DATA 607 Week Five: Working with Tidy Data

Kevin Kirby

2024-09-24

Week Five Homework Overview

		Los Angeles	Phoenix	San Diego	San Francisco	Seattle
ALASKA	on time	497	221	212	503	1,841
	delayed	62	12	20	102	305
AM WEST	on time	694	4,840	383	320	201
	delayed	117	415	65	129	61

Source: *Numbersense*, Kaiser Fung, McGraw Hill, 2013

Figure 1: Airline Delays Chart

The image above was provided by the assignment and is basis for the work below. It describes arrival delays for two airlines across five destinations.

For this assignment, I need to:

- Create a .CSV file based on the image above
  - I was "encouraged to use a 'wide' structure similar to how the information appears above so that [I] can practice tidying and transformations. . . "
- Read the information from your .CSV file into R, and use tidyr and dplyr as needed to tidy and transform your data.
- Perform analysis to compare the arrival delays for the two airlines

Import and Tidy Data

To tidy the data, I had to spin my tires for a bit to find the right combination of tidyverse code to unlock the format I needed. It took me a long minute to realize that "col" needed to equal the entire comma separated header as a string. Once I figured that out, I was able to break it from the comma separated values in single cells into a traditional column/row setup.

The data initially came in as non-numeric, which I learned when I went to go do the summary that's next and I got an error saying the values were characters. I returned back and turned the city columns of number of flights into numbers.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(dplyr)
library(ggplot2)

delays_csv <- "https://storage.googleapis.com/data_science_masters_files/2024_fall/data_607_data_manage
delays_csv_raw <- read_csv(delays_csv)
```

```
## Rows: 4 Columns: 1
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Airline,Status,Los Angeles,Phoenix,San Diego,San Francisco,Seattle
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
delays_split <- delays_csv_raw %>%
  separate(col = 'Airline,Status,Los Angeles,Phoenix,San Diego,San Francisco,Seattle', into = c('Airlin

delays_split <- delays_split %>%
  mutate(across(c('Los Angeles', 'Phoenix', 'San Diego', 'San Francisco', 'Seattle'), as.numeric))
```

Analysis

For the final analysis piece, I first moved the data around a bit so I had the different cities as values in a column instead of columns on their own. It's a lot easier to group different values from the same column together to do analysis than to have a setup where some key data is diffferent row values and others are different column values. In my experience, it's best to keep like with like.

delays_city produces a basic table that shows on-time percent by airline, status, and city.

```
delays_analysis <- delays_split %>%
  pivot_longer(cols = c('Los Angeles', 'Phoenix', 'San Diego', 'San Francisco', 'Seattle'),
               names_to = "city", values_to = "flights")

delays_city <- delays_analysis %>%
  group_by(Airline, city) %>%
  mutate(total_flights = sum(flights)) %>%
  ungroup() %>%
  mutate(ontime_percent = (flights / total_flights) * 100) %>%
  select(Airline, city, Status, ontime_percent)
```

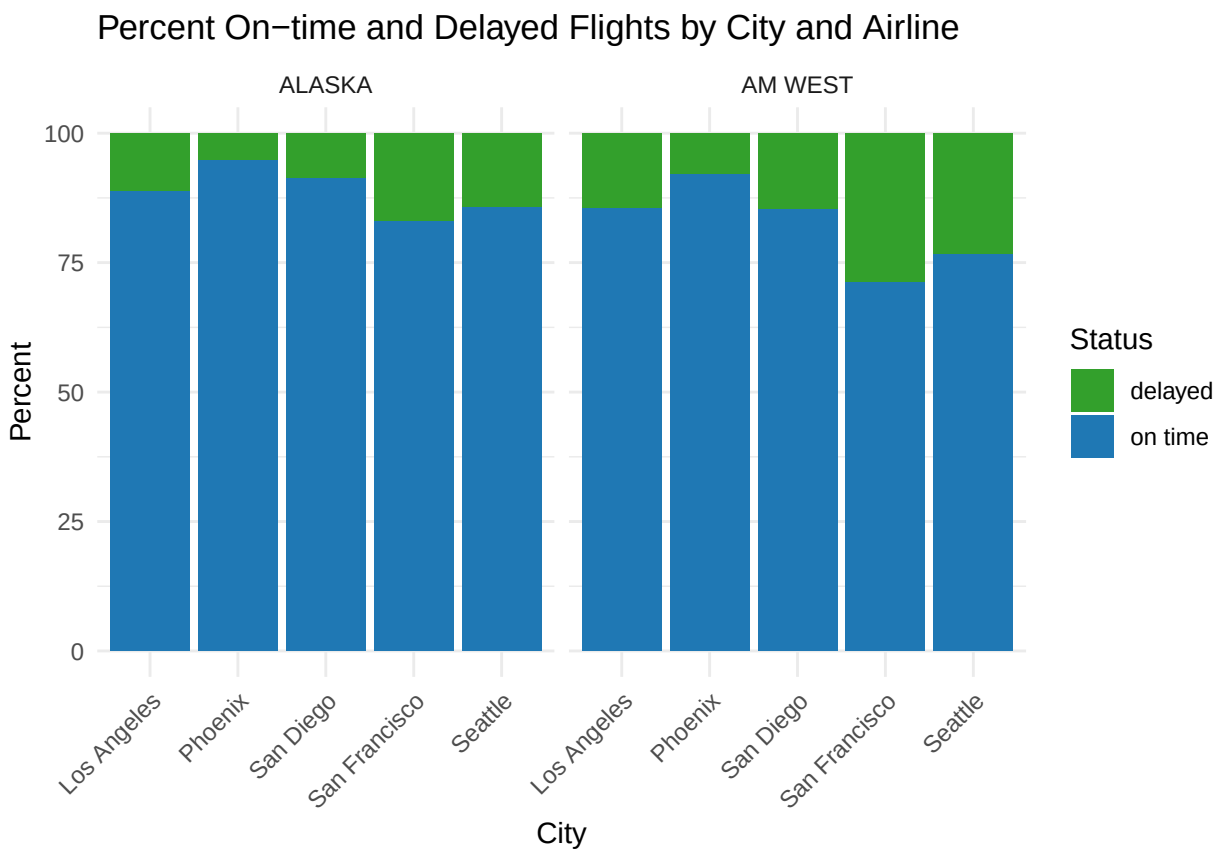
```r
print(delays_city)
```

```
## # A tibble: 20 x 4
##    Airline city          Status  ontime_percent
##    <chr>   <chr>         <chr>            <dbl>
##  1 ALASKA  Los Angeles   on time          88.9
##  2 ALASKA  Phoenix       on time          94.8
##  3 ALASKA  San Diego     on time          91.4
##  4 ALASKA  San Francisco on time          83.1
##  5 ALASKA  Seattle       on time          85.8
##  6 ALASKA  Los Angeles   delayed          11.1
##  7 ALASKA  Phoenix       delayed           5.15
##  8 ALASKA  San Diego     delayed           8.62
##  9 ALASKA  San Francisco delayed          16.9
## 10 ALASKA  Seattle       delayed          14.2
## 11 AM WEST Los Angeles   on time          85.6
## 12 AM WEST Phoenix       on time          92.1
## 13 AM WEST San Diego     on time          85.5
## 14 AM WEST San Francisco on time          71.3
## 15 AM WEST Seattle       on time          76.7
## 16 AM WEST Los Angeles   delayed          14.4
## 17 AM WEST Phoenix       delayed           7.90
## 18 AM WEST San Diego     delayed          14.5
## 19 AM WEST San Francisco delayed          28.7
## 20 AM WEST Seattle       delayed          23.3
```

This chart breaks out flight status by city and airline. I like to use stacked bar charts for these types of percent comparisons because in early EDA, which is what this is, I'm looking for patterns that stand out right away or don't quite make sense. Scanning across this, the question that immediately came to mind was: why does AM West struggle more in San Francisco than Alaska?

```r
status_colors <- c("on time" = "#1f78b4", "delayed" = "#33a02c")

ggplot(delays_city, aes(x = city, y = ontime_percent, fill = Status)) +
  geom_bar(stat = "identity", position = "stack") +
  facet_grid(~ Airline, scales = "free_x", space = "free_x") +  # Group by airline
  scale_fill_manual(values = status_colors) +  # Use status-specific colors
  labs(title = "Percent On-time and Delayed Flights by City and Airline",
       x = "City", y = "Percent") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


Percent On−time and Delayed Flights by City and Airline
ALASKA
AM WEST
100
75
50
25
0
Percent
Los Angeles
Phoenix
San Diego
San Francisco
Seattle
City
Status
delayed
on time