Week 4 / Assignment – Tidying and Transforming Data

Alina Vikhnevich

2025-02-23

Introduction

When working with data, one of the most important steps is ensuring it is structured in a way that makes analysis easy and insightful. This is where tidy data comes in - where each row represents a single observation, each column represents a variable, and each cell contains a single value. Having data in this format helps avoid confusion and allows for smooth transformations, visualizations, and comparisons.

In this report, we take flight status data from two airlines and transform it from a wide format (where each city has its own column) into a long format (where each row represents a single airline's flight status to a given city). This transformation allows us to efficiently summarize flight trends and explore potential patterns in delays and on-time flights.

Tidy data is a structured format where each row represents an observation, each column represents a variable, and each cell contains a single value. This structure makes data easier to manipulate, analyze, and visualize. In this report, we reshape flight data from wide to long format to facilitate meaningful comparisons between airlines and flight statuses. This report focuses on tidying and transforming data in R. Using tidyr and dplyr, we will reshape data between wide and long formats, making it more suitable for analysis.

Loading the Data

To begin, we first load our dataset, which contains flight status information for Alaska Airlines and AM West, across five destinations. We use raed_csv() to read the dataset and print its contents to ensure it was loaded correctly.

```
flight_data <- read_csv("/Users/alina_vikhnevich/Desktop/Spring 2025/DATA 607/DATA607/flight_data.csv")
print(flight_data)
```

```
## # A tibble: 4 x 7
##   Airline Status  `Los Angeles` Phoenix `San Diego` `San Francisco` Seattle
##   <chr>   <chr>           <dbl>   <dbl>       <dbl>           <dbl>   <dbl>
## 1 ALASKA  on time           497     221         212             503    1841
## 2 ALASKA  delayed            62      12          20             102     305
## 3 AM WEST on time           694    4840         383             320     201
## 4 AM WEST delayed           117     415          65             129      61
```

Tidying the Data

Currently, the dataset is in a wide format, where each destination is represented as a separate column. While this format is easy for humans to read, it's not ideal for analysis in R. Using the pivot_longer() function,

we reshape the data into a long format, where each row contains the airline, flight status, destination, and number of flights.

Reshaping Wide to Long Format

```
tidy_flight_data <- flight_data %>%
  pivot_longer(cols = -c(Airline, Status),
               names_to = "Destination",
               values_to = "Count")
print(tidy_flight_data)
```

```
## # A tibble: 20 x 4
##    Airline Status  Destination   Count
##    <chr>   <chr>   <chr>         <dbl>
##  1 ALASKA  on time Los Angeles     497
##  2 ALASKA  on time Phoenix         221
##  3 ALASKA  on time San Diego       212
##  4 ALASKA  on time San Francisco   503
##  5 ALASKA  on time Seattle        1841
##  6 ALASKA  delayed Los Angeles      62
##  7 ALASKA  delayed Phoenix          12
##  8 ALASKA  delayed San Diego        20
##  9 ALASKA  delayed San Francisco   102
## 10 ALASKA  delayed Seattle         305
## 11 AM WEST on time Los Angeles     694
## 12 AM WEST on time Phoenix        4840
## 13 AM WEST on time San Diego       383
## 14 AM WEST on time San Francisco   320
## 15 AM WEST on time Seattle         201
## 16 AM WEST delayed Los Angeles     117
## 17 AM WEST delayed Phoenix         415
## 18 AM WEST delayed San Diego        65
## 19 AM WEST delayed San Francisco   129
## 20 AM WEST delayed Seattle          61
```

Summarizing the Data

With our data now in long format, we can more easily perform aggregations and comparisons. In its original format, comparing flights across multiple cities was difficult, as each city had a separate column. Now, we can group by airline and flight status to summarize total flights for each airline.

By converting the data to long format, we enable easier analysis and visualization. In its wide format, comparisons between airlines and destinations were difficult since each city had its own column. The long format allows us to efficiently group, filter, and summarize flight data, revealing trends in delays and on-time performances. This approach is particularly useful for modeling and visualization, as tools like ggplot2 work best with tidy data.

Total Flights by Airline and Status

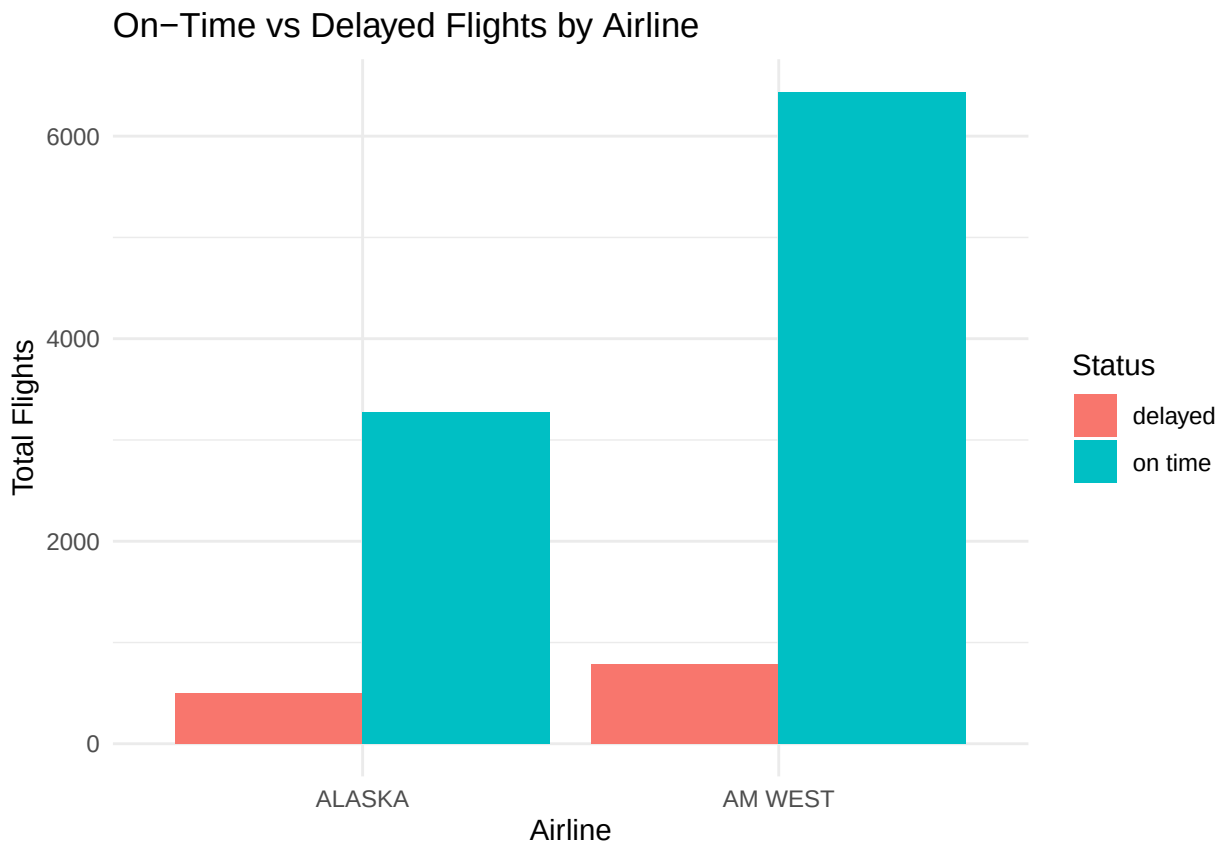
```
summarized_data <- tidy_flight_data %>%
  group_by(Airline, Status) %>%
  summarise(Total_Flights = sum(Count), .groups = 'drop')
print(summarized_data)
```

```
## # A tibble: 4 x 3
##   Airline Status  Total_Flights
##   <chr>   <chr>           <dbl>
## 1 ALASKA  delayed           501
## 2 ALASKA  on time          3274
## 3 AM WEST delayed           787
## 4 AM WEST on time          6438
```

Visualizing the Data

A bar chart is one of the best ways to compare flight trends across airlines. Below, we plot the number of on-time vs. delayed flights for each airline, using ggplot2. The **facet_wrap()** function allows us to visually separate data by airline, making it easier to spot differences in flight performance.

Bar Plot: On-Time vs Delayed Flights



Conclusion

Through this analysis, we gained insights into flight trends across different destinations. Converting the dataset from wide to long format allowed for easier data manipulation and visualization.

Key Takeaways:

1. **Alaska Airlines** had a higher percentage of on-time flights, making it a better choice for reliability.

2. Seattle and Los Angeles had the highest total flights, while San Diego had the fewest.
3. **AM West** experienced more delays, which could indicate potential scheduling inefficiencies.

By restructuring our data into a tidy format, we were able to quickly analyze patterns and trends, making it easier to derive meaningful insights for decision-making.

This assignment demonstrated how to reshape and analyze flight data in R. Using tidyr and dplyr, we converted wide-format data into a tidy structure, allowing for better insights into flight trends.