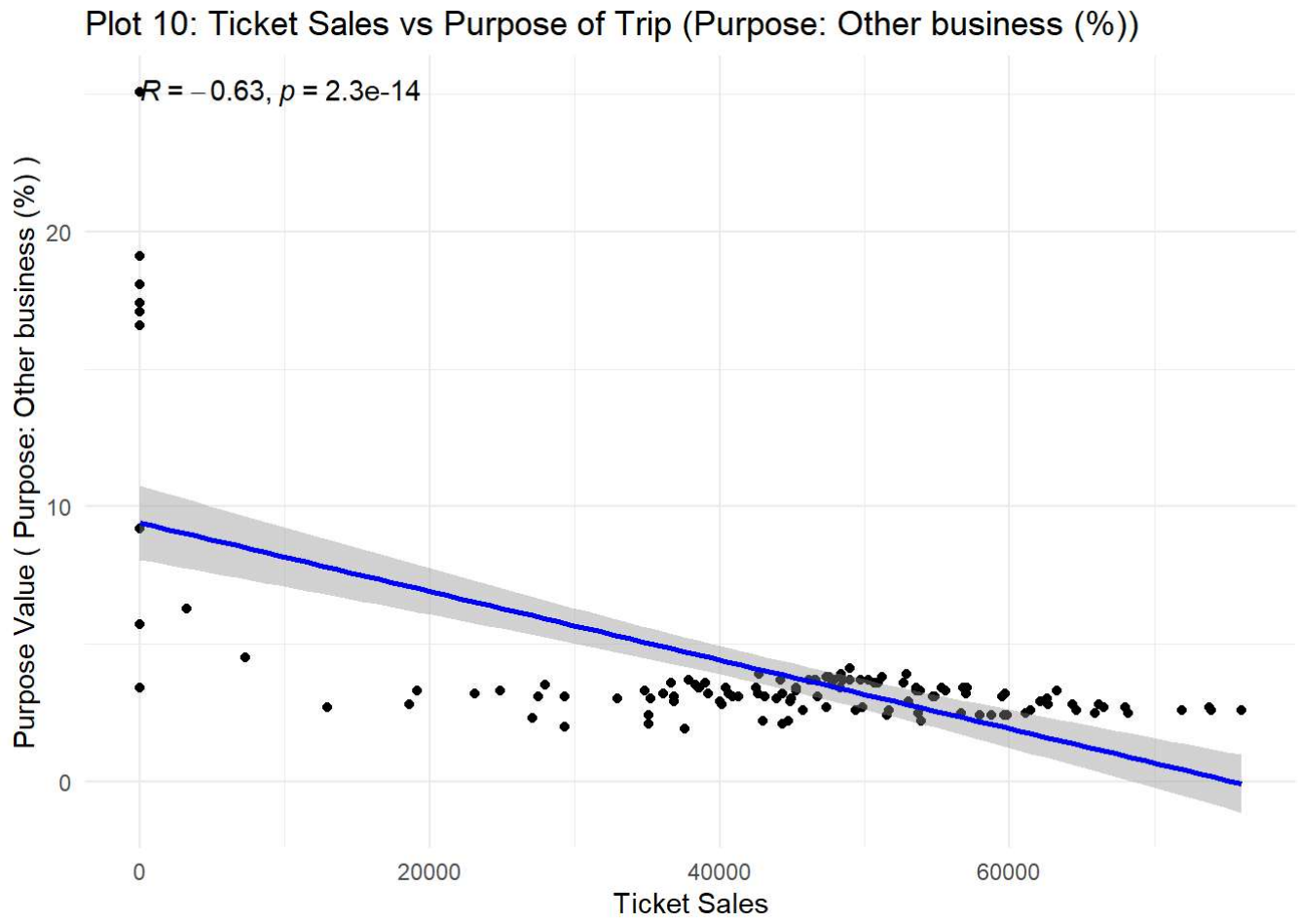

```
## Plot 9 - Ticket Sales vs Purpose: MCI-net (%) - Pearson Correlation: 0.566
```

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



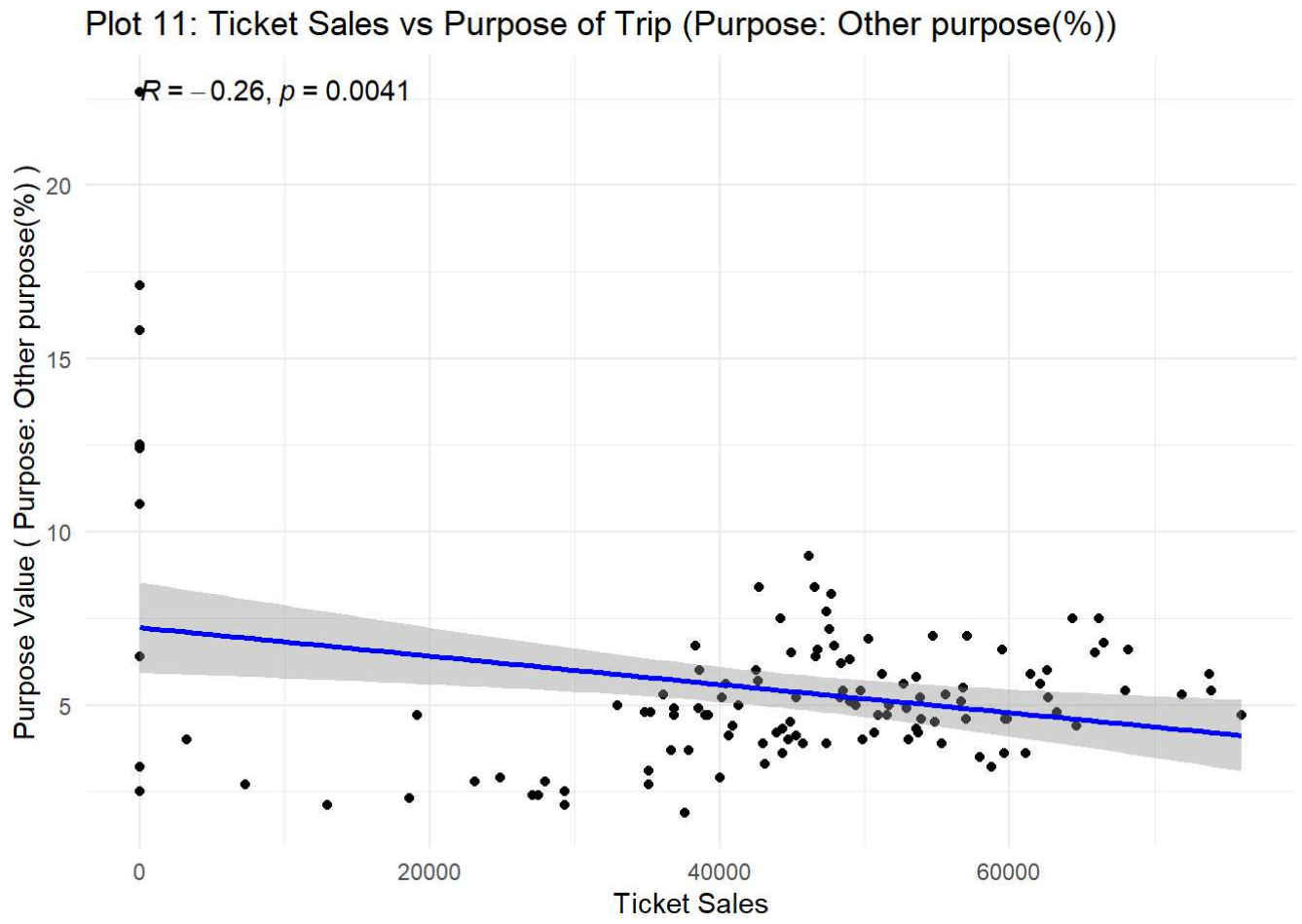

```
## Plot 10 - Ticket Sales vs Purpose: Other business (%) - Pearson Correlation: -0.625
```

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



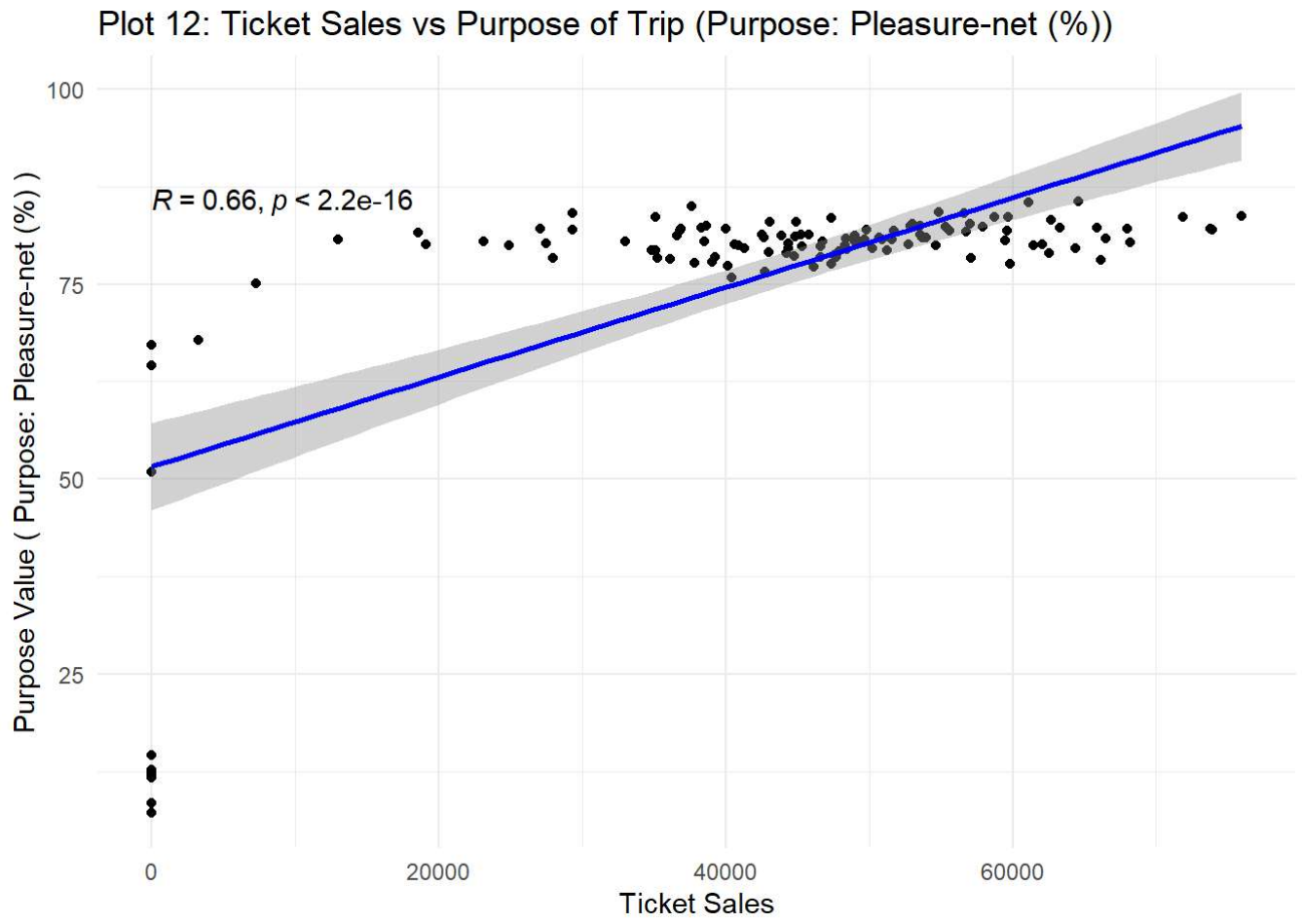

```
## Plot 11 - Ticket Sales vs Purpose: Other purpose(%) - Pearson Correlation: -0.260
```

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



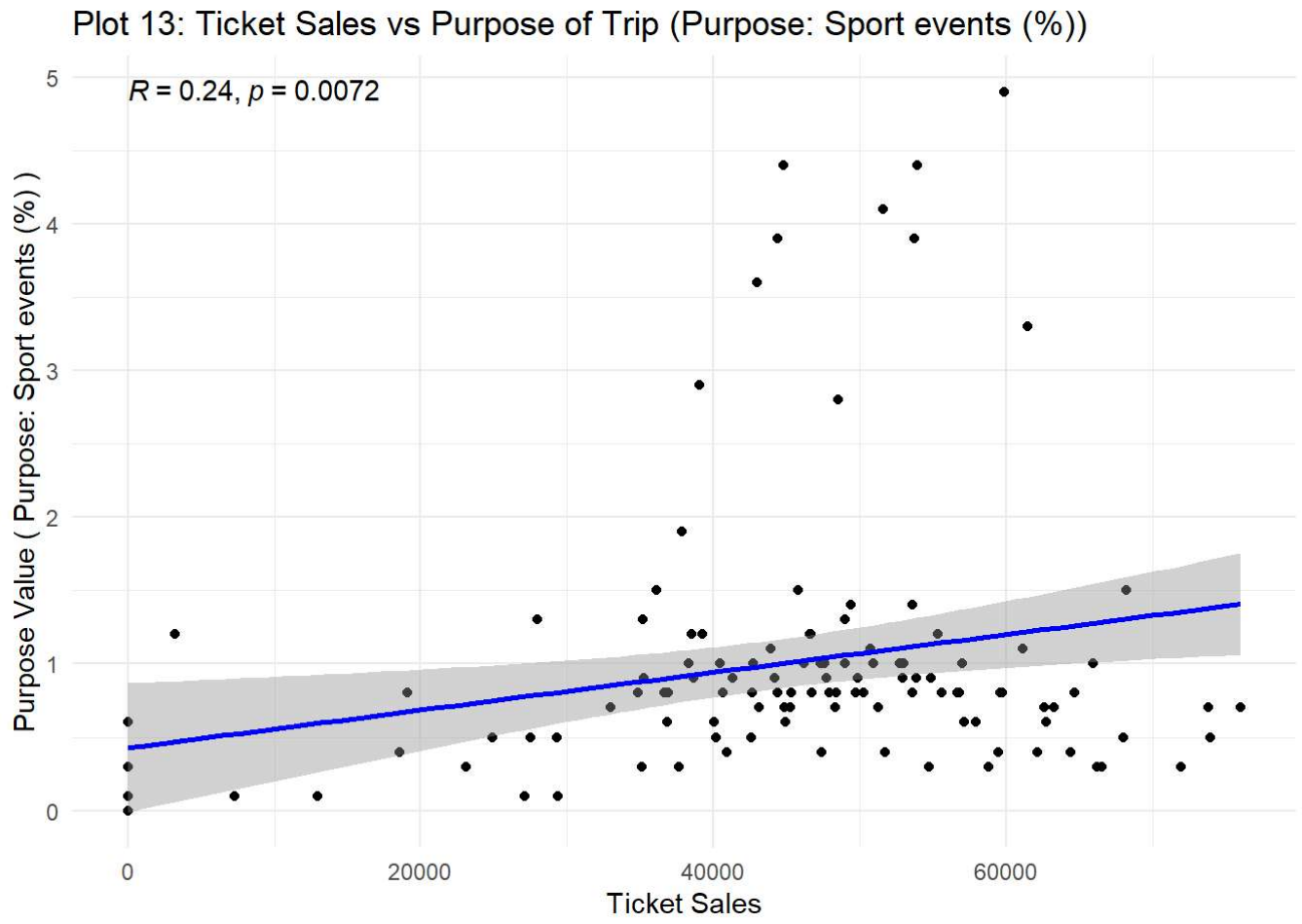

```
## Plot 12 - Ticket Sales vs Purpose: Pleasure-net (%) - Pearson Correlation: 0.663
```

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



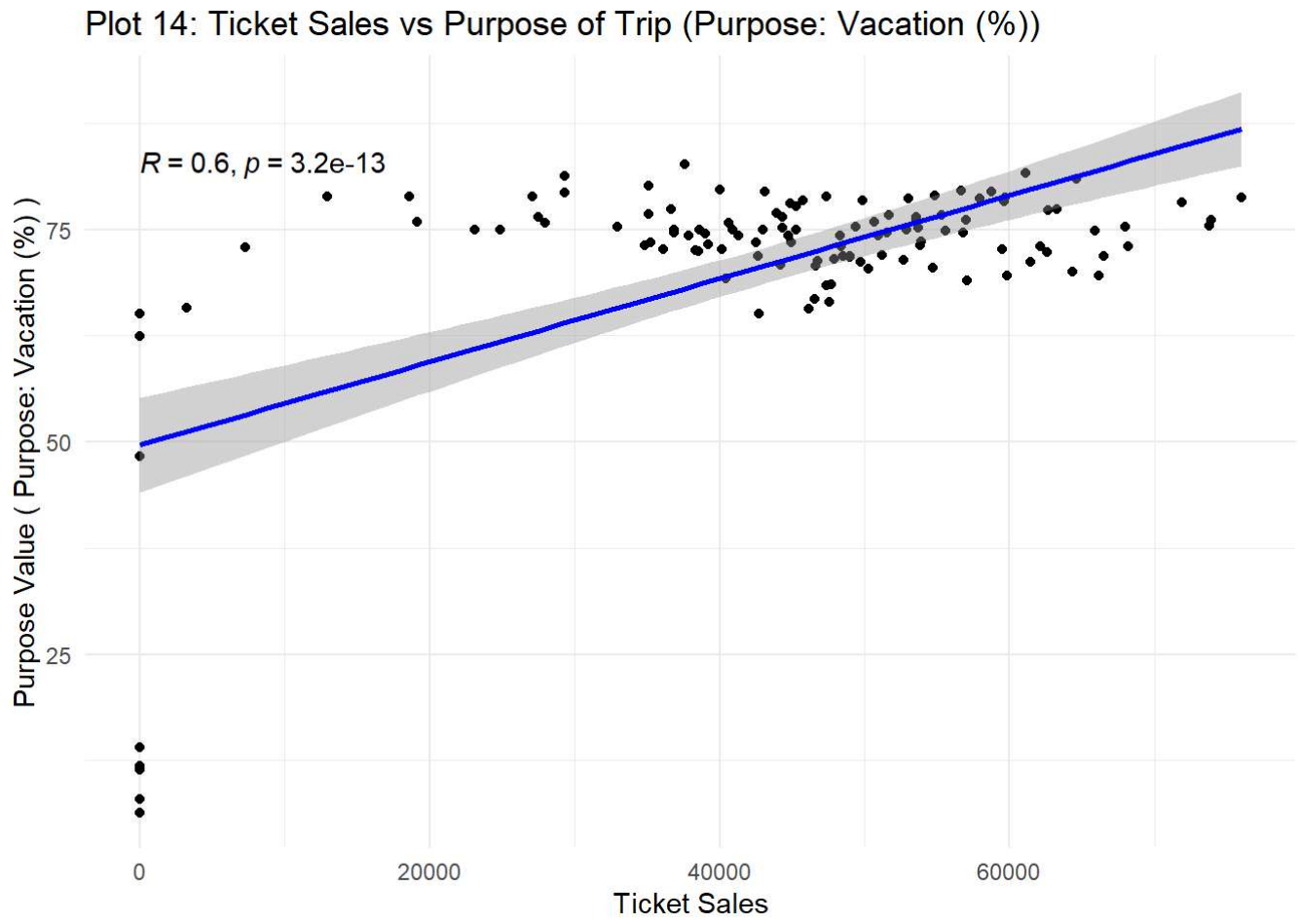

```
## Plot 13 - Ticket Sales vs Purpose: Sport events (%) - Pearson Correlation: 0.244
```

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



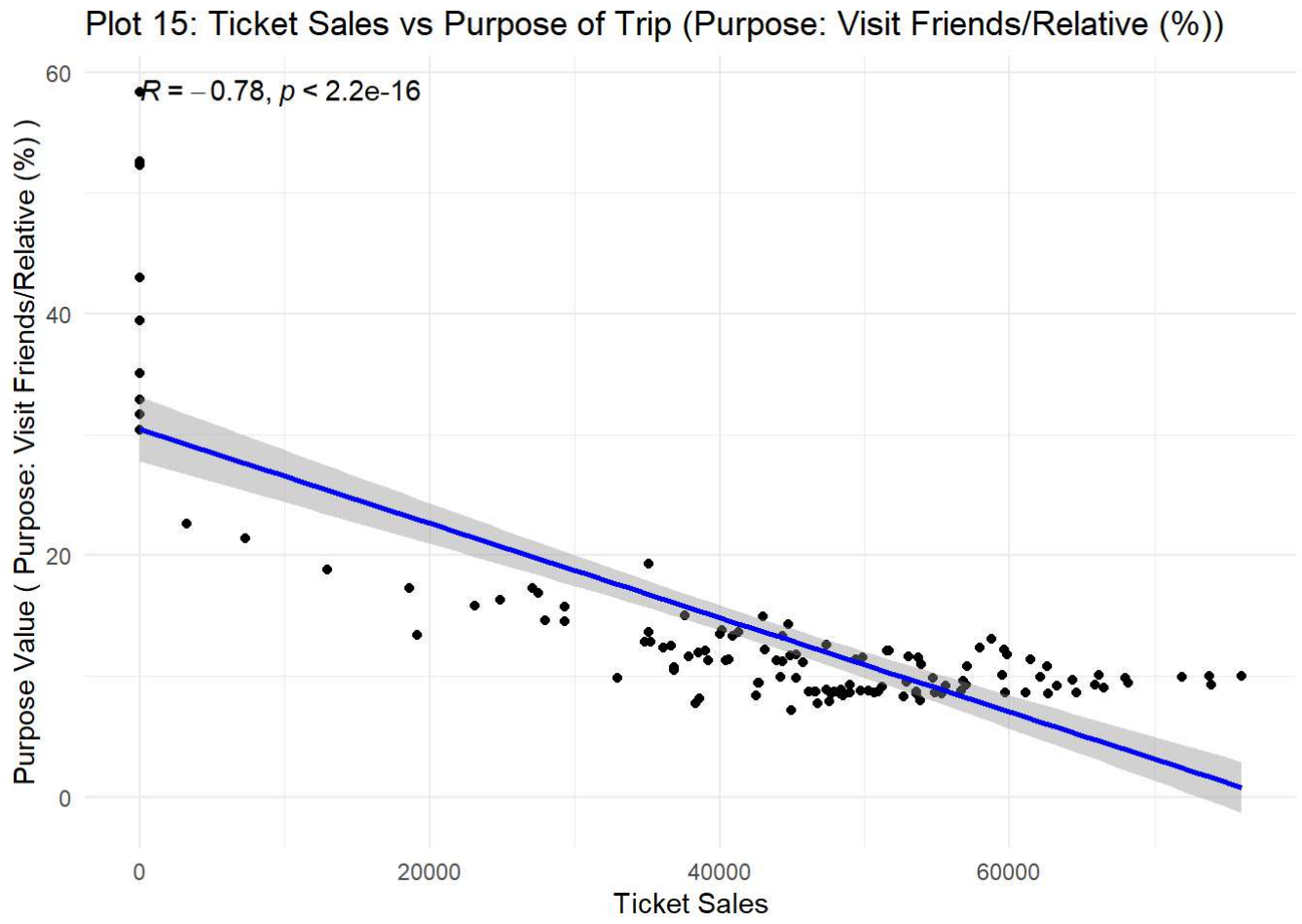

```
## Plot 14 - Ticket Sales vs Purpose: Vacation (%) - Pearson Correlation: 0.603
```

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




```
## Plot 15 - Ticket Sales vs Purpose: Visit Friends/Relative (%) - Pearson Correlation: -0.781
```

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



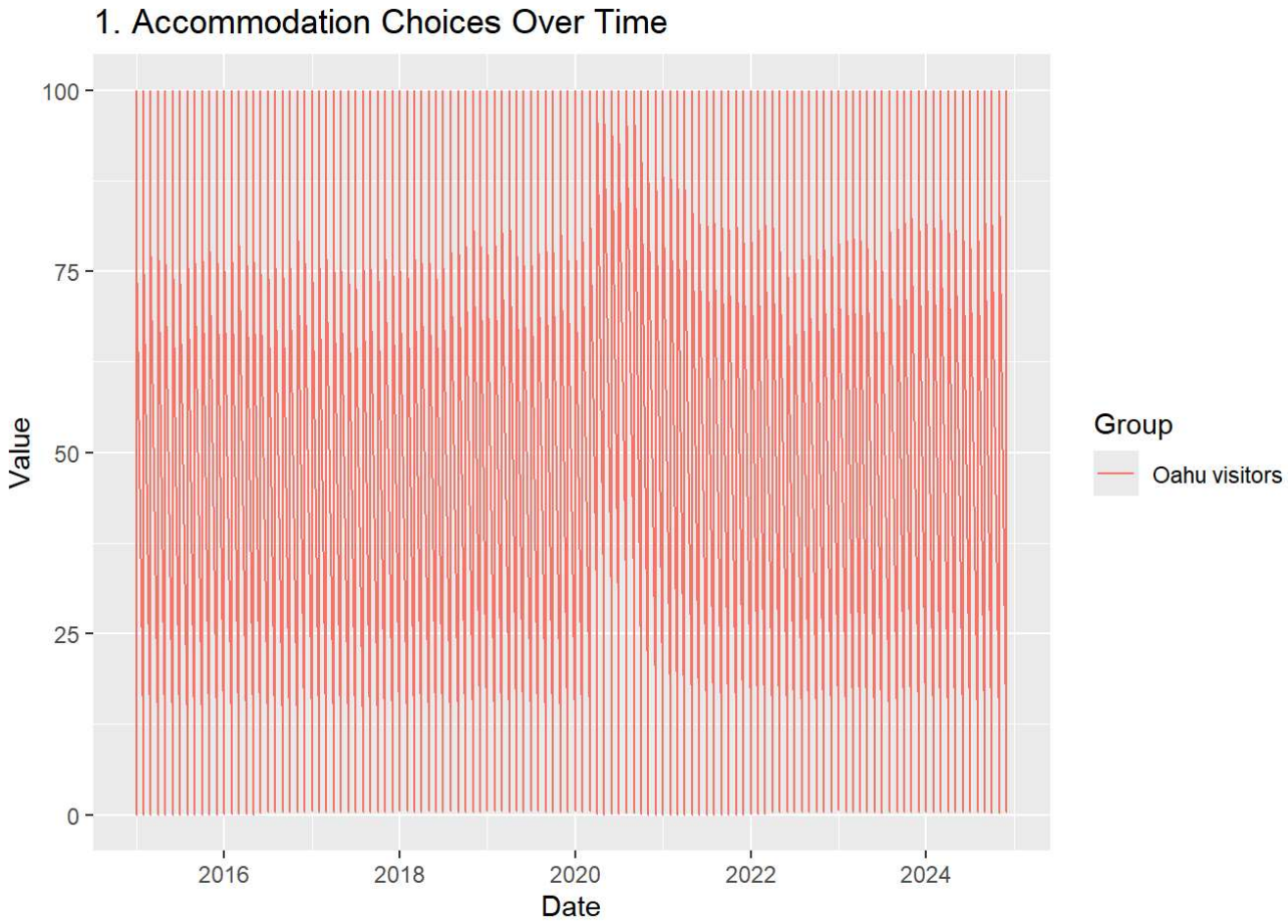

```
# Pad with NA if fewer than 15 Indicators
if(length(unique_indicators) < 15) {
  for(i in (length(unique_indicators) + 1):15) {
    cat(sprintf("Plot %d - No additional Indicator available\n", i))
    correlation_values[i] <- NA
  }
}

# Summary of correlation values
cat("\nSummary of Pearson Correlation Coefficients:\n")
```

```
##
## Summary of Pearson Correlation Coefficients:
```

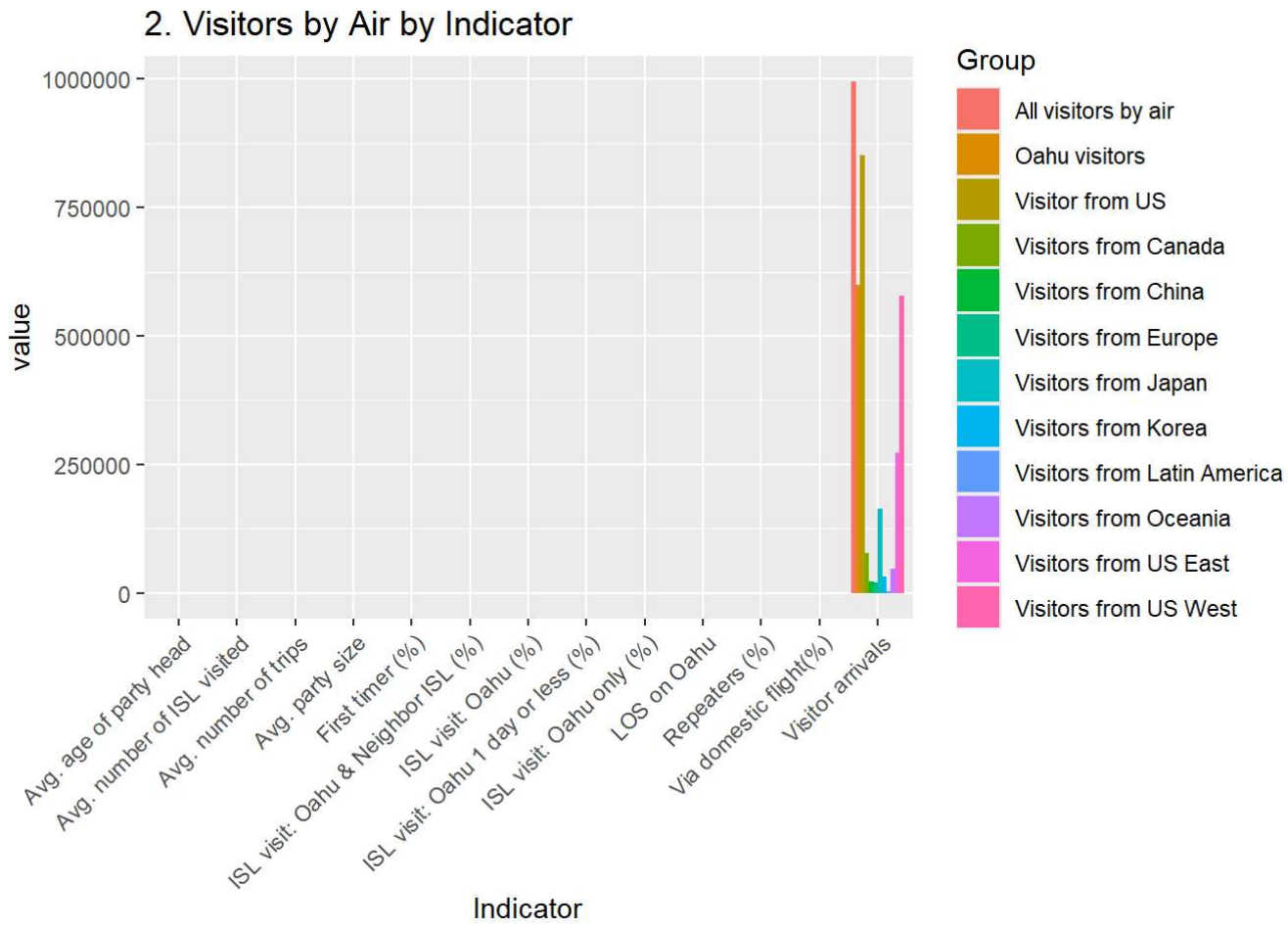
```
for(i in 1:15) {
  if(i <= length(unique_indicators)) {
    indicator <- unique_indicators[i]
    cor_val <- correlation_values[i]
    if(!is.na(cor_val)) {
      cat(sprintf("Plot %d - Ticket Sales vs %s: %.3f\n", i, indicator, cor_val))
    } else {
      cat(sprintf("Plot %d - Ticket Sales vs %s: Insufficient data\n", i, indicator))
    }
  } else {
    cat(sprintf("Plot %d - No Indicator available: NA\n", i))
  }
}
```

```
## Plot 1 - Ticket Sales vs Purpose: Attend school (%): -0.271
## Plot 2 - Ticket Sales vs Purpose: Conventions (%): 0.468
## Plot 3 - Ticket Sales vs Purpose: Corporate meetings (%): 0.240
## Plot 4 - Ticket Sales vs Purpose: Get married (%): 0.515
## Plot 5 - Ticket Sales vs Purpose: Govt/Military (%): -0.592
## Plot 6 - Ticket Sales vs Purpose: Honeymoon (%): 0.575
## Plot 7 - Ticket Sales vs Purpose: Honeymoon/Get married (%): 0.571
## Plot 8 - Ticket Sales vs Purpose: Incentive (%): 0.566
## Plot 9 - Ticket Sales vs Purpose: MCI-net (%): 0.566
## Plot 10 - Ticket Sales vs Purpose: Other business (%): -0.625
## Plot 11 - Ticket Sales vs Purpose: Other purpose(%): -0.260
## Plot 12 - Ticket Sales vs Purpose: Pleasure-net (%): 0.663
## Plot 13 - Ticket Sales vs Purpose: Sport events (%): 0.244
## Plot 14 - Ticket Sales vs Purpose: Vacation (%): 0.603
## Plot 15 - Ticket Sales vs Purpose: Visit Friends/Relative (%): -0.781
```

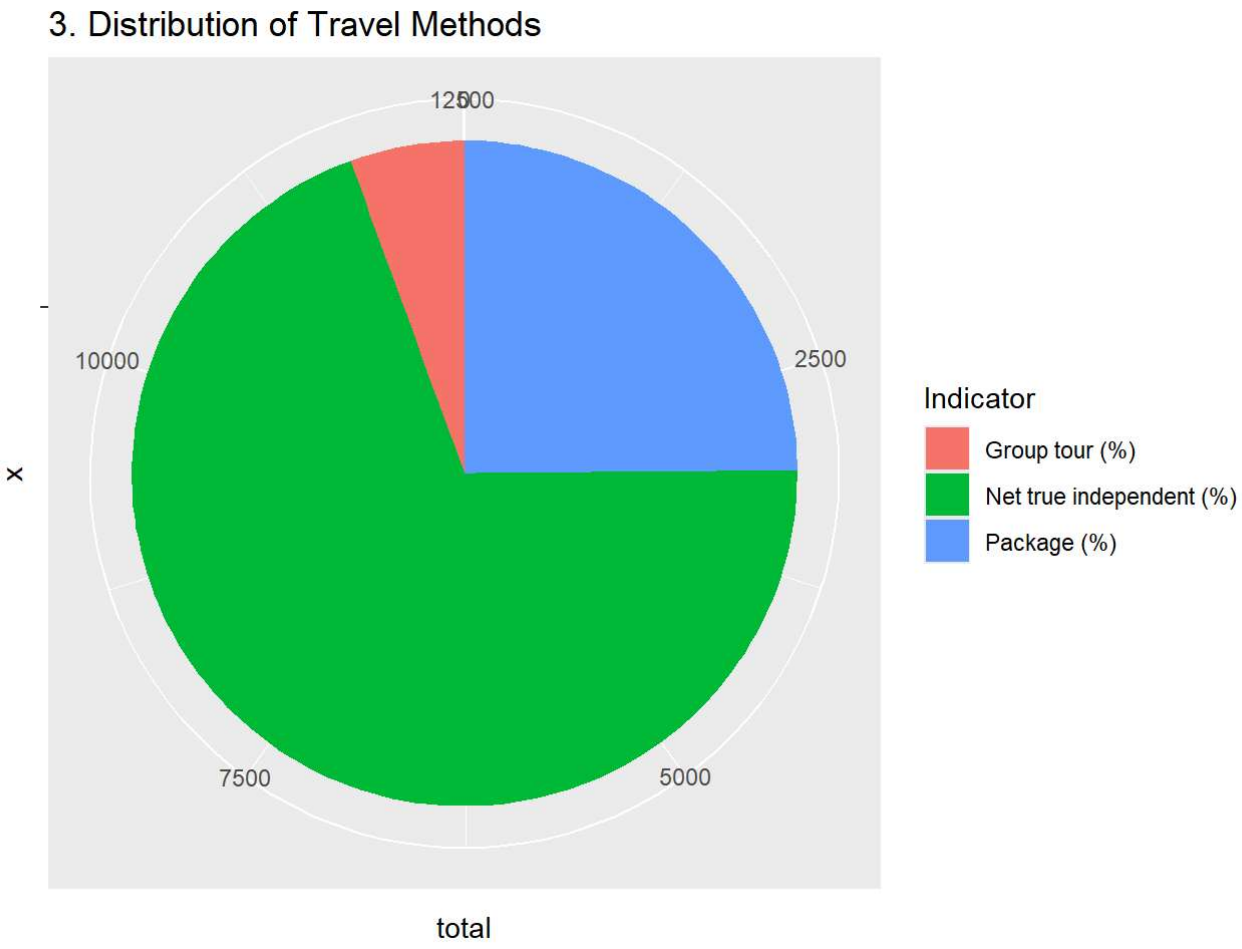
```
# 1. Accommodation Choices Time Series
if(nrow(accom_data) > 0) {
  ggplot(accom_data, aes(x = as.Date(paste0(`YYYY-MM`, "-01")), y = value, color = Group)) +
    geom_line() +
    labs(title = "1. Accommodation Choices Over Time", x = "Date", y = "Value")
}
```



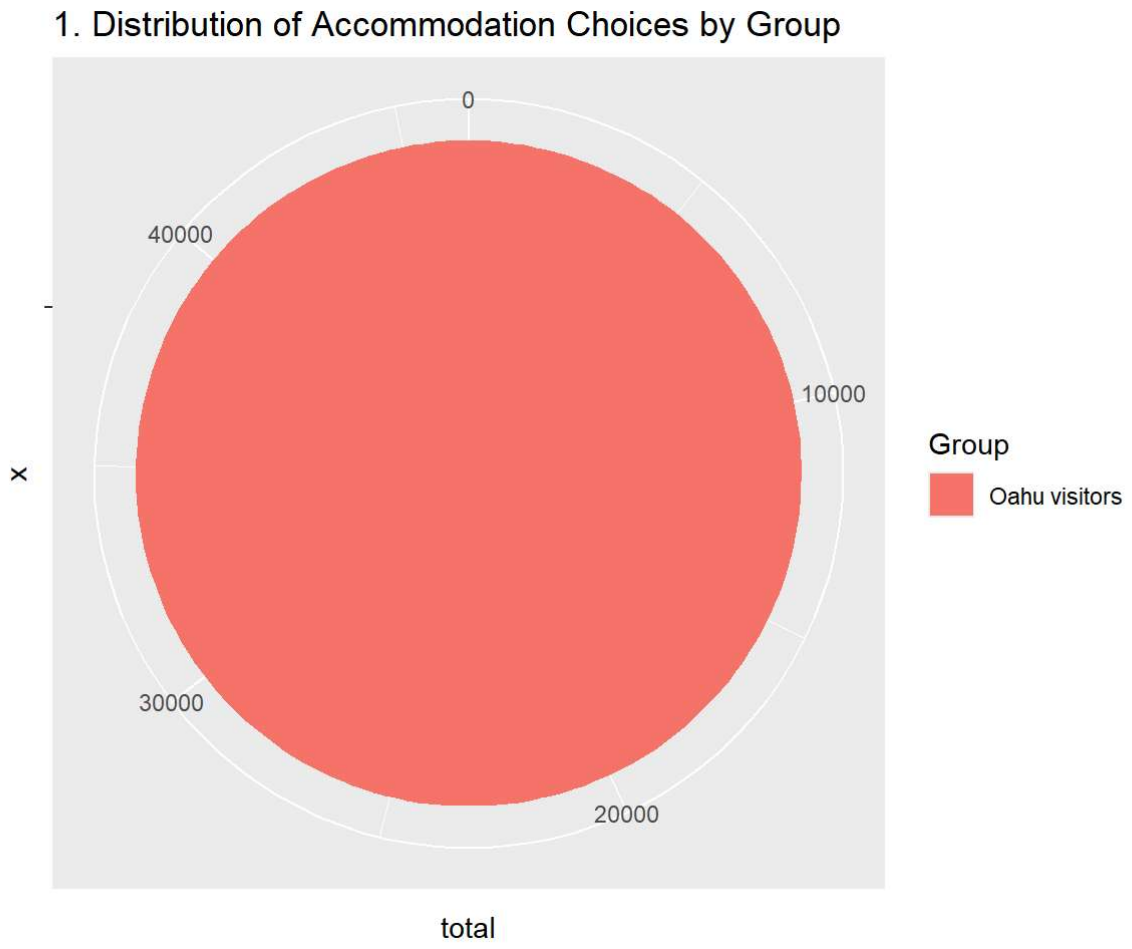

```
# 2. Visitors by Air Bar Plot
if(nrow(air_data) > 0) {
  ggplot(air_data, aes(x = Indicator, y = value, fill = Group)) +
    geom_bar(stat = "identity", position = "dodge") +
    theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
    labs(title = "2. Visitors by Air by Indicator")
}
```



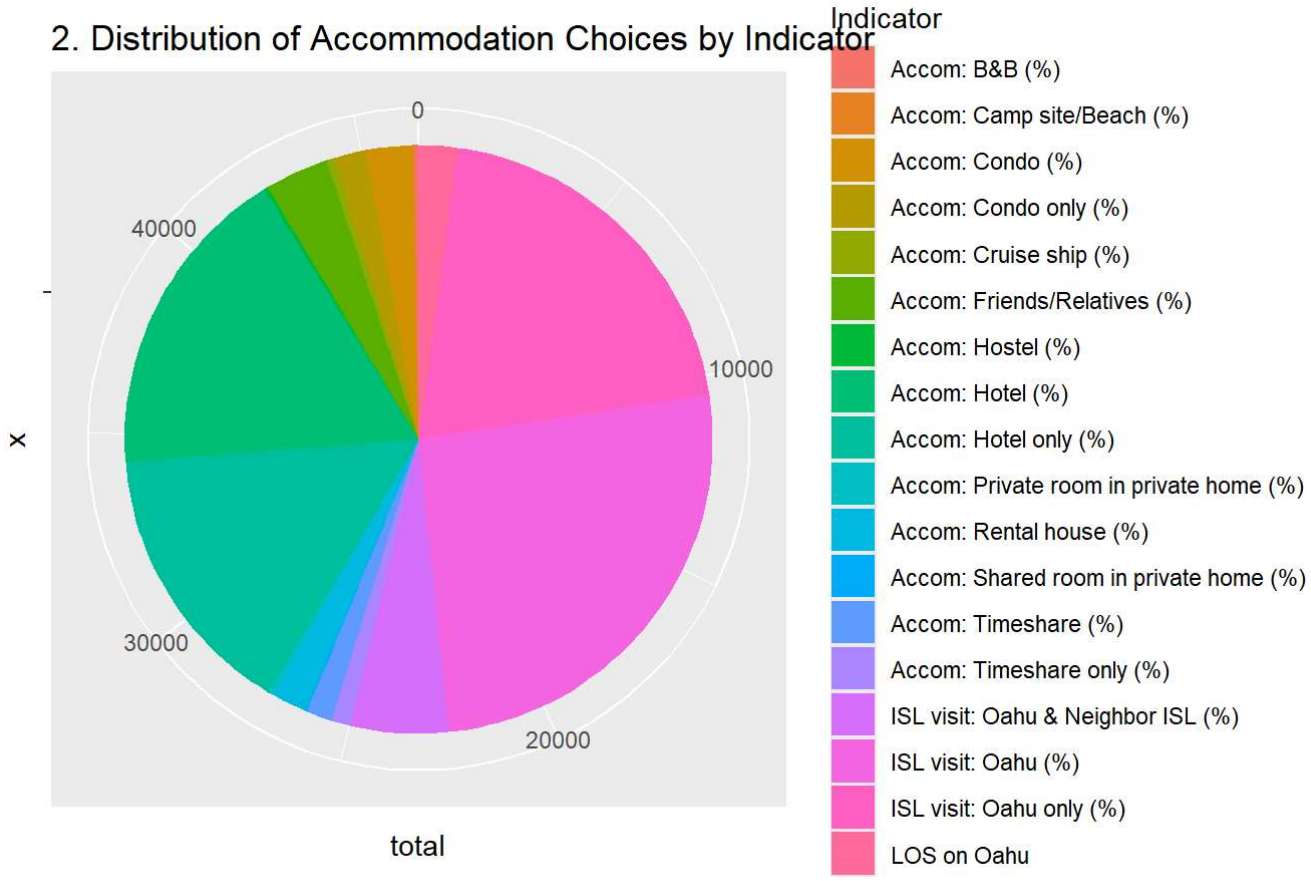

```
# 3. Method of Travel Pie Chart
if(nrow(travel_data) > 0) {
  travel_sum <- travel_data %>% group_by(Indicator) %>% summarise(total = sum(value, na.rm = TRU
E))
  ggplot(travel_sum, aes(x = "", y = total, fill = Indicator)) +
    geom_bar(stat = "identity") +
    coord_polar("y") +
    labs(title = "3. Distribution of Travel Methods")
}
```



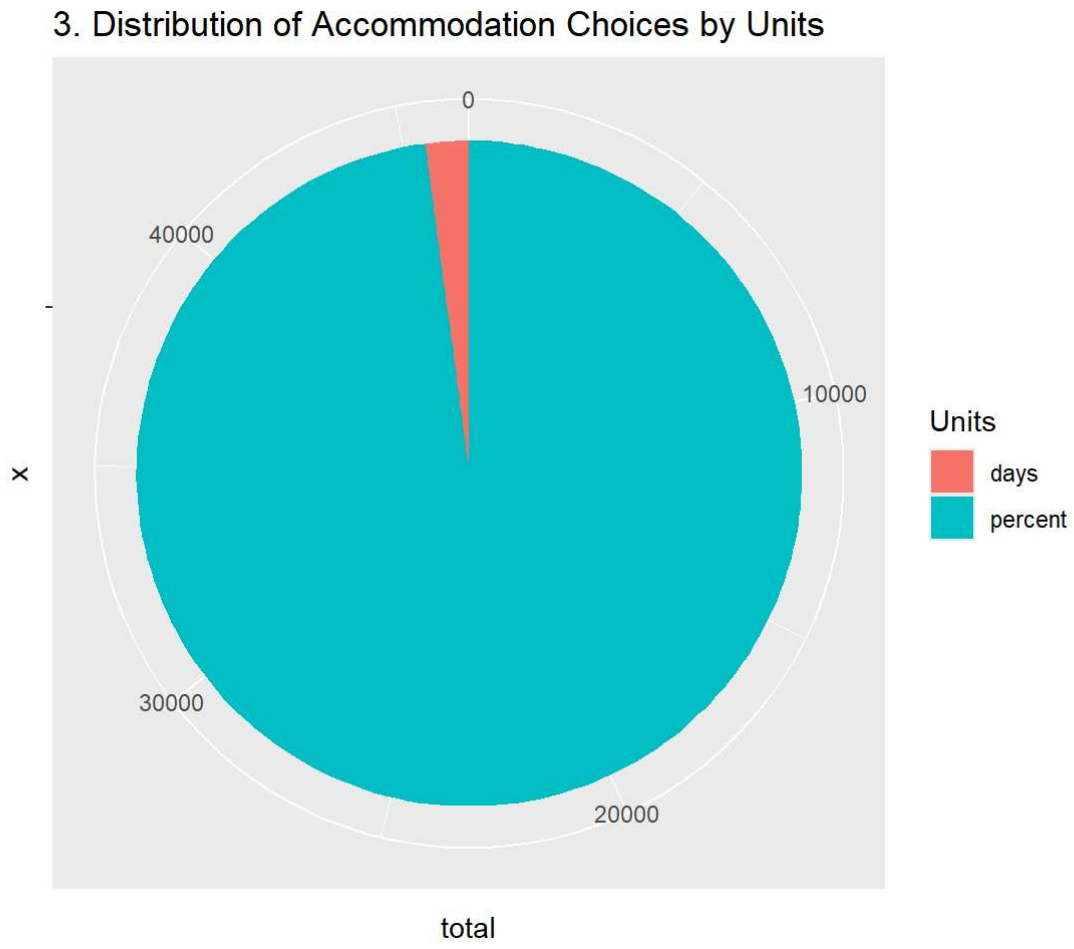

```
# 1. Accommodation Choices by Group
if(nrow(accom_data) > 0) {
  accom_sum <- accom_data %>% group_by(Group) %>% summarise(total = sum(value, na.rm = TRUE))
  ggplot(accom_sum, aes(x = "", y = total, fill = Group)) +
    geom_bar(stat = "identity") +
    coord_polar("y") +
    labs(title = "1. Distribution of Accommodation Choices by Group")

}
```



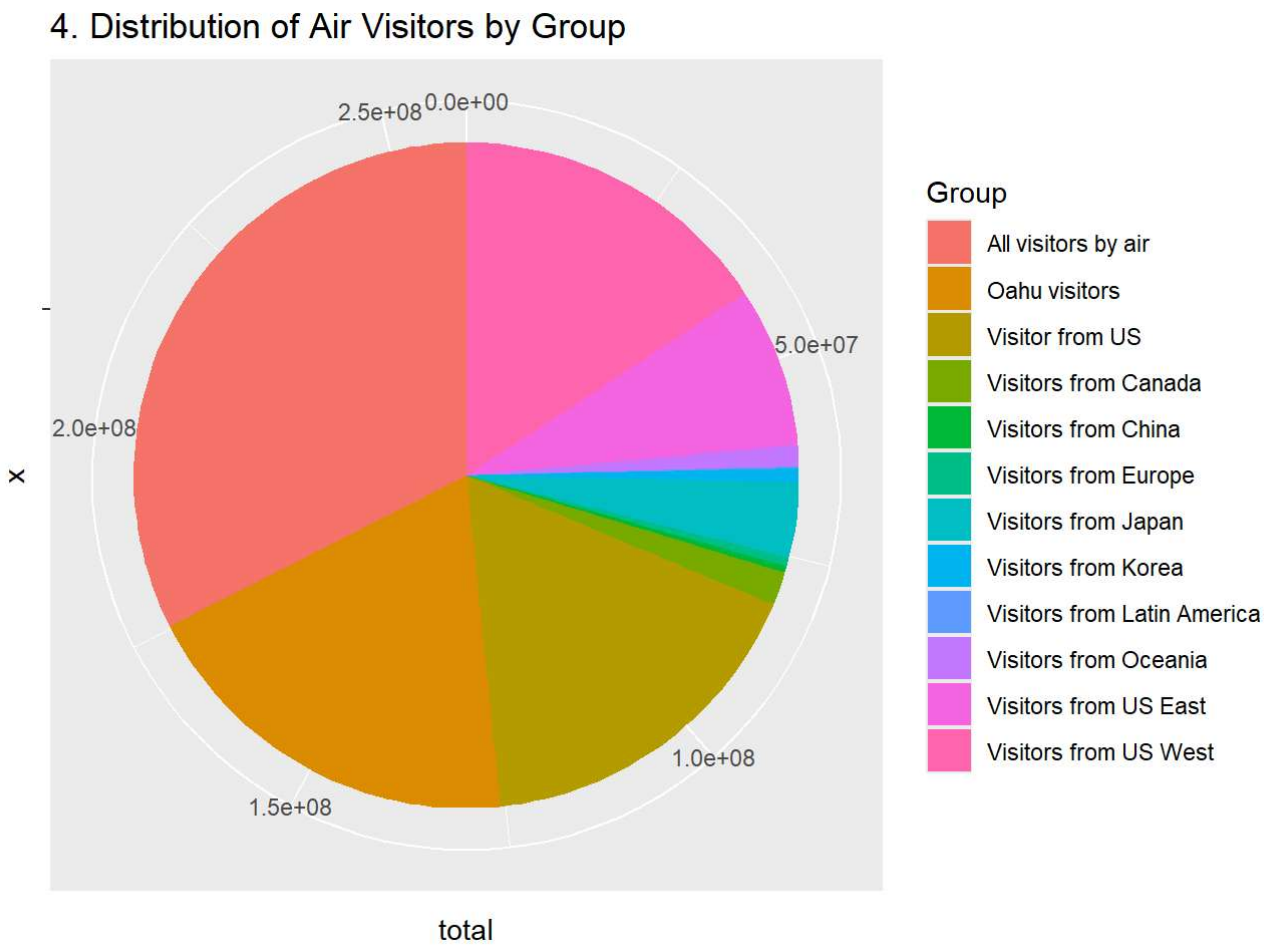

```
# 2. Accommodation Choices by Indicator
if(nrow(accom_data) > 0) {
  accom_sum <- accom_data %>% group_by(Indicator) %>% summarise(total = sum(value, na.rm = TRU
E))
  ggplot(accom_sum, aes(x = "", y = total, fill = Indicator)) +
    geom_bar(stat = "identity") +
    coord_polar("y") +
    labs(title = "2. Distribution of Accommodation Choices by Indicator")

}
```



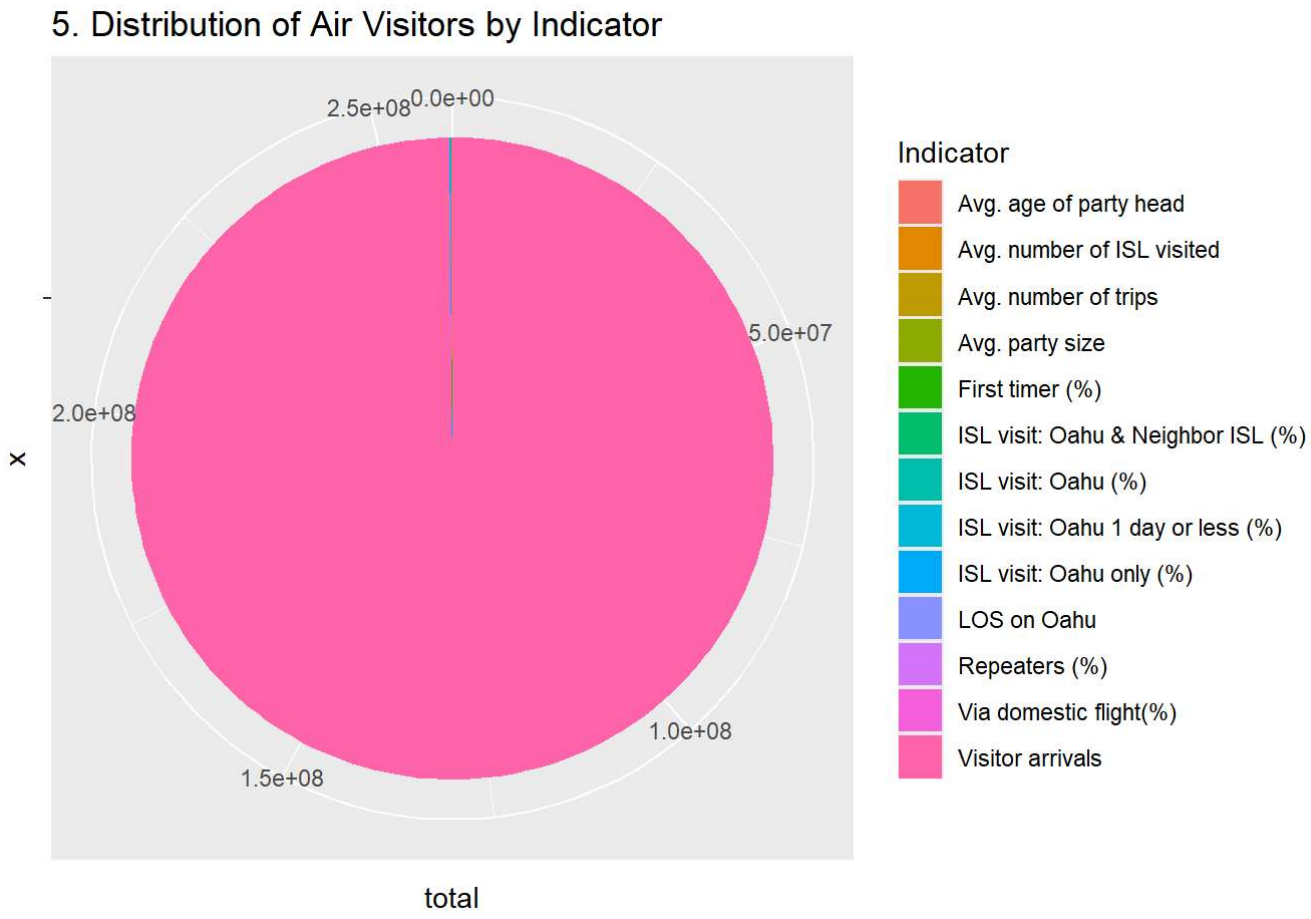

```
# 3. Accommodation Choices by Units
if(nrow(accom_data) > 0) {
  accom_sum <- accom_data %>% group_by(Units) %>% summarise(total = sum(value, na.rm = TRUE))
  ggplot(accom_sum, aes(x = "", y = total, fill = Units)) +
    geom_bar(stat = "identity") +
    coord_polar("y") +
    labs(title = "3. Distribution of Accommodation Choices by Units")

}
```



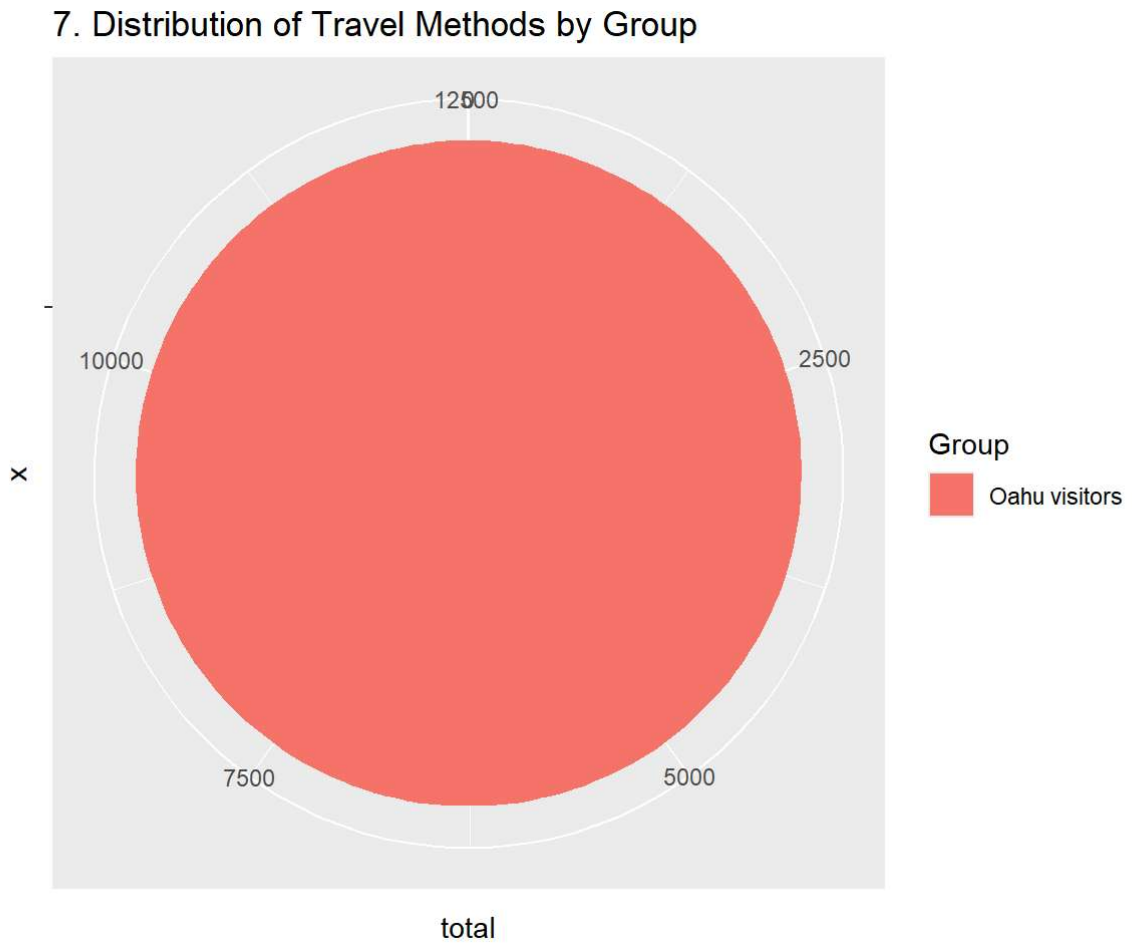

```
# 4. Air Visitors by Group
if(nrow(air_data) > 0) {
  air_sum <- air_data %>% group_by(Group) %>% summarise(total = sum(value, na.rm = TRUE))
  ggplot(air_sum, aes(x = "", y = total, fill = Group)) +
    geom_bar(stat = "identity") +
    coord_polar("y") +
    labs(title = "4. Distribution of Air Visitors by Group")

}
```



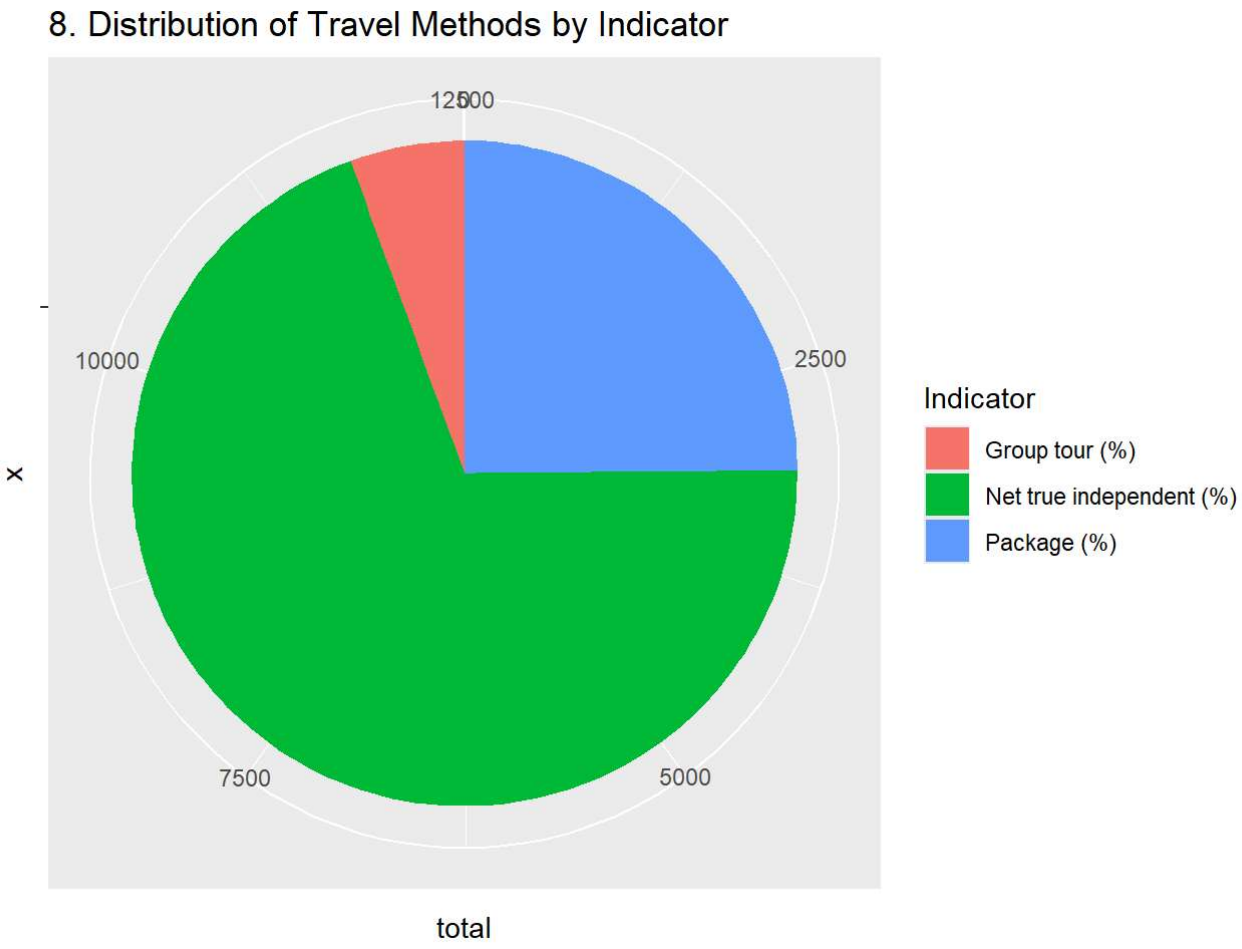

```
# 5. Air Visitors by Indicator
if(nrow(air_data) > 0) {
  air_sum <- air_data %>% group_by(Indicator) %>% summarise(total = sum(value, na.rm = TRUE))
  ggplot(air_sum, aes(x = "", y = total, fill = Indicator)) +
    geom_bar(stat = "identity") +
    coord_polar("y") +
    labs(title = "5. Distribution of Air Visitors by Indicator")

}
```



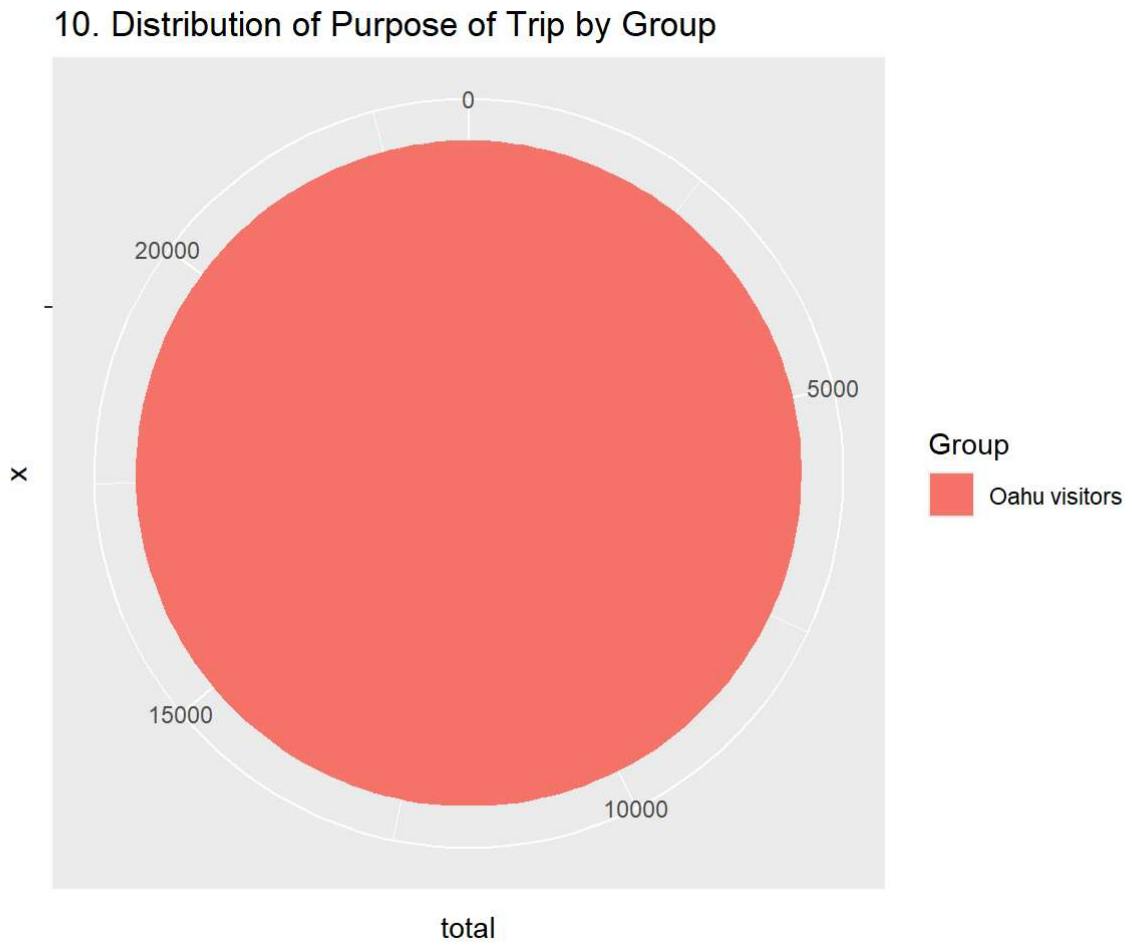

```
# 7. Travel Method by Group
if(nrow(travel_data) > 0) {
  travel_sum <- travel_data %>% group_by(Group) %>% summarise(total = sum(value, na.rm = TRUE))
  ggplot(travel_sum, aes(x = "", y = total, fill = Group)) +
    geom_bar(stat = "identity") +
    coord_polar("y") +
    labs(title = "7. Distribution of Travel Methods by Group")

}
```



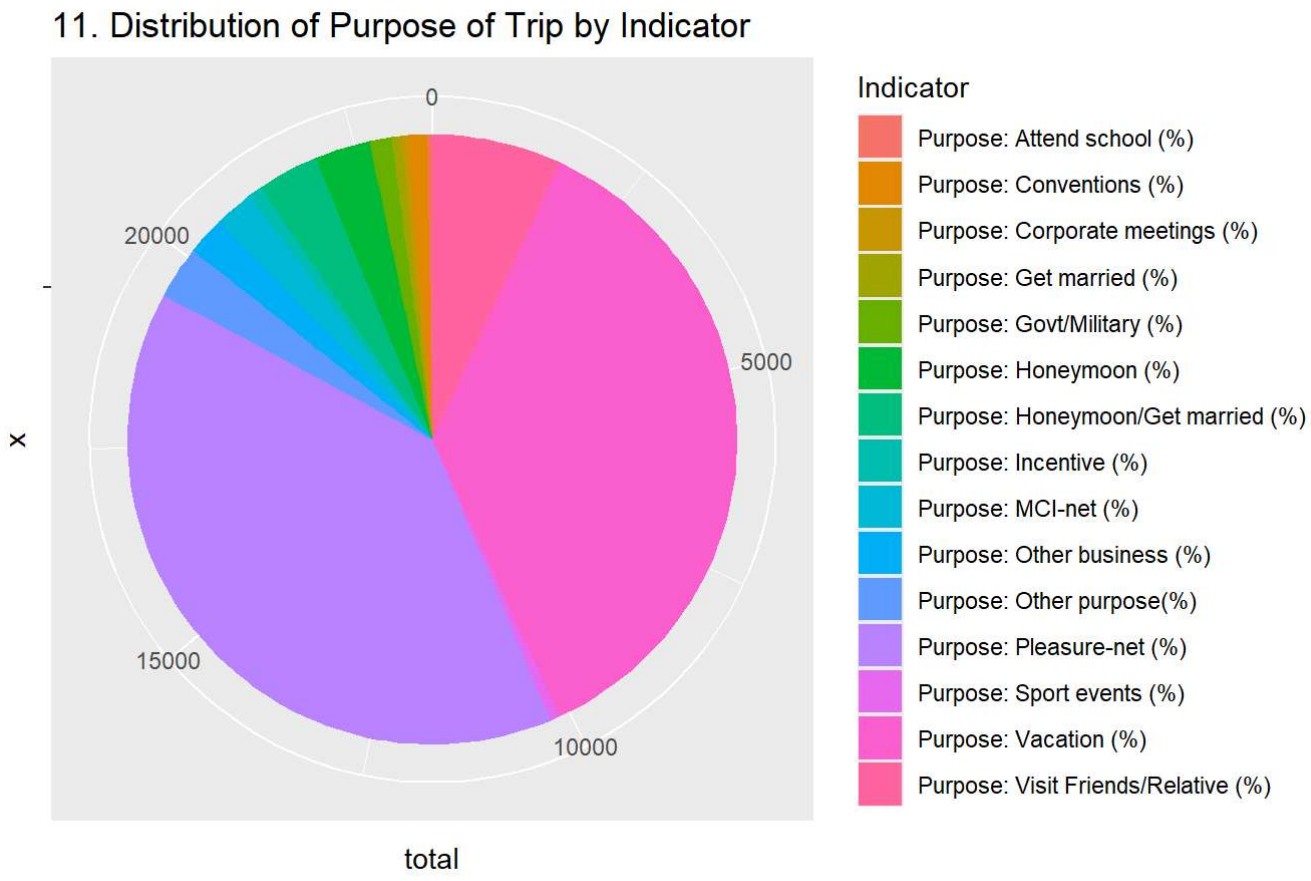

```
# 8. Travel Method by Indicator
if(nrow(travel_data) > 0) {
  travel_sum <- travel_data %>% group_by(Indicator) %>% summarise(total = sum(value, na.rm = TRU
E))
  ggplot(travel_sum, aes(x = "", y = total, fill = Indicator)) +
    geom_bar(stat = "identity") +
    coord_polar("y") +
    labs(title = "8. Distribution of Travel Methods by Indicator")

}
```



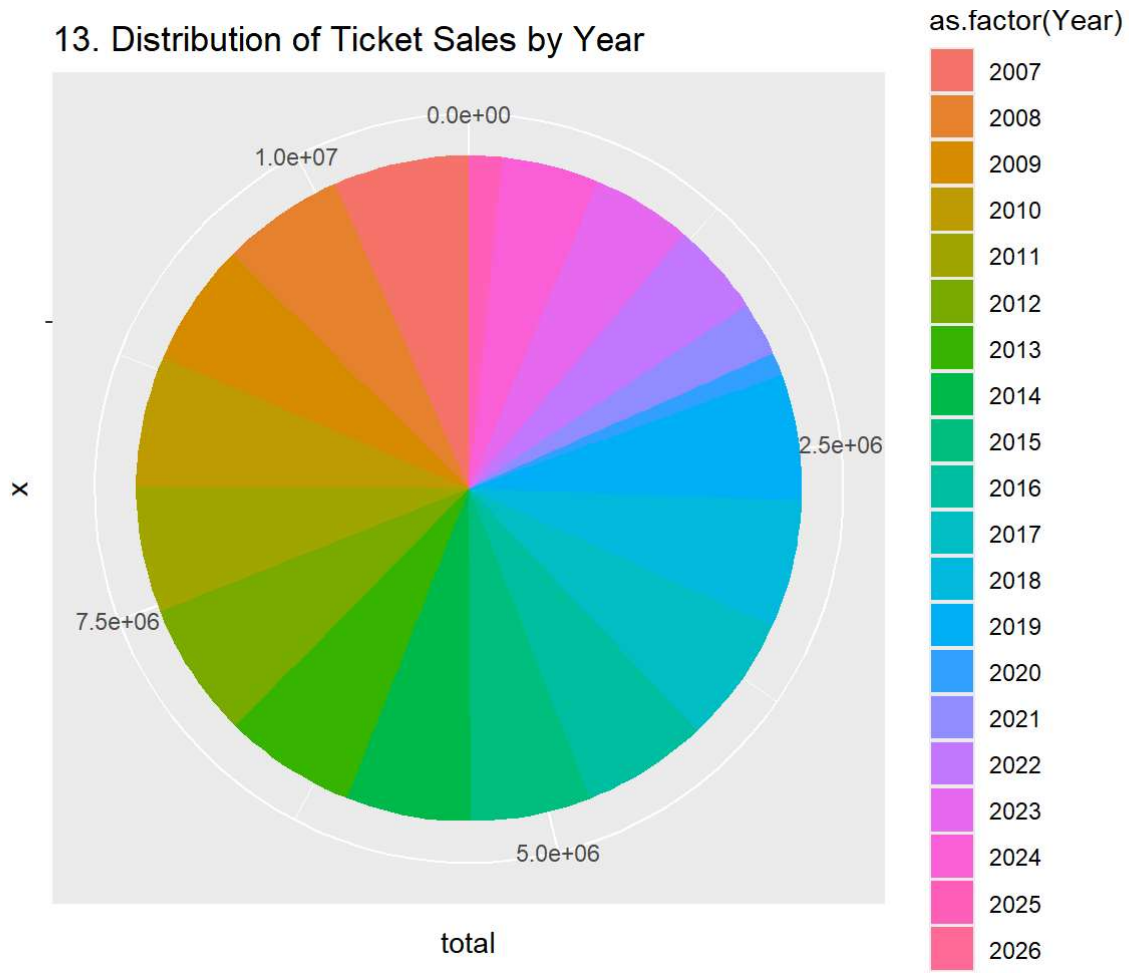

```
# 10. Purpose of Trip by Group
if(nrow(purpose_data) > 0) {
  purpose_sum <- purpose_data %>% group_by(Group) %>% summarise(total = sum(value, na.rm = TRU
E))
  ggplot(purpose_sum, aes(x = "", y = total, fill = Group)) +
    geom_bar(stat = "identity") +
    coord_polar("y") +
    labs(title = "10. Distribution of Purpose of Trip by Group")

}
```



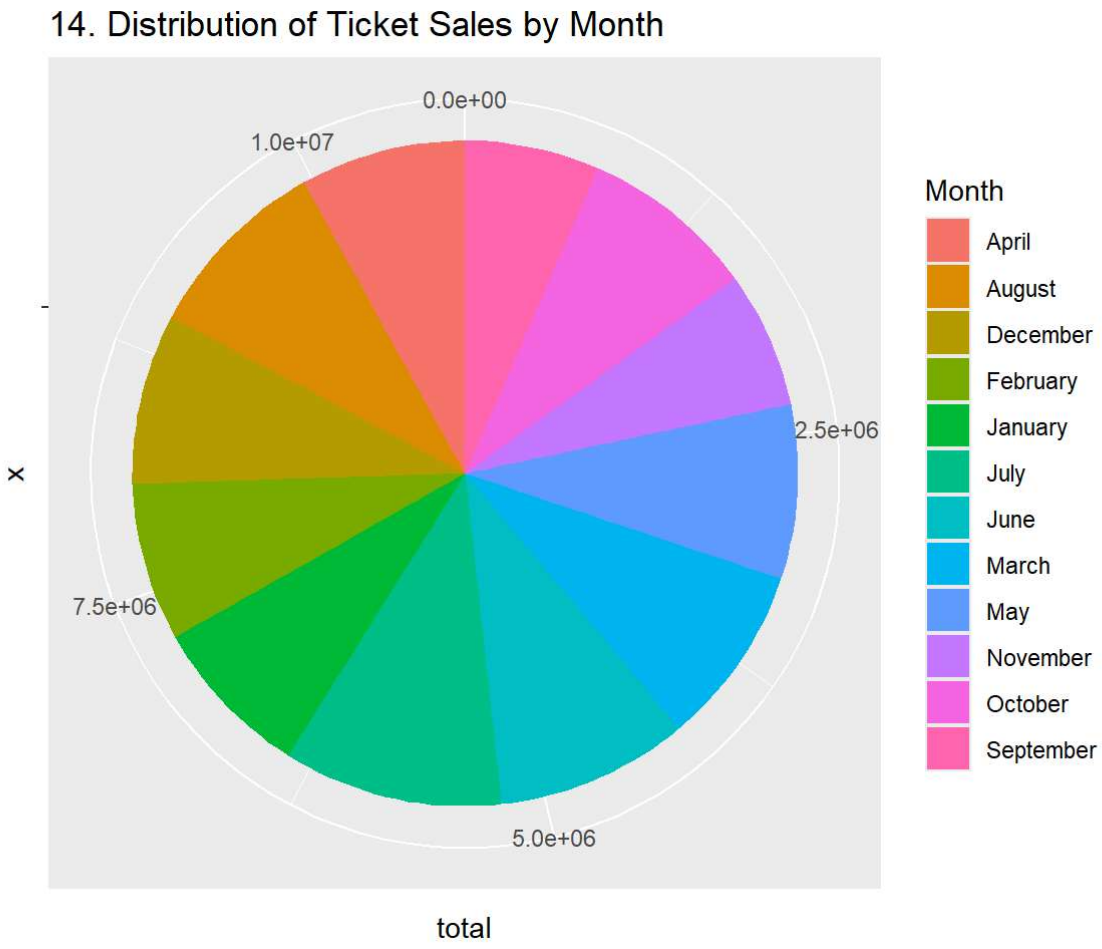

```
# 11. Purpose of Trip by Indicator
if(nrow(purpose_data) > 0) {
  purpose_sum <- purpose_data %>% group_by(Indicator) %>% summarise(total = sum(value, na.rm = T
RUE))
  ggplot(purpose_sum, aes(x = "", y = total, fill = Indicator)) +
    geom_bar(stat = "identity") +
    coord_polar("y") +
    labs(title = "11. Distribution of Purpose of Trip by Indicator")

}
```



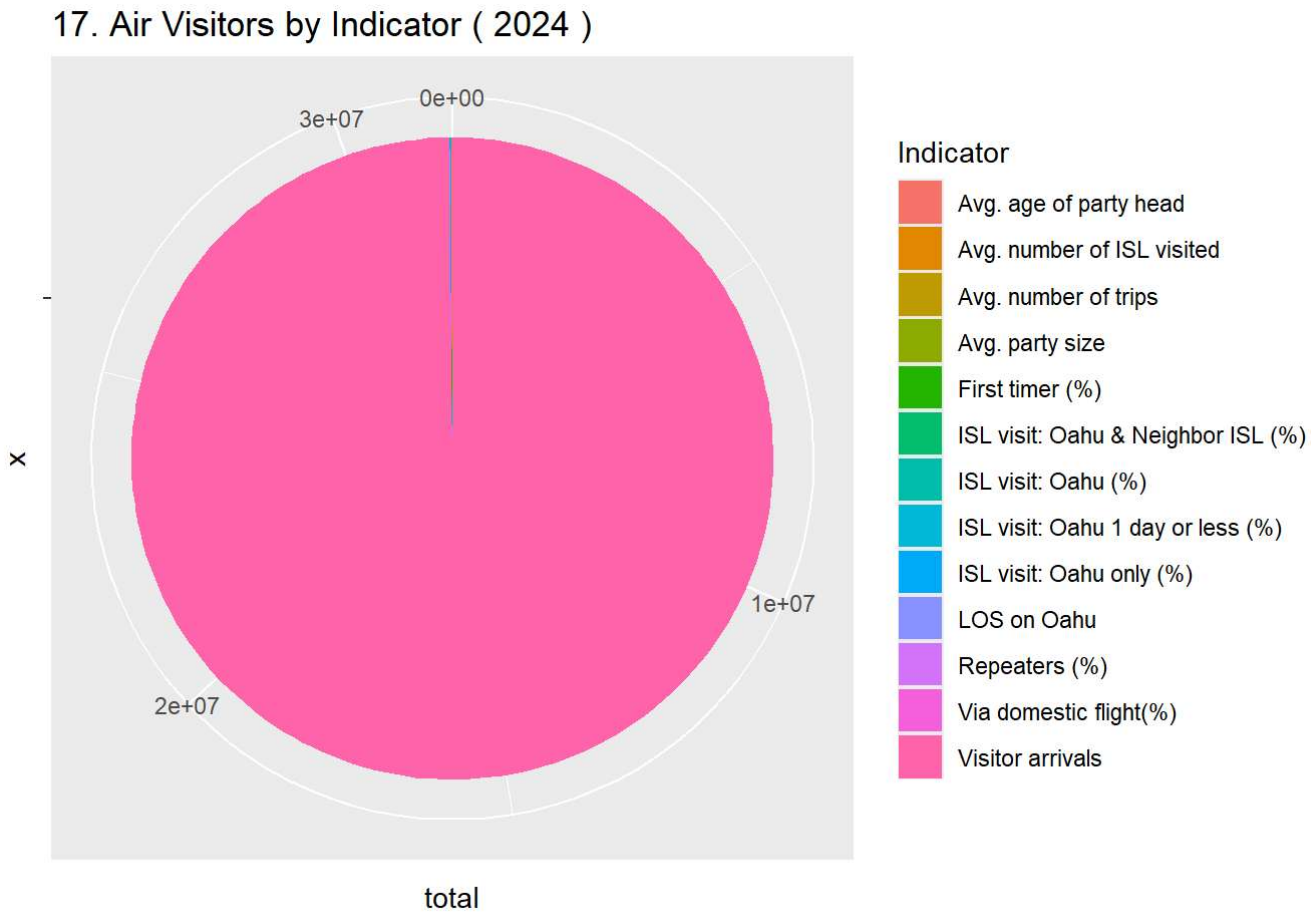

```
# 13. Ticket Sales by Year
if(nrow(ticket_data) > 0) {
  ticket_sum <- ticket_data %>% group_by(Year) %>% summarise(total = sum(TicketSell, na.rm = TRU
E))
  ggplot(ticket_sum, aes(x = "", y = total, fill = as.factor(Year))) +
    geom_bar(stat = "identity") +
    coord_polar("y") +
    labs(title = "13. Distribution of Ticket Sales by Year")

}
```



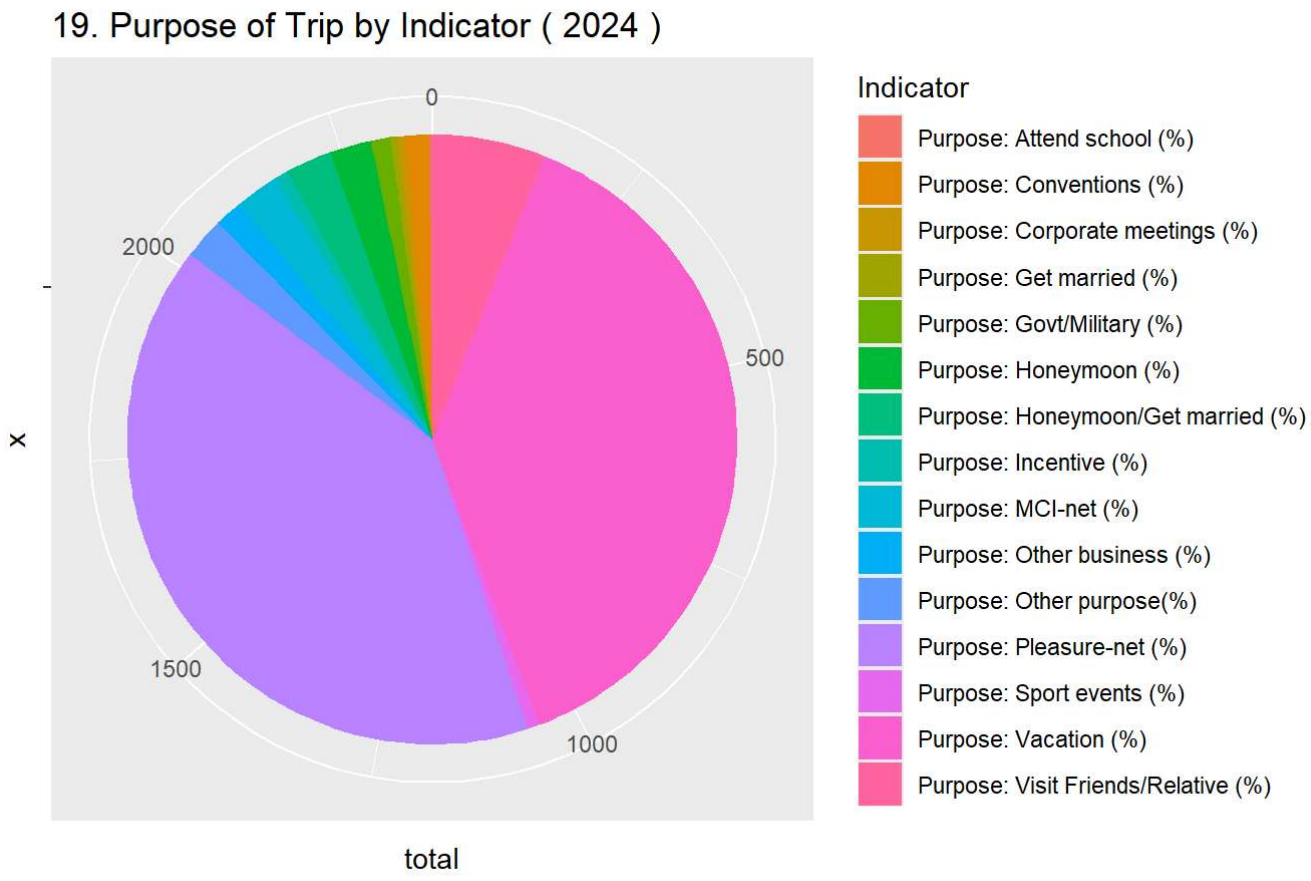

```
# 14. Ticket Sales by Month
if(nrow(ticket_data) > 0) {
  ticket_sum <- ticket_data %>% group_by(Month) %>% summarise(total = sum(TicketSell, na.rm = TR
UE))
  ggplot(ticket_sum, aes(x = "", y = total, fill = Month)) +
    geom_bar(stat = "identity") +
    coord_polar("y") +
    labs(title = "14. Distribution of Ticket Sales by Month")

}
```



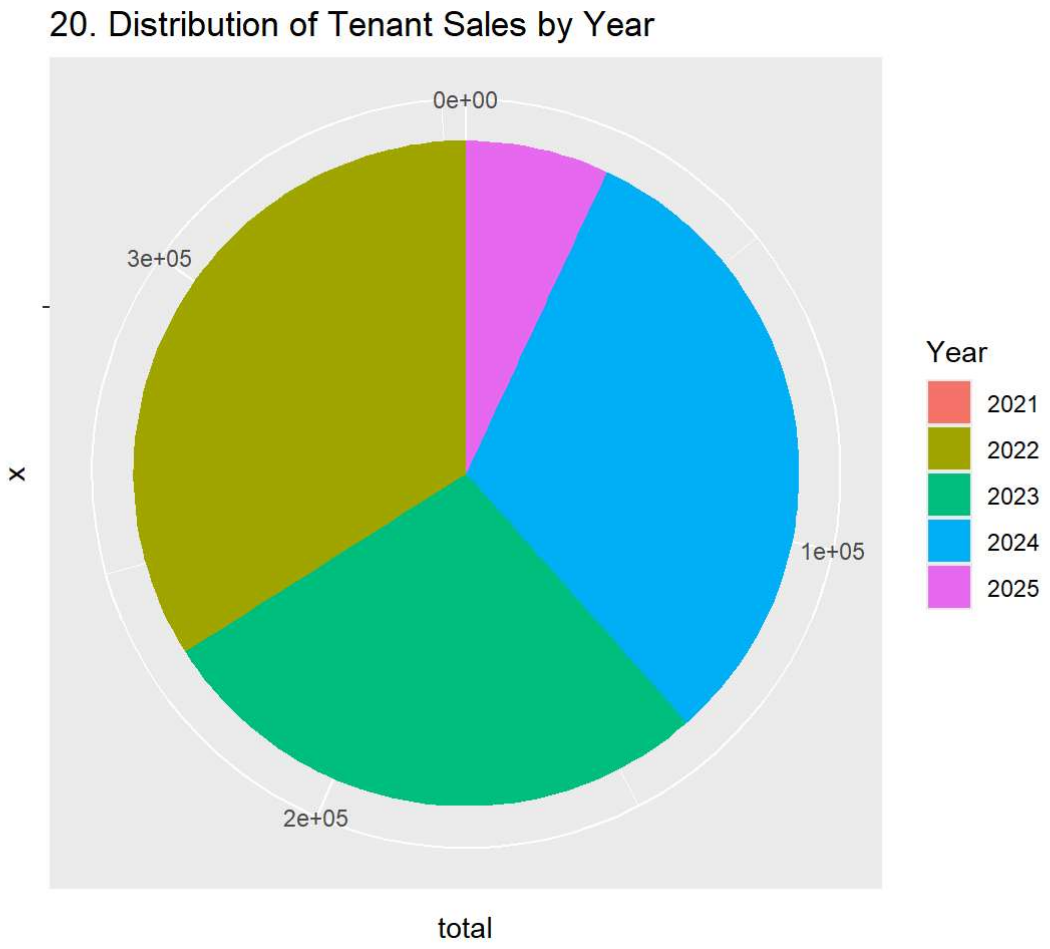

```
# 17. Air Visitors by Indicator (Latest Year)
if(nrow(air_data) > 0) {
  latest_year <- max(substr(air_data$`YYYY-MM`, 1, 4))
  air_sum <- air_data %>%
    filter(substr(`YYYY-MM`, 1, 4) == latest_year) %>%
    group_by(Indicator) %>%
    summarise(total = sum(value, na.rm = TRUE))
  ggplot(air_sum, aes(x = "", y = total, fill = Indicator)) +
    geom_bar(stat = "identity") +
    coord_polar("y") +
    labs(title = paste("17. Air Visitors by Indicator (", latest_year, ")"))

}
```



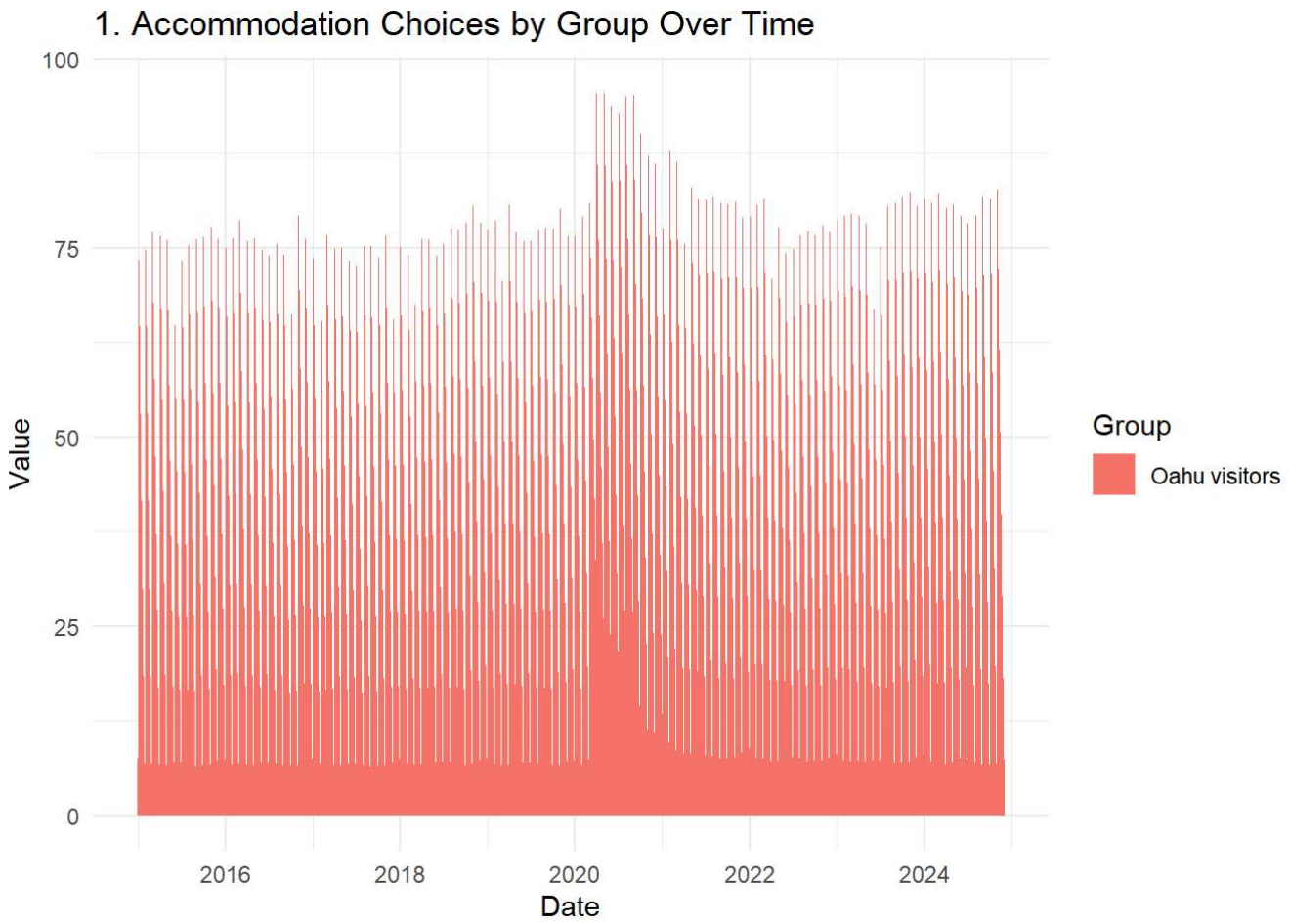

```
# 19. Purpose of Trip by Indicator (Latest Year)
if(nrow(purpose_data) > 0) {
  latest_year <- max(substr(purpose_data$`YYYY-MM`, 1, 4))
  purpose_sum <- purpose_data %>%
    filter(substr(`YYYY-MM`, 1, 4) == latest_year) %>%
    group_by(Indicator) %>%
    summarise(total = sum(value, na.rm = TRUE))
  ggplot(purpose_sum, aes(x = "", y = total, fill = Indicator)) +
    geom_bar(stat = "identity") +
    coord_polar("y") +
    labs(title = paste("19. Purpose of Trip by Indicator (", latest_year, ")"))

}
```



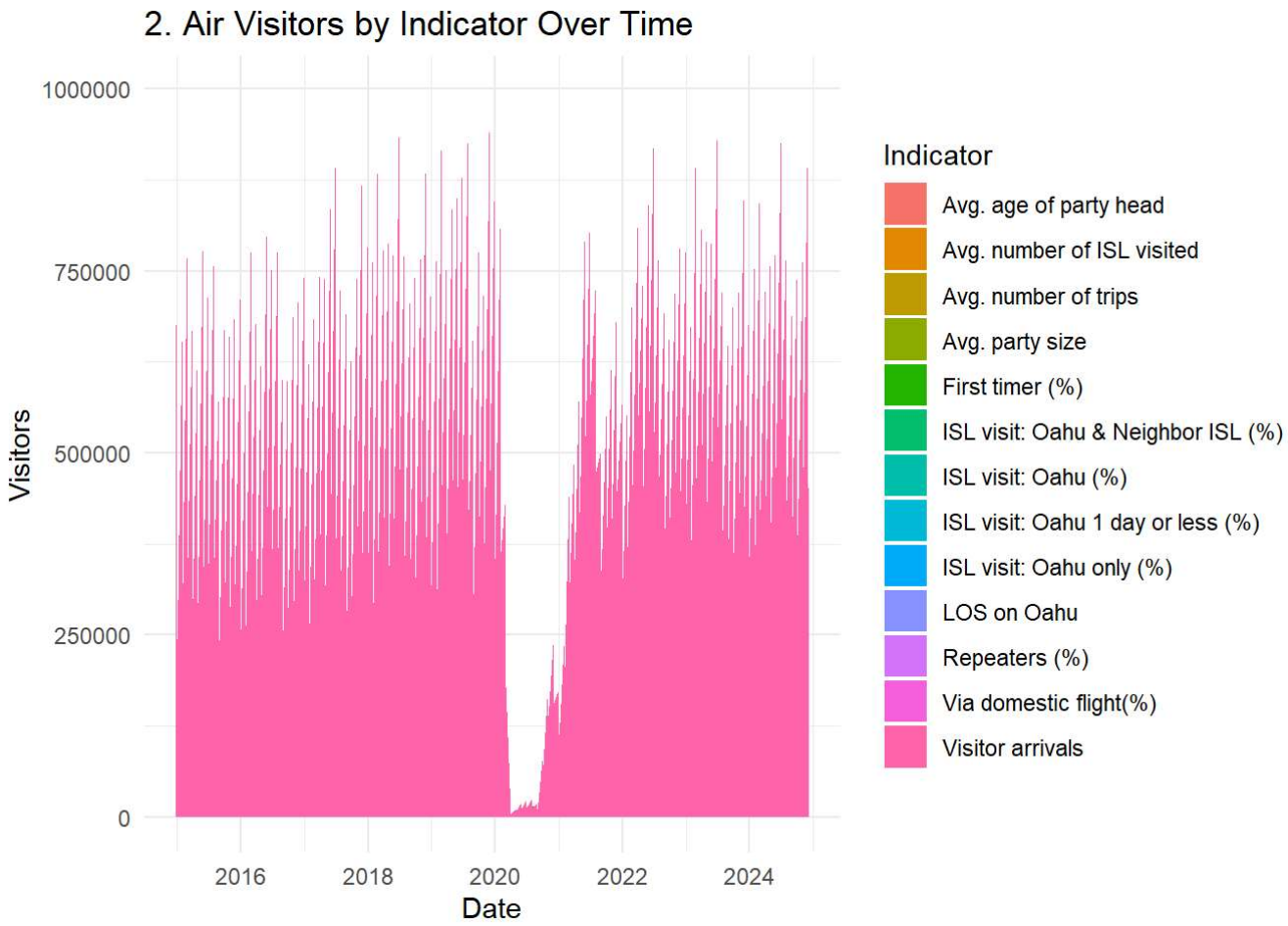

```
# 20. Tenant Sales by Year
if(nrow(tenant_data) > 0) {
  tenant_sum <- tenant_data %>%
    mutate(Year = substr(Date, 1, 4)) %>%
    group_by(Year) %>%
    summarise(total = sum(`Net Sales`, na.rm = TRUE))
  ggplot(tenant_sum, aes(x = "", y = total, fill = Year)) +
    geom_bar(stat = "identity") +
    coord_polar("y") +
    labs(title = "20. Distribution of Tenant Sales by Year")

}
```



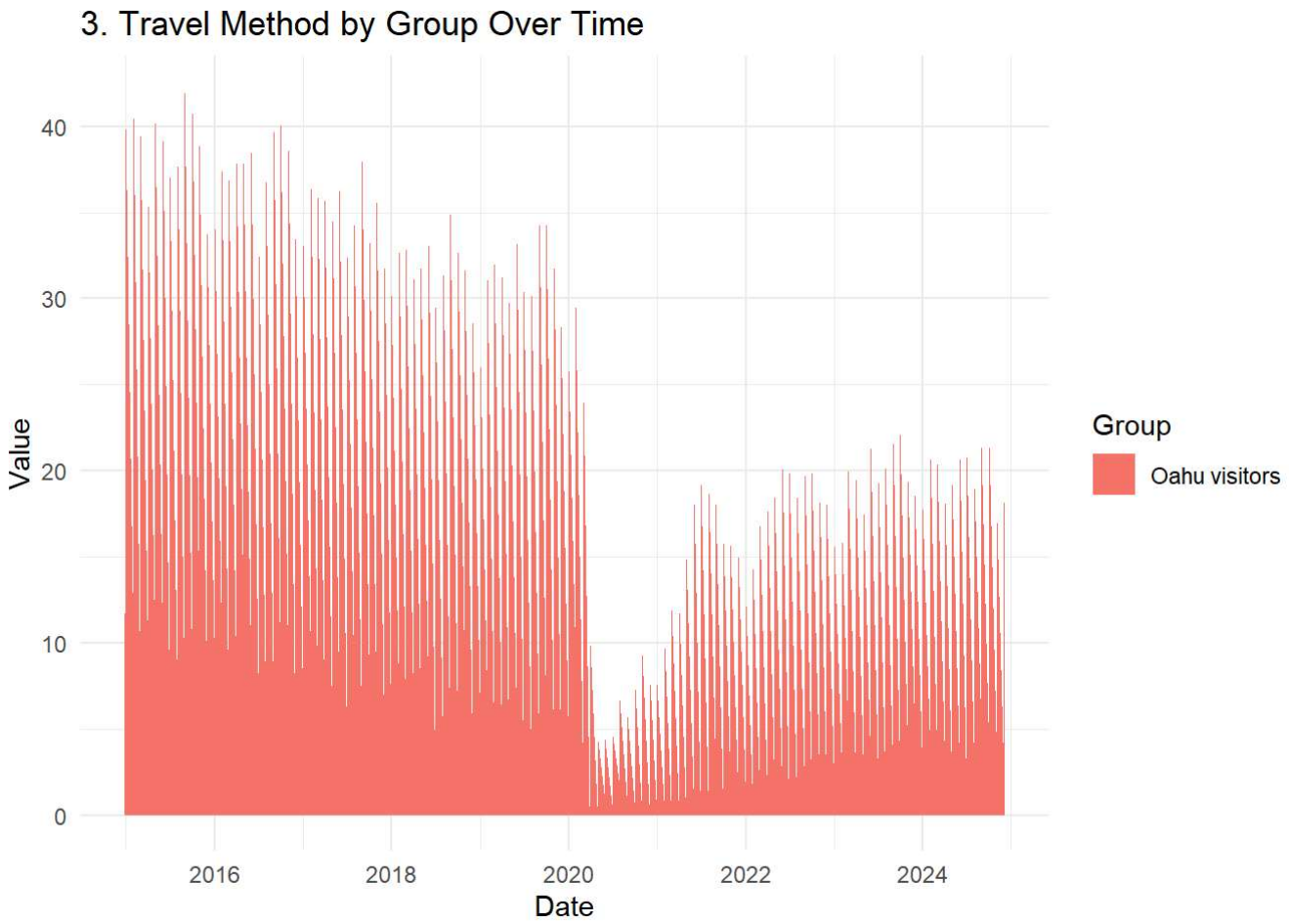

```
# 1. Accommodation Choices by Group Over Time
# Business Value: Understand which accommodation types drive demand to optimize inventory or par
tnerships.
if(nrow(accom_data) > 0) {
  ggplot(accom_data, aes(x = as.Date(paste0(`YYYY-MM`, "-01")), y = value, fill = Group)) +
    geom_area() +
    labs(title = "1. Accommodation Choices by Group Over Time",
         x = "Date", y = "Value") +
    theme_minimal()

}
```



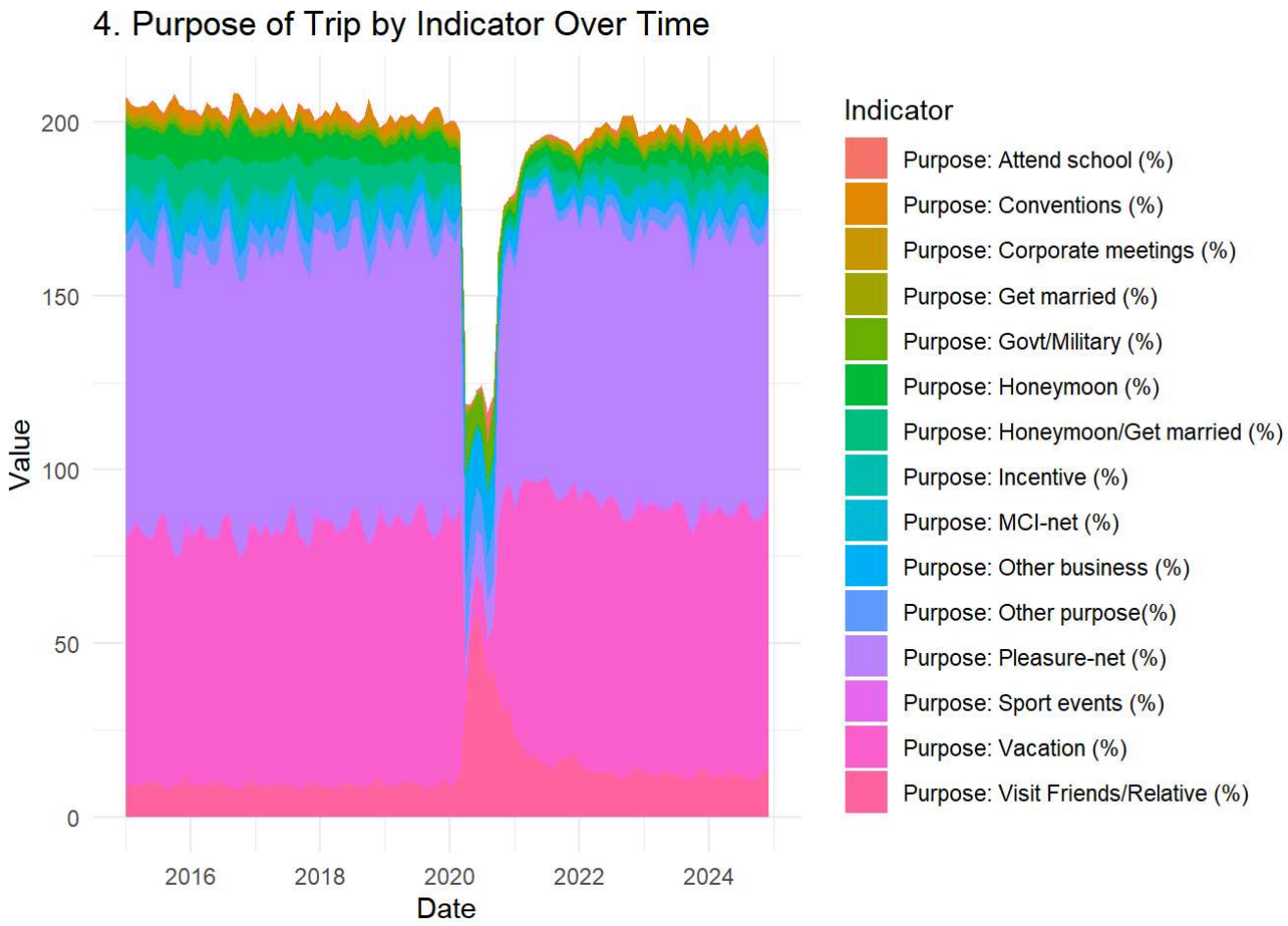

```
# 2. Air Visitors by Indicator Over Time
# Business Value: Track visitor trends by purpose or source to tailor marketing campaigns.
if(nrow(air_data) > 0) {
  ggplot(air_data, aes(x = as.Date(paste0(`YYYY-MM`, "-01")), y = value, fill = Indicator)) +
    geom_area() +
    labs(title = "2. Air Visitors by Indicator Over Time",
         x = "Date", y = "Visitors") +
    theme_minimal()

}
```




```
# 3. Travel Method by Group Over Time
# Business Value: Identify preferred travel methods to optimize transportation partnerships or s
ervices.
if(nrow(travel_data) > 0) {
  ggplot(travel_data, aes(x = as.Date(paste0(`YYYY-MM`, "-01")), y = value, fill = Group)) +
    geom_area() +
    labs(title = "3. Travel Method by Group Over Time",
         x = "Date", y = "Value") +
    theme_minimal()

}
```

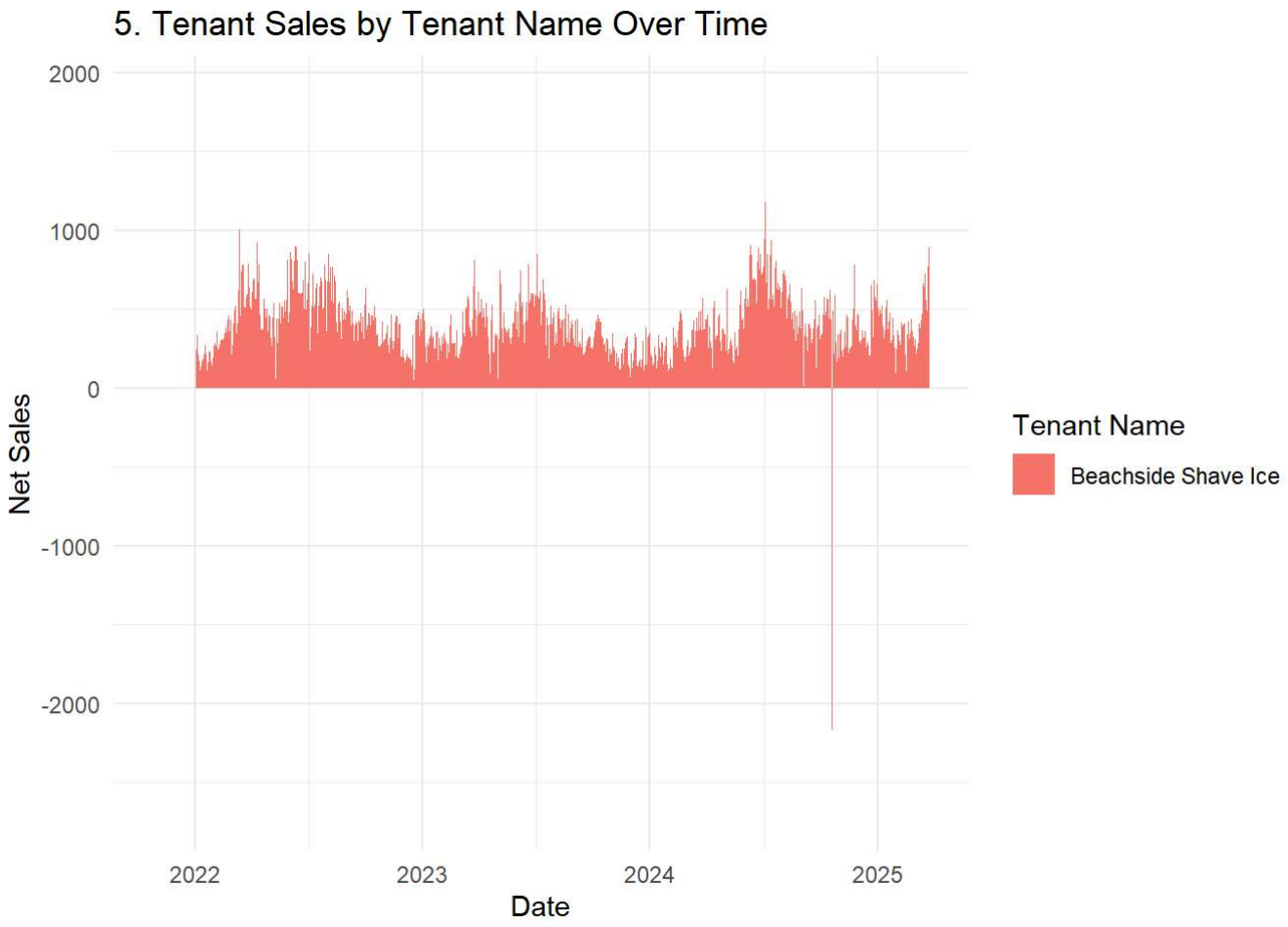



```
# 4. Purpose of Trip by Indicator Over Time
# Business Value: Understand traveler motivations (e.g., leisure, business) to target promotions
effectively.
if(nrow(purpose_data) > 0) {
  ggplot(purpose_data, aes(x = as.Date(paste0(`YYYY-MM`, "-01")), y = value, fill = Indicator))
+
    geom_area() +
    labs(title = "4. Purpose of Trip by Indicator Over Time",
         x = "Date", y = "Value") +
    theme_minimal()

}
```

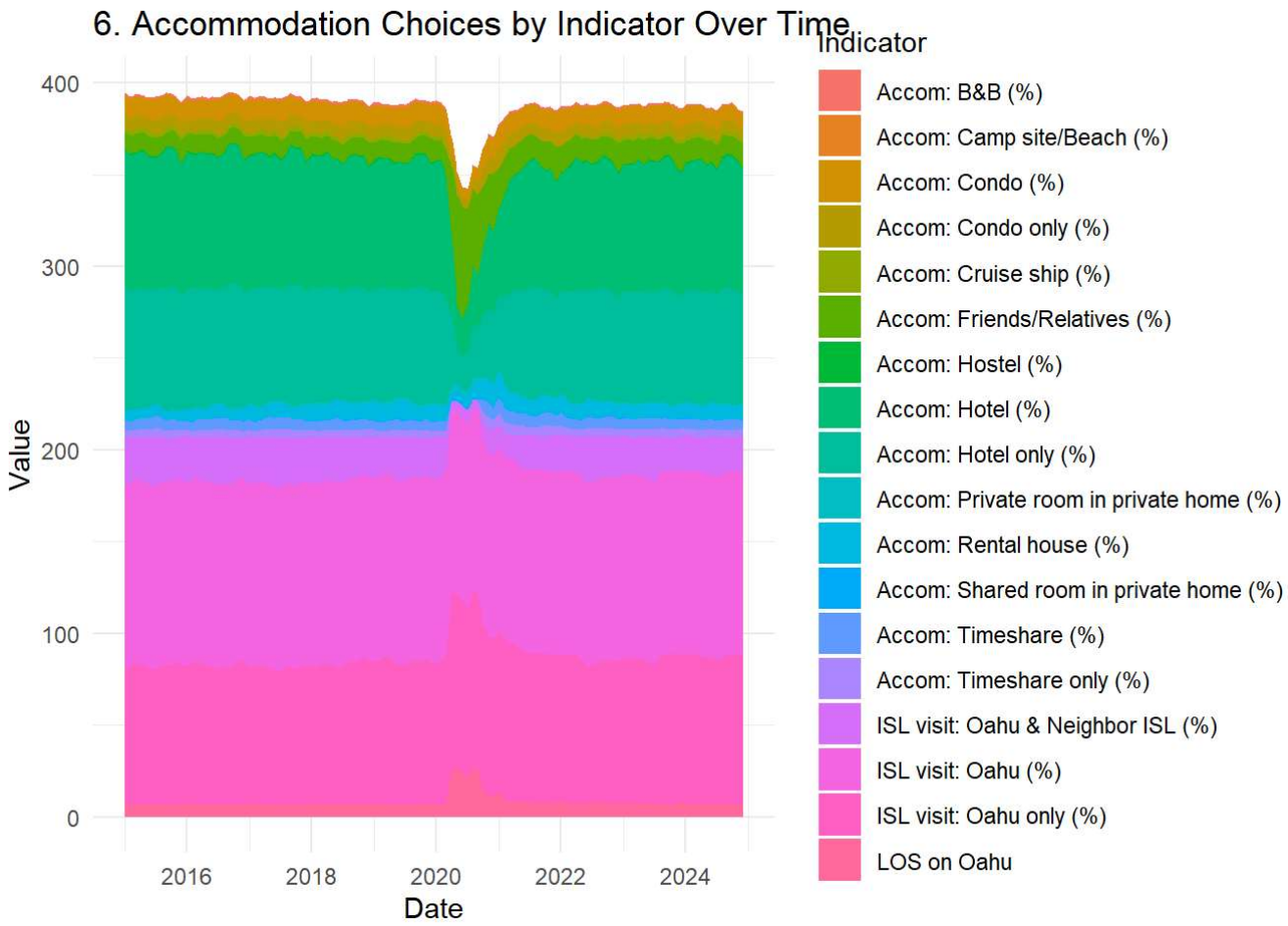



```
# 5. Tenant Sales by Tenant Name Over Time
# Business Value: Monitor tenant performance to inform lease renewals or retail strategy adjustm
ents.
if(nrow(tenant_data) > 0) {
  ggplot(tenant_data, aes(x = as.Date(Date), y = `Net Sales`, fill = `Tenant Name`)) +
    geom_area() +
    labs(title = "5. Tenant Sales by Tenant Name Over Time",
         x = "Date", y = "Net Sales") +
    theme_minimal()

}
```

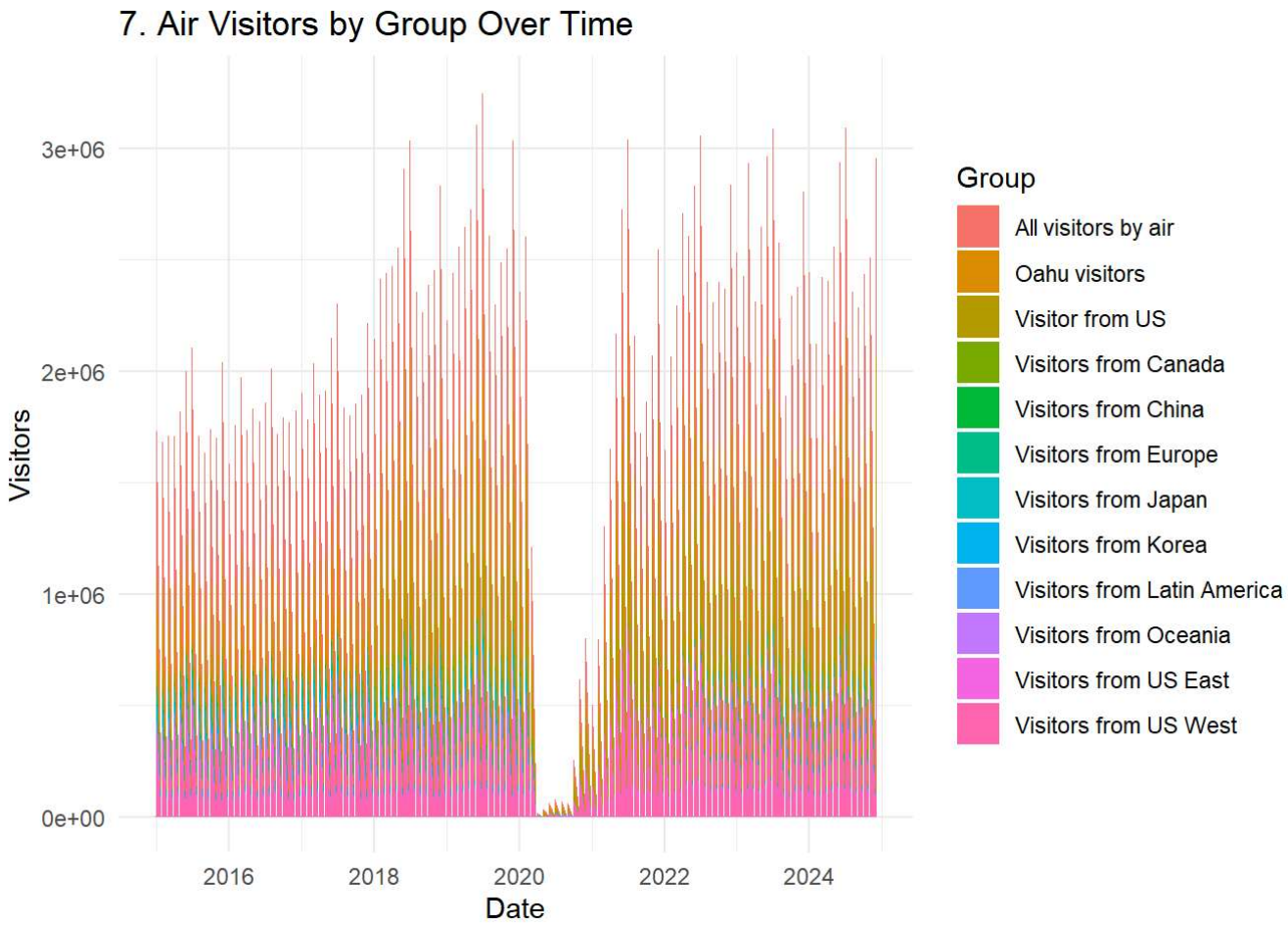
```
## Warning: Removed 374 rows containing non-finite outside the scale range
## (`stat_align()`).
```



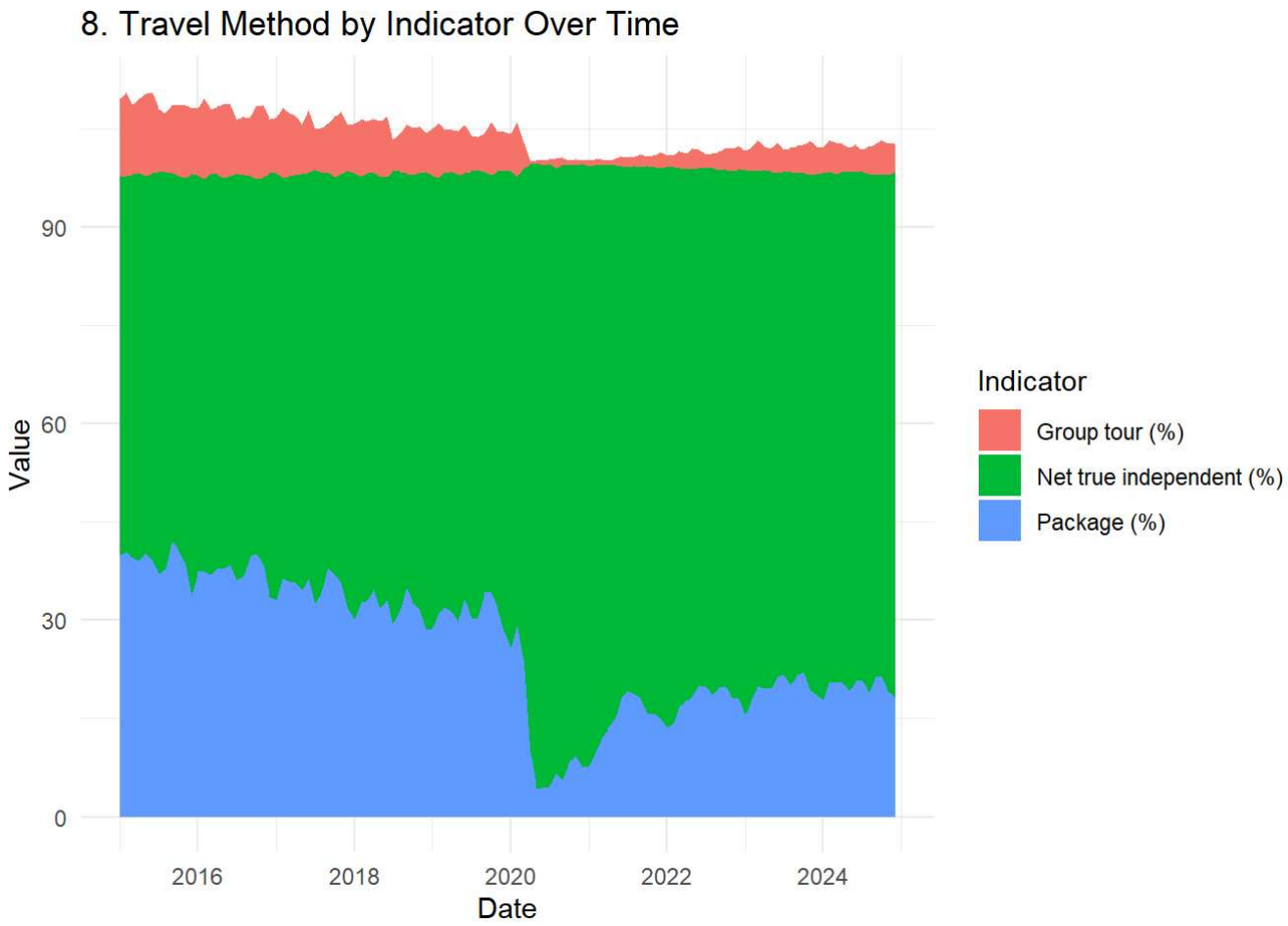

```
# 6. Accommodation Choices by Indicator Over Time
# Business Value: Analyze specific accommodation metrics (e.g., bookings, revenue) to identify g
rowth areas.
if(nrow(accom_data) > 0) {
  ggplot(accom_data, aes(x = as.Date(paste0(`YYYY-MM`, "-01")), y = value, fill = Indicator)) +
    geom_area() +
    labs(title = "6. Accommodation Choices by Indicator Over Time",
         x = "Date", y = "Value") +
    theme_minimal()

}
```



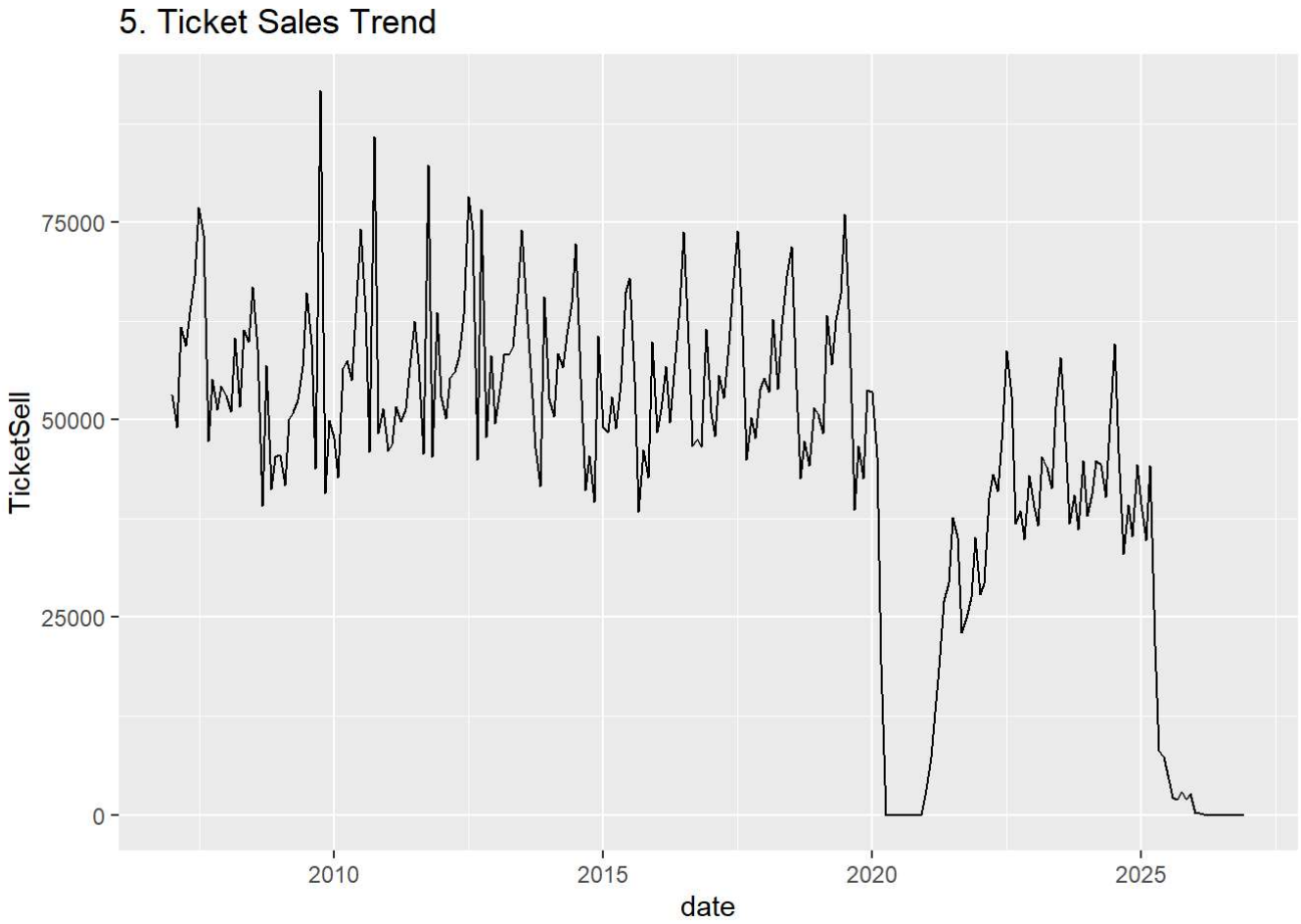

```
# 7. Air Visitors by Group Over Time
# Business Value: Compare visitor demographics or regions to allocate advertising budgets effici
ently.
if(nrow(air_data) > 0) {
  ggplot(air_data, aes(x = as.Date(paste0(`YYYY-MM`, "-01")), y = value, fill = Group)) +
    geom_area() +
    labs(title = "7. Air Visitors by Group Over Time",
         x = "Date", y = "Visitors") +
    theme_minimal()

}
```




```
# 8. Travel Method by Indicator Over Time
# Business Value: Assess trends in travel methods (e.g., air, cruise) to negotiate better suppli
er contracts.
if(nrow(travel_data) > 0) {
  ggplot(travel_data, aes(x = as.Date(paste0(`YYYY-MM`, "-01")), y = value, fill = Indicator)) +
    geom_area() +
    labs(title = "8. Travel Method by Indicator Over Time",
         x = "Date", y = "Value") +
    theme_minimal()

}
```

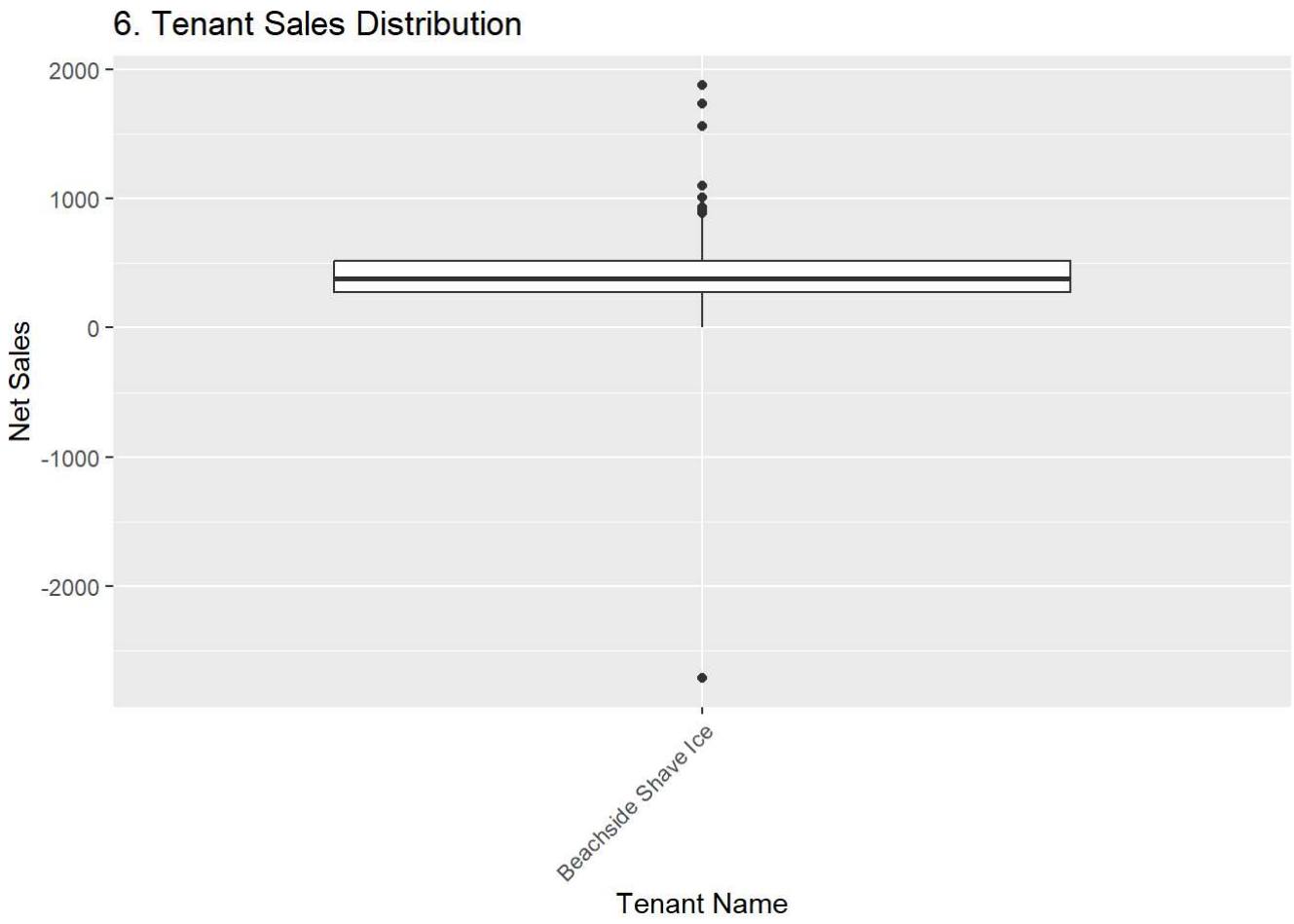



```
# 5. Ticket Sales Trend
if(nrow(ticket_data) > 0) {
  ticket_data$date <- as.Date(paste(ticket_data$Year, ticket_data$Month, "01", sep = "-"), forma
t = "%Y-%B-%d")
  ggplot(ticket_data, aes(x = date, y = TicketSell)) +
    geom_line() +
    labs(title = "5. Ticket Sales Trend")
}
```



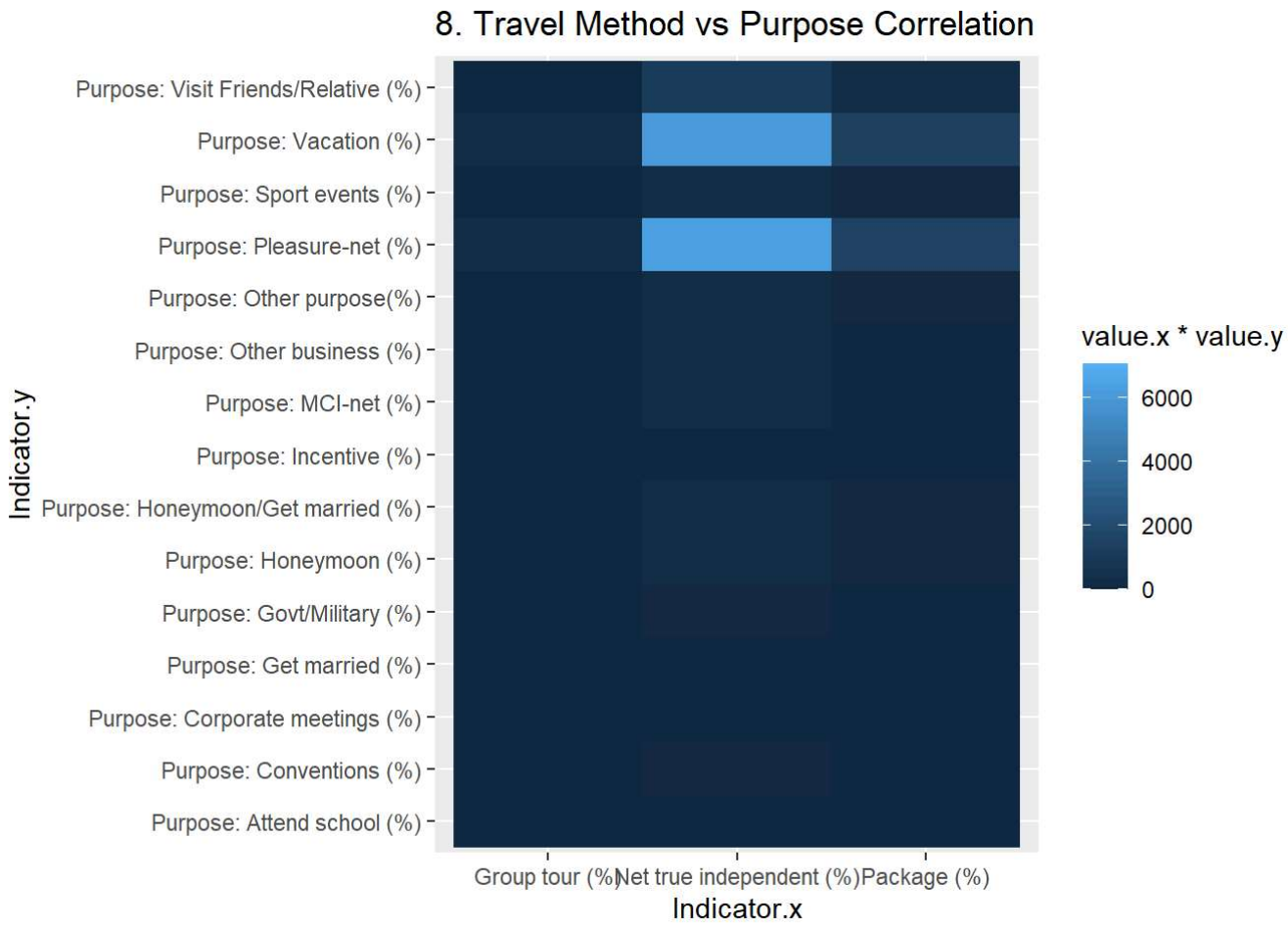

```
# 6. Tenant Sales Box Plot
if(nrow(tenant_data) > 0) {
  ggplot(tenant_data, aes(x = `Tenant Name`, y = `Net Sales`)) +
    geom_boxplot() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
    labs(title = "6. Tenant Sales Distribution")
}
```

```
## Warning: Removed 374 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



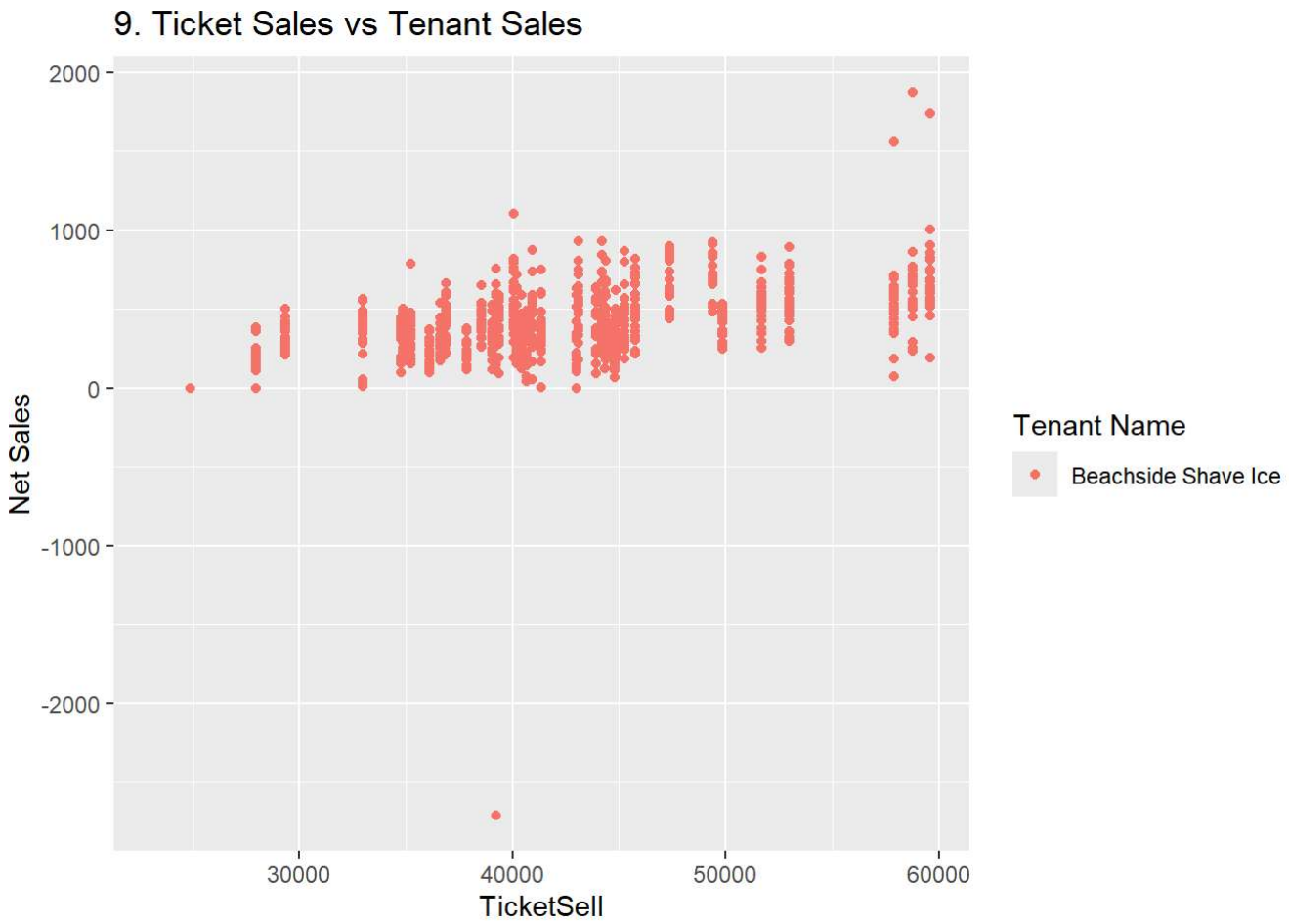

```
# 7. Accommodation vs Air Visitors Scatter
if(nrow(accom_data) > 0 && nrow(air_data) > 0) {
  combined <- merge(accom_data, air_data, by = "YYYY-MM", suffixes = c("_accom", "_air"))
  ggplot(combined, aes(x = value_accom, y = value_air, color = Group_accom)) +
    geom_point() +
    labs(title = "7. Accommodation vs Air Visitors")
}
```


```
# 8. Travel Method vs Purpose Heatmap
if(nrow(travel_data) > 0 && nrow(purpose_data) > 0) {
  combined_tp <- merge(travel_data, purpose_data, by = "YYYY-MM")
  ggplot(combined_tp, aes(x = Indicator.x, y = Indicator.y, fill = value.x * value.y)) +
    geom_tile() +
    labs(title = "8. Travel Method vs Purpose Correlation")
}
```

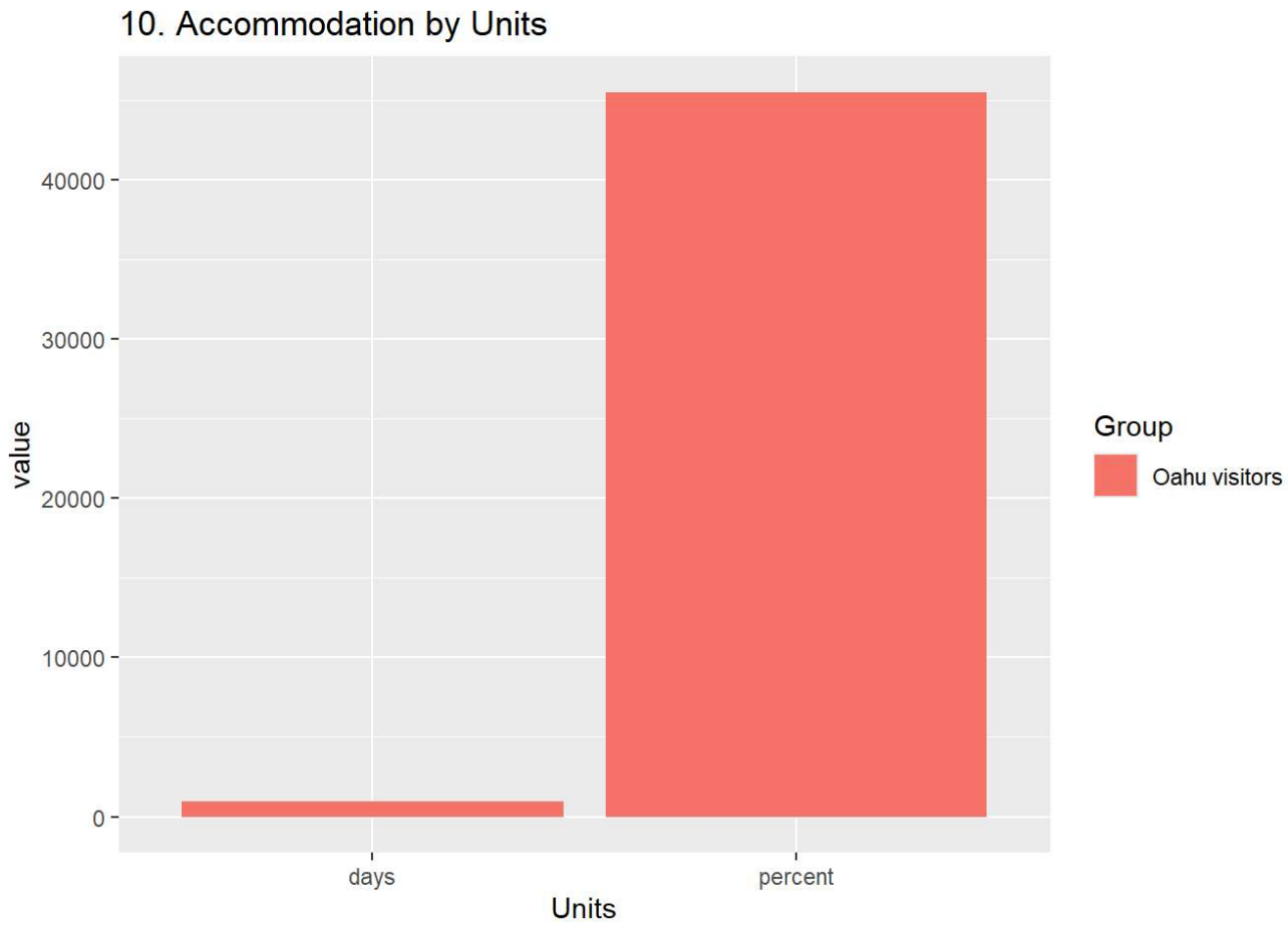



```
# 9. Ticket Sales vs Tenant Sales Scatter
if(nrow(tenant_data) > 0 && nrow(ticket_data) > 0) {
  tenant_data$date <- as.Date(tenant_data$Date)
  ticket_data$month_year <- format(ticket_data$date, "%Y-%m")
  tenant_data$month_year <- format(tenant_data$date, "%Y-%m")
  combined_ts <- merge(tenant_data, ticket_data, by = "month_year")
  ggplot(combined_ts, aes(x = TicketSell, y = `Net Sales`, color = `Tenant Name`)) +
    geom_point() +
    labs(title = "9. Ticket Sales vs Tenant Sales")
}
```

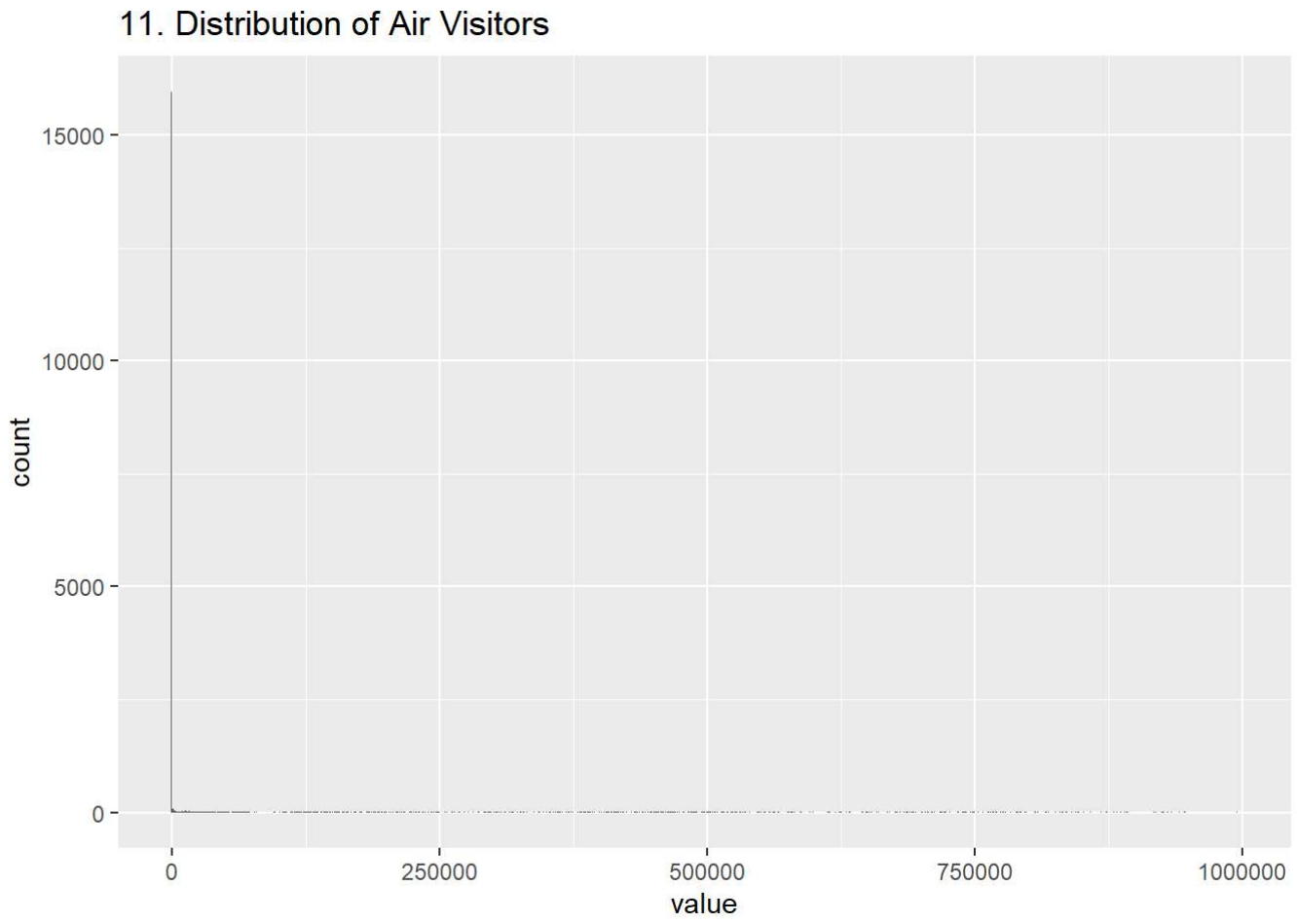
```
## Warning: Removed 374 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



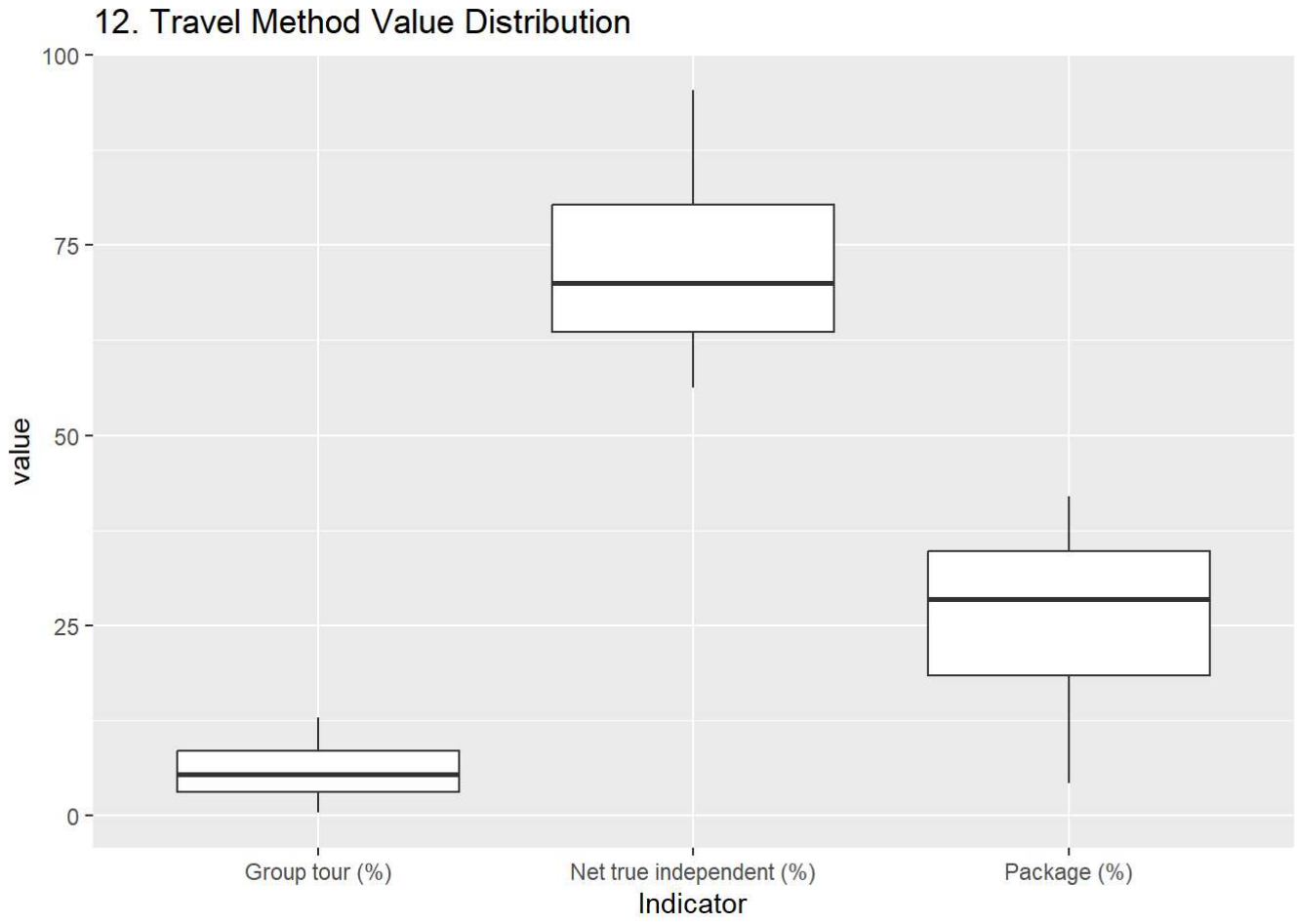

```
# 10. Accommodation Units Bar Plot
if(nrow(accom_data) > 0) {
  ggplot(accom_data, aes(x = Units, y = value, fill = Group)) +
    geom_bar(stat = "identity") +
    labs(title = "10. Accommodation by Units")
}
```



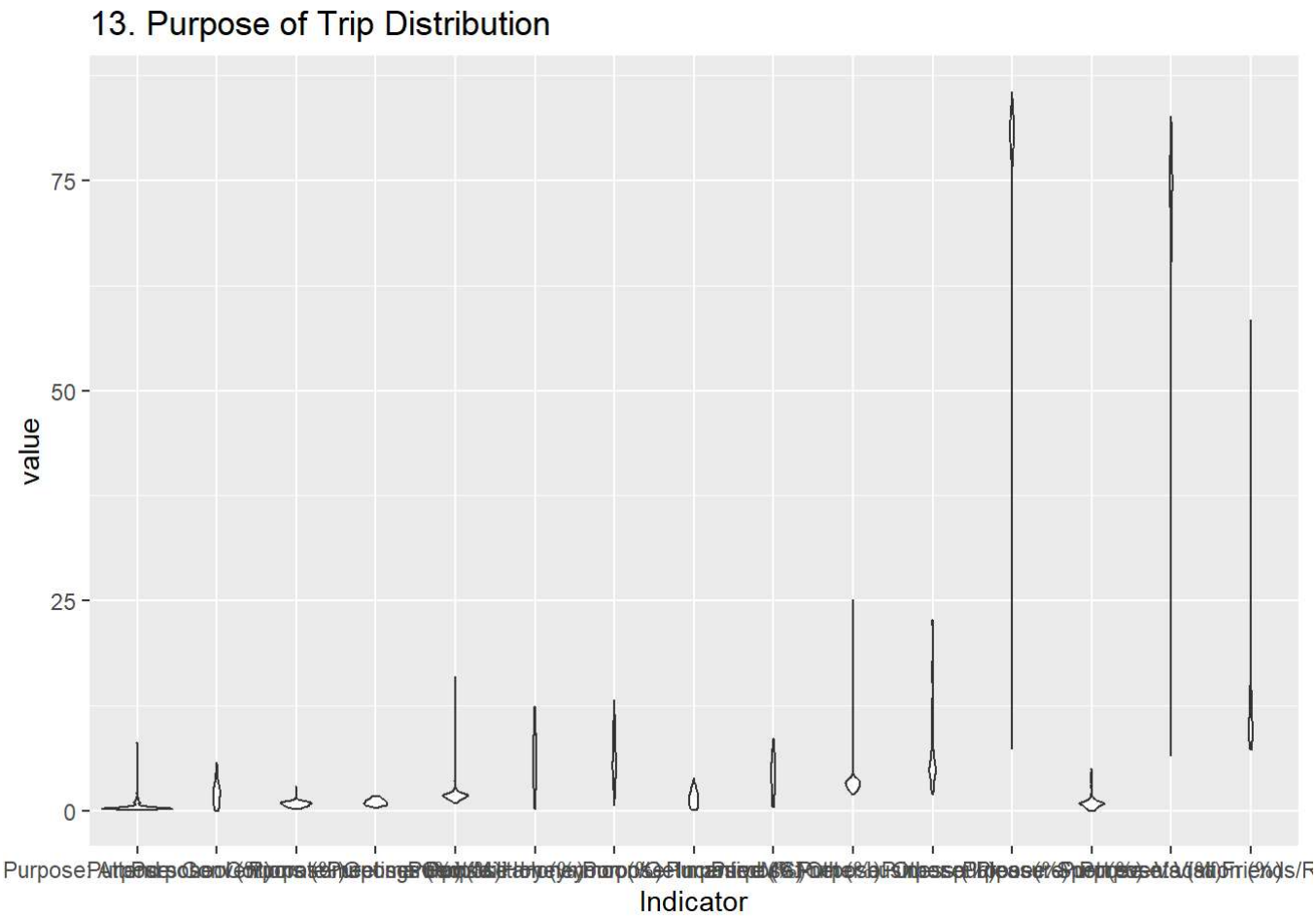

```
# 11. Air Visitors Histogram
if(nrow(air_data) > 0) {
  ggplot(air_data, aes(x = value)) +
    geom_histogram(binwidth = 1000) +
    labs(title = "11. Distribution of Air Visitors")
}
```



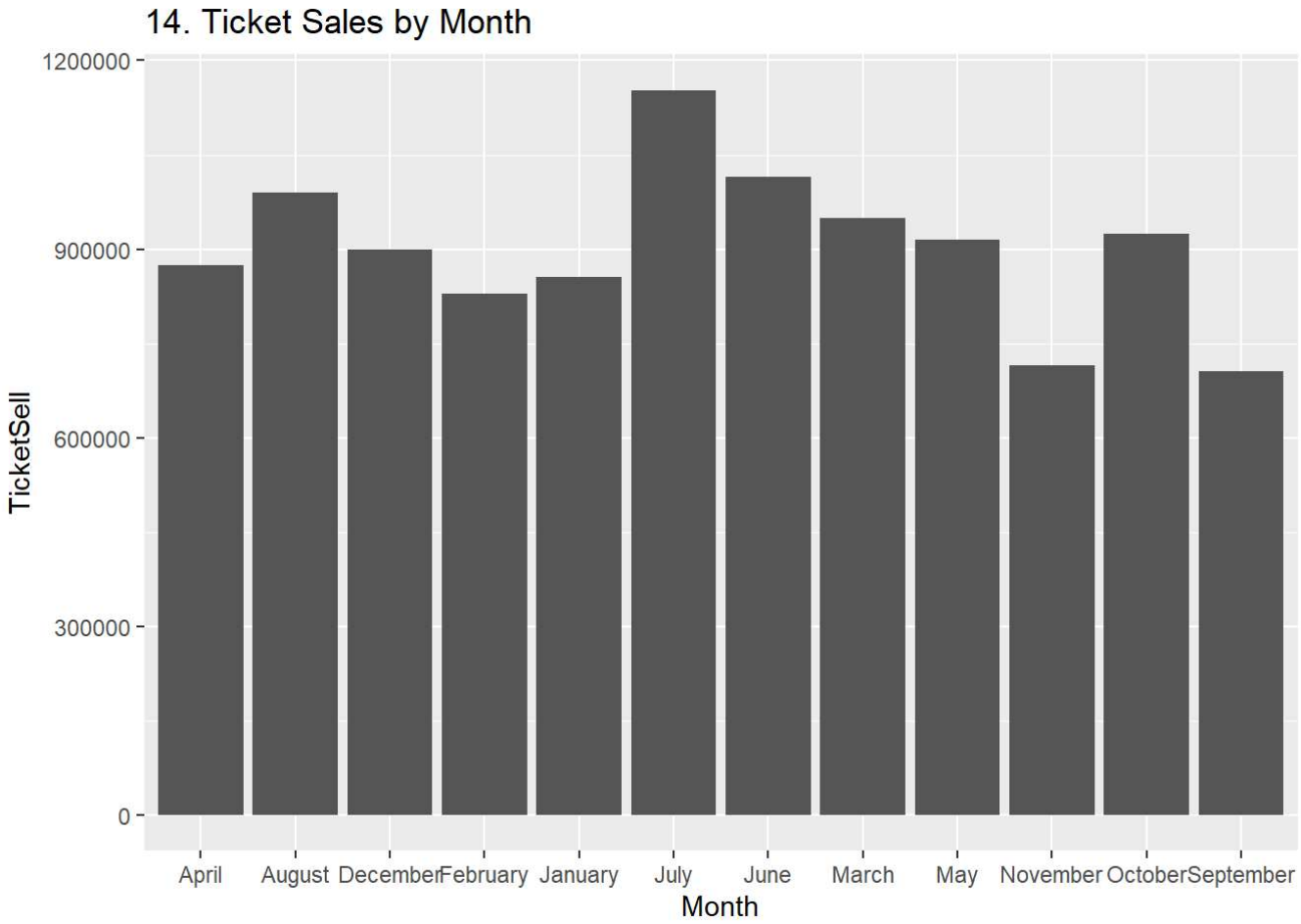

```
# 12. Travel Method Box Plot
if(nrow(travel_data) > 0) {
  ggplot(travel_data, aes(x = Indicator, y = value)) +
    geom_boxplot() +
    labs(title = "12. Travel Method Value Distribution")
}
```




```
# 13. Purpose of Trip Violin Plot
if(nrow(purpose_data) > 0) {
  ggplot(purpose_data, aes(x = Indicator, y = value)) +
    geom_violin() +
    labs(title = "13. Purpose of Trip Distribution")
}
```

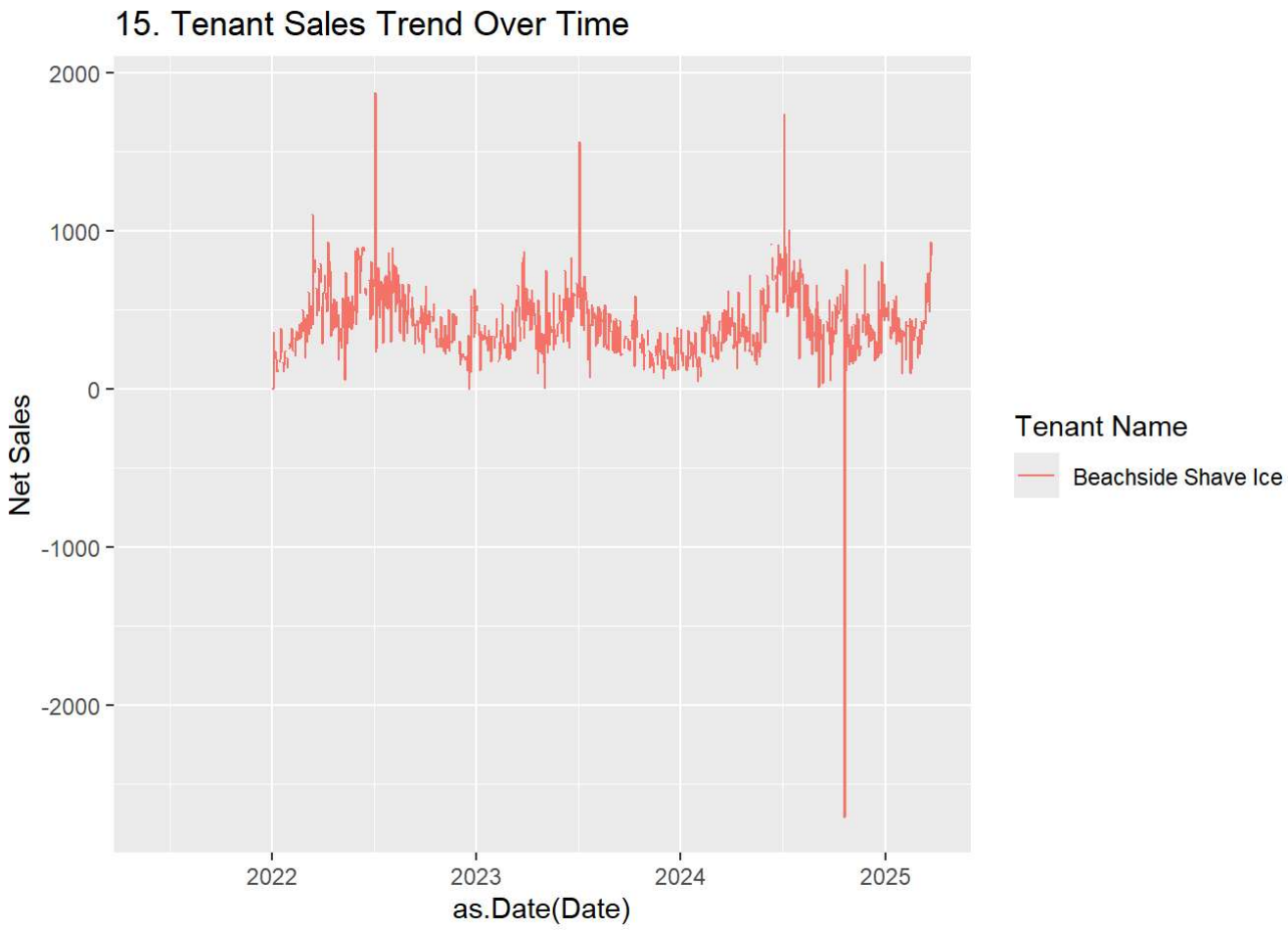



```
# 14. Ticket Sales by Month Bar
if(nrow(ticket_data) > 0) {
  ggplot(ticket_data, aes(x = Month, y = TicketSell)) +
    geom_bar(stat = "identity") +
    labs(title = "14. Ticket Sales by Month")
}
```

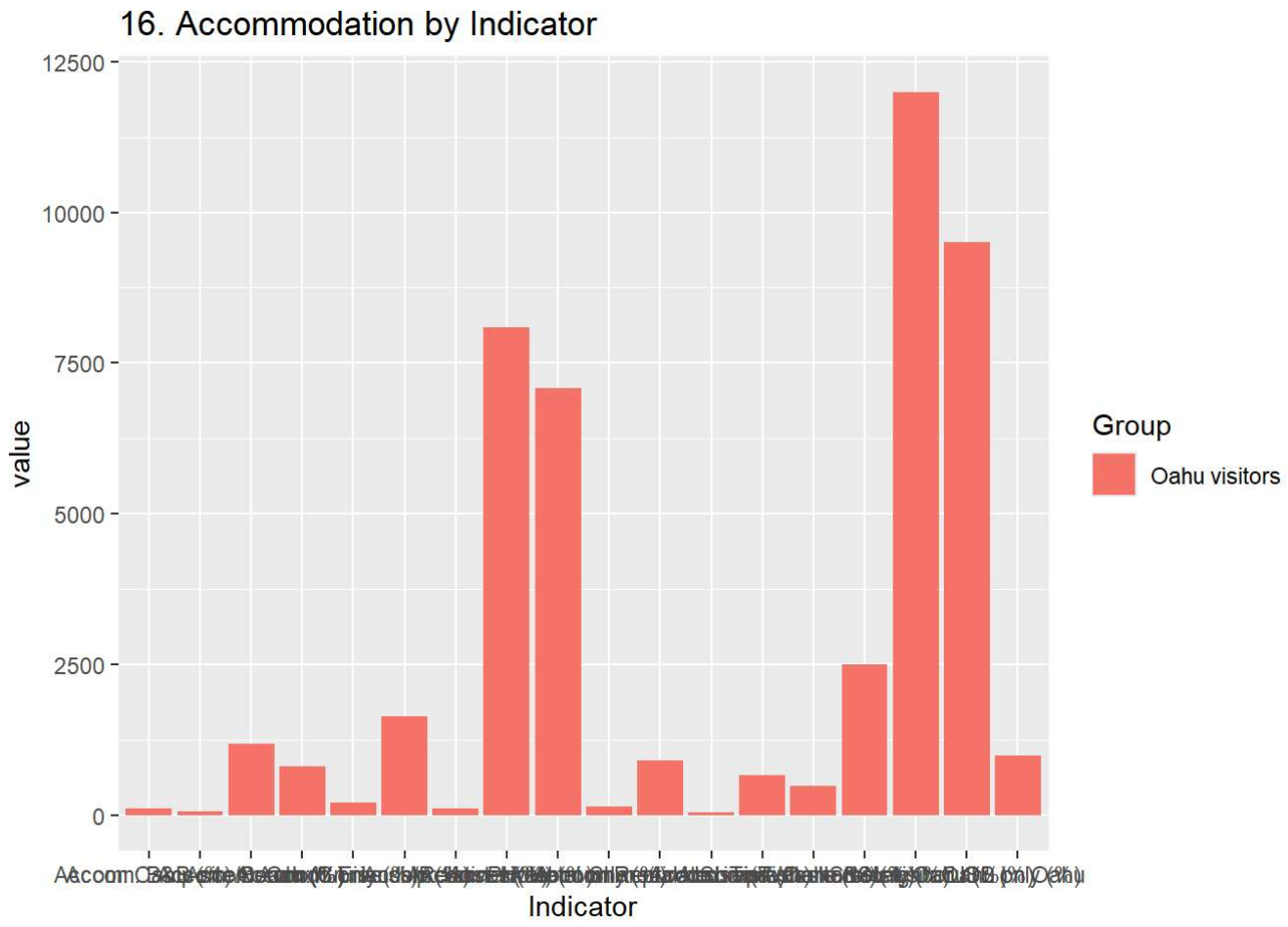



```
# 15. Tenant Sales Trend
if(nrow(tenant_data) > 0) {
  ggplot(tenant_data, aes(x = as.Date(Date), y = `Net Sales`, color = `Tenant Name`)) +
    geom_line() +
    labs(title = "15. Tenant Sales Trend Over Time")
}
```

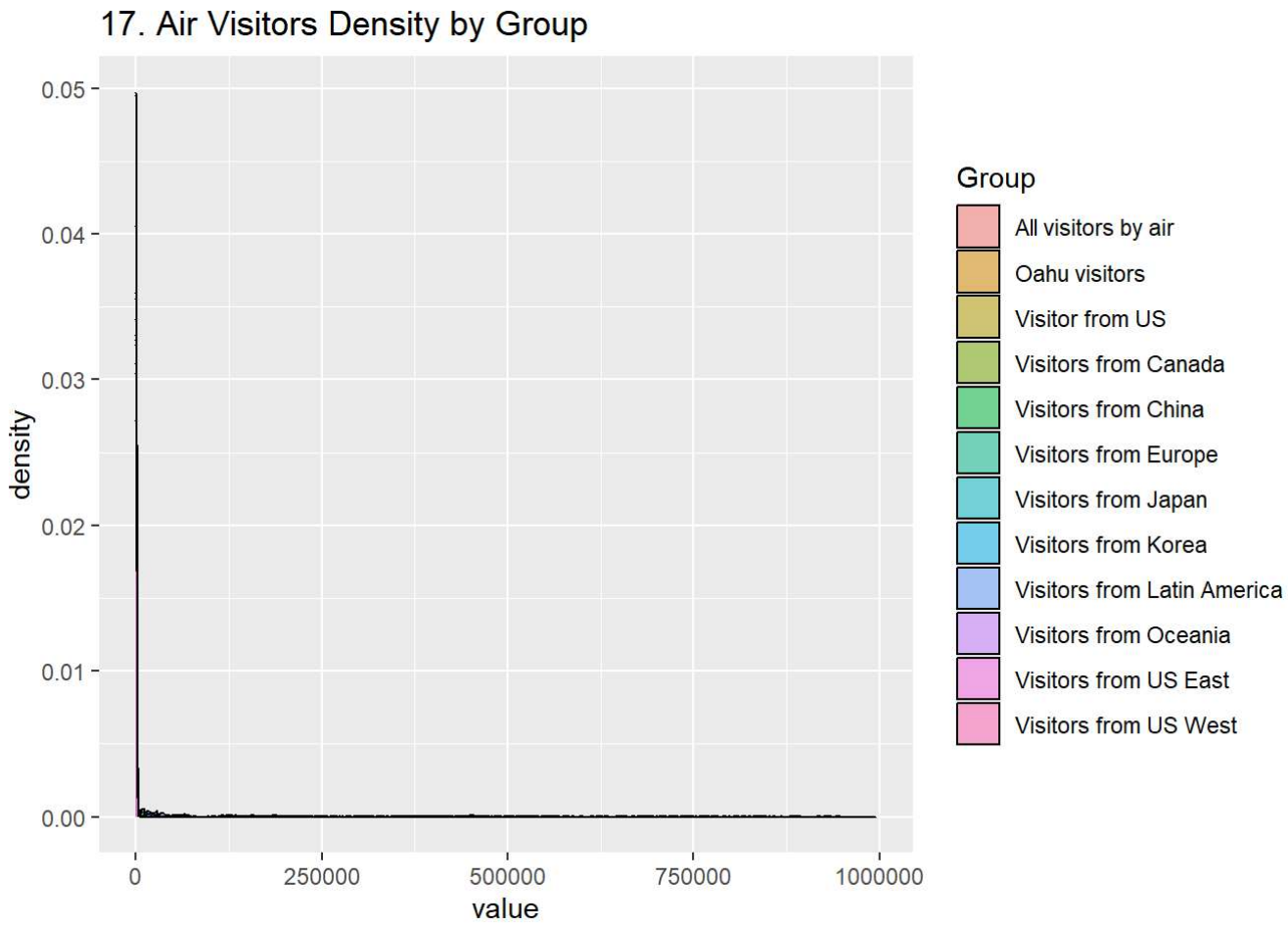
```
## Warning: Removed 128 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```
# 16. Accommodation by Indicator Stacked Bar
if(nrow(accom_data) > 0) {
  ggplot(accom_data, aes(x = Indicator, y = value, fill = Group)) +
    geom_bar(stat = "identity") +
    labs(title = "16. Accommodation by Indicator")
}
```




```
# 17. Air Visitors Density Plot
if(nrow(air_data) > 0) {
  ggplot(air_data, aes(x = value, fill = Group)) +
    geom_density(alpha = 0.5) +
    labs(title = "17. Air Visitors Density by Group")
}
```




```
# Disconnect from database
dbDisconnect(con)
```

```
## [1] TRUE
```