

# Streamlining Practice Operations with R: A Data-Driven Approach for Aoife's GP Practice

Nur Alya Maisarah Abd Karim, A00268252

## Contents

<b>1</b>	<b>Executive Summary</b>	<b>2</b>
<b>2</b>	<b>Overview of the GP Practice</b>	<b>2</b>
<b>3</b>	<b>Objectives and Approach</b>	<b>2</b>
<b>4</b>	<b>Introduction</b>	<b>2</b>
<b>5</b>	<b>Data Management</b>	<b>3</b>
5.1	Challenges . . . . .	3
5.2	R Solution . . . . .	3
<b>6</b>	<b>Data Analysis and Visualization</b>	<b>8</b>
6.1	Challenges . . . . .	8
6.2	R Solution . . . . .	8
6.3	Benefits of Visualizing Data with ggplot2 . . . . .	8
<b>7</b>	<b>Automated Reporting</b>	<b>11</b>
7.1	Challenges . . . . .	11
7.2	R Solution . . . . .	11
<b>8</b>	<b>End of the Year Tax Report</b>	<b>15</b>
8.1	End-of-Year Tax Reports . . . . .	15
8.2	Benefits of Automating Tax Reports: . . . . .	15
<b>9</b>	<b>Financial Analysis: Monitoring Revenue and Costs</b>	<b>18</b>
9.1	Revenue Breakdown by Source . . . . .	18
9.2	Breakdown of Analysis: . . . . .	19
<b>10</b>	<b>Compliance and Monitoring</b>	<b>20</b>
10.1	Key Monitoring Areas: . . . . .	21
<b>11</b>	<b>Conclusion</b>	<b>25</b>
11.1	Broader Impacts . . . . .	25

# 1 Executive Summary

Given the increasing administrative workload and fragmented data storage systems, this proposal presents tailored data management solutions for Aoife’s GP practice, with a focus on utilizing R Studio and other recommended tools. The primary objectives are to streamline data access, enhance the accuracy of reports, and automate routine tasks, ultimately easing the burden on the front desk and improving patient interactions. By implementing a more organized approach to data management, this report aims to help Aoife make informed decisions when selecting systems that boost efficiency, ensure compliance, and support consistent and reliable reporting across all areas of the practice.

## 2 Overview of the GP Practice

Aoife’s practice faces typical yet challenging operational hurdles: four doctors and three receptionists manage patient data, financial records, and operational metrics scattered across formats and devices. Manual tracking of patient and financial data is particularly challenging given the busy, hands-on nature of GP practices. The current data environment demands a shift to a centralized, structured approach that offers real-time accessibility and reliability.

## 3 Objectives and Approach

This report outlines how R can be used to improve the company’s operations. R offers powerful tools for automating repetitive tasks, organizing large amounts of data, and generating insightful reports. By using R, the company can streamline processes, making it easier to manage data and reduce the time spent on manual tasks. The use of R for data analysis and reporting will also help improve the accuracy and consistency of the reports, leading to more informed decision-making. This approach aims to enhance overall efficiency by simplifying workflows, improving productivity, and supporting better business decisions.

## 4 Introduction

Aoife’s GP practice serves as a vital healthcare provider in the community, staffed by four doctors and three receptionists who manage a steady flow of patients and administrative tasks. Like many healthcare practices, the clinic faces the challenge of balancing high-quality patient care with the increasing demands of data management. Addressing these challenges is crucial to maintaining efficiency and service quality, even with limited resources.

Key issues include scheduling appointments, managing patient records, distributing workloads, and generating reports for internal and external purposes. The secure handling, frequent updates, and meaningful analysis of healthcare data further complicate these tasks, making manual methods less viable in a modern setting. These challenges underscore the need for robust, automated data management solutions.

This report explores how the R programming language can help address these challenges. R provides powerful tools for data analysis, organization, and visualization, making it ideal for converting raw data into actionable insights. By leveraging R, the practice can automate processes such as generating patient flow reports, identifying trends in appointments, and optimizing resource allocation. These enhancements reduce the administrative burden on staff, allowing them to focus more on patient care.

Adopting a data-driven approach also positions the practice to work more effectively with data analytics consultancies. With access to preliminary analyses and clear visualizations, the practice can better communicate its requirements, ensuring alignment with consultancy services and potentially lowering costs. Ultimately, integrating R into the practice’s operations will enhance efficiency, streamline reporting, and support evidence-based decision-making, enabling Aoife’s clinic to meet the demands of a data-driven future.

## 5 Data Management

### 5.1 Challenges

Aoife’s GP practice faces several challenges in managing its data, which affect daily operations and decision-making. These challenges include:

**Data stored in various formats and locations:** The practice’s data is spread across different formats like CSV, TXT, and Excel files, and stored in multiple places, such as cloud storage, local computers, and even mobile devices. This scattered setup makes it hard to quickly access and analyze data, and staff end up spending too much time searching for and organizing files, which slows down operations and increases the chances of mistakes.

**Redundant data and inconsistent naming conventions:** Duplicate records and inconsistent file naming make the data difficult to manage. For example, patient records might appear in different files with slight differences in names or formats, like “Patient\_Record\_Aoife\_Hutton.xlsx” and “AoifeHutton\_PatientRecord.csv.” These inconsistencies create confusion and make it hard to track down duplicates, which can lead to errors in reporting and decision-making.

**High administrative workload for receptionists:** Receptionists have to manage appointments, patient records, and a lot of other tasks manually. With the current fragmented data setup, this workload becomes overwhelming. Tasks like gathering appointment statistics or searching for patient information take up time that could be better spent on improving patient care and satisfaction.

These challenges highlight the need for a more organized, centralized, and automated approach to managing data, which would improve efficiency and lighten the load on staff.

### 5.2 R Solution

R offers a comprehensive and flexible solution to address these challenges by centralizing, cleaning, and analyzing data effectively. With its extensive ecosystem of packages designed for data manipulation and analysis, R simplifies the process of transforming messy, scattered data into a unified and actionable format.

#### 5.2.1 Key Features of the R Solution

##### Data Centralization:

- Using R, data from various formats (CSV, TXT, Excel) and sources (cloud or local storage) can be imported and consolidated into a single, organized dataset.
- Packages like **readr** and **readxl** provide seamless integration for reading different file types, while **googledrive** or **aws.s3** can help fetch files from cloud storage.

##### Data Cleaning:

- One of R’s key strengths is its ability to clean and transform data. For Aoife’s GP practice, R can address issues like redundant records and inconsistent naming conventions using packages like **dplyr** and **tidyr**.

##### Removing Duplicates:

- The **distinct()** function in **dplyr** can be used to remove duplicate rows from the dataset, ensuring that no patient or appointment is counted more than once.

## Standardizing Columns and Formats:

- Using functions from the **dplyr** package, such as `mutate()` and `filter()`, R can standardize column names and formats. For instance, you can rename columns (e.g., changing “Patient\_ID” to “patient\_id”) or convert payment types to lowercase for consistency
- The `group_by()` function in **dplyr** is invaluable for aggregating data, enabling easy identification of patterns or summaries across categories like payment types or appointment statuses
- The **stringr** package provides tools such as `str_to_lower()` to clean up inconsistent text entries, such as standardizing case in payment types or ensuring patient names follow the same format

## Handling Dates Efficiently:

- The **lubridate** package simplifies working with dates by parsing, formatting, and manipulating date fields. For example, `ymd()` can convert “20240317” into a standard date format like “2024-03-17,” making appointment scheduling and analysis easier

## Aggregating Data for Summaries:

- The `group_by()` function in **dplyr** helps in aggregating data to identify patterns or summarize metrics across categories, such as calculating total payments per type (e.g., HSE, private)

Example:

```
# Load necessary libraries
library(dplyr)
library(tidyr)
library(knitr)
library(lubridate) # For date manipulation
library(stringr)   # For string manipulation

# Example data
data <- data.frame(
  patient_id = c(1, 2, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12,
                13, 14, 14, 15, 16),
  payment_type = c(
    "HSE", "private", "HSE", "private", "medical card",
    "HSE", "medical card", "private", "HSE", "medical card",
    "private", "HSE", "HSE", "medical card", "private",
    "private", "HSE"
  ),
  amount_paid = c(50, 100, 100, 50, 20, 80, 100, 120, 60, 30,
                 90, 70, 110, 40, 130, 60, 85),
  payment_date = c(
    "20241101", "20241102", "20241102", "20241105", "20241107",
    "20241109", "20241110", "20241111", "20241115", "20241117",
    "20241120", "20241121", "20241122", "20241123", "20241123",
    "20241124", "20241125"
  )
)
```

*# Convert payment\_date to Date format*

Table 1: Summary of Payments by Payment Type (Filtered and Converted)

Payment Type	Total Amount Paid (€)	Total Amount Paid (USD)
Hse	555	610.5
Private	550	605.0
Medical Card	140	154.0

```
data <- data %>%
  mutate(payment_date = ymd(payment_date)) # Use lubridate to parse dates

cat("Code to generate the table:\n\n")
```

Code to generate the table:

```
# Cleaning
cleaned_data <- data %>%
  distinct() %>%
  mutate(
    amount_paid_usd = amount_paid * 1.1, # Convert amount to USD
    payment_type = str_to_title(payment_type) # Format payment type
  ) %>%
  filter(
    amount_paid > 30, # Filter payments above 30
    month(payment_date) == 11 # Filter for November payments
  ) %>%
  group_by(payment_type) %>%
  summarize(
    total_amount = sum(amount_paid),
    total_amount_usd = sum(amount_paid_usd),
    .groups = "drop"
  )
```

The table for the code is as below:

```
# Formatted table with LaTeX formatting
cleaned_data %>%
  arrange(desc(total_amount)) %>%
  kable(
    caption = "Summary of Payments by Payment Type (Filtered and Converted)",
    col.names = c("Payment Type", "Total Amount Paid (€)", "Total Amount Paid (USD)"),
    format = "latex", # For PDF compatibility
    booktabs = TRUE
  ) %>%
  knitr::knit_print()
```

Below is an additional section which demonstrates how to handle missing values using imputation and how to deal with outliers in a dataset.

### Handling Missing Values:

- The `is.na()` function from the **base** package identifies missing values in the `amount_paid` column, and the `ifelse()` function can replace them with the column's median. The median is used for imputation because it is less sensitive to outliers compared to the mean.

## Outlier Detection:

- The **dplyr** package can be used with the Interquartile Range (IQR) method to detect outliers. Values outside  $Q3 + 3 \times IQR$  are considered outliers and can be removed using the `filter()` function in `dplyr`.

## Visualization:

A bar chart created with `ggplot2` visually confirms the cleaning process, showing the adjusted amounts without outliers or missing values

## Output:

- **Before Cleaning:** Missing values and extreme outliers are present in the data
- **After Cleaning:** Missing values are imputed, and outliers are removed

## Examples:

```
# Load necessary libraries
library(dplyr)
library(tidyr)
library(ggplot2)
library(lubridate)

# Example data with missing values and outliers
data_with_missing <- data.frame(
  patient_id = c(1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010),
  amount_paid = c(50, NA, 70, 5000, 40, 35, NA, 80, 30, 25),
  # Contains missing values and an outlier
  payment_date = c(
    "20241101", "20241102", "20241105", "20241107",
    "20241109", "20241110", "20241111", "20241115",
    "20241117", "20241120"
  )
)

# Convert payment_date to Date format
data_with_missing <- data_with_missing %>%
  mutate(payment_date = ymd(payment_date))

# Handling missing values: Using imputation
# Replacing missing 'amount_paid' with median
data_imputed <- data_with_missing %>%
  mutate(amount_paid = ifelse(is.na(amount_paid), median(amount_paid, na.rm = TRUE), amount_paid))

# Detecting and handling outliers
# Define an outlier as any value greater than 3 times the interquartile range (IQR)
Q1 <- quantile(data_imputed$amount_paid, 0.25, na.rm = TRUE)
Q3 <- quantile(data_imputed$amount_paid, 0.75, na.rm = TRUE)
IQR <- Q3 - Q1
outlier_threshold <- Q3 + 3 * IQR

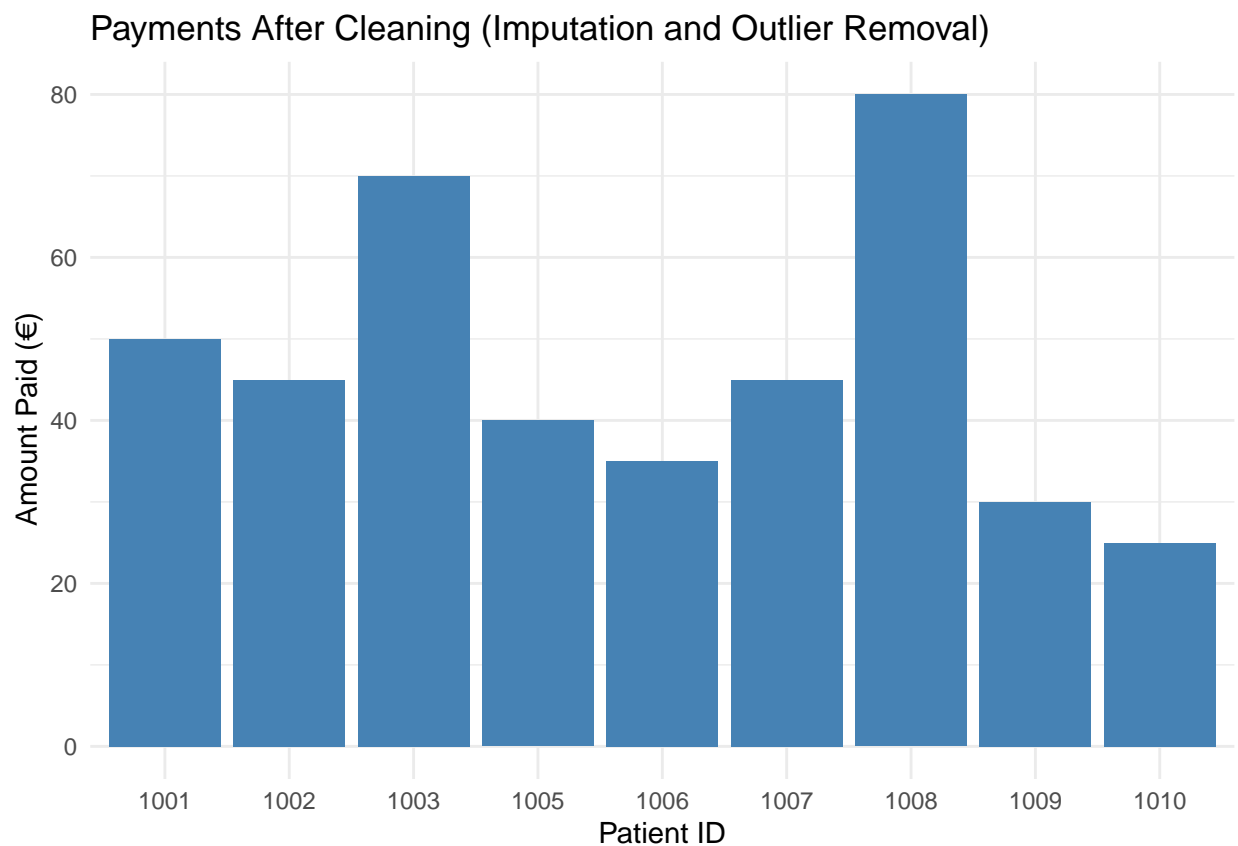
# Remove outliers
data_cleaned <- data_imputed %>%
  filter(amount_paid <= outlier_threshold)
```

```

# Summary of the statistics before and after cleaning
summary_before <- summary(data_with_missing$amount_paid)
summary_after <- summary(data_cleaned$amount_paid)

# Plot
ggplot(data_cleaned, aes(x = factor(patient_id), y = amount_paid)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(
    title = "Payments After Cleaning (Imputation and Outlier Removal)",
    x = "Patient ID",
    y = "Amount Paid (€)"
  ) +
  theme_minimal()

```



## 6 Data Analysis and Visualization

### 6.1 Challenges

Data analysis is critical in any GP practice to identify performance trends, improve operational efficiency, and support informed decision-making. However, several challenges can complicate these tasks:

- **Monitoring GP Performance:** It can be difficult to track individual doctor performance, including the number of appointments handled, patient satisfaction, and outcomes over time
- **Appointment Trends:** Identifying trends such as peak appointment times, common types of consultations, or recurring cancellations often requires manual work and is prone to errors
- **Simplifying Data Presentation:** Presenting raw data during monthly reviews can overwhelm staff and stakeholders, making it harder to draw actionable conclusions quickly.

### 6.2 R Solution

The **ggplot2** package, part of the **Tidyverse** collection of R packages, is an effective tool for turning raw data into clear, informative visualizations. Using ggplot2, Aoife can create bar plots, line charts, and heatmaps to uncover trends in appointments, identify peak times, and evaluate doctor performance. These visualizations make it easier for staff to spot patterns and make decisions based on data.

To further improve the look and clarity of the charts, the **RColorBrewer** package can be used. This package offers a range of color palettes, such as “Dark2,” which can be applied with `scale_fill_brewer()` to make the charts more visually appealing and easy to read, ensuring that data is presented in a clear and accessible way during reviews.

### 6.3 Benefits of Visualizing Data with ggplot2

- **Intuitive Visuals:** Presenting data in the form of graphs and charts makes it easier for all stakeholders to understand key insights at a glance
- **Customization:** R’s ggplot2 offers extensive customization options to ensure the visualizations are tailored to specific needs, including adding labels, legends, and annotations
- **Automated Reports:** Visualization scripts can be reused monthly, saving time and ensuring consistency in reporting

```
library(ggplot2)

# Example appointment data
appointment_data <- data.frame(
  day = c(
    "Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
    "Saturday", "Sunday", "Monday (Week 2)", "Tuesday (Week 2)",
    "Wednesday (Week 2)", "Thursday (Week 2)", "Friday (Week 2)",
    "Saturday (Week 2)", "Sunday (Week 2)"
  ),
  num_appointments = c(30, 40, 35, 50, 45, 20, 25, 32, 42, 38, 55, 48, 22, 28)
)

# Bar chart for appointment trends
custom_colors <- c(
```

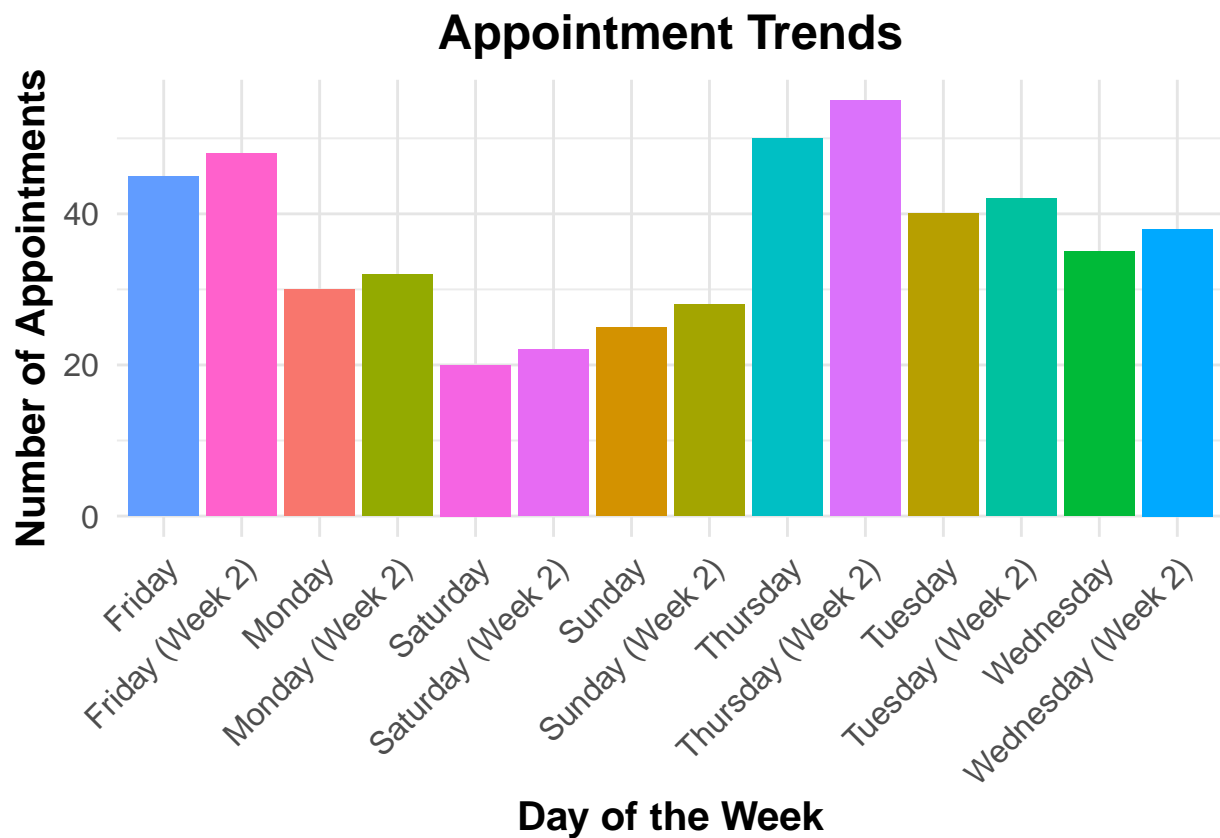


```

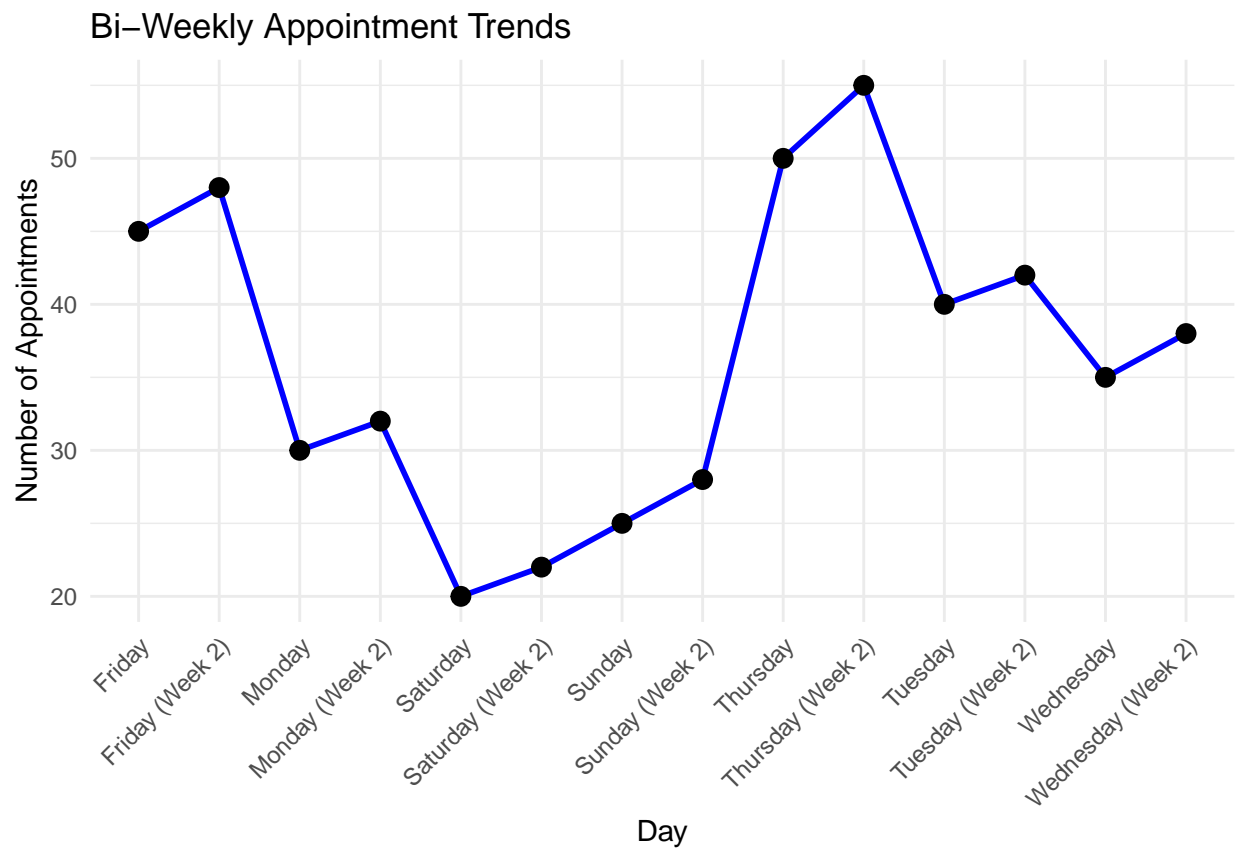
"Monday" = "#F8766D", "Tuesday" = "#B79F00", "Wednesday" = "#00BA38",
"Thursday" = "#00BFC4", "Friday" = "#619CFF", "Saturday" = "#F564E3",
"Sunday" = "#D39200", "Monday (Week 2)" = "#93AA00", "Tuesday (Week 2)" = "#00C19F",
"Wednesday (Week 2)" = "#00A9FF", "Thursday (Week 2)" = "#DB72FB",
"Friday (Week 2)" = "#FF61CC", "Saturday (Week 2)" = "#E76BF3",
"Sunday (Week 2)" = "#A3A500"
)

ggplot(appointment_data, aes(x = day, y = num_appointments, fill = day)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  labs(
    title = "Appointment Trends",
    x = "Day of the Week",
    y = "Number of Appointments"
  ) +
  scale_fill_manual(values = custom_colors) +
  theme_minimal(base_size = 15) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1), # Rotate x-axis labels for better fit
    plot.title = element_text(hjust = 0.5, size = 18, face = "bold"),
    axis.title = element_text(face = "bold"), # Bold axis labels
    panel.grid.major = element_line(color = "gray90", size = 0.5) # Lighter grid lines
  )

```



```
# Line chart
ggplot(appointment_data, aes(x = day, y = num_appointments, group = 1)) +
  geom_line(color = "blue", size = 1) +
  geom_point(size = 3) +
  labs(
    title = "Bi-Weekly Appointment Trends",
    x = "Day",
    y = "Number of Appointments"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Rotate x-axis labels
```



## 7 Automated Reporting

### 7.1 Challenges

Automated reporting is a critical need for Aoife's GP practice due to the repetitive and time-consuming nature of compiling important documents. The key challenges include:

**Manual generation of monthly HSE reimbursement reports:** Each month, the practice must prepare detailed reports for the Health Service Executive (HSE) to claim reimbursements. This involves manually aggregating data from various sources, organizing it into the required format, and ensuring accuracy. This process is prone to human error, especially under time pressure, and consumes significant administrative time

**End-of-year tax reports for patients:** Many patients request detailed summaries of their medical expenses for tax purposes. Generating these reports requires aggregating transaction data that is often scattered across multiple systems or files. Ensuring the accuracy and completeness of these reports is essential, as errors can lead to patient dissatisfaction or even legal complications

### 7.2 R Solution

R provides an efficient and reliable solution for automating these recurring reporting tasks. By leveraging R's powerful scripting and data manipulation capabilities, Aoife's practice can streamline the generation of monthly and annual reports, reducing the administrative burden and minimizing errors.

#### 7.2.1 Key Benefits of Using R for Automated Reporting:

##### 1. Reproducible Scripts:

- R scripts enable the automation of repetitive tasks like data cleaning, aggregation, and formatting, ensuring that reports are consistently generated without manual intervention. Once written, these scripts can be reused for future reports, saving time and increasing accuracy
- In the provided examples, R demonstrates its power for automated reporting through the use of packages such as `dplyr` and `knitr`. Functions like `mutate()`, `summarise()`, and `kable()` allow for streamlined workflows. For instance, in the HSE Claims Report, the script uses `mutate()` from **dplyr** to calculate the total reimbursement per consultation type by multiplying the number of patients by the reimbursement rate. The results are then displayed in a LaTeX-compatible table using `kable()` from **knitr**, ensuring that the final report is not only accurate but also formatted in a clean, professional manner
- Example:

```
# Load necessary libraries
library(dplyr)
library(knitr)

# Example extended HSE claims data
claims_data <- data.frame(
  consultation_type = c(
    "Routine Checkup", "Specialist Referral", "Vaccination", "Blood Test",
    "X-ray", "MRI Scan", "Consultation - Emergency", "Mental Health Check",
    "Physical Therapy", "Dental Consultation", "Eye Test", "Chronic Disease Management",
    "Diabetes Care", "Pediatric Care", "Geriatric Care", "Laboratory Test",
    "Surgical Consultation"
```

Table 2: Summary of Payments by Payment Type

Consultation Type	Number of Patients	Reimbursement Rate	Total Reimbursement (€)
Routine Checkup	150	40	6000
Specialist Referral	30	60	1800
Vaccination	50	30	1500
Blood Test	100	50	5000
X-ray	40	70	2800
MRI Scan	60	90	5400
Consultation - Emergency	20	100	2000
Mental Health Check	70	80	5600
Physical Therapy	80	75	6000
Dental Consultation	30	55	1650
Eye Test	60	45	2700
Chronic Disease Management	110	60	6600
Diabetes Care	85	50	4250
Pediatric Care	95	65	6175
Geriatric Care	120	55	6600
Laboratory Test	45	70	3150
Surgical Consultation	55	95	5225

```

),
num_patients = c(
  150, 30, 50, 100, 40, 60, 20, 70, 80, 30, 60, 110, 85, 95, 120, 45, 55
),
reimbursement_rate = c(
  40, 60, 30, 50, 70, 90, 100, 80, 75, 55, 45, 60, 50, 65, 55, 70, 95
)
)

# Total reimbursement for each consultation type
monthly_report <- claims_data %>%
  mutate(total_reimbursement = num_patients * reimbursement_rate)

# Table of HSE Claims Summary with LaTeX formatting
monthly_report %>%
  kable(
    caption = "Summary of Payments by Payment Type",
    format = "latex",
    col.names = c("Consultation Type", "Number of Patients",
                  "Reimbursement Rate", "Total Reimbursement (€)"),
    booktabs = TRUE
  )

# Cleaned data
cleaned_data <- data.frame(
  payment_type = c(
    "HSE", "Private", "Medical Card", "Insurance",
    "Corporate Accounts", "Specialist Consultations", "Emergency Services",
    "Outpatient Care"
  )

```

Table 3: Detailed Monthly Revenue by Payment Type

Payment Type	Total Monthly Revenue (€)
HSE	8000
Private	12000
Medical Card	5000
Insurance	4000
Corporate Accounts	7000
Specialist Consultations	9500
Emergency Services	6500
Outpatient Care	7200

```

),
total_revenue_monthly = c(
  8000, 12000, 5000, 4000,
  7000, 9500, 6500, 7200
)
)

# Summary Report
monthly_summary <- cleaned_data %>%
  summarise(total_revenue = sum(total_revenue_monthly))

# Displaying the payment data with LaTeX-compatible table format
cleaned_data %>%
  kable(
    caption = "Detailed Monthly Revenue by Payment Type",
    col.names = c("Payment Type", "Total Monthly Revenue (€)"),
    format = "latex", # Use LaTeX format for PDF
    booktabs = TRUE
  )

```

## 2. Automated Scheduling:

- R scripts can be seamlessly integrated with scheduling tools such as **cronR** or **Windows Task Scheduler** to automate report generation. This allows reports to be generated and saved at regular intervals without requiring manual input. For example, monthly or quarterly reports can be automatically created using scheduled R scripts, ensuring timely updates and reducing the administrative workload

## 3.Data Validation:

- R's data manipulation capabilities, particularly through functions like `filter()` and `mutate()` from the **dplyr** package, make it easy to validate and clean data before report generation. This ensures that inconsistencies, such as missing values or outliers, are addressed beforehand. For instance, invalid payment records can be filtered out, or errors in payment amounts can be corrected, improving the accuracy and reliability of reports

## 4.Customizable Output Formats:

- R offers exceptional flexibility in output formats, enabling reports to be generated as CSV files, PDFs, or Excel sheets. Packages like **knitr**, **writexl**, and **rmarkdown** facilitate this versatility. For example,

the `kable()` function from **knitr** can produce professional LaTeX-compatible tables, ideal for inclusion in polished documents. This flexibility allows reports to be customized to meet specific presentation and distribution needs

## 5. Scalable Reporting:

- As the practice expands and the volume of data increases, R scripts can be modified to accommodate larger datasets. With functions like `group_by()` and `summarise()` from **dplyr**, R can efficiently aggregate and process data, even when dealing with thousands of patient records. This ensures that reporting processes remain smooth and effective, regardless of the growth in data size

## 6. Example Workflow:

For the **Monthly HSE Reimbursement Report**, R can filter transactions by payment type, calculate aggregated totals for each month, and export the final report to a predefined location in the required format. This process ensures timely and accurate submission of reimbursement claims.

For the **End-of-Year Tax Report for Patients**, R can group transactions by patient using the `group_by()` function and calculate annual totals with `summarise()`. These reports can then be automatically generated and either saved for patient pickup or sent directly to patients via email using packages like **blastula**.

By automating these workflows, R reduces the administrative burden on staff, minimizes errors, and allows them to focus on more critical tasks. This approach improves operational efficiency and ensures the practice meets external reporting requirements with professionalism and ease.

## 8 End of the Year Tax Report

### 8.1 End-of-Year Tax Reports

Patients frequently request detailed medical expense summaries for tax purposes, which involves gathering and consolidating data about their appointments, treatments, and payments. With data scattered across different sources (e.g., spreadsheets, databases, and manual records), manually creating these reports is not only time-consuming but also increases the risk of errors.

R provides a robust solution to automate this process. By consolidating patient records, cleaning the data, and summarizing it into a well-structured report for each patient, R can save significant time and improve accuracy. These reports can then be exported in widely used formats such as PDF or Excel, ensuring they are easy to share and compliant with tax authority requirements.

### 8.2 Benefits of Automating Tax Reports:

- **Time Efficiency:** Automating recurring report generation significantly reduces the workload for administrative staff.
- **Accuracy:** Data cleaning and processing in R minimize errors in calculations.
- **Professional Output:** Reports can be customized and formatted for a professional look, instilling confidence in patients.
- **Scalability:** The process can handle large datasets, making it feasible even for larger clinics.

#### 8.2.1 Example: Generating Tax Reports for Patients

In the case of End-of-Year Tax Reports, R automates the task of consolidating patient expenses and generating detailed reports. The `mutate()` function is used to create a custom report for each patient, which includes their total visits and the amount paid. This report can be exported or printed in a professional format using `kable()`.

```
# Example patient expenses
expense_data <- data.frame(
  patient_id = 1:14, # Patient IDs from 1 to 14
  patient_name = c(
    "Haley Scott", "Arisa Khalid", "Alice Brown", "Leong Chi",
    "Emily Dina Matthew", "Nur Halid", "Sophia Lee", "Hong Xie Li",
    "Olivia Taylor", "Daniel Bond", "OKalawi Beng", "Emma Scott",
    "Jia Lee", "Ava Clark"
  ),
  total_visits = c(5, 3, 2, 4, 7, 6, 1, 8, 4, 2, 5, 3, 9, 6),
  total_paid = c(200, 150, 100, 180, 300, 250, 50, 320, 190, 90, 210, 140, 400, 280)
)

# Generating tax report
tax_reports <- expense_data %>%
  mutate(
    report_text = paste(
      "Patient:", patient_name,
      "\nTotal Visits:", total_visits,
      "\nTotal Amount Paid: €", total_paid
    )
  )
```

Table 4: Summary of Patient Tax Reports

ID	Name	Visits	Amount Paid (€)
13	Jia Lee	9	400
8	Hong Xie Li	8	320
5	Emily Dina Matthew	7	300
14	Ava Clark	6	280
6	Nur Halid	6	250
11	OKalawi Beng	5	210
1	Haley Scott	5	200
9	Olivia Taylor	4	190
4	Leong Chi	4	180
2	Arisa Khalid	3	150
12	Emma Scott	3	140
3	Alice Brown	2	100
10	Daniel Bond	2	90
7	Sophia Lee	1	50

```

)
)

# View individual reports
print(tax_reports$report_text)

## [1] "Patient: Haley Scott \nTotal Visits: 5 \nTotal Amount Paid: € 200"
## [2] "Patient: Arissa Khalid \nTotal Visits: 3 \nTotal Amount Paid: € 150"
## [3] "Patient: Alice Brown \nTotal Visits: 2 \nTotal Amount Paid: € 100"
## [4] "Patient: Leong Chi \nTotal Visits: 4 \nTotal Amount Paid: € 180"
## [5] "Patient: Emily Dina Matthew \nTotal Visits: 7 \nTotal Amount Paid: € 300"
## [6] "Patient: Nur Halid \nTotal Visits: 6 \nTotal Amount Paid: € 250"
## [7] "Patient: Sophia Lee \nTotal Visits: 1 \nTotal Amount Paid: € 50"
## [8] "Patient: Hong Xie Li \nTotal Visits: 8 \nTotal Amount Paid: € 320"
## [9] "Patient: Olivia Taylor \nTotal Visits: 4 \nTotal Amount Paid: € 190"
## [10] "Patient: Daniel Bond \nTotal Visits: 2 \nTotal Amount Paid: € 90"
## [11] "Patient: OKalawi Beng \nTotal Visits: 5 \nTotal Amount Paid: € 210"
## [12] "Patient: Emma Scott \nTotal Visits: 3 \nTotal Amount Paid: € 140"
## [13] "Patient: Jia Lee \nTotal Visits: 9 \nTotal Amount Paid: € 400"
## [14] "Patient: Ava Clark \nTotal Visits: 6 \nTotal Amount Paid: € 280"

# Creating a nicely formatted table for LaTeX output
expense_data %>%
  arrange(desc(total_paid)) %>% # Sort by total paid for better presentation
  kable(
    caption = "Summary of Patient Tax Reports",
    col.names = c("ID", "Name", "Visits", "Amount Paid (€)"),
    format = "latex", # Change to LaTeX format for PDF
    booktabs = TRUE
  )

```

Key Steps:



- **Data Aggregation:**

- The `group_by()` function groups data by `patient_id`, ensuring each patient's information is consolidated.
- `summarize()` calculates the total payments and the number of appointments for each patient.

- **Automation:**

- This script can be extended to loop through each `patient_id` and generate a separate, customized report for every patient, either as an Excel file or a PDF.

- **Professional Output:**

- The `kable()` function creates a professional summary table in LaTeX format, which can be used as part of a larger document or report.

By automating tax report generation, clinics can ensure that patients receive accurate and detailed summaries promptly, reducing administrative burdens and enhancing patient satisfaction.

## 9 Financial Analysis: Monitoring Revenue and Costs

Financial analysis plays a pivotal role in maintaining the long-term success and growth of Aoife's GP practice. By leveraging R for financial tracking and analysis, Aoife can gain a clearer understanding of her practice's revenue streams, assess costs, and improve profitability. This data-driven approach allows her to make informed decisions about pricing strategies, revenue diversification, and cost management, ensuring the practice remains financially healthy and sustainable.

Below are key areas of financial analysis that can be enhanced using R:

### 9.1 Revenue Breakdown by Source

Aoife's GP practice generates income from various sources, with the primary ones being the Health Service Executive (HSE), private consultations, and medical card consultations. By regularly analyzing how much revenue each of these sources contributes, Aoife can gain valuable insights into which areas are the most profitable and where there may be room for growth.

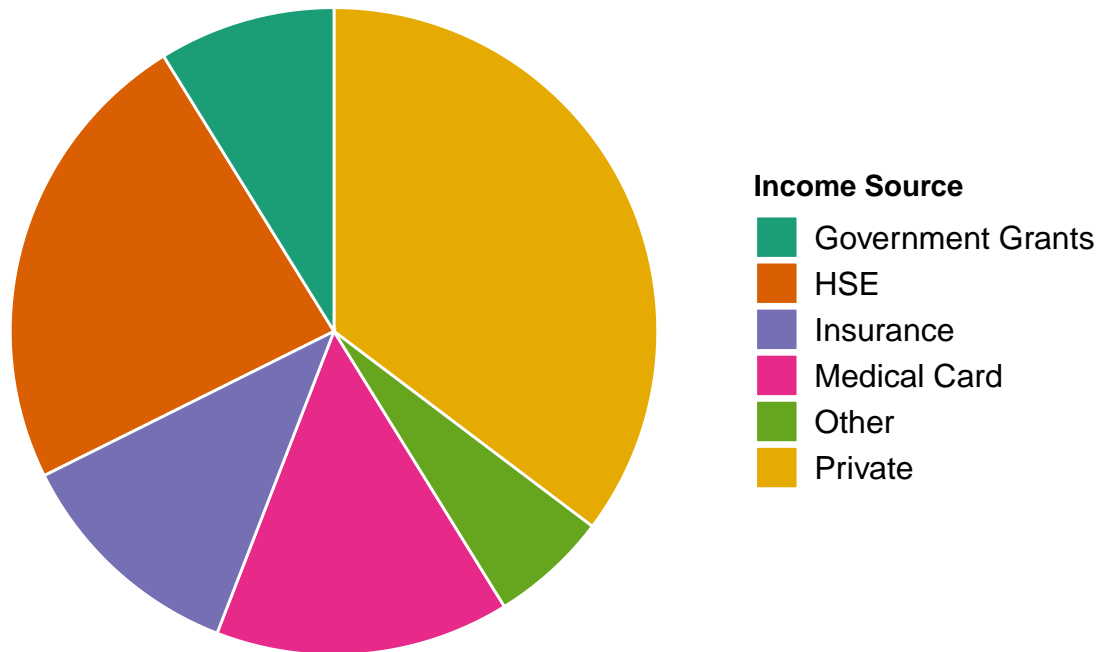
If the practice relies too heavily on a single source of income, such as HSE or private consultations, it may be at risk if that revenue stream declines. To mitigate this, Aoife could look for opportunities to diversify the practice's services, reducing dependence on one particular income source and ensuring financial stability.

To help visualize this, the following pie chart illustrates the distribution of revenue across the various income sources in the practice. This provides a clear overview of where the majority of income is coming from, and can guide Aoife's decisions regarding resource allocation and growth opportunities.

```
# Revenue Data
revenue_data <- data.frame(
  source = c("HSE", "Private", "Medical Card", "Insurance", "Government Grants", "Other"),
  total_income = c(8000, 12000, 5000, 4000, 3000, 2000)
)

# Pie chart for revenue breakdown
ggplot(revenue_data, aes(x = "", y = total_income, fill = source)) +
  geom_bar(stat = "identity", width = 1, color = "white") +
  coord_polar("y") +
  labs(
    title = "Revenue Breakdown by Source",
    fill = "Income Source"
  ) +
  theme_void() +
  theme(
    legend.position = "right", # Legend to the right of the bar chart
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    axis.text = element_blank(), # Remove axis text
    axis.title = element_blank(), # Remove axis title
    panel.grid = element_blank(), # Remove background grid lines
    legend.title = element_text(face = "bold"),
    legend.text = element_text(size = 12) # Adjust font size of legend
  ) +
  scale_fill_brewer(palette = "Dark2")
```

## Revenue Breakdown by Source



### 9.2 Breakdown of Analysis:

#### 9.2.1 Revenue Data Setup:

- A dataset `revenue_data` is created with columns for `source` (the different income streams) and `total_income` (the respective income for each source)

#### 9.2.2 Pie Chart Creation:

- The `ggplot()` function is used to create the pie chart, where the income sources are plotted as segments of a circle
- `geom_bar()` creates the bars, and `coord_polar()` transforms the bar chart into a pie chart
- The chart is customized with a clean, professional design using `theme_void()` to remove axes and gridlines for a sleek appearance. The legend is positioned to the right for better readability

#### 9.2.3 Color Customization:

- The `scale_fill_brewer(palette = "Dark2")` function applies a soft, professional color palette from the **RColorBrewer** package, making charts visually appealing and easy to interpret by distinguishing categorical data with muted yet distinct colors

## 10 Compliance and Monitoring

Monitoring and compliance are critical aspects of Aoife's GP practice, particularly in relation to regulatory requirements and accurate reimbursement from state programs such as the Health Service Executive (HSE). Effective tracking of appointments, patient billing, and documentation ensures that Aoife's practice complies with relevant healthcare standards while minimizing the risk of errors or financial discrepancies.

Using R, Aoife can automate and streamline the monitoring of key metrics, including appointment trends, patient demographics, and financial transactions. Regular compliance checks, such as verifying the number of consultations covered under state schemes, can help ensure that the practice receives the correct reimbursement and remains compliant with legal and financial regulations.

```
# Load necessary libraries
library(dplyr)
library(ggplot2)

# Data for appointments
appointments_data <- data.frame(
  consultation_type = c(
    "Annual Health Check", "Cardiology Review", "Vaccination", "Blood Test",
    "X-ray", "MRI Scan", "Emergency Consultation", "Mental Health Consultation",
    "Rehabilitation Therapy", "Dental Cleaning", "Vision Check",
    "Chronic Illness Management", "Diabetes Screening",
    "Pediatric Consultation", "Senior Care Checkup",
    "Lab Diagnostics", "Surgical Assessment"
  ),
  num_patients = c(150, 30, 50, 100, 40, 60, 20, 70, 80, 30, 60, 110, 85, 95, 120, 45, 55),
  payment_type = c(
    "HSE", "Private", "Medical Card", "Private", "HSE", "Private", "HSE", "Medical Card",
    "Private", "HSE", "Private", "Private", "Medical Card", "HSE",
    "Private", "Private", "Medical Card"
  )
)

# Billing data for reimbursement
billing_data <- data.frame(
  payment_type = c("HSE", "Private", "Medical Card", "Private", "HSE", "Private"),
  total_billed = c(8000, 12000, 5000, 4000, 7000, 9500)
)

# Compliance check: To verify the number of consultations under state schemes
hse_appointments <- appointments_data %>%
  filter(payment_type == "HSE") %>%
  summarise(total_hse_appointments = sum(num_patients))

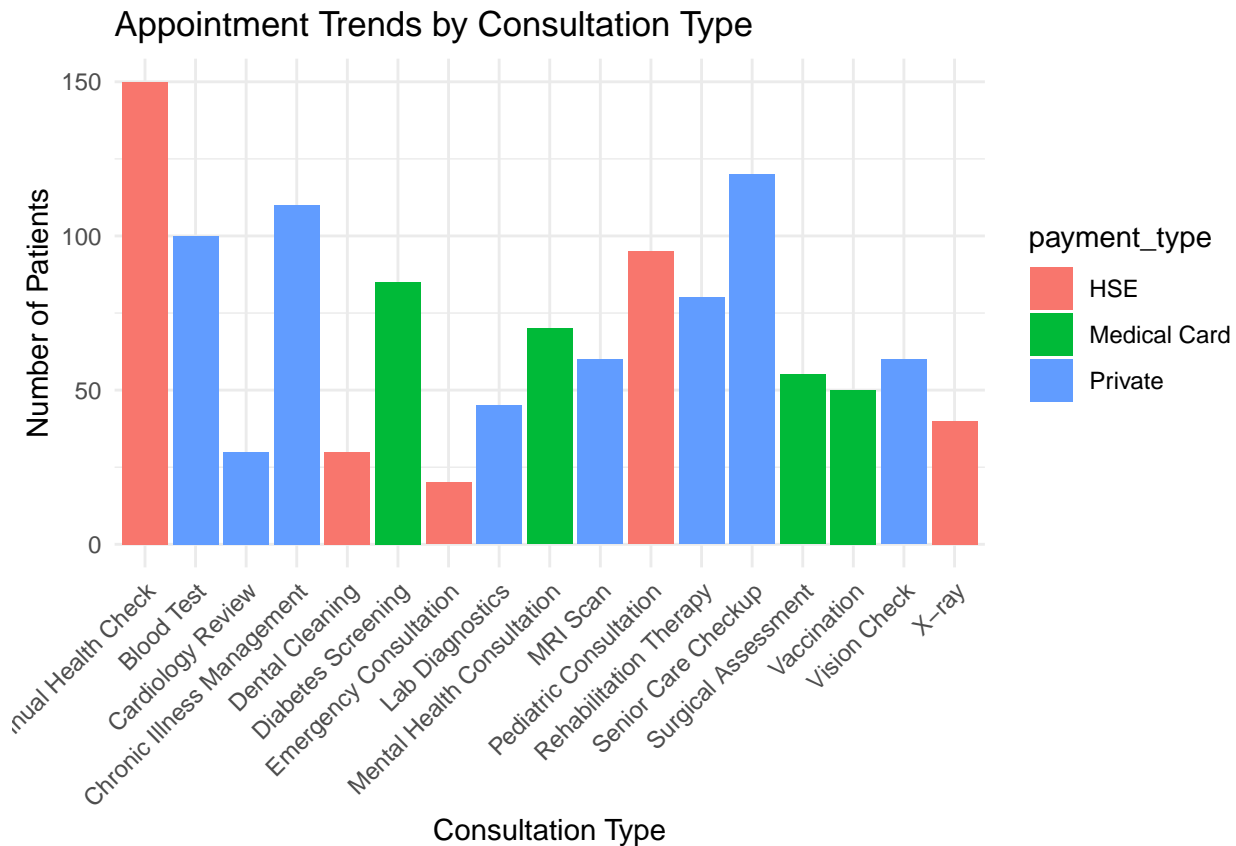
# Compliance check: To ensure the billing aligns with the HSE consultations
hse_billing_check <- billing_data %>%
  filter(payment_type == "HSE") %>%
  summarise(total_hse_billed = sum(total_billed))

# Compare the total number of HSE consultations with the total billed
compliance_status <- ifelse(hse_appointments$total_hse_appointments * 50
== hse_billing_check$total_hse_billed,
  "Compliance Met", "Compliance Issues Detected")
```

```
# Output compliance status
print(compliance_status)
```

```
## [1] "Compliance Issues Detected"
```

```
# Bar chart for appointment trends by consultation type
ggplot(appointments_data, aes(x = consultation_type, y = num_patients,
fill = payment_type)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(
    title = "Appointment Trends by Consultation Type",
    x = "Consultation Type",
    y = "Number of Patients"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



## 10.1 Key Monitoring Areas:

- **Appointment Tracking:** Keeping accurate records of the number of consultations each month to ensure proper billing
- **Patient Demographics:** Ensuring all patient data is complete, up-to-date, and in compliance with privacy regulations

- **Financial Compliance:** Monitoring payments from HSE, private patients, and other sources to avoid overpayments or underpayments, and to facilitate tax reporting

The use of R allows Aoife to automate routine tasks, track patient interactions efficiently, and generate compliance reports at regular intervals, thereby reducing administrative burden and improving overall operational efficiency.

The use of R in Aoife's practice allows for automation of key tasks, including tracking appointments, generating compliance reports, and visualizing performance. By automating routine tasks and using visual tools to monitor key metrics, Aoife can ensure her practice operates smoothly and complies with all necessary regulations, all while minimizing the administrative burden and focusing on providing quality care.

```
# GP performance data
library(ggplot2)
library(RColorBrewer)
library(stringr)

# GP performance data
gp_data <- data.frame(
  gp_name = c(
    "Dr. Jamie", "Dr. Linda", "Dr. Callum", "Dr. Damien", "Dr. Eva", "Dr. Emily",
    "Dr. Gareth", "Dr. Xi", "Dr. Ilya",
    "Dr. Melissa", "Dr. Najwa", "Dr. Lori", "Dr. Jonathan", "Dr. Oro", "Dr. Mila",
    "Dr. Weatherspoon", "Dr. Qi"
  ),
  patients_seen = c(
    120, 150, 130, 140, 110, 125, 135, 140, 160, 145, 125, 135, 110, 115, 145, 130, 138
  ),
  cancellations = c(
    5, 10, 8, 6, 7, 4, 3, 6, 9, 5, 6, 7, 4, 5, 6, 4, 8
  )
)

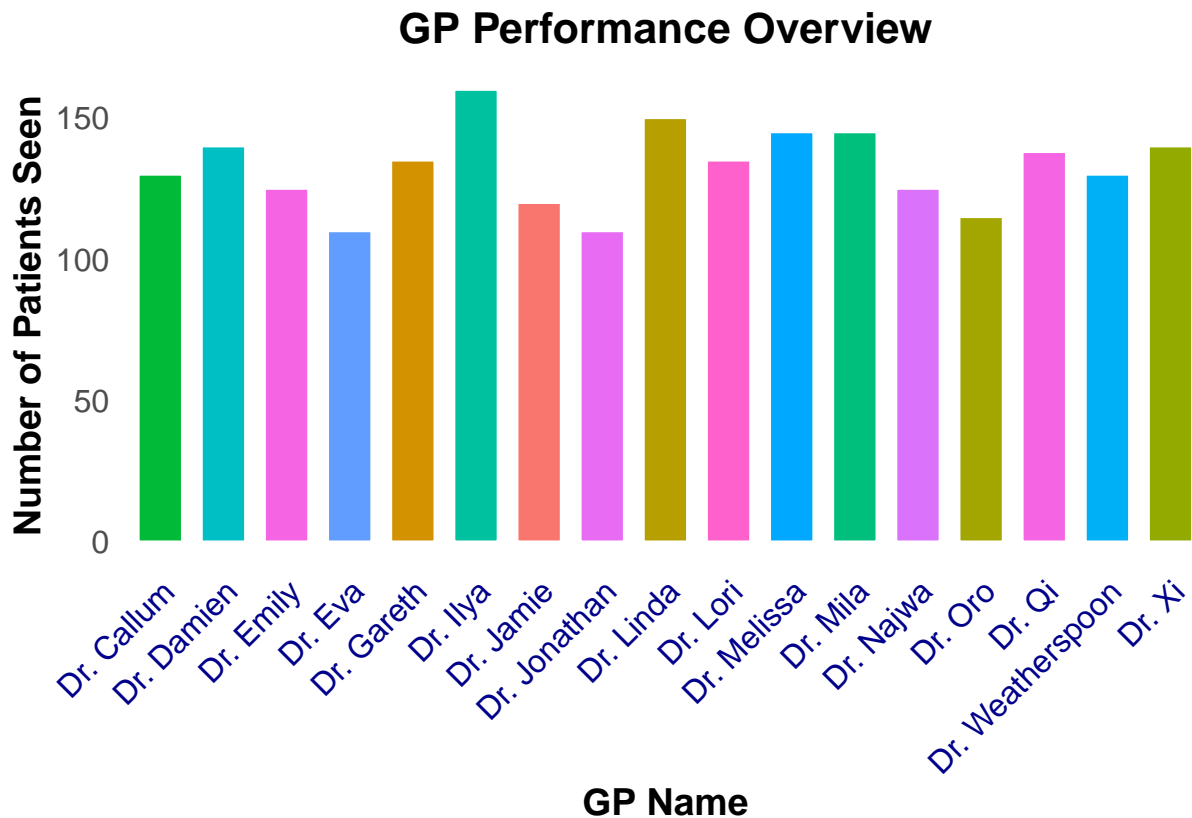
# Bar chart for GP performance
custom_colors <- c(
  "Dr. Jamie" = "#F8766D", "Dr. Linda" = "#B79F00", "Dr. Callum" = "#00BA38",
  "Dr. Damien" = "#00BFC4", "Dr. Eva" = "#619CFF", "Dr. Emily" = "#F564E3",
  "Dr. Gareth" = "#D39200", "Dr. Xi" = "#93AA00", "Dr. Ilya" = "#00C19F",
  "Dr. Melissa" = "#00A9FF", "Dr. Najwa" = "#DB72FB", "Dr. Lori" = "#FF61CC",
  "Dr. Jonathan" = "#E76BF3", "Dr. Oro" = "#A3A500", "Dr. Mila" = "#00BF7D",
  "Dr. Weatherspoon" = "#00B0F6", "Dr. Qi" = "#F564E3"
)

ggplot(gp_data, aes(x = gp_name, y = patients_seen, fill = gp_name)) +
  geom_bar(stat = "identity", show.legend = FALSE, color = "white", width = 0.7) +
  labs(
    title = "GP Performance Overview",
    x = "GP Name",
    y = "Number of Patients Seen"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1, size = 12, color = "darkblue"),
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
```

```

axis.title = element_text(size = 14, face = "bold"),
axis.text.y = element_text(size = 12),
panel.grid.major = element_blank(),
panel.grid.minor = element_blank(),
plot.margin = margin(10, 10, 10, 10)
) +
scale_fill_manual(values = custom_colors)

```



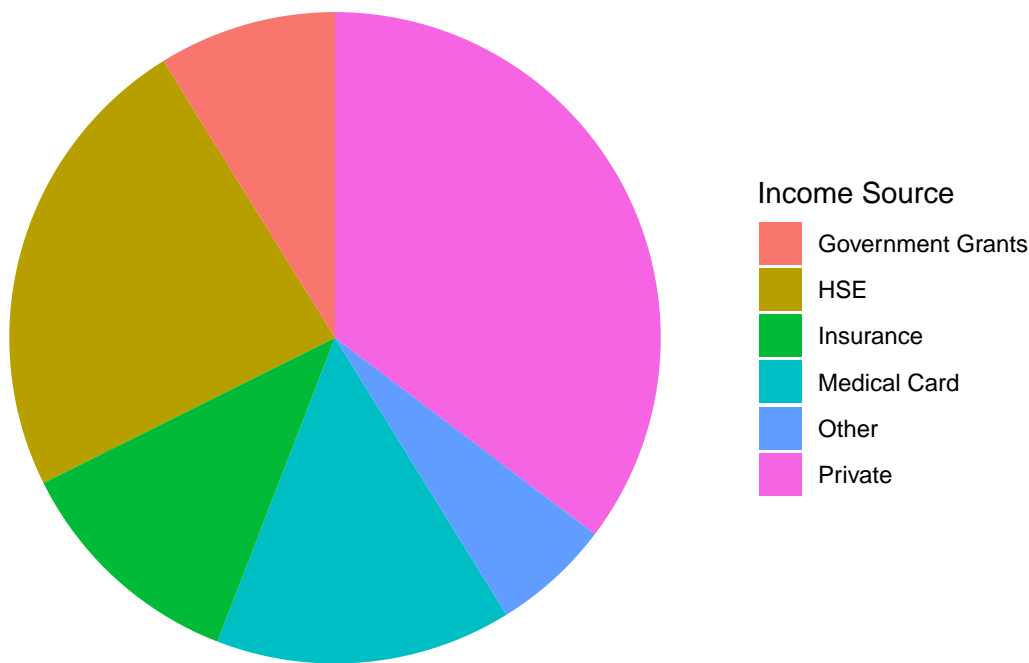
```

# Revenue data for compliance monitoring
revenue_data <- data.frame(
  source = c("HSE", "Private", "Medical Card", "Insurance", "Government Grants", "Other"),
  total_income = c(8000, 12000, 5000, 4000, 3000, 2000)
)

# Pie chart for revenue breakdown
ggplot(revenue_data, aes(x = "", y = total_income, fill = source)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar("y") +
  labs(title = "Revenue Breakdown by Source", fill = "Income Source") +
  theme_void() +
  theme(
    legend.position = "right",
    plot.title = element_text(size = 14, face = "bold")
  )

```

Revenue Breakdown by Source





## 11 Conclusion

By integrating R into daily operations, Aoife’s GP practice can achieve significant improvements in how data is managed, analyzed, and utilized. The following key benefits demonstrate how R can address the challenges the practice faces:

- **Centralizing and Cleaning Data**  
R’s powerful packages, such as `dplyr` and `tidyr`, make it possible to consolidate data from various sources into a single, structured format. This eliminates redundancy, resolves inconsistencies, and ensures that the data is accurate and ready for analysis
- **Automating Labor-Intensive Reporting Tasks**  
Reports such as monthly HSE reimbursement claims and end-of-year patient tax summaries, which previously required significant manual effort, can now be automated using reproducible R scripts. This reduces the workload, minimizes errors, and ensures consistent, high-quality outputs. Additionally, the `knitr` package, combined with `kable` for table formatting, allows for the seamless generation of clean, professional-looking reports in PDF or HTML formats. These tools ensure that Aoife’s practice can deliver high-quality, polished documents efficiently
- **Providing Actionable Insights Through Data Analysis and Visualization**  
The `ggplot2` package enables the creation of intuitive, professional-quality visualizations that make complex data easier to interpret. Whether it’s monitoring GP performance or analyzing appointment trends, these insights empower data-driven decision-making and operational improvements
- **Interactive Data Dashboards with Shiny** Additionally, another suggestion is that by using the Shiny package, Aoife’s GP practice can create interactive dashboards to visualize trends in real time. These dashboards could track patient appointments, financial metrics, or monitor compliance with healthcare regulations. The ability to interact with data through a web-based interface enhances accessibility and allows for quick adjustments in response to changing conditions

### 11.1 Broader Impacts

Implementing R will bring measurable improvements to operational efficiency, allowing staff to dedicate more time to patient care and less time to administrative tasks. Receptionists will experience reduced workloads, enabling better support for doctors and patients alike. Furthermore, the practice will have the ability to use well-organized data as a basis for strategic discussions with external consultants or stakeholders.

By leveraging R’s capabilities, Aoife’s GP practice can transform its operational model into one that is proactive, strategic, and resilient—positioning itself for continued growth and success in an increasingly data-driven healthcare landscape.